

# Controle Preditivo Baseado em Modelo - Volume 2: Casos Multivariável e Não-linear

Julio Elias Normey-Rico  
Rodolfo César Costa Flesch  
Daniel Martins Lima  
Vinícius Berndsen Peccin

22 de fevereiro de 2023



# Sumário

<b>1</b>	<b>Introdução</b>	<b>7</b>
<b>2</b>	<b>Revisitando o MPC SISO</b>	<b>11</b>
2.1	MPC SISO – Ideias básicas e notação . . . . .	11
2.2	Um exemplo motivador . . . . .	15
2.3	Comentários finais . . . . .	19
2.4	Exercícios propostos . . . . .	19
<b>3</b>	<b>DMC e GPC Multivariáveis</b>	<b>21</b>
3.1	Obtenção da lei de controle . . . . .	25
3.2	DMC multivariável . . . . .	36
3.2.1	Cálculo das predições no caso estável . . . . .	36
3.2.2	Ação antecipativa - <i>Feed-forward</i> . . . . .	43
3.2.3	DMC multivariável recursivo . . . . .	46
3.2.4	DMC MIMO para plantas integradoras . . . . .	50
3.2.5	DMC Generalizado multivariável . . . . .	53
3.3	GPC multivariável . . . . .	59
3.3.1	Cálculo das predições (caso $C(z^{-1}) \neq 1$ ) . . . . .	73
3.4	Estudos de caso . . . . .	83
3.4.1	Coluna de destilação água-metanol . . . . .	83
3.4.2	Fracionador de óleo pesado . . . . .	86
3.4.3	Sistema de quatro tanques . . . . .	91
3.4.4	Reator Continuamente Agitado (CSTR) . . . . .	95
3.5	Exercícios propostos . . . . .	101
<b>4</b>	<b>MPC em Variáveis de Estado</b>	<b>105</b>
4.1	Formulação do SSMPC . . . . .	108
4.2	Ação antecipativa - <i>Feed-forward</i> . . . . .	113
4.3	Obtenção da lei de controle . . . . .	117
4.3.1	Inclusão de restrições . . . . .	120
4.4	Exemplos . . . . .	126

4.5	Controlador equivalente . . . . .	131
4.6	Comentários finais . . . . .	135
<b>5</b>	<b>Controle Preditivo Não Linear</b>	<b>137</b>
5.1	Controle preditivo não linear: Uma visão geral . . . . .	138
5.2	Controle NMPC direto . . . . .	142
5.2.1	CasADi – Generalidades . . . . .	143
5.2.2	Montando o NMPC em CasADi . . . . .	145
5.2.3	Um exemplo completo . . . . .	146
5.3	O algoritmo NEPSAC . . . . .	150
5.3.1	Modelo de predição . . . . .	151
5.3.2	Decomposição da sequência de controle . . . . .	158
5.3.3	Extensão para o caso multivariável . . . . .	169
5.4	Practical Nonlinear Model Predictive Control - PNMPC . . .	171
5.4.1	Obtendo a resposta livre . . . . .	171
5.4.2	Obtendo a resposta forçada . . . . .	172
5.4.3	Algoritmo PNMPC . . . . .	177
5.4.4	Extensão para o caso multivariável . . . . .	179
5.5	Estudo de Caso Multivariável – CSTR . . . . .	180
5.6	Comentários finais . . . . .	186
5.7	Exercícios propostos . . . . .	186
<b>6</b>	<b>MPC para Sistemas com Atraso</b>	<b>189</b>
6.1	Entendendo o efeito do atraso no MPC . . . . .	189
6.2	Interpretação do GPC como um DTC e o algoritmo DTC-GPC	210
6.2.1	Análise do GPC para sistemas com atraso . . . . .	211
6.2.2	O algoritmo DTC-GPC . . . . .	221
6.2.3	Exemplos . . . . .	226
6.2.4	O caso multivariável . . . . .	232
6.3	DMC para sistemas com atraso . . . . .	236
6.3.1	DMC como um DTC: plantas estáveis . . . . .	237
6.3.2	DMC Filtrado: Plantas estáveis . . . . .	241
6.3.3	GDMC: plantas integradoras e instáveis . . . . .	244
6.3.4	DMC filtrado MIMO: plantas estáveis . . . . .	249
6.3.5	O caso não linear . . . . .	250
6.4	Exercícios propostos . . . . .	252
<b>7</b>	<b>Métodos de Otimização e Algoritmos para o MPC</b>	<b>255</b>
7.1	Algoritmo DMC recursivo . . . . .	257
7.2	Algoritmo GPC recursivo . . . . .	260
7.3	Método de barreira logarítmica . . . . .	263

7.3.1	Método de Newton . . . . .	263
7.3.2	Barreira logarítmica . . . . .	264
7.3.3	Caminho central . . . . .	267
7.3.4	Sintonização . . . . .	268
7.3.5	Algoritmo de barreira logarítmica para MPC . . . . .	270
7.4	Método de projeção de gradiente . . . . .	272
7.4.1	Projeção ortogonal . . . . .	276
7.4.2	Aceleração do método de projeção de gradiente . . . . .	278
7.4.3	Algoritmo GPAD para MPC . . . . .	279
7.5	Métodos de operador de partição . . . . .	279
7.5.1	Precursos . . . . .	281
7.5.2	Formulação ADMM . . . . .	283
7.5.3	Sintonização . . . . .	283
7.5.4	Condições de otimalidade . . . . .	284
7.5.5	Convergência . . . . .	285
7.5.6	Forma escalonada . . . . .	285
7.5.7	Algoritmo ADMM para MPC . . . . .	285
7.6	Considerações finais . . . . .	288
7.7	Exercícios propostos . . . . .	288
<b>A</b>	<b>Apêndice: Simuladores</b>	<b>291</b>
A.1	Simulador GPC MIMO . . . . .	291
<b>B</b>	<b>Definições de otimização convexa</b>	<b>295</b>



# Capítulo 1

## Introdução

Este livro é o Volume 2 da obra que tem como objetivo principal apresentar aos estudantes de engenharia (e de áreas afins) os fundamentos do controle preditivo baseado em modelo, também denominado controle preditivo (MPC, do inglês *Model Predictive Control*), que é uma das metodologias de controle mais usadas na indústria de processos.

Seguindo a mesma linha do Volume 1, neste volume a abordagem também é teórico-prática, apresentando o desenvolvimento da teoria clássica do MPC, aspectos de implementação dos algoritmos de controle e estudos de caso, alguns deles retirados do ambiente industrial, como do setor de energia solar, e da indústria de petróleo e gás. Neste volume, são apresentadas as versões multivariáveis (MIMO, do inglês *Multi-Input Multi-Output*) dos controladores DMC (do inglês *Dynamic Matrix Control*) e GPC (do inglês *Generalized Predictive Controller*) estudados no Volume 1, e estudaremos também outras estratégias de MPC linear e não linear. Ainda, neste volume analisamos aspectos relacionados com o controle MPC de processos com atraso e com os algoritmos de otimização utilizados.

Dando continuidade ao apresentado no Volume 1, neste volume, o desenvolvimento teórico apresentado segue, também, uma abordagem no domínio do tempo discreto, recorrendo a conceitos e ferramentas básicas da teoria de controle digital clássico, como equações a diferenças e transformada Z, e a ferramentas de sistemas multivariáveis como descrições em variáveis de estado e matriz função de transferência. O livro é orientado a estudantes de engenharia que possuem conhecimentos básicos de teoria de controle em tempo contínuo e tempo discreto, e que conhecem a teoria de controle MPC SISO e os controladores GPC e DMC na sua versão monovariável, seja através do estudo do conteúdo do Volume 1 desta obra ou de outro material bibliográfico. Assim, este Volume 2 é mais orientado a disciplinas de mestrado e doutorado, mas também pode ser usado em cursos de graduação, nas

fases finais.

Assim, os próximos capítulos estão organizados da seguinte maneira:

- No Capítulo 3 é apresentada a generalização dos algoritmos DMC e GPC do caso SISO para o caso MIMO, mas inicia o estudo discutindo aspectos particulares do tratamento de sistemas com múltiplas entradas e saídas, que são válidas para todo tipo de sistema de controle MIMO.
- O Capítulo 4 é dedicado ao SSMPC (do inglês *State Space Model Predictive Control*), que, assim como o DMC e GPC, também é uma abordagem muito usada, principalmente no estudo de problemas mais teóricos de MPC. Este capítulo apresenta o SSMPC diretamente para o caso MIMO, sendo os sistemas SISO tratados como um caso particular. Isto se justifica pelas próprias características da representação de sistemas em variáveis de estado, que serve de forma unificada para sistemas SISO e MIMO.
- No Capítulo 5 é apresentado um panorama geral sobre o MPC não linear (NMPC, do inglês *Nonlinear Model Predictive Control*) e como implementar algoritmos que sejam passíveis de uso na prática industrial, onde muitos processos apresentam dinâmicas não lineares. O capítulo trata inicialmente o caso SISO e generaliza depois os resultados para o caso MIMO.
- No Capítulo 6 é apresentado o problema de controle preditivo de processos com atraso. A estrutura do MPC é analisada de forma tal a entender como a compensação do atraso dentro da estrutura do controlador é realizada. Assim, traçando um paralelo com as estruturas de compensadores de tempo morto (*Dead-time compensators* - DTC) analisamos diversos aspectos relacionados com a robustez do MPC e a rejeição de perturbações. Finalmente, estudamos uma forma de modificar o preditor para conseguir controladores preditivos lineares e não lineares mais robustos e mais simples de ajustar quando controlam processos com atraso,
- No Capítulo ?? são tratados aspectos práticos dos controladores MPC, generalizando alguns dos tópicos apresentados no Volume 1 e adicionado outros específicos do caso MIMO. O foco do capítulo é analisar estratégias usadas na prática para uso industrial do MPC.
- Por fim, no Capítulo 7, são tratados aspectos relacionados com os algoritmos de otimização e metodologias para poder implementar o MPC



em sistemas rápidos, que precisam de períodos de amostragem muito pequenos.

Para fins de revisão dos conteúdos apresentados, no final de cada capítulo, são propostos exercícios e/ou problemas e atividades de simulação. Ainda, ferramentas de simulação gratuitas são apresentadas no apêndice e códigos dos controladores estudados são disponibilizados para descarga na página web do livro <sup>1</sup>.

---

<sup>1</sup><https://danielml.paginas.ufsc.br/livro-mpc/>



## Capítulo 2

# Revisitando o MPC SISO

Como já comentado, na apresentação do conteúdo deste livro consideramos que o leitor conhece a teoria de controle preditivo SISO mas, para facilitar o entendimento, neste capítulo revisamos as principais ideias do MPC SISO apresentadas no Volume 1, assim como definimos alguns aspectos de notação, de forma tal que o leitor que conhece o assunto possa prosseguir com a análise dos próximos capítulos sem necessidade de recorrer constantemente àquele volume. Ainda, para motivar o estudo dos próximos capítulos, também apresentamos um exemplo simulado do uso do controle GPC MIMO, que mostra a potencialidade do MPC.

### 2.1 MPC SISO – Ideias básicas e notação

No caso SISO, consideramos um processo com saída ou variável controlada  $y$ , variável de controle ou manipulada  $u$  e perturbação mensurável  $q$ . O MPC SISO estudado no Volume 1 utiliza um modelo matemático linear e em tempo discreto que aproxima o comportamento do processo, de forma tal que se pode escrever  $y(k)$  como função de  $u(k)$  e  $q(k)$ . Usando esse modelo, o MPC calcula as previsões da saída do processo em uma janela de tempo futura (denominada horizonte de predição), e usando um otimizador, encontra as ações de controle futuras (no horizonte de controle) que minimizam uma determinada função objetivo, considerando as restrições de operação do processo (como por exemplo valores máximos e mínimos das variáveis envolvidas).

O MPC é uma estratégia de controle que utiliza realimentação, dado que o modelo utiliza os valores passados das variáveis controladas para calcular as previsões que, por sua vez, são usadas para encontrar a ação de controle, isto é, a ação de controle depende da saída medida. Por outro lado, o MPC

pode ter também uma ação antecipativa ou de pre-alimentação, caso utilize informações das perturbações medidas e tenha conhecimento das referências futuras que a variável controlada deve seguir.

De forma geral, o MPC calcula a ação de controle a ser aplicada no instante atual,  $k$ , a partir da predição do comportamento da planta, baseado em um modelo dinâmico da mesma, em um horizonte finito  $N$ . São computados  $N_u$  controles futuros por meio da otimização de uma função objetivo que representa algum critério de comportamento do sistema e condições de operação da planta. Apenas a primeira ação de controle é aplicada ao sistema e o MPC recalcula uma nova sequência de controle a cada novo instante de amostragem, uma vez que novas informações da planta são adquiridas.

O MPC não é uma estratégia de controle específica, mas uma filosofia de controle que contempla uma família de estratégias ou técnicas de controle que têm alguns elementos em comum [CB04]:

- modelo de predição;
- função objetivo (ou função custo);
- procedimento para a obtenção da lei de controle.

O que diferencia cada tipo de técnica é basicamente como as perturbações são consideradas, o modelo utilizado na predição e a função objetivo, como visto no Volume 1 para o DMC e GPC. Vamos aqui apenas revisar as ideias principais.

O DMC SISO utiliza um modelo de resposta ao degrau do processo:

$$y(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i), \quad (2.1)$$

onde  $\Delta u$  é o incremento de controle, e os coeficientes  $g_i$  representam a resposta ao degrau que relaciona a variável manipulada com a saída. O modelo para as perturbações não medidas considera que elas são constantes no horizonte:  $\hat{n}(k+j | k) = \hat{n}(k | k)$  para todo  $j$ . Assim, as predições pode ser escritas, para um elemento  $j$  no horizonte, como:

$$\hat{y}(k+j | k) = \sum_{i=1}^{\infty} g_i \Delta u(k+j-i) + \hat{n}(k | k),$$

onde  $\hat{y}(k+j | k)$  é uma predição da saída em um tempo futuro  $k+j$ , dadas as informações em  $k$ , e  $\hat{n}(k | k) = y(k) - \hat{y}_o(k | k)$ , sendo  $\hat{y}_o$  a estimação sem correção da saída do sistema obtida a partir de (2.1).

Já o GPC SISO utiliza um modelo de função de transferência para representar as relações entre a saída do processo, a variável manipulada  $u$  e as perturbações não medidas  $n$ :

$$y(k) = \frac{B(z^{-1})z^{-d}}{A(z^{-1})}u(k-1) + \frac{C(z^{-1})}{A(z^{-1})}n(k),$$

com  $A$ ,  $B$  e  $C$  polinômios no operador atraso  $z^{-1}$  e  $d$  o atraso do processo, As perturbações são modeladas por:

$$n(k) = \frac{C(z^{-1})}{1 - z^{-1}}e(k),$$

com  $e(k)$  um ruído branco de média nula. Assim, as predições são calculadas, para um elemento  $j$  no horizonte, usando:

$$\hat{y}(k+j | k) = \frac{B(z^{-1})z^{-d}}{A(z^{-1})}u(k+j-1) + \frac{C(z^{-1})}{(1 - z^{-1})A(z^{-1})}e(k+j),$$

que pode ser escrita como:

$$\tilde{A}(z^{-1})\hat{y}(k+j | k) = B(z^{-1})z^{-d}\Delta u(k+j-1) + C(z^{-1})e(k+j),$$

com  $\tilde{A}(z^{-1}) = A(z^{-1})(1 - z^{-1})$  e  $\Delta u(k) = (1 - z^{-1})u(k)$ . Este modelo é chamado de CARIMA, do inglês *Controlled Autoregressive Integrated Moving Average*.

A função objetivo para o caso SISO (apresentada no Volume 1, e usada tanto no DMC quanto no GPC) é dada por:

$$J = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(k+j | k) - r(k+j)]^2 + \sum_{j=0}^{N_u-1} \lambda(j) [\Delta u(k+j)]^2, \quad (2.2)$$

onde  $r(k+j)$  é a referência em  $k+j$ ,  $\Delta u(k+j)$  é o incremento de controle em  $k+j$ ,  $\delta(j)$  e  $\lambda(j)$  ponderam o comportamento futuro do erro e do esforço de controle, respectivamente,  $N_1$  e  $N_2$  definem uma janela de horizonte de predição, sendo que  $N = N_2 - N_1 + 1$ , e  $N_u$  é o horizonte de controle.

A função objetivo descrita em (2.2) busca, por um lado, que a saída do processo siga uma dada referência e, por outro lado, ajustar o esforço de controle necessário para esse objetivo. Note que o compromisso entre estes dois objetivos é ajustado utilizando  $\delta(j)$  e  $\lambda(j)$ . Neste objetivo  $N_1$  e  $N_2$ , são usados para definir que janela de tempo desejamos considerar em  $J$ , e  $N_u$  ajusta o compromisso entre flexibilidade para definir os sinais de controle e a complexidade computacional.

A função objetivo pode ser colocada de forma vetorial, definindo os vetores de incrementos de controles futuros  $\Delta \mathbf{u} \in \mathbb{R}^{N_u}$ , de referências futuras  $\mathbf{r} \in \mathbb{R}^N$ , e de predições futuras  $\hat{\mathbf{y}} \in \mathbb{R}^N$ :

$$J = [\mathbf{r} - \hat{\mathbf{y}}]^T \mathbf{Q}_y [\mathbf{r} - \hat{\mathbf{y}}] + \Delta \mathbf{u}^T \mathbf{Q}_u \Delta \mathbf{u} \quad (2.3)$$

com

$$\mathbf{r} = \begin{bmatrix} r(k + N_1) \\ r(k + N_1 + 1) \\ \vdots \\ r(k + N_2) \end{bmatrix}, \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k + N_1 | k) \\ \hat{y}(k + N_1 + 1 | k) \\ \vdots \\ \hat{y}(k + N_2 | k) \end{bmatrix}, \quad \Delta \mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \\ \vdots \\ \Delta u(k + N_u - 1) \end{bmatrix}$$

e se as ponderações são constantes no horizonte  $\delta(j) = \delta$  e  $\lambda(j) = \lambda$ :

$$\mathbf{Q}_y = \delta \mathbf{I}_N, \quad \mathbf{Q}_u = \lambda \mathbf{I}_{N_u},$$

sendo  $\mathbf{I}_N$  e  $\mathbf{I}_{N_u}$  matrizes identidade de dimensão  $N$  e  $N_u$  respectivamente.

Para expressar  $J$  apenas em função das variáveis de decisão, que são os valores futuros da ação de controle, tanto o DMC como o GPC usam o conceito de resposta livre e resposta forçada, que foi apresentado no Volume 1 e que é resumido na sequência.

Para o DMC, usando o modelo de predição, os valores das predições nos instantes  $j$  da janela de predição podem ser escritos em função dos controles futuros e passados. Se separamos estas duas partes podemos escrever o vetor de predições da saída do processo  $\hat{\mathbf{y}}$  como  $\hat{\mathbf{y}} = \mathbf{G}\Delta \mathbf{u} + \mathbf{f}$ , sendo  $\mathbf{G}$  a matriz dinâmica ou de resposta ao degrau do sistema, e  $\mathbf{f}$  a resposta livre, que expressa a dependência de  $\hat{\mathbf{y}}$  com os controles passados. **Como visto no Volume 1, a resposta livre é obtida se mantemos o controle futuro constante, ou seja todos os incrementos futuros são nulos e assim  $\Delta \mathbf{u} = \mathbf{0}_{N_u}$ , e  $\hat{\mathbf{y}} = \mathbf{f}$  ( $\mathbf{0}_{N_u}$  é um vetor nulo de dimensão  $N_u$ ). Já se consideramos o sistema com uma condição de equilíbrio inicial  $y_0$  com os controles passados constantes,  $\mathbf{f} = y_0 \mathbf{1}_N$  ( $\mathbf{1}_N$  é um vetor cujos elementos são todos 1 de dimensão  $N$ ), e assim obtemos  $\hat{\mathbf{y}} = \mathbf{G}\Delta \mathbf{u} + y_0 \mathbf{1}_N$  que mostra a dependência das predições apenas com as variações de controle que executaremos no futuro, pois o termo  $y_0 \mathbf{1}_N$  é constante.**

Para o GPC, usando o modelo CARIMA recursivamente, podemos calcular os valores das predições nos instantes  $j$  da janela de predição em função dos controles futuros e dos valores passados da saída e do controle. Assim, da mesma forma que no DMC, é possível separar as predições nas partes correspondentes à resposta forçada e à resposta livre, que têm a mesma interpretação que no DMC.

Usando então a equação  $\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f}$  para substituir  $\hat{\mathbf{y}}$  na expressão de  $J$  dada por (2.3), podemos transformar a função objetivo em uma expressão dependente apenas do controle futuro e de informações conhecidas (da referência e dos valores passados das variáveis). Então, o objetivo do controle preditivo SISO é resolver, a cada amostra, um processo de otimização da função objetivo do tipo:

$$\begin{aligned} \min_{\Delta\mathbf{u}} \quad & J \\ \text{s.a.} \quad & \bar{\mathbf{R}}\Delta\mathbf{u} \leq \bar{\mathbf{r}} \\ & \bar{\mathbf{C}}\Delta\mathbf{u} = \bar{\mathbf{c}}, \end{aligned} \tag{2.4}$$

onde  $\bar{\mathbf{R}}\Delta\mathbf{u} \leq \bar{\mathbf{r}}$  e  $\bar{\mathbf{C}}\Delta\mathbf{u} = \bar{\mathbf{c}}$  representam as restrições de desigualdade e igualdade, respectivamente, colocadas em função do vetor de incrementos de controles futuros,  $\Delta\mathbf{u}$ , que tem dimensão  $N_u$ . As matrizes  $\bar{\mathbf{C}}$ ,  $\bar{\mathbf{R}}$ ,  $\bar{\mathbf{r}}$  e  $\bar{\mathbf{c}}$  são calculadas em função do modelo e dos limites impostos para as diferentes variáveis. Estas restrições podem ser de todo tipo e sobre todas as variáveis do problema mas, em geral, podem ser colocadas em função da variável de decisão do problema, que é o vetor de controle futuro, usando novamente a equação  $\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f}$  para substituir  $\hat{\mathbf{y}}$  em todas as restrições que se apliquem à saída futura do processo.

O problema de controle (2.4) definido no MPC, que é um problema de otimização, tipicamente deve ser resolvido a cada instante de amostragem. Há casos em que o problema de otimização tem solução analítica mas, em geral, ele precisa ser resolvido com métodos numéricos. A solução encontrada nesse problema pode variar dependendo dos parâmetros de ajuste do controle MPC, as ponderações e horizontes.

## 2.2 Um exemplo motivador

Para motivar os leitores e mostrar o potencial da estratégia de controle MPC no caso MIMO, vamos considerar aqui um estudo de caso de um processo multivariável relativamente complexo que deve operar dentro de determinadas especificações. Trata-se do controle do fracionador de óleo pesado da *Shell*, um sistema muito conhecido na literatura, e utilizado para estudar diferentes estratégias de controle para colunas de destilação, inclusive de MPC [CB04]. Este exemplo ilustra o problema de controle de um processo MIMO com atrasos diferentes nas variáveis de saída  $y$  sujeito a restrições de operação.

O processo, mostrado na Figura 2.1, possui três variáveis que devem ser controladas: as composições dos produtos superior (ou de topo,  $y_1$ ) e lateral ( $y_2$ ), que são medidas por analisadores, e a temperatura no fundo (ou base)

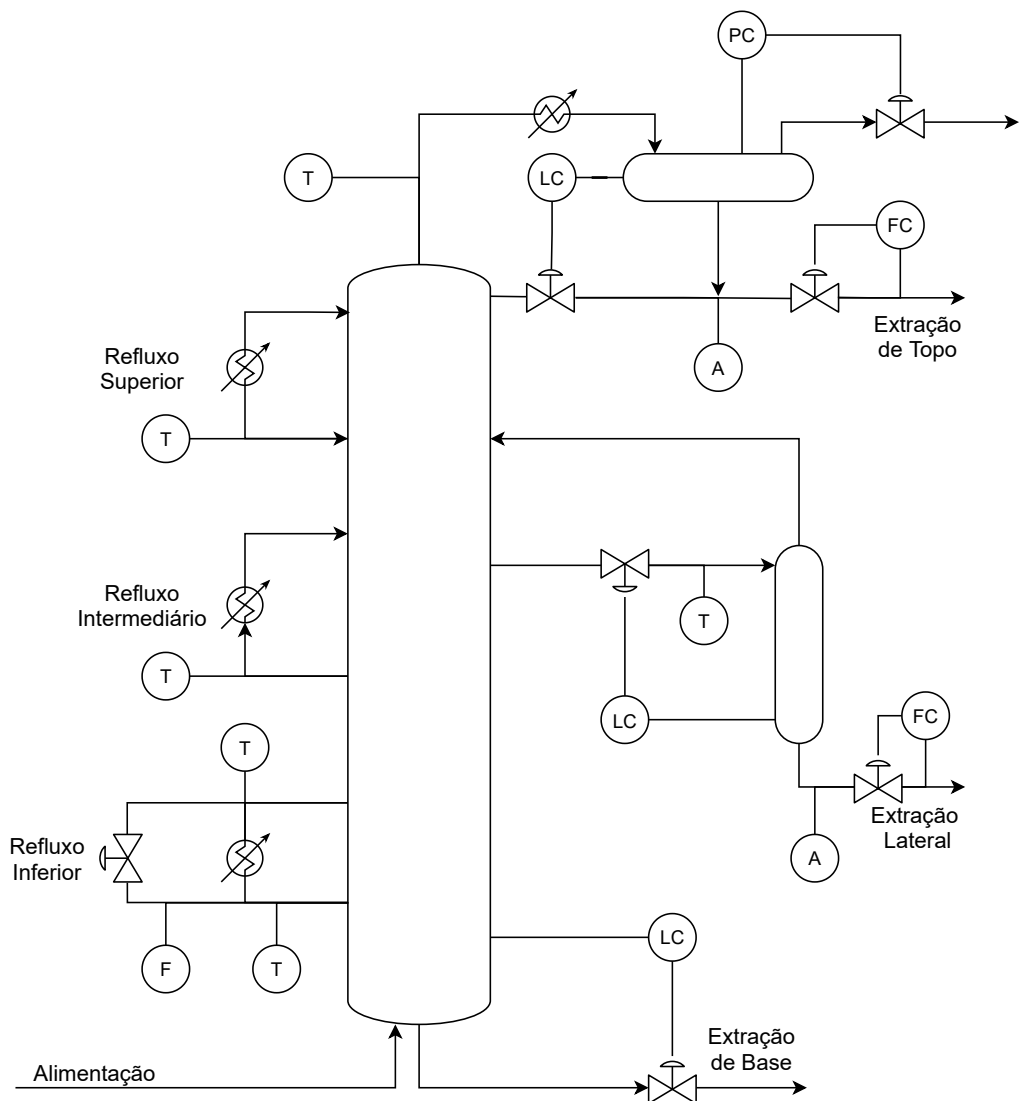


Figura 2.1: Esquema do fracionador de óleo pesado.

da torre ( $y_3$ ). As variáveis manipuladas são a vazão de extração superior (ou de topo,  $u_1$ ), a vazão de extração lateral ( $u_2$ ) e a vazão de refluxo inferior ( $u_3$ ). Neste processo, a alimentação fornece todos os requisitos de calor para a coluna, não necessitando de adição de energia externa. A dinâmica do processo pode ser descrita pelo seguinte modelo, que representa as variações das



variáveis manipuladas e controladas nas vizinhanças do ponto de operação:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} = \begin{bmatrix} \frac{4,05e^{-27s}}{1+50s} & \frac{1,77e^{-28s}}{1+60s} & \frac{5,88e^{-27s}}{1+50s} \\ \frac{5,39e^{-18s}}{1+50s} & \frac{5,72e^{-14s}}{1+60s} & \frac{6,9e^{-15s}}{1+40s} \\ \frac{4,38e^{-20s}}{1+33s} & \frac{4,42e^{-22s}}{1+44s} & \frac{7,2}{1+19s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{bmatrix}$$

O modelo de tempo discreto obtido com uma amostragem de 4 min é dado por:

$$\begin{bmatrix} \frac{0,08(z^{-1} + 2,88z^{-2})}{1 - 0,923z^{-1}}z^{-6} & \frac{0,114z^{-1}}{1 - 0,936z^{-1}}z^{-7} & \frac{0,116(z^{-1} + 2,883z^{-2})}{1 - 0,923z^{-1}}z^{-6} \\ \frac{0,211(z^{-1} + 0,96z^{-2})}{1 - 0,923z^{-1}}z^{-4} & \frac{0,187(z^{-1} + 0,967z^{-2})}{1 - 0,936z^{-1}}z^{-3} & \frac{0,17(z^{-1} + 2,854z^{-2})}{1 - 0,905z^{-1}}z^{-3} \\ \frac{0,5z^{-1}}{1 - 0,886z^{-1}}z^{-5} & \frac{0,196z^{-1} + 0,955z^{-2}}{1 - 0,913z^{-1}}z^{-5} & \frac{1,367z^{-1}}{1 - 0,81z^{-1}} \end{bmatrix}.$$

Um GPC MIMO foi projetado para este sistema usando horizontes de predição da ordem da maior constante de tempo do modelo e considerando o primeiro elemento do horizonte depois do atraso. Observe que esta escolha parece intuitivamente adequada, pois como o sinal de controle que vamos calcular somente afeta a saída depois do atraso, não faz sentido considerar na função objetivo as predições das saídas antes do atraso. Assim, para a primeira saída escolhemos  $N_1 = 6$  e  $N_2 = 15$ , para a segunda saída escolhemos  $N_1 = 3$  e  $N_2 = 15$ , e para a terceira saída escolhemos  $N_1 = 1$  e  $N_2 = 5$  (observe que consideramos o menor atraso de cada saída). O horizonte de controle foi ajustado com  $N_u = 5$  para todas as variáveis manipuladas. Assumiremos que o controlador tem conhecimento das referências futuras.

Vamos considerar que inicialmente não temos restrições nas variáveis e que todas as saídas têm a mesma importância. Assim, as ponderações de todas as variáveis manipuladas são fixadas em  $\lambda = 2$  e as das controladas em  $\delta = 1$ . O resultado de aplicar este MPC no processo, para mudanças tipo degrau nas referências das três variáveis controladas, é mostrado na Figura 2.2. As figuras mostradas são do simulador interativo usado para estudo do controle GPC MIMO, descrito no Apêndice A.

O sistema parte do ponto de operação, considerado zero na simulação pois são mostradas as variações das variáveis do processo. Como podemos observar, o sistema apresenta uma resposta com pouco acoplamento, e transitórios sem grandes oscilações, o que é adequado para este processo. As ações

de controle também têm um comportamento suave e o sistema de controle permite que as referências (mostradas em linhas pontilhadas) sejam alcançadas no regime permanente. Ainda, pode ser observado que, como o controle conhece a referência futura para as variáveis 2 e 3, o MPC executa a mudança de referência antecipadamente. O mesmo não acontece para a variável 1 porque a referência sofreu alteração no primeiro instante da simulação.

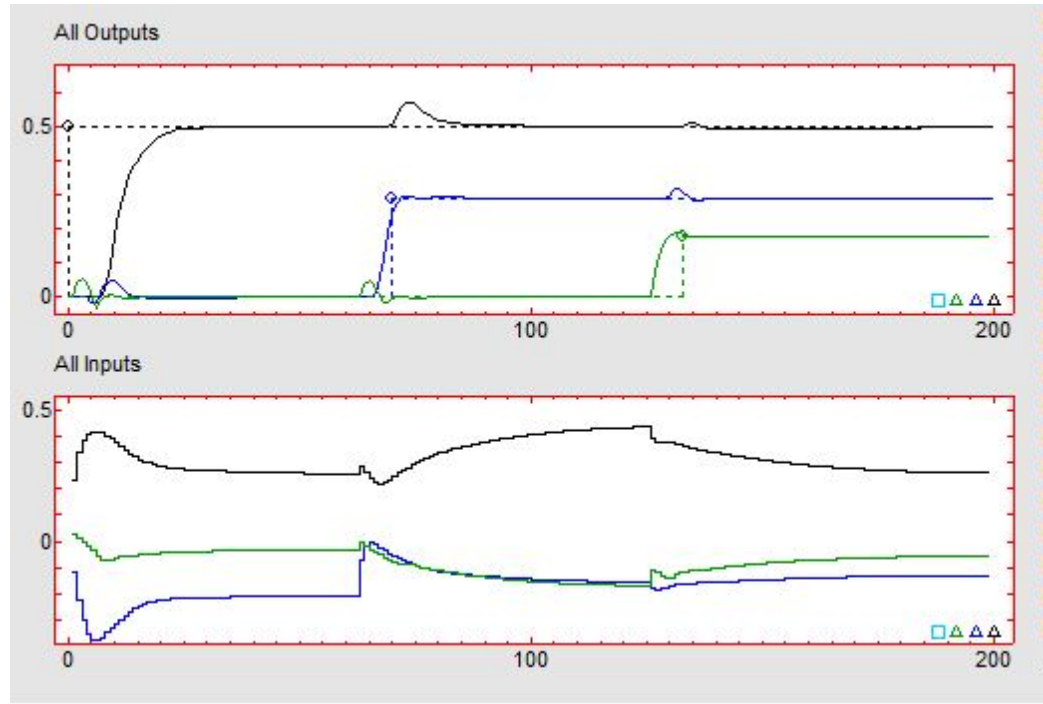


Figura 2.2: Respostas do fracionador de óleo pesado para o caso sem restrições.

Para mostrar o potencial do MPC, uma segunda simulação inclui restrições nas variáveis controladas. Vamos supor que não se deseja que a variável  $y_1$  se afaste mais do que 0,5 do ponto de operação, mas que um operador envia um comando de mudança de referência de valor 0,7 (as outras mudanças de referência são mantidas como na simulação anterior). Na Figura 2.3 é mostrado o resultado desta simulação. Observe que o controlador MPC não permite que a variável  $y_1$  viole a restrição imposta quando a mudança de referência é aplicada, e tenta manter, durante todo o tempo, a variável na restrição imposta, garantindo, assim, o correto funcionamento do processo. Como podemos concluir deste estudo de caso, o MPC é extremamente flexível e permite atender diversas especificações. Estudaremos com mais detalhes

estes aspectos nos próximos capítulos.

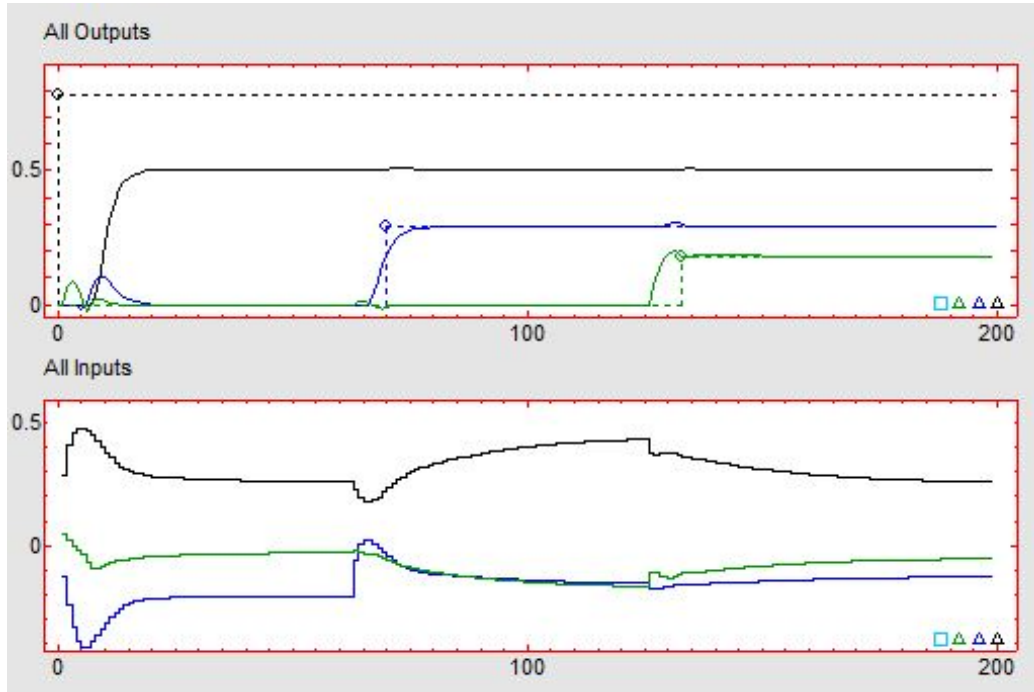


Figura 2.3: Respostas do fracionador de óleo pesado para o caso com restrições.

## 2.3 Comentários finais

Este capítulo apresentou as principais ideias e notação usada no MPC SISO, que foram desenvolvidas com detalhe no Volume 1, e que servirão de base para a apresentação do MPC MIMO e não linear dos próximos capítulos. Também apresentamos um exemplo para motivar o estudo do MPC MIMO, mostrando a potencialidade da técnica.

## 2.4 Exercícios propostos

1. Utilizando a ferramenta de simulação para GPC MIMO apresentada no Apêndice, escolha o processo  $3 \times 3$  da ferramenta. Ajuste o controle GPC sem restrições de forma intuitiva usando a análise do efeito dos parâmetros de sintonia apresentados no capítulo. Mude os parâmetros

e observe o efeito nas respostas. Inclua restrições no problema e analise os resultados, considerando restrições apenas em uma das manipuladas. Mude a restrição para as outras manipuladas e finalmente considere restrições em todas elas de forma simultânea. Observe a flexibilidade no ajuste do sistema de controle que o MPC proporciona.

## Capítulo 3

# DMC e GPC Multivariáveis

Na prática, é comum que se necessite controlar simultaneamente mais de uma variável de um dado processo utilizando, para isso, mais de uma variável manipulada. Se cada variável manipulada apenas afeta uma controlada, então o projeto do controle pode ser realizado usando vários controladores monovariáveis independentes. Porém, as variáveis desses processos usualmente estão acopladas, ou seja, a variação de uma variável manipulada pode causar efeitos em mais de uma controlada simultaneamente. Nesse caso, o uso de vários controladores independentes pode não ser adequado e, tipicamente, faz mais sentido considerar o processo como um sistema multivariável, que tem múltiplas variáveis manipuladas e múltiplas controladas sendo consideradas como um conjunto, e não como a união de vários sistemas monovariáveis.

Os controladores monovariáveis, como os PIDs, que têm muito sucesso no controle de sistemas monovariáveis (controle local), em geral, têm dificuldades para lidar com essa interação de forma adequada. Por isso, existem técnicas para melhorar o desempenho dos PIDs no caso multivariável, tal qual o uso de desacopladores, mas a complexidade do projeto do sistema de controle aumenta consideravelmente quando o sistema é de dimensão elevada. Além disso, nem sempre é possível desacoplar completamente a resposta de um processo multivariável e, em muitos casos, desacopladores introduzem ganhos elevados em altas frequências, que acabam levando a outros problemas do ponto de vista prático, como a amplificação de ruídos. Nesses casos, técnicas avançadas como a de controle preditivo permitem, em geral, obter melhores resultados que os obtidos com controladores PIDs, dado que, além de serem mais interessantes para tratar o problema multivariável, podem incluir diversos objetivos e restrições no problema. Cabe aqui lembrar que o MPC teve sua origem no controle de plantas multivariáveis no setor industrial petroquímico [CR80, RRJP76].

Outro aspecto importante que deve ser destacado dos sistemas multi-

variáveis é que, dependendo das características do processo, nem sempre é possível que o sistema alcance um ponto de operação arbitrariamente definido pelo usuário. Nos sistemas de controle monovariáveis estudados no Volume 1, mostramos que sob estabilidade em malha fechada e com um controle que possui ação integral, é possível levar a saída do sistema para uma referência desejada constante com erro zero contanto que a ação de controle esteja dentro da faixa de atuação do atuador. No caso de sistemas multivariáveis, para que esta propriedade seja verdadeira, precisamos adicionar uma condição extra: ter graus de liberdade suficientes no controle para chegar nas referências desejadas. Por exemplo, se consideramos um sistema linear estável de 2 saídas e 1 entrada, que tem modelo estático dado por:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} K_{e1} \\ K_{e2} \end{bmatrix} u$$

não é possível encontrar um valor de  $u$  capaz de levar  $y_1$  a uma referência  $r_1$  e  $y_2$  a uma referência  $r_2$  simultaneamente, a menos que  $r_1/r_2 = K_{e1}/K_{e2}$ . De forma geral, para um sistema multivariável estável com modelo estático

$$\mathbf{y} = \mathbf{K}_e \mathbf{u},$$

uma dada referência vetorial  $\mathbf{r}$  somente será alcançável se existe um controle  $\mathbf{u}$  tal que  $\mathbf{r} = \mathbf{K}_e \mathbf{u}$ . Assim, principalmente no caso de plantas com menor número de variáveis manipuladas que controladas, a referência passada ao controle tem que ser factível quando se deseja erro de seguimento nulo. Uma outra alternativa é que o objetivo de controle seja buscar o ponto de equilíbrio mais próximo da referência escolhida. Veja que em alguns problemas pode ser requerido, por exemplo, que uma ou mais variáveis atinjam uma referência enquanto outras apenas sejam mantidas em faixas preestabelecidas de segurança.

Considerando a discussão apresentada, as técnicas de controle preditivo são capazes de lidar muito facilmente com sistemas multivariáveis, pois: (i) os conceitos utilizados na formulação do controle não dependem da dimensão do processo; (ii) podem utilizar o modelo completo do processo explicitamente, assim, as interações entre as variáveis ficam refletidas nas predições; (iii) podem ser usadas com as mais diversas funções custo, o que dá muita flexibilidade ao controlador para, inclusive, considerar problemas como o de chegar a um ponto de operação próximo de uma referência não factível; e (iv) em termos de sintonia, a influência dos parâmetros de ajuste do controle na resposta do sistema é similar àquela observada no caso monovariável, apesar de que a quantidade de parâmetros para ajustar cresce linearmente com a quantidade de entradas e de saídas.

Neste capítulo, serão apresentadas as versões multivariáveis dos algoritmos DMC e GPC. Como as diferenças entre estas formulações para uma dada função custo são as mesmas que no caso monovariável, isto é, são dadas pela forma como se calcula a resposta livre do sistema, iniciamos a apresentação deste capítulo mostrando como obter a lei de controle no caso multivariável para depois analisar cada uma das particularidades dos algoritmos no que se refere ao cálculo das predições e da resposta livre. Posteriormente, no Capítulo 4, estudaremos a formulação do MPC para modelos dados por equações de estado, tratando os casos monovariável e multivariável no mesmo capítulo.

Antes de iniciar com aspectos mais formais, vamos introduzir um exemplo de um sistema multivariável simples, para ilustrar a relação entre as entradas e as saídas do processo, e como as predições podem ser calculadas.

**Exemplo 3.1.** *Considere um sistema de controle de vazão e temperatura de água em uma tubulação de um processo industrial. A água com temperatura controlada é obtida pela mistura de duas vazões de água com temperaturas diferentes, uma quente e outra fria. Duas bombas são usadas, uma para a linha de água quente e outra para a linha de água fria, e a mistura se realiza por uma conexão tipo Y. Transdutores de temperatura e vazão são adequadamente instalados na seção final da tubulação, onde a água será usada. Neste sistema, pode-se atuar no sinal de controle das duas bombas. Logicamente, neste processo, a temperatura e a vazão são duas variáveis acopladas fisicamente, dado que uma variação em qualquer um dos atuadores trará uma variação de temperatura e de vazão na água entregue ao próximo processo. Assim, este processo pode ser considerado multivariável. Um modelo simplificado deste processo, funcionando nas vizinhanças de um ponto de operação, foi construído usando equações diferenciais, obtendo:*

$$20 \frac{dT(t)}{dt} + T(t) = 5u_q(t) - 3u_f(t),$$

$$5 \frac{dV(t)}{dt} + V(t) = 1,5u_q(t) + u_f(t),$$

*considerando  $u_q$  e  $u_f$ , respectivamente, os sinais de controle das bombas de água quente e fria, e  $T(t)$  e  $V(t)$ , respectivamente, a temperatura e a vazão da água no ponto de medição. Os tempos no modelo estão em segundos.*

*Um modelo em tempo discreto deste processo, obtido usando um sustentador de ordem zero e um tempo de amostragem de 1 segundo, é dado por:*

$$T(k) = 0,95T(k-1) + 0,25u_q(k-1) - 0,15u_f(k-1),$$

$$V(k) = 0,82V(k-1) + 0,27u_q(k-1) + 0,18u_f(k-1).$$

A partir deste modelo, podemos inferir que os valores futuros de temperatura e de vazão serão afetados pelos valores passados dessas variáveis e pelos valores passados e futuros das variáveis manipuladas. Um modelo de resposta livre e forçada poderia ser obtido de forma simples como no estudo dos controles preditivos monovariáveis apresentados no Volume 1 e discutido no Capítulo 2, apenas usando o princípio de superposição. Por exemplo, as respostas forçadas de  $T$  e  $V$  poderiam ser obtidas para a variável manipulada  $u_f$  considerando  $u_q = 0$  e depois para  $u_q$  considerando  $u_f = 0$ . Para cada saída, os resultados podem então ser somados com a resposta livre do sistema, obtida com todos os controles futuros nulos. Finalmente, podemos organizar os resultados obtidos de forma vetorial para ter uma visão da forma multivariável de calcular as previsões da saída deste processo.

Vamos supor que escolhemos um horizonte de predição de 3 amostras para  $T$ , de 2 amostras para  $V$  e um horizonte de controle de 2 amostras para  $u_q$  e  $u_f$ . Assim, usando os conceitos vistos no caso monovariável para as previsões de temperatura,  $\hat{T}$ , poderíamos calcular:

$$\begin{bmatrix} \hat{T}(k+1 | k) \\ \hat{T}(k+2 | k) \\ \hat{T}(k+3 | k) \end{bmatrix} = \begin{bmatrix} 0,250 & 0 \\ 0,490 & 0,250 \\ 0,715 & 0,490 \end{bmatrix} \begin{bmatrix} \Delta u_q(k) \\ \Delta u_q(k+1) \end{bmatrix} \\ + \begin{bmatrix} -0,150 & 0 \\ -0,294 & -0,150 \\ -0,429 & -0,294 \end{bmatrix} \begin{bmatrix} \Delta u_f(k) \\ \Delta u_f(k+1) \end{bmatrix} + \begin{bmatrix} f_T(k+1) \\ f_T(k+2) \\ f_T(k+3) \end{bmatrix},$$

sendo  $f_T$  os valores da resposta livre de temperatura.

Já para as previsões de vazão,  $\hat{V}$ , poderíamos calcular:

$$\begin{bmatrix} \hat{V}(k+1 | k) \\ \hat{V}(k+2 | k) \end{bmatrix} = \begin{bmatrix} 0,270 & 0 \\ 0,409 & 0,270 \end{bmatrix} \begin{bmatrix} \Delta u_q(k) \\ \Delta u_q(k+1) \end{bmatrix} \\ + \begin{bmatrix} 0,180 & 0 \\ 0,327 & 0,180 \end{bmatrix} \begin{bmatrix} \Delta u_f(k) \\ \Delta u_f(k+1) \end{bmatrix} + \begin{bmatrix} f_V(k+1) \\ f_V(k+2) \end{bmatrix},$$

sendo  $f_V$  os valores da resposta livre de vazão.

Podemos, agora, agrupar as previsões em um vetor  $\hat{\mathbf{y}}$ , os incrementos de controle em um segundo vetor  $\Delta \mathbf{u}$ , e as respostas livres em um terceiro vetor  $\mathbf{f}$ , definidos como:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{T}(k+1 | k) \\ \hat{T}(k+2 | k) \\ \hat{T}(k+3 | k) \\ \hat{V}(k+1 | k) \\ \hat{V}(k+2 | k) \end{bmatrix}, \quad \Delta \mathbf{u} = \begin{bmatrix} \Delta u_q(k) \\ \Delta u_q(k+1) \\ \Delta u_f(k) \\ \Delta u_f(k+1) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_T(k+1) \\ f_T(k+2) \\ f_T(k+3) \\ f_V(k+1) \\ f_V(k+2) \end{bmatrix}.$$



Finalmente, é possível escrever:

$$\hat{\mathbf{y}} = \begin{bmatrix} 0,250 & 0 & -0,150 & 0 \\ 0,490 & 0,250 & -0,294 & -0,150 \\ 0,715 & 0,490 & -0,429 & -0,294 \\ 0,270 & 0 & 0,180 & 0 \\ 0,409 & 0,270 & 0,327 & 0,180 \end{bmatrix} \Delta \mathbf{u} + \mathbf{f}.$$

Assim, observamos que a forma de escrever as predições do sistema segue a mesma lógica do caso monovariável.

Como ilustramos no exemplo anterior, conceitualmente, a forma de escrever as predições de um modelo linear usando:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta \mathbf{u} + \mathbf{f} \quad (3.1)$$

não é afetada pelo fato de o sistema ser mono ou multivariável. Nesta equação, se consideramos um sistema multivariável,  $\hat{\mathbf{y}}$  representará o vetor de predições de todas as saídas do processo organizadas de forma sequencial, da primeira até a última saída, e  $\Delta \mathbf{u}$  o vetor dos incrementos de controle futuros de todas as variáveis manipuladas, também organizadas de forma sequencial, da primeira até a última variável manipulada. Observe que quando os incrementos do controle futuros são nulos, então  $\hat{\mathbf{y}} = \mathbf{f}$ , como acontece no caso monovariável. Além disso, se consideramos todo o passado nulo, as predições apenas dependem do controle futuro. Assim, é esperado que a obtenção do sinal de controle a partir da minimização da função custo seja feita de forma semelhante ao que ocorre no caso monovariável.

Na sequência deste capítulo, vamos considerar processos multivariáveis com  $m$  entradas de controle (ou variáveis manipuladas) e  $n$  saídas (ou variáveis controladas). Assim, poderemos tratar tanto processos quadrados, com  $m = n$ , ou retangulares,  $m \neq n$ . No caso de considerar as perturbações medidas no modelo, isto é, para o uso da ação antecipativa do MPC, consideraremos um número de perturbações medidas  $m_q$ .

### 3.1 Obtenção da lei de controle

A função custo clássica<sup>1</sup> do caso multivariável é muito similar àquela utilizada no caso monovariável mostrada no Capítulo 2 em (2.2). O que muda é o fato de que agora são levados em conta os erros futuros preditos de todas as  $n$

---

<sup>1</sup>Estudaremos funções de custo mais gerais no Capítulo ??.

saídas e também o esforço de controle de todas as  $m$  entradas:

$$J = \sum_{l=1}^n \sum_{i=N_{1,l}}^{N_{2,l}} \delta_l(i) (r_l(k+i) - \hat{y}_l(k+i | k))^2 + \sum_{p=1}^m \sum_{i=0}^{N_{u,p}-1} \lambda_p(i) \Delta u_p(k+i)^2, \quad (3.2)$$

onde  $r_l$ ,  $N_{1,l}$ ,  $N_{2,l}$ ,  $\delta_l(i)$  são a referência, o horizonte inicial de predição, o horizonte final de predição, e as ponderações, respectivamente, da saída  $l$ , e  $N_{u,p}$  e  $\lambda_p(i)$  são, respectivamente, o horizonte de controle e as ponderações do esforço de controle da entrada  $p$ . Assim como no caso monovariável, usualmente utilizam-se ponderações constantes ao longo do horizonte, ou seja,  $\delta_l(i) = \delta_l$  e  $\lambda_p(i) = \lambda_p$ ,  $\forall i, l, p$ . No caso monovariável é comum fixar uma das ponderações com valor unitário (tipicamente  $\delta = 1$ ) e ajustar apenas a outra. No caso multivariável, note que ainda é possível fixar uma das ponderações como sendo unitária, porém, isso não pode ser feito para todos os pesos do erro de seguimento de referência, pois é justamente por meio de diferentes ponderações que se torna possível priorizar o seguimento de referência de uma saída em detrimento das outras. Assim, no caso multivariável, além da priorização entre seguimento rápido de referência e suavidade no sinal de controle, que já ocorre no caso monovariável, é importante definir a importância relativa entre as saídas e entre cada uma das saídas e cada um dos esforços de controle. Note que o número de parâmetros de sintonia do controlador no caso multivariável cresce consideravelmente, visto que cada saída e cada entrada tem seu conjunto de horizontes e ponderações, cuja alteração modificará o controlador resultante e, por consequência, a resposta de malha fechada. Vamos ilustrar este ponto com um exemplo de escolha de parâmetros de ajuste de um MPC no caso multivariável.

**Exemplo 3.2.** *Considere o sistema de controle de uma coluna de destilação multi-produto, onde se deseja controlar a temperatura de extração de quatro produtos diferentes. Nesta coluna, o erro de seguimento de temperatura em cada uma das extrações de produto é inversamente proporcional à qualidade do produto extraído. Assim, manter o erro abaixo de um dado valor implica manter a qualidade do produto acima de uma certa especificação. Ainda, considere que os produtos 1 e 3 têm valor comercial muito maior que os produtos 2 e 4. Por outro lado, as dinâmicas das temperaturas associadas aos produtos 1 e 2 são mais lentas que as das temperaturas associadas aos produtos 3 e 4, e ainda possuem um atraso de transporte de 5 amostras. Finalmente, entre as variáveis manipuladas do processo  $\Delta u_p$ ,  $p = 1 \dots 4$  (consideramos aqui um processo quadrado),  $\Delta u_1$  é a que tem maior impacto*

econômico, por se tratar da regulação da alimentação de combustível na base da coluna. Suponha, ainda, que todas as temperaturas estão normalizadas para a mesma faixa de variação de 0% a 100% e que todas as variáveis manipuladas estão também entre 0% e 100%.

Neste problema simples, de dimensão  $4 \times 4$ , precisamos ajustar 20 parâmetros, sendo 12 referentes aos horizontes e 8 a ponderações. Como discutido anteriormente, até seria possível fixar uma das ponderações a um valor unitário, mas ainda restariam 7 ponderações para serem ajustadas, além dos parâmetros referentes aos horizontes. Seguindo as indicações práticas de escolha de parâmetros já discutidas no caso SISO, os horizontes das variáveis controladas 1 e 3 serão escolhidos como  $N_{1,1} = N_{1,3} = 5 + 1 = 6$  por causa do atraso (não faz sentido colocar na função objetivo um termo que não depende dos sinais de controle a serem calculados) e com  $N_{2,1}$  e  $N_{2,3}$  compatíveis com a dinâmica lenta dessas variáveis (usamos uma janela que capture o regime transitório das variáveis). Como as variáveis 2 e 4 não têm atraso, utilizamos  $N_{1,2} = N_{1,4} = 1$  e  $N_{2,2}$  e  $N_{2,4}$  menores que os das outras variáveis controladas, dado que são mais rápidas que aquelas. Os horizontes de controle podem ser escolhidos inicialmente como um terço da janela média de predição das controladas e depois um ajuste fino pode ser feito.

Com as considerações sobre a importância das variáveis para o processo, o ajuste do controle multivariável deve considerar  $\delta_1$  e  $\delta_3$  com valores maiores que  $\delta_2$  e  $\delta_4$ , para refletir a importância dos produtos 1 e 3 com relação aos produtos 2 e 4. Para considerar corretamente o peso na função custo, é interessante também considerar um ajuste que leve em conta o tamanho do horizonte de predição.

Com relação às manipuladas, as ponderações de controle usarão  $\lambda_1$  muito maior que  $\lambda_i$ ,  $i = 2 \dots 4$  para tentar minimizar o consumo de combustível, que é a variável que mais impacta o custo de operação. Novamente aqui, caso os horizontes de controle sejam diferentes para cada variável manipulada, podemos calcular os pesos para levar em consideração o efeito na função custo a ser minimizada.

E, por fim, vale ressaltar que o valor relativo entre as ponderações da saída e do esforço de controle também influencia o resultado final. Se, por exemplo, as ponderações das saídas forem muito maiores do que as dos controles a resposta tende a ser mais agressiva, um comportamento idêntico ao caso monovariável (veja por exemplo os casos simulados na ferramenta interativa do Capítulo 2 do Volume 1).

No exemplo anterior, consideramos um caso simples no qual todas as variáveis controladas e manipuladas estavam no mesmo intervalo de operação. Isso nem sempre acontece na prática, onde, por exemplo, uma temperatura

pode ter uma faixa de variação com valores entre  $100^\circ\text{C}$  e  $200^\circ\text{C}$  e uma concentração entre  $1\text{ g mol}^{-1}$  e  $2\text{ g mol}^{-1}$ . Neste caso, o erro de temperatura terá valores da ordem de 100 vezes do valor do erro de concentração. Por exemplo, um erro de 1% de temperatura será, em valores absolutos, 100 vezes maior que um erro de 1% na concentração. Desta forma, o peso na função custo do erro na concentração será muito menor que o do erro na temperatura, quando na prática, podem, por exemplo, ter a mesma importância para o processo. Esta situação pode ser corrigida utilizando um modelo de processo com variáveis normalizadas, no qual, por exemplo, cada variável seja dividida pelo seu valor máximo, ficando então normalizada a um valor máximo de 1. Se todas as variáveis controladas e manipuladas têm o mesmo horizonte, esta modificação resolve o problema. Entretanto, se os horizontes de predição são diferentes, então também precisamos ajustar as ponderações colocando um fator inversamente proporcional a cada horizonte em cada termo.

De forma geral, se uma variável controlada  $y_l$  tem valor máximo  $y_{l,max}$  e horizonte de predição  $N_{2,l} - N_{1,l} + 1$ , podemos escolher uma normalização do peso  $\delta_l$  como:

$$\delta_l = \delta_{ln} \frac{1}{(y_{l,max})^2 (N_{2,l} - N_{1,l} + 1)},$$

com  $\delta_{ln}$  sendo o peso normalizado a ser usado na sintonia.

Da mesma forma, se uma variável manipulada  $\Delta u_p$  tem valor máximo  $\Delta u_{p,max}$  e horizonte de controle  $N_u$ , podemos escolher uma normalização do peso  $\lambda_p$  como:

$$\lambda_p = \lambda_{pn} \frac{1}{(\Delta u_{p,max})^2 N_u},$$

com  $\lambda_{pn}$  sendo o peso normalizado a ser usado na sintonia. Vejamos um exemplo deste ajuste.

**Exemplo 3.3.** Considere um processo com 2 variáveis controladas,  $y_1$  e  $y_2$ , e 2 variáveis manipuladas,  $u_1$  e  $u_2$ .  $y_1$  tem horizonte de 20 amostras e valores no intervalo  $[0, 100]$ ,  $y_2$  tem horizonte de 10 amostras e valores no intervalo  $[0, 10]$ ,  $\Delta u_1$  tem horizonte de 5 amostras e valores no intervalo  $[-1, 1]$  e  $\Delta u_2$  tem horizonte de 3 amostras e valores no intervalo  $[-5, 5]$ . Se queremos que uma variação em quaisquer uma das variáveis tenha o mesmo peso na função custo, precisamos trabalhar com pesos normalizados pelo máximo de cada variável ao quadrado e pelo horizonte, como:

$$\delta_1 = \delta_{1n} \frac{1}{100^2} \frac{1}{20} = \delta_{1n} \frac{1}{2 \cdot 10^5}$$

$$\delta_2 = \delta_{2n} \frac{1}{10^2} \frac{1}{10} = \delta_{2n} \frac{1}{10^3}$$

$$\lambda_1 = \lambda_{1n} \frac{1}{1^2} \frac{1}{5} = \lambda_{1n} \frac{1}{5}$$

$$\lambda_2 = \lambda_{2n} \frac{1}{5^2} \frac{1}{3} = \lambda_{2n} \frac{1}{75}$$

Assim, se os pesos normalizados,  $\delta_{in}$  e  $\lambda_{in}$ , com  $i = 1, 2$ , forem todos iguais e aplicarmos os pesos resultantes,  $\delta_i$  e  $\lambda_i$ , com  $i = 1, 2$ , na função custo, todas as variáveis terão contribuições iguais na função custo. Da mesma forma, quando um peso normalizado for, por exemplo, o dobro do outro, o termo correspondente ao primeiro peso estará contribuindo duas vezes mais que o segundo termo na função custo.

Estes exemplos mostram como, apesar de intuitiva, a definição dos horizontes e dos pesos pode ser um problema complexo em sistemas com grandes dimensões. Além disso, o último exemplo ilustra que a normalização das variáveis e pesos pode simplificar o problema. Vamos agora desenvolver a forma da função custo multivariável na notação matricial, assim como foi feito para o caso monovariável no Volume 1, e mostrado no Capítulo 2, na Equação 2.3.

É possível escrever a função custo na forma matricial assumindo que o vetor de predições futuras é organizado da forma padrão apresentada em (3.1). Neste caso, a função custo pode ser escrita como:

$$J = [\mathbf{r} - \hat{\mathbf{y}}]^T \mathbf{Q}_y [\mathbf{r} - \hat{\mathbf{y}}] + \Delta \mathbf{u}^T \mathbf{Q}_u \Delta \mathbf{u} \quad (3.3)$$

com

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_n \end{bmatrix}, \quad \mathbf{r}_l = \begin{bmatrix} r_l(k + N_{1,l}) \\ r_l(k + N_{1,l} + 1) \\ \vdots \\ r_l(k + N_{2,l}) \end{bmatrix}, \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \\ \hat{\mathbf{y}}_n \end{bmatrix},$$

$$\hat{\mathbf{y}}_l = \begin{bmatrix} \hat{y}_l(k + N_{1,l} | k) \\ \hat{y}_l(k + N_{1,l} + 1 | k) \\ \vdots \\ \hat{y}_l(k + N_{2,l} | k) \end{bmatrix}, \quad \Delta \mathbf{u} = \begin{bmatrix} \Delta \mathbf{u}_1 \\ \Delta \mathbf{u}_2 \\ \vdots \\ \Delta \mathbf{u}_m \end{bmatrix}, \quad \Delta \mathbf{u}_p = \begin{bmatrix} \Delta u_p(k) \\ \Delta u_p(k + 1) \\ \vdots \\ \Delta u_p(k + N_{u,p} - 1) \end{bmatrix}$$

$$\mathbf{Q}_y = \begin{bmatrix} \delta_1 \mathbf{I}_{N_{2,1}-N_{1,1}+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \delta_2 \mathbf{I}_{N_{2,2}-N_{1,2}+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \delta_n \mathbf{I}_{N_{2,n}-N_{1,n}+1} \end{bmatrix},$$

$$\mathbf{Q}_u = \begin{bmatrix} \lambda_1 \mathbf{I}_{N_{u,1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \lambda_2 \mathbf{I}_{N_{u,2}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \lambda_m \mathbf{I}_{N_{u,m}} \end{bmatrix}.$$

Nesta formulação, os vetores de predição de cada uma das saídas são empilhados no vetor  $\hat{\mathbf{y}}$ , que tem dimensão  $\sum_{l=1}^n (N_{2,l} - N_{1,l} + 1)$ , e os controles são empilhados no vetor  $\Delta \mathbf{u}$ , que tem dimensão  $\sum_{p=1}^m N_{u,p}$ .

A função custo  $J$  no caso multivariável, em sua forma matricial, é quase idêntica àquela obtida no caso monovariável, apresentada em 2.3. As alterações se dão apenas nas definições das matrizes e vetores, e no uso de  $\mathbf{Q}_y$  e  $\mathbf{Q}_u$ , que são matrizes diagonais com as ponderações. Note que na formulação apresentada, foram empregadas ponderações constantes ao longo dos horizontes para cada variável, como forma de simplificar a notação das matrizes  $\mathbf{Q}_y$  e  $\mathbf{Q}_u$ , porém, é possível empregar valores diferentes para cada posição do horizonte, definindo matrizes diagonais com essas ponderações no lugar da ponderação constante multiplicada pela matriz identidade. Na definição das matrizes de ponderação, a notação  $\mathbf{0}$  é usada para designar matrizes com todos os elementos nulos, de dimensão apropriada, como forma de não complicar a notação.

Como o formato final da função custo é semelhante ao já apresentado no caso monovariável, é possível reescrever a função custo no formato padrão de programação quadrática (do inglês, *Quadratic Programming* - QP):

$$J = \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H}_{qp} \Delta \mathbf{u} + \mathbf{f}_{qp}^T \Delta \mathbf{u} + c_{qp}, \quad (3.4)$$

com

$$\begin{aligned} \mathbf{H}_{qp} &= 2 (\mathbf{G}^T \mathbf{Q}_y \mathbf{G} + \mathbf{Q}_u) \\ \mathbf{f}_{qp} &= -2 \mathbf{G}^T \mathbf{Q}_y [\mathbf{r} - \mathbf{f}] \\ c_{qp} &= [\mathbf{r} - \mathbf{f}]^T \mathbf{Q}_y [\mathbf{r} - \mathbf{f}]. \end{aligned}$$

Para chegar em 3.4 basta substituir a expressão de  $\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f}$  na função custo e manipular <sup>2</sup>. Da mesma forma que com a função custo e as predições, para as restrições seguimos o mesmo procedimento do caso monovariável, considerando cada restrição associada a cada variável manipulada e transformando-a em uma expressão vetorial. Finalmente, a equação que representa todas as restrições de desigualdade do problema pode ser colocada como:

$$\bar{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}.$$

---

<sup>2</sup>Os detalhes desta manipulação se mostram no Capítulo 3 do Volume 1.

Vamos ilustrar a montagem das matrizes  $\bar{\mathbf{R}}$  e  $\bar{\mathbf{r}}$  para o caso estudado no exemplo anterior por meio de um novo exemplo.

**Exemplo 3.4.** *Considere o mesmo caso do Exemplo 3.3, um processo multivariável  $2 \times 2$ , considerando que as ações de controle estão restritas ao intervalo  $[0, 10]$  e os incrementos das ações de controle estão restritos ao intervalo  $[-1, 1]$ . Como os horizontes de controle são  $N_{u,1} = 5$  e  $N_{u,2} = 3$ , teremos:*

$$0 \leq u_1(k+j) \leq 10, \quad -1 \leq \Delta u_1(k+j) \leq 1 \quad j = 1, \dots, 5$$

$$0 \leq u_2(k+j) \leq 10, \quad -1 \leq \Delta u_2(k+j) \leq 1 \quad j = 1, \dots, 3.$$

*O vetor de incremento de ações de controle a ser usado pelo otimizador é:*

$$\Delta \mathbf{u} = [\Delta u_1(k), \Delta u_1(k+1), \dots, \Delta u_1(k+4), \Delta u_2(k), \Delta u_2(k+1), \Delta u_2(k+2)]^T,$$

*que tem dimensão  $8 \times 1$ . Assim, vamos escrever as restrições em função deste vetor. Para as restrições no incremento, podemos escrever:*

$$-\mathbf{1}_8 \leq \Delta \mathbf{u} \leq \mathbf{1}_8,$$

*ou*

$$\mathbf{I}_8 \Delta \mathbf{u} \leq \mathbf{1}_8,$$

$$-\mathbf{I}_8 \Delta \mathbf{u} \leq \mathbf{1}_8.$$

*Para as restrições no controle podemos usar que:*

$$\mathbf{u}_1 = \mathbf{T}_5 \Delta \mathbf{u}_1 + \mathbf{1}_5 u_1(k-1),$$

$$\mathbf{u}_2 = \mathbf{T}_3 \Delta \mathbf{u}_2 + \mathbf{1}_3 u_2(k-1),$$

*sendo  $\mathbf{T}_5$  e  $\mathbf{T}_3$  as matrizes quadradas triangulares inferiores com elementos não nulos todos iguais a um, de dimensão 5 e 3, respectivamente. Assim, podemos escrever:*

$$\mathbf{0}_5 \leq \mathbf{T}_5 \Delta \mathbf{u}_1 + \mathbf{1}_5 u_1(k-1) \leq 10 \mathbf{1}_5$$

$$\mathbf{0}_3 \leq \mathbf{T}_3 \Delta \mathbf{u}_2 + \mathbf{1}_3 u_2(k-1) \leq 10 \mathbf{1}_3$$

*ou*

$$\mathbf{T}_5 \Delta \mathbf{u}_1 \leq 10 \mathbf{1}_5 - \mathbf{1}_5 u_1(k-1) = \mathbf{1}_5 (10 - u_1(k-1))$$

$$\mathbf{T}_3 \Delta \mathbf{u}_2 \leq 10 \mathbf{1}_3 - \mathbf{1}_3 u_2(k-1) = \mathbf{1}_3 (10 - u_2(k-1))$$

$$-\mathbf{T}_5 \Delta \mathbf{u}_1 \leq \mathbf{1}_5 u_1(k-1)$$

$$-\mathbf{T}_3 \Delta \mathbf{u}_2 \leq \mathbf{1}_3 u_2(k-1)$$

Agrupando agora todas as restrições temos como colocar a equação na forma  $\overline{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}$ :

$$\begin{bmatrix} \mathbf{I}_5 & 0 \\ 0 & \mathbf{I}_3 \\ -\mathbf{I}_5 & 0 \\ 0 & -\mathbf{I}_3 \\ \mathbf{T}_5 & 0 \\ 0 & \mathbf{T}_3 \\ -\mathbf{T}_5 & 0 \\ 0 & -\mathbf{T}_3 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_1 \\ \Delta \mathbf{u}_2 \end{bmatrix} \leq \begin{bmatrix} \mathbf{1}_5 \\ \mathbf{1}_3 \\ \mathbf{1}_5 \\ \mathbf{1}_3 \\ \mathbf{1}_5(10 - u_1(k-1)) \\ \mathbf{1}_3(10 - u_2(k-1)) \\ \mathbf{1}_5 u_1(k-1) \\ \mathbf{1}_3 u_2(k-1) \end{bmatrix},$$

obtendo as matrizes  $\overline{\mathbf{R}}$  e  $\bar{\mathbf{r}}$  usadas no otimizador.

A minimização da função custo anterior considerando as restrições, como explicado no caso monovariável no Volume 1, nos dá o vetor de incrementos futuros a ser aplicado no processo a cada instante de amostragem, que no caso sem restrições se reduz a  $\Delta \mathbf{u} = \mathbf{K} [\mathbf{r} - \mathbf{f}]$  com  $\mathbf{K} = (\mathbf{G}^T \mathbf{Q}_y \mathbf{G} + \mathbf{Q}_u)^{-1} \mathbf{G}^T \mathbf{Q}_y$ .

Note que  $\Delta \mathbf{u}$  define um vetor de incrementos futuros de controle, porém, apenas os  $\Delta u_p(k)$  com  $p = 1 \dots m$  são aplicados ao processo no instante  $k$ , pois  $\Delta \mathbf{u}$  deve ser recalculado a cada instante de amostragem, seguindo o princípio de horizonte deslizante. Assim, é necessário obter apenas o valor de  $\Delta u_p(k)$  ótimo, dado pelo vetor de incrementos de controle para o instante atual, definido como:

$$\Delta \mathbf{u}_a = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \vdots \\ \Delta u_p(k) \end{bmatrix}, \quad (3.5)$$

com elementos iguais aos primeiros elementos de cada um dos sub-vetores do vetor  $\Delta \mathbf{u}$ .

Por fim, as ações de controle a serem aplicadas no processo são:

$$u_p(k) = u_p(k-1) + \Delta u_p(k), \quad \forall p = 1 \dots m. \quad (3.6)$$

Agora que temos a forma da lei de controle estabelecida para o caso multivariável, podemos retomar a análise do funcionamento do sistema em malha fechada para o caso de referências constantes. Assim como no caso monovariável, por ter incluído um modelo de controle incremental e um modelo de perturbação de grau nas predições, sabemos que, assumindo a estabilidade da malha fechada, a resposta livre converge para a saída do processo e o



sistema terá um controle  $\Delta \mathbf{u} = \mathbf{0}$ , o que significa que  $\mathbf{K}(\mathbf{r} - \mathbf{f}) = \mathbf{0}$ . Mas esta igualdade não garante que, em regime permanente, os valores de cada uma das respostas livres  $\mathbf{f}_{l,ss}$  alcancem a correspondente referência  $\mathbf{r}_l$ , dado que a condição imposta em regime permanente é:

$$\mathbf{K}(\mathbf{r} - \mathbf{f}_{ss}) = \sum_{l=1}^n \mathbf{K}_l(\mathbf{r}_l - \mathbf{f}_{l,ss}) = \sum_{l=1}^n \mathbf{K}_l \mathbf{e}_{l,ss} = \mathbf{0},$$

na qual o vetor de ganho foi escrito como  $\mathbf{K} = [\mathbf{K}_1 \ \mathbf{K}_2 \ \dots \ \mathbf{K}_n]$ . Observe que a equação anterior pode ser satisfeita com valores de erro estático  $\mathbf{e}_{l,ss}$  não nulos, bastando que uma combinação linear de erros que satisfaça a equação seja encontrada.

Uma forma mais direta de visualizar este problema é analisar a característica estática do sistema, assumindo um processo estável de ganho estático  $\mathbf{K}_e$ . Temos que  $\mathbf{y} = \mathbf{K}_e \mathbf{u}$ , e desejamos que  $\mathbf{y} = \mathbf{r}$ , então,

$$\mathbf{r} = \mathbf{K}_e \mathbf{u}.$$

Podemos interpretar essa equação como um sistema linear de equações em que a incógnita é o vetor de controle  $\mathbf{u}$ . Se o sistema for quadrado,  $n = m$ , então temos um número de equações iguais ao número de incógnitas e, se o posto de  $\mathbf{K}_e$  for completo (todas as linhas, ou colunas, são linearmente independentes), há uma única solução que é  $\mathbf{u} = \mathbf{K}_e^{-1} \mathbf{r}$ . Se há um número de entradas maior do que saídas,  $m > n$ , então há um número de equações menor do que o de incógnitas, assim, se o posto de  $\mathbf{K}_e$  for igual a  $n$ , há um número infinito de combinações de  $\mathbf{u}$  que satisfaz a equação. Em malha fechada, o ponto a ser escolhido vai depender da sintonia do controlador, das restrições, e do estado atual do sistema. Porém, se o número de saídas é maior que o número de variáveis manipuladas,  $n > m$ , o número de equações é maior do que o número de incógnitas, assim, se o posto de  $\mathbf{K}_e$  for igual a  $m$ , não há solução que satisfaça todas as equações simultaneamente. Desta forma, o controlador convergirá a um ponto em que  $\Delta \mathbf{u} = \mathbf{K}(\mathbf{r} - \mathbf{f}_{ss}) = \mathbf{0}$ , mesmo que os erros de rastreamento individuais não sejam nulos, e este ponto dependerá da sintonia utilizada e do estado atual do sistema (por conta das restrições). O próximo exemplo ilustra esse caso.

**Exemplo 3.5.** *Considere um processo com duas saídas e uma entrada, com horizonte de predição 2 para as duas saídas e horizonte de controle 1,  $\delta_1 = \delta_2 = 1$  e  $\lambda = 0$ . Assim, o MPC tem liberdade total para escolher a ação de controle e os dois erros têm o mesmo peso na função objetivo. Considere, ainda, que o ganho estático de  $u$  para  $y_1$  é 2 e o ganho estático de  $u$  para*

$y_2$  é 3. O controle sem restrições em regime permanente neste caso pode ser colocado como:

$$\Delta \mathbf{u}_{ss} = [\mathbf{K}_1 \quad \mathbf{K}_2] \begin{bmatrix} \mathbf{e}_{1,ss} \\ \mathbf{e}_{2,ss} \end{bmatrix} = 0,$$

ou:

$$[K_{11} \ K_{12} \ K_{21} \ K_{22}][e_{1,ss} \ e_{1,ss} \ e_{2,ss} \ e_{2,ss}]^T = (K_{11}+K_{12})e_{1,ss} + (K_{21}+K_{22})e_{2,ss} = 0.$$

Assim, qualquer par de erros que satisfaça a igualdade anterior pode ser uma solução do problema. Se a referência escolhida for  $r_1 = 1$  e  $r_2 = 1$ , sabemos que não existe um valor de  $u$  que leve as duas saídas para a referência, pois  $r_1/r_2 = 1$  é diferente de  $K_{e1}/K_{e2} = 2/3$ . O MPC vai encontrar, em regime permanente, o valor de  $u$  tal que obtenha o mínimo de  $J$ , porém a igualdade dos ganhos estáticos sempre será obedecida, dado que é uma característica do sistema.

Neste problema simples, podemos calcular a solução de forma analítica se aceitarmos que o sistema é estável em malha fechada para o ajuste proposto. Veja que, em regime permanente, todas as variáveis estarão em valores constantes, então o incremento de controle será nulo e a função custo proposta será:

$$J = 2(r_1 - y_{1,ss})^2 + 2(r_2 - y_{2,ss})^2,$$

sendo  $y_{1,ss}$  e  $y_{2,ss}$  os valores de estado estacionário das saídas, que verificam  $y_{1,ss} = 2u_{ss}$  e  $y_{2,ss} = 3u_{ss}$ , com  $u_{ss}$  sendo o valor de equilíbrio do controle. Assim:

$$J = 2(1 - 2u_{ss})^2 + 2(1 - 3u_{ss})^2 = 2(2 - 10u_{ss} + 13u_{ss}^2).$$

O mínimo de  $J$  é obtido com  $u_{ss} = 5/13$ , valor esse que levará a saída 1 ao valor  $y_{1,ss} = 10/13$  e a saída 2 ao valor  $y_{2,ss} = 15/13$ . Note que os erros em regime permanente são similares em módulo (com valor  $3/13$ ) e de diferente sinal. Isso ocorre porque os pesos e os horizontes de ambas as saídas são iguais na função custo.

Observe que como o MPC encontrará sempre aquela combinação que minimize  $J$  e respeite o modelo estático do sistema, o resultado vai depender do ajuste de parâmetros do controlador. Se os pesos das saídas mudam para  $\delta_1 = 1$  e  $\delta_2 = 2$ , obtemos:

$$J = 2(1 - 2u_{ss})^2 + 4(1 - 3u_{ss})^2 = 2(3 - 16u_{ss} + 22u_{ss}^2),$$

e o mínimo de  $J$  é obtido com  $u_{ss} = 4/11$ , que leva a saída 1 ao valor  $y_{1,ss} = 8/11$  e a saída 2 ao valor  $y_{2,ss} = 12/11$ , melhorando o erro da segunda saída, de maior peso, em detrimento do erro da primeira saída, de menor peso.

Do ponto de vista da formulação do MPC, o modelo do processo sempre pode ser visto como uma restrição de igualdade que deve ser cumprida pelas entradas e saídas do sistema, de forma que a solução do problema de otimização sempre chegará a uma solução que satisfaça as condições de operação do processo.

Finalmente, outro ponto deve ser destacado. Apesar de a solução do problema de otimização neste caso ter a mesma forma que no caso monovariável, o aumento da dimensão do problema a ser resolvido tem grande impacto no tempo de processamento, aumentando consideravelmente para um sistema multivariável com grande número de variáveis manipuladas e controladas. Como o tempo necessário para obter a solução do problema de otimização pode ser um limitante em uma dada aplicação prática, discutiremos, em um capítulo específico do livro (Capítulo 7), alguns métodos para diminuir o tempo computacional e conseguir aplicar o MPC em processos rápidos. O próximo exemplo ilustra o cálculo do tamanho do problema a ser resolvido.

**Exemplo 3.6.** *Considere o mesmo processo  $2 \times 2$  do Exemplo 3.4, com os mesmos horizontes de controle  $N_{u,1} = 5$  e  $N_{u,2} = 3$ . A solução do problema de otimização, neste caso, fornece um vetor de incrementos de controle dado por:*

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \\ \Delta u_1(k+2) \\ \Delta u_1(k+3) \\ \Delta u_1(k+4) \\ -\Delta u_2(k) \\ \Delta u_2(k+1) \\ \Delta u_2(k+2) \end{bmatrix}, \quad (3.7)$$

*mas apenas os valores de  $\Delta u_1(k)$  e  $\Delta u_2(k)$  são usados no instante  $k$  para calcular  $u_1(k) = u_1(k-1) + \Delta u_1(k)$  e  $u_2(k) = u_2(k-1) + \Delta u_2(k)$  para serem enviados ao processo.*

*Ainda para este exemplo, podemos calcular as dimensões do problema de otimização. A dimensão do vetor  $\hat{\mathbf{y}}$  é  $\sum_{l=1}^2 (N_{2,l} - N_{1,l} + 1) = 20 + 10 = 30$  e a do vetor  $\Delta \mathbf{u}$  é  $\sum_{p=1}^2 N_{u,p} = 5 + 3 = 8$ . Assim, temos um problema com 8 variáveis de decisão e 30 predições a serem calculadas em cada amostra.*

Apresentada a forma como se obtém a lei de controle do MPC a partir do conhecimento do modelo compacto das predições, da função custo e das restrições, analisaremos, na sequência, as particularidades de cada formulação de MPC multivariável no que diz respeito ao cálculo das predições, começando com o DMC.

## 3.2 DMC multivariável

Nesta seção, estudaremos aspectos relacionados com o cálculo das predições no DMC multivariável, sempre para um processo com  $m$  entradas de controle e  $n$  saídas a serem controladas.

### 3.2.1 Cálculo das predições no caso estável

Considere que o sistema multivariável é estável e que não há perturbações medidas (o caso multivariável com perturbações medidas será apresentado na Seção 3.2.2). Assim, o modelo de malha aberta de cada saída  $l$  é dado por

$$y_l(k) = \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} \Delta u_p(k-i). \quad (3.8)$$

onde  $\Delta u_p$  é o incremento de controle associado à variável manipulada  $p$ , e os coeficientes  $g_{l,p,i}$  representam a resposta ao degrau que relaciona a variável manipulada  $p$  com a saída  $l$ . Observe aqui que, como o modelo é linear, podemos obter o modelo completo apenas somando as respostas obtidas de forma individual para um degrau aplicado em cada uma das entradas. Esta simplicidade do modelo usado pelo DMC é uma das suas vantagens pois, na prática industrial, basta realizar um ensaio de resposta ao degrau para cada uma das entradas de controle e armazenar as repostas obtidas em todas as saídas do processo, para depois organizar e compor as somas.

A partir deste modelo de malha aberta, considerando condições iniciais nulas, as predições da saída  $l$  no tempo  $k+j$  podem ser calculadas da seguinte forma:

$$\hat{y}_l(k+j | k) = \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} \Delta u_p(k+j-i) + \hat{n}_l(k+j | k), \quad (3.9)$$

com  $\hat{n}_l(k+j | k)$  representando uma predição da perturbação não mensurável em  $k+j$  observada na saída  $l$ . Assim como no caso monovariável, consideramos um modelo de perturbação constante, isto é,  $\hat{n}_l(k+j | k) = \hat{n}_l(k | k) = y_l(k) - \hat{y}_{ol}(k | k)$ , com  $y_l(k)$  a saída real do processo e  $\hat{y}_{ol}(k | k)$  calculado a partir de (3.8). Separando as parcelas do somatório que dependem de valores

futuros de  $\Delta u_p$ , é possível reescrever a predição como:

$$\begin{aligned}\hat{y}_l(k+j | k) &= \sum_{p=1}^m \sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) \\ &\quad + \sum_{p=1}^m \sum_{i=j+1}^{\infty} g_{l,p,i} \Delta u(k+j-i) + \hat{n}(k | k).\end{aligned}\quad (3.10)$$

Utilizando a definição de  $\hat{n}_l(k | k)$  e (3.8) para obter  $\hat{y}_{ol}(k | k)$ , temos

$$\begin{aligned}\hat{y}_l(k+j | k) &= \sum_{p=1}^m \sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) \\ &\quad + \sum_{p=1}^m \sum_{i=1}^{\infty} (g_{l,p,j+i} - g_{l,p,i}) \Delta u(k-i) + y_l(k),\end{aligned}\quad (3.11)$$

ou, de forma mais compacta,

$$\hat{y}_l(k+j | k) = \sum_{p=1}^m \sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) + f_l(k+j | k), \quad (3.12)$$

onde  $f_l(k+j | k)$  é a resposta livre da saída  $l$  do sistema, ou seja, a parte da resposta que não depende das ações de controle futuras, e, após o truncamento dos somatórios infinitos, é dada por:

$$f_l(k+j | k) = \sum_{p=1}^m \sum_{i=1}^{N_{ss,l,p}} (g_{l,p,j+i} - g_{l,p,i}) \Delta u(k-i) + y_l(k). \quad (3.13)$$

Note que o truncamento é possível pois, para sistemas assintoticamente estáveis, a resposta ao degrau de um dado par de entrada-saída tende a se tornar constante após um número suficiente de amostras  $N_{ss,l,p}$ , portanto,  $(g_{l,p,j+i} - g_{l,p,i}) \approx 0$ ,  $\forall i > N_{ss,l,p}$ . Uma observação importante é que, no caso geral, é possível definir um horizonte de modelo  $N_{ss,l,p}$  para cada modelo de resposta ao degrau do sistema multivariável. Isso passa a ser interessante quando se tem modelos com tempos de respostas muito diferentes. Assim, para os modelos com tempo de assentamento baixo, é possível escolher valores de  $N_{ss,l,p}$  menores, o que pode impactar o tamanho de algumas matrizes, tornando mais eficiente a implementação do algoritmo. Comentaremos este ponto com mais detalhes ainda nesta seção, ao apresentar a forma matricial das predições.

Considerando o cálculo das predições de 1 a  $N_{2,l}$  as predições futuras da saída  $l$ , dadas por (3.12), podem então ser escritas de forma matricial da seguinte maneira:

$$\hat{\mathbf{y}}_l = \sum_{p=1}^m \mathbf{G}_{l,p} \Delta \mathbf{u}_p + \mathbf{f}_l, \quad (3.14)$$

onde  $\mathbf{G}_{l,p} \in \mathbb{R}^{N_{2,l} \times N_{u,p}}$  é a matriz de coeficientes da resposta ao degrau da entrada  $p$  para a saída  $l$ ,  $N_{2,l}$  é o horizonte de predição da saída  $l$ ,  $N_{u,p}$  é o horizonte de controle da entrada  $p$ , e  $\mathbf{f}_l \in \mathbb{R}^{N_{2,l}}$  a resposta livre da saída  $l$  do sistema. Estes vetores e matrizes são dados por:

$$\mathbf{f}_l = \begin{bmatrix} f_l(k+1 | k) \\ f_l(k+2 | k) \\ \vdots \\ f_l(k+N_{2,l} | k) \end{bmatrix} = \sum_{p=1}^m \mathbf{H}_{l,p} \underline{\Delta \mathbf{u}}_p + \mathbf{1}_{N_{2,l}} y_l(k), \quad \underline{\Delta \mathbf{u}}_p = \begin{bmatrix} \Delta u_p(k-1) \\ \Delta u_p(k-2) \\ \vdots \\ \Delta u_p(k-N_{ss,l,p}) \end{bmatrix},$$

$$\mathbf{H}_{l,p} = \begin{bmatrix} h_{l,p,1,1} & h_{l,p,1,2} & \cdots & h_{l,p,1,N_{ss,l,p}-1} & h_{l,p,1,N_{ss,l,p}} \\ h_{l,p,2,1} & h_{l,p,2,2} & \cdots & h_{l,p,2,N_{ss,l,p}-1} & h_{l,p,2,N_{ss,l,p}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{l,p,N_{2,l},1} & h_{l,p,N_{2,l},2} & \cdots & h_{l,p,N_{2,l},N_{ss,l,p}-1} & h_{l,p,N_{2,l},N_{ss,l,p}} \end{bmatrix},$$

com

$$h_{l,p,i,j} = g_{l,p,i+j} - g_{l,p,j}, \quad j+i \leq N_{ss,l,p},$$

$$h_{l,p,i,j} = g_{l,p,N_{ss,l,p}} - g_{l,p,j}, \quad j+i > N_{ss,l,p},$$

e

$$\mathbf{G}_{l,p} = \begin{bmatrix} g_{l,p,1} & 0 & \cdots & 0 \\ g_{l,p,2} & g_{l,p,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{l,p,N_{2,l}} & g_{l,p,N_{2,l}-1} & \cdots & g_{l,p,N_{2,l}-N_{u,p}+1} \end{bmatrix}.$$

Da mesma forma que no caso monovariável, se o horizonte inicial de predição da saída  $l$  for maior do que um ( $N_{1,l} > 1$ ), as  $N_{1,l} - 1$  linhas iniciais de  $\mathbf{G}_{l,p}$ ,  $\mathbf{H}_{l,p}$ ,  $\hat{\mathbf{y}}_l$  e  $\mathbf{f}_l$  devem ser removidas.

Combinando as predições de todas as  $n$  saídas, dadas por (3.14), obtemos o formato padrão de predição

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f}, \quad (3.15)$$

com  $\mathbf{f} = \mathbf{H} \underline{\Delta \mathbf{u}} + \mathbb{I} \mathbf{y}(k)$  e

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{1,m} \\ \mathbf{G}_{2,1} & \cdots & \mathbf{G}_{2,m} \\ \vdots & \vdots & \vdots \\ \mathbf{G}_{n,1} & \cdots & \mathbf{G}_{n,m} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \cdots & \mathbf{H}_{1,m} \\ \mathbf{H}_{2,1} & \cdots & \mathbf{H}_{1,m} \\ \vdots & \vdots & \vdots \\ \mathbf{H}_{n,1} & \cdots & \mathbf{H}_{n,m} \end{bmatrix}, \quad \underline{\Delta \mathbf{u}} = \begin{bmatrix} \underline{\Delta \mathbf{u}}_1 \\ \underline{\Delta \mathbf{u}}_2 \\ \vdots \\ \underline{\Delta \mathbf{u}}_m \end{bmatrix},$$

$$\mathbb{I} = \begin{bmatrix} \mathbf{1}_{N_{2,1}-N_{1,1}+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{N_{2,2}-N_{1,2}+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{N_{2,n}-N_{1,n}+1} \end{bmatrix}, \quad \mathbf{y}(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_n(k) \end{bmatrix}.$$

Observe que a matriz  $\mathbf{H}$  resulta do empilhamento das matrizes  $\mathbf{H}_{l,p}$ , assim, para cada  $p$ , todas as matrizes  $\mathbf{H}_{l,p}$  devem ter o mesmo número de colunas, independentemente do  $N_{ss,l,p}$  escolhido. Dessa forma, é necessário adicionar colunas nulas em cada sub-matriz para ajustar a dimensão de todas ao maior dos  $N_{ss,l,p}$  para  $l = 1 \dots n$ . Já para o número de linhas não é necessário ajustar dimensões, pois todas as  $\mathbf{H}_{l,p}$ ,  $p = 1 \dots m$  têm o mesmo número de linhas, igual ao horizonte de predição da saída  $l$ .

Vamos apresentar um exemplo de cálculo manual deste modelo multivariável considerando valores de  $N_{ss,l,p}$  pequenos para poder mostrar os resultados numéricos de forma mais clara.

**Exemplo 3.7.** *Considere um processo de duas entradas e duas saídas e suponha que todas as variáveis estão normalizadas no intervalo  $[-10, 10]$ . Os ensaios de resposta ao degrau realizados permitiram obter os coeficientes de quatro respostas, representados pelos vetores:*

$$\mathbf{g}_{1,1} = [0,3 \ 0,45 \ 0,78 \ 0,95 \ 1,0 \ 1,0 \ 1,0 \ \dots],$$

$$\mathbf{g}_{1,2} = [0,14 \ 0,41 \ 0,76 \ 1,2 \ 1,2 \ 1,2 \ \dots],$$

$$\mathbf{g}_{2,1} = [0,2 \ 0,6 \ 1,4 \ 1,6 \ 2,0 \ 2,0 \ 2,0 \ \dots],$$

$$\mathbf{g}_{2,2} = [-0,1 \ -0,3 \ -0,7 \ -1 \ -1 \ -1 \ \dots].$$

Assim, podemos definir  $N_{ss,1,1} = 4$ ,  $N_{ss,2,1} = 4$ ,  $N_{ss,1,2} = 3$  e  $N_{ss,2,2} = 3$ , dado que para valores maiores teremos coeficientes nulos nas colunas das matrizes  $\mathbf{H}_{l,p}$ . Suponha que os horizontes de predição e de controle foram escolhidos como  $N_{1,1} = 1$ ,  $N_{2,1} = 3$ ,  $N_{1,2} = 1$ ,  $N_{2,2} = 2$ ,  $N_{u,1} = 2$ , e  $N_{u,2} = 1$ . Desta forma, a matriz  $\mathbf{G}$  é composta por 4 sub matrizes com número de linhas igual ao horizonte de predição correspondente à saída considerada e de colunas igual ao horizonte de controle da variável manipulada considerada. Neste caso:

$$\mathbf{G}_{1,1} = \begin{bmatrix} 0,30 & 0 \\ 0,45 & 0,3 \\ 0,78 & 0,45 \end{bmatrix}, \quad \mathbf{G}_{1,2} = \begin{bmatrix} 0,14 \\ 0,41 \\ 0,76 \end{bmatrix},$$

$$\mathbf{G}_{2,1} = \begin{bmatrix} 0,20 & 0 \\ 0,60 & 0,20 \end{bmatrix}, \quad \mathbf{G}_{2,2} = \begin{bmatrix} -0,10 \\ -0,30 \end{bmatrix},$$

que têm dimensões  $3 \times 2$ ,  $3 \times 1$ ,  $2 \times 2$  e  $2 \times 1$ . Assim,  $\mathbf{G}$  tem dimensão  $5 \times 3$  e é dada por:

$$\mathbf{G} = \left[ \begin{array}{cc|c} 0,30 & 0 & 0,14 \\ 0,45 & 0,3 & 0,41 \\ 0,78 & 0,45 & 0,76 \\ \hline 0,20 & 0 & -0,10 \\ 0,60 & 0,20 & -0,30 \end{array} \right].$$

A matriz  $\mathbb{I}$ , que é bloco diagonal, tem dois blocos  $\mathbf{1}_3$  e  $\mathbf{1}_2$ , do tamanho do horizonte de predição de cada saída, respectivamente 3 e 2, o que resulta em:

$$\mathbb{I} = \left[ \begin{array}{c|c} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \end{array} \right].$$

Finalmente, a matriz  $\mathbf{H}$  é composta por 4 submatrizes de dimensões que dependem dos horizontes de predição e dos horizontes de modelo. Assim, neste caso, temos blocos de dimensões  $3 \times 4$ ,  $3 \times 3$ ,  $2 \times 4$  e  $2 \times 3$ :

$$\mathbf{H}_{1,1} = \begin{bmatrix} g_{1,1,2} - g_{1,1,1} & g_{1,1,3} - g_{1,1,2} & g_{1,1,4} - g_{1,1,3} & g_{1,1,5} - g_{1,1,4} \\ g_{1,1,3} - g_{1,1,1} & g_{1,1,4} - g_{1,1,2} & g_{1,1,5} - g_{1,1,3} & g_{1,1,5} - g_{1,1,4} \\ g_{1,1,4} - g_{1,1,1} & g_{1,1,5} - g_{1,1,2} & g_{1,1,5} - g_{1,1,3} & g_{1,1,5} - g_{1,1,4} \end{bmatrix},$$

$$\mathbf{H}_{1,2} = \begin{bmatrix} g_{1,2,2} - g_{1,2,1} & g_{1,2,3} - g_{1,2,2} & g_{1,2,4} - g_{1,2,3} \\ g_{1,2,3} - g_{1,2,1} & g_{1,2,4} - g_{1,2,2} & g_{1,2,4} - g_{1,2,3} \\ g_{1,2,4} - g_{1,2,1} & g_{1,2,4} - g_{1,2,2} & g_{1,2,4} - g_{1,2,3} \end{bmatrix},$$

$$\mathbf{H}_{2,1} = \begin{bmatrix} g_{2,1,2} - g_{2,1,1} & g_{2,1,3} - g_{2,1,2} & g_{2,1,4} - g_{2,1,3} & g_{2,1,5} - g_{2,1,4} \\ g_{2,1,3} - g_{2,1,1} & g_{2,1,4} - g_{2,1,2} & g_{2,1,5} - g_{2,1,3} & g_{2,1,5} - g_{2,1,4} \end{bmatrix},$$

$$\mathbf{H}_{2,2} = \begin{bmatrix} g_{2,2,2} - g_{2,2,1} & g_{2,2,3} - g_{2,2,2} & g_{2,2,4} - g_{2,2,3} \\ g_{2,2,3} - g_{2,2,1} & g_{2,2,4} - g_{2,2,2} & g_{2,2,4} - g_{2,2,3} \end{bmatrix},$$

resultando em:

$$\mathbf{H}_{1,1} = \begin{bmatrix} 0,15 & 0,33 & 0,17 & 0,05 \\ 0,48 & 0,50 & 0,22 & 0,05 \\ 0,65 & 0,55 & 0,22 & 0,05 \end{bmatrix},$$



$$\mathbf{H}_{1,2} = \begin{bmatrix} 0,27 & 0,35 & 0,44 \\ 0,62 & 0,55 & 0,44 \\ 0,82 & 0,55 & 0,44 \end{bmatrix},$$

$$\mathbf{H}_{2,1} = \begin{bmatrix} 0,40 & 0,80 & 0,60 & 0,40 \\ 1,20 & 1,40 & 0,60 & 0,40 \end{bmatrix},$$

$$\mathbf{H}_{2,2} = \begin{bmatrix} -0,20 & -0,40 & -0,30 \\ -0,60 & -0,70 & -0,30 \end{bmatrix}.$$

Assim:

$$\mathbf{H} = \begin{bmatrix} 0,15 & 0,33 & 0,17 & 0,05 & | & 0,27 & 0,35 & 0,44 \\ 0,48 & 0,50 & 0,22 & 0,05 & | & 0,62 & 0,55 & 0,44 \\ 0,65 & 0,55 & 0,22 & 0,05 & | & 0,82 & 0,55 & 0,44 \\ 0,40 & 0,80 & 0,60 & 0,40 & | & -0,20 & -0,40 & -0,30 \\ 1,20 & 1,40 & 0,60 & 0,40 & | & -0,60 & -0,70 & -0,30 \end{bmatrix}.$$

Finalizando o exemplo, montamos os vetores de predição, controles futuros e controles passados, como:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1(k+1) \\ \hat{y}_1(k+2) \\ \hat{y}_1(k+3) \\ \hat{y}_2(k+1) \\ \hat{y}_2(k+2) \end{bmatrix}, \quad \Delta \mathbf{u} = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k) \end{bmatrix}, \quad \overleftarrow{\Delta \mathbf{u}} = \begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_1(k-2) \\ \Delta u_1(k-3) \\ \Delta u_1(k-4) \\ \Delta u_2(k-1) \\ \Delta u_2(k-2) \\ \Delta u_2(k-3) \end{bmatrix},$$

e o vetor de saídas medidas é dado por:

$$\mathbf{y}(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix}.$$

Assim, as predições são calculadas como:

$$\begin{bmatrix} \hat{y}_1(k+1) \\ \hat{y}_1(k+2) \\ \hat{y}_1(k+3) \\ \hat{y}_2(k+1) \\ \hat{y}_2(k+2) \end{bmatrix} = \mathbf{G} \begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k) \end{bmatrix} + \mathbf{H} \begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_1(k-2) \\ \Delta u_1(k-3) \\ \Delta u_1(k-4) \\ \Delta u_2(k-1) \\ \Delta u_2(k-2) \\ \Delta u_2(k-3) \end{bmatrix} + \mathbb{I} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix},$$

e os dois últimos termos representam a resposta livre  $\mathbf{f}$ :

$$\begin{bmatrix} f_1(k+1) \\ f_1(k+2) \\ f_1(k+3) \\ f_2(k+1) \\ f_2(k+2) \end{bmatrix} = \mathbf{H} \begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_1(k-2) \\ \Delta u_1(k-3) \\ \Delta u_1(k-4) \\ \Delta u_2(k-1) \\ \Delta u_2(k-2) \\ \Delta u_2(k-3) \end{bmatrix} + \mathbb{I} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix}.$$

No próximo exemplo usaremos estas predições para obter a lei de controle.

**Exemplo 3.8.** Neste exemplo, vamos calcular a lei de controle sem restrições do sistema analisado no Exemplo 3.7, utilizando o desenvolvimento apresentado neste capítulo.

Considere que desejamos limitar 3 vezes mais a variação do controle  $u_2$  que a variação do controle  $u_1$ , e que o erro da variável  $y_2$  é duas vezes mais importante que o da variável  $y_1$ . Assim, como as variáveis do modelo já estavam normalizadas, apenas consideramos divisão das ponderações pelo horizonte de cada variável, como já explicado. Considerando ponderações  $\delta_{1n} = 1$ ,  $\delta_{2n} = 2$ ,  $\lambda_{1n} = 1$  e  $\lambda_{2n} = 3$ , as matrizes de ponderação são dadas por:

$$\mathbf{Q}_y = \begin{bmatrix} 1/3 & 0 & 0 & | & 0 & 0 \\ 0 & 1/3 & 0 & | & 0 & 0 \\ 0 & 0 & 1/3 & | & 0 & 0 \\ 0 & 0 & 0 & | & 2/2 & 0 \\ 0 & 0 & 0 & | & 0 & 2/2 \end{bmatrix}, \quad \mathbf{Q}_u = \begin{bmatrix} 1/2 & 0 & | & 0 \\ 0 & 1/2 & | & 0 \\ 0 & 0 & | & 3/1 \end{bmatrix}.$$

Observe que colocamos explicitamente a divisão das ponderações pelo horizonte de cada variável. Para esse ajuste, o ganho do controle é:

$$\begin{aligned} \mathbf{K} &= (\mathbf{G}^T \mathbf{Q}_y \mathbf{G} + \mathbf{Q}_u)^{-1} \mathbf{G}^T \mathbf{Q}_y \\ &= \begin{bmatrix} 0,092596 & 0,097308 & 0,17809 & | & 0,18687 & 0,47858 \\ -0,042913 & 0,10852 & 0,14646 & | & -0,077941 & 0,1174 \\ 0,013107 & 0,035541 & 0,06748 & | & -0,03167 & -0,10317 \end{bmatrix} \end{aligned}$$

e a lei de controle final pode ser escrita como:

$$\begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k) \end{bmatrix} = \mathbf{K} \left( \begin{bmatrix} r_1(k+1) \\ r_1(k+2) \\ r_1(k+3) \\ r_2(k+1) \\ r_2(k+2) \end{bmatrix} - \begin{bmatrix} f_1(k+1) \\ f_1(k+2) \\ f_1(k+3) \\ f_2(k+1) \\ f_2(k+2) \end{bmatrix} \right),$$

sendo que apenas  $\Delta u_1(k)$  e  $\Delta u_2(k)$  são aplicados no processo no instante  $k$ , pelo princípio do horizonte deslizante.

### 3.2.2 Ação antecipativa - *Feed-forward*

Assim como analisado no DMC monovariável apresentado no Volume 1, é possível incluir as perturbações medidas no cálculo das predições do sistema multivariável. A partir de (3.9), mas considerando a predição da saída  $l$  e um conjunto de perturbações medidas  $m_q$  temos:

$$\begin{aligned} \hat{y}_l(k+j | k) = & \sum_{p=1}^m \sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) \\ & + \sum_{p=1}^m \sum_{i=1}^{N_{ss,l,p}} (g_{l,p,j+i} - g_{l,p,i}) \Delta u(k-i) + y_l(k) \\ & + \sum_{s=1}^{m_q} \sum_{i=0}^{j-1} g_{q,l,s,i} \Delta q_s(k+j-i) \\ & + \sum_{s=1}^{m_q} \sum_{i=0}^{N_{ssq,l,s}} (g_{q,l,s,j+i} - g_{q,l,s,i}) \Delta q(k-i). \end{aligned} \quad (3.16)$$

Note que esta equação tem os primeiros três termos iguais ao caso sem perturbações medidas, apenas foram adicionados os dois últimos termos, correspondentes às contribuições das perturbações medidas futuras e passadas, respectivamente. Os coeficientes da resposta ao degrau que relaciona a saída  $y_l$  com a perturbação medida  $q_s$  são definidos como  $g_{q,l,s,i}$ . Note que a soma do último termo foi truncada em  $N_{ssq,l,s}$ , que é o horizonte de modelo da resposta ao degrau da saída  $l$  relativa à perturbação medida  $q_s$ , considerando um modelo estável. Note, ainda, que consideramos uma perturbação como futura a partir de  $k+1$  e passada até o tempo atual  $k$ , pois seu valor em  $k$  é conhecido.

Podemos, ainda, reescrever a predição de forma mais compacta como:

$$\hat{y}_l(k+j | k) = \sum_{p=1}^m \sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) + f_l(k+j | k). \quad (3.17)$$

Nesta equação, a nova resposta livre,  $f_l(k+j | k)$ , inclui informações do passado e futuro da perturbação. Quando os valores futuros da perturbação não são conhecidos (na maioria dos casos práticos), usualmente se considera

que a perturbação manterá o último valor medido  $q_s(k+j) = q_s(k), \forall j > 0$ , ou seja,  $\Delta q_s(k+j) = 0, \forall j > 0$ , e o termo futuro pode ser descartado.

Considerando novamente as predições de  $y_l$  de 1 a  $N_{2,l}$ , os termos da resposta livre  $\mathbf{f}_l$  que dependem do passado e do futuro da perturbação medida podem ser colocados de forma matricial da mesma forma que foi feito para o sinal de controle:

$$\sum_{s=1}^{m_q} \mathbf{G}_{q,l,s} \Delta \mathbf{q}_s + \sum_{s=1}^{m_q} \mathbf{H}_{q,l,s} \overleftarrow{\Delta \mathbf{q}_s},$$

com:

$$\mathbf{H}_{q,l,s} = \begin{bmatrix} h_{q,l,s,1,0} & h_{q,l,s,1,1} & \cdots & h_{q,l,s,1,N_{ssq,l,s}-1} & h_{q,l,s,1,N_{ssq,l,s}} \\ h_{q,l,s,2,0} & h_{q,l,s,2,1} & \cdots & h_{q,l,s,2,N_{ssq,l,s}-1} & h_{q,l,s,2,N_{ssq,l,s}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{q,l,s,N_{2,l},0} & h_{q,l,s,N_{2,l},1} & \cdots & h_{q,l,s,N_{2,l},N_{ssq,l,s}-1} & h_{q,l,s,N_{2,l},N_{ssq,l,s}} \end{bmatrix},$$

$$h_{q,l,s,i,j} = g_{q,l,s,i+j} - g_{q,l,s,j}, \quad j+i \leq N_{ssq,l,s},$$

$$h_{q,l,s,i,j} = g_{q,l,s,N_{ssq,l,s}} - g_{q,l,s,j}, \quad j+i > N_{ssq,l,s},$$

$$\mathbf{G}_{q,l,s} = \begin{bmatrix} g_{q,l,s,0} & 0 & \cdots & 0 \\ g_{q,l,s,1} & g_{q,l,s,0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{q,l,s,N_{2,l}-1} & g_{q,l,s,N_{2,l}-2} & \cdots & g_{q,l,s,N_{2,l}-N_{q,s}} \end{bmatrix}.$$

Nestas equações,

$$\Delta \mathbf{q}_s = [\Delta q_s(k+1), \Delta q_s(k+2), \dots, \Delta q_s(k+N_{q,s})]^T,$$

$$\overleftarrow{\Delta \mathbf{q}_s} = [\Delta q_s(k), \Delta q_s(k-1), \dots, \Delta q_s(k-N_{ssq,s})]^T,$$

$\mathbf{H}_{q,l,s} \in \mathbb{R}^{N_{2,l} \times N_{ssq,l,s}+1}$  e  $\mathbf{G}_{q,l,s} \in \mathbb{R}^{N_{2,l} \times N_{q,s}}$ , e caso  $N_{1,l} > 1$  as primeiras  $N_{1,l} - 1$  linhas das matrizes  $\mathbf{H}_{q,l,s}$  e  $\mathbf{G}_{q,l,s}$  devem ser eliminadas.

Combinando as predições de todas as  $n$  saídas, obtemos, novamente, o formato padrão já visto no caso sem perturbação medida:

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f} \quad (3.18)$$

mas, agora,  $\mathbf{f} = \mathbf{H} \overleftarrow{\Delta \mathbf{u}} + \mathbb{I} \mathbf{y}(k) + \mathbf{H}_q \overleftarrow{\Delta \mathbf{q}} + \mathbf{G}_q \Delta \mathbf{q}$ , sendo que  $\mathbf{G}$  e  $\mathbb{I}$  são os mesmos calculados no caso sem perturbação medida e

$$\mathbf{G}_q = \begin{bmatrix} \mathbf{G}_{q,1,1} & \cdots & \mathbf{G}_{q,1,m_q} \\ \mathbf{G}_{q,2,1} & \cdots & \mathbf{G}_{q,2,m_q} \\ \vdots & \vdots & \vdots \\ \mathbf{G}_{q,n,1} & \cdots & \mathbf{G}_{q,n,m_q} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_{q,1,1} & \cdots & \mathbf{H}_{q,1,m_q} \\ \mathbf{H}_{q,2,1} & \cdots & \mathbf{H}_{q,2,m_q} \\ \vdots & \vdots & \vdots \\ \mathbf{H}_{q,n,1} & \cdots & \mathbf{H}_{q,n,m_q} \end{bmatrix},$$

$$\overleftarrow{\Delta \mathbf{q}} = \begin{bmatrix} \overleftarrow{\Delta \mathbf{q}_1} \\ \overleftarrow{\Delta \mathbf{q}_2} \\ \vdots \\ \overleftarrow{\Delta \mathbf{q}_{m_q}} \end{bmatrix}, \quad \Delta \mathbf{q} = \begin{bmatrix} \Delta \mathbf{q}_1 \\ \Delta \mathbf{q}_2 \\ \vdots \\ \Delta \mathbf{q}_{m_q} \end{bmatrix}.$$

Observe que a matriz  $\mathbf{H}_{\mathbf{q}}$  resulta do empilhamento das matrizes  $\mathbf{H}_{q,l,s}$ . Assim, para cada  $s$  todas as matrizes  $\mathbf{H}_{q,l,s}$  devem ter o mesmo número de colunas, independentemente do  $N_{ssq,l,s}$  escolhido. Dessa forma, de forma similar ao caso da matriz  $\mathbf{H}$  apresentado na seção anterior, é preciso corrigir as dimensões com colunas nulas, se necessário. O próximo exemplo ilustra o cálculo da parte da resposta livre que depende da perturbação medida.

**Exemplo 3.9.** *Considere o processo de duas entradas e duas saídas do Exemplo 3.8, no qual adicionamos uma perturbação medida  $q(k)$ . Suponha que os horizontes do exemplo anterior são mantidos e o horizonte da perturbação é  $N_q = 2$ , dado que supomos que a perturbação futura é conhecida. Considere, ainda, que no ensaio de resposta ao degrau para  $q$  foram obtidos os coeficientes de duas respostas (dado que neste caso  $m_q = 1$ ) representados pelos vetores:*

$$\mathbf{g}_{q,1,1} = [0,2 \ 0,4 \ 0,8 \ 1,1 \ 1,1 \ 1,1 \ \dots],$$

$$\mathbf{g}_{q,2,1} = [0,1 \ 0,3 \ 0,7 \ 1,0 \ 1,0 \ 1,0 \ \dots],$$

e suponha que as respostas para a perturbação não são instantâneas, assim  $g_{l,p,0} = 0, \forall l, p$ .

Precisamos, neste problema, calcular a matriz  $\mathbf{G}_{\mathbf{q}}$ , a matriz  $\mathbf{H}_{\mathbf{q}}$  e os vetores  $\Delta \mathbf{q}$  e  $\overleftarrow{\Delta \mathbf{q}}$  necessários para calcular a parte da resposta livre que depende da perturbação.

Analisando as respostas ao degrau, observamos que podemos escolher  $N_{ssq,1,1} = N_{ssq,2,1} = 3$ . Observamos que a matriz  $\mathbf{G}_{\mathbf{q}}$  é composta por duas submatrizes com número de linhas igual ao horizonte de predição e de colunas igual a  $N_q = 2$ , e a matriz  $\mathbf{H}_{\mathbf{q}}$ , composta por duas submatrizes com número de linhas igual ao horizonte de predição e de colunas igual a  $N_{ssq} + 1 = 3 + 1 = 4$ . Essas matrizes são dadas por:

$$\mathbf{G}_{q,1,1} = \begin{bmatrix} 0 & 0 \\ 0,2 & 0 \\ 0,4 & 0,2 \end{bmatrix}, \quad \mathbf{G}_{q,2,1} = \begin{bmatrix} 0 & 0 \\ 0,1 & 0 \end{bmatrix},$$

$$\mathbf{H}_{q,1,1} = \begin{bmatrix} g_{q,1,1,1} - g_{q,1,1,0} & g_{q,1,1,2} - g_{q,1,1,1} & g_{q,1,1,3} - g_{q,1,1,2} & g_{q,1,1,3} - g_{q,1,1,2} \\ g_{q,1,1,2} - g_{q,1,1,0} & g_{q,1,1,3} - g_{q,1,1,1} & g_{q,1,1,3} - g_{q,1,1,2} & g_{q,1,1,3} - g_{q,1,1,2} \\ g_{q,1,1,3} - g_{q,1,1,0} & g_{q,1,1,3} - g_{q,1,1,1} & g_{q,1,1,3} - g_{q,1,1,2} & g_{q,1,1,3} - g_{q,1,1,2} \end{bmatrix},$$

$$\mathbf{H}_{q,2,1} = \begin{bmatrix} g_{q,2,1,1} - g_{q,2,1,0} & g_{q,2,1,2} - g_{q,2,1,1} & g_{q,2,1,3} - g_{q,2,1,2} & g_{q,2,1,3} - g_{q,2,1,3} \\ g_{q,2,1,2} - g_{q,2,1,0} & g_{q,2,1,3} - g_{q,2,1,1} & g_{q,2,1,3} - g_{q,2,1,2} & g_{q,2,1,3} - g_{q,2,1,3} \end{bmatrix},$$

$$\mathbf{H}_{q,1,1} = \begin{bmatrix} 0,2 & 0,2 & 0,4 & 0 \\ 0,4 & 0,6 & 0,4 & 0 \\ 0,8 & 0,6 & 0,4 & 0 \end{bmatrix}, \quad \mathbf{H}_{q,2,1} = \begin{bmatrix} 0,1 & 0,2 & 0,4 & 0 \\ 0,3 & 0,6 & 0,4 & 0 \end{bmatrix},$$

ou seja:

$$\mathbf{G}_q = \begin{bmatrix} 0 & 0 \\ 0,2 & 0 \\ 0,4 & 0,2 \\ 0 & 0 \\ 0,1 & 0 \end{bmatrix}, \quad \mathbf{H}_q = \begin{bmatrix} 0,2 & 0,2 & 0,4 & 0 \\ 0,4 & 0,6 & 0,4 & 0 \\ 0,8 & 0,6 & 0,4 & 0 \\ 0,1 & 0,2 & 0,4 & 0 \\ 0,3 & 0,6 & 0,4 & 0 \end{bmatrix},$$

e

$$\Delta \mathbf{q} = \begin{bmatrix} \Delta q(k+1) \\ \Delta q(k+2) \end{bmatrix}, \quad \overleftarrow{\Delta \mathbf{q}} = \begin{bmatrix} \Delta q(k) \\ \Delta q(k-1) \\ \Delta q(k-2) \\ \Delta q(k-3) \end{bmatrix}.$$

### 3.2.3 DMC multivariável recursivo

Seguindo passos similares aos apresentados no Volume 1 para se obter o algoritmo recursivo do DMC no caso monovariável, é possível mostrar que, para implementar a versão recursiva do DMC multivariável, precisamos apenas alterar o cálculo da resposta livre. No caso multivariável, é mais eficiente armazenar cada vetor de predição de malha aberta,  $\hat{\mathbf{y}}_{ol}$ , individualmente. Esse vetor é tal que  $\hat{\mathbf{y}}_{ol} \in \mathbb{R}^{N_{ss,l}}$  e é definido da seguinte forma

$$\hat{\mathbf{y}}_{ol}(k | k-1) = \begin{bmatrix} \hat{y}_{ol}(k | k-1) \\ \hat{y}_{ol}(k+1 | k-1) \\ \vdots \\ \hat{y}_{ol}(k+N_{ss,l}-1 | k-1) \end{bmatrix}.$$

Na primeira iteração do algoritmo, em  $k=0$ , assume-se que o sistema está em regime permanente. Assim,  $\hat{y}_{ol}(k+j | k-1) = y_l(0)$ ,  $\forall j \geq 0$  e  $\forall l$ .  $N_{ss,l}$  é definido como o maior de todos os  $N_{ss,l,p}$  correspondentes à saída  $l$ .

A cada nova iteração do algoritmo DMC, é possível atualizar os vetores de predição de malha aberta,  $\hat{\mathbf{y}}_{ol}$ , pois agora os valores de  $\Delta u_p(k-1)$  são conhecidos. Essa atualização é dada por

$$\hat{\mathbf{y}}_{ol}(k | k) = \hat{\mathbf{y}}_{ol}(k | k-1) + \sum_{p=1}^m \begin{bmatrix} g_{l,p,1} \\ g_{l,p,2} \\ \vdots \\ g_{l,p,N_{ss,l}} \end{bmatrix} \Delta u_p(k-1) \quad (3.19)$$

com:

$$\hat{\mathbf{y}}_{ol}(k | k) = \begin{bmatrix} \hat{y}_{ol}(k | k) \\ \hat{y}_{ol}(k + 1 | k) \\ \vdots \\ \hat{y}_{ol}(k + N_{ss,l} - 1 | k) \end{bmatrix}.$$

Assim como no caso monovariável, o primeiro elemento de  $\hat{\mathbf{y}}_{ol}(k | k)$  é  $\hat{y}_{ol}(k | k)$ , e é utilizado para o cálculo do erro de predição  $\hat{n}_l(k | k) = y_l(k) - \hat{y}_{ol}(k | k)$ , de acordo com (3.9). Esse primeiro elemento de cada  $\hat{\mathbf{y}}_{ol}(k | k)$  é então descartado, uma vez que apenas as predições a partir de  $k + 1$  serão utilizadas a partir de agora. Devido ao horizonte deslizando, os elementos internos de  $\hat{\mathbf{y}}_{ol}(k | k)$  são então deslocados, dando origem a  $\hat{\mathbf{y}}_{ol}(k + 1 | k)$ , da seguinte forma:

$$\hat{\mathbf{y}}_{ol}(k + 1 | k) = \begin{bmatrix} \hat{y}_{ol}(k + 1 | k) \\ \hat{y}_{ol}(k + 2 | k) \\ \vdots \\ \hat{y}_{ol}(k + N_{ss,l} - 1 | k) \\ \hat{y}_{ol}(k + N_{ss,l} | k) \end{bmatrix} \approx \begin{bmatrix} \hat{y}_{ol}(k + 1 | k) \\ \hat{y}_{ol}(k + 2 | k) \\ \vdots \\ \hat{y}_{ol}(k + N_{ss,l} - 1 | k) \\ \hat{y}_{ol}(k + N_{ss,l} - 1 | k) \end{bmatrix}. \quad (3.20)$$

Veja que o último elemento do vetor  $\hat{\mathbf{y}}_{ol}(k | k)$  é repetido para se obter  $\hat{\mathbf{y}}_{ol}(k + 1 | k)$ , pois assim como no caso monovariável,  $\hat{y}_{ol}(k + N_{ss,l} | k) \approx \hat{y}_{ol}(k + N_{ss,l} - 1 | k)$  dado que podemos considerar que a resposta do modelo converge a um valor constante em  $N_{ss,l}$ . Assim, finalmente, cada uma das respostas livres do sistema (considerando as predições de 1 a  $N_{2,l}$ ) pode ser obtida com:

$$\mathbf{f}_l = \begin{bmatrix} \hat{y}_{ol}(k + 1 | k) \\ \vdots \\ \hat{y}_{ol}(k + N_{2,l} | k) \end{bmatrix} + \mathbf{1}_{N_{2,l}} (y_l(k) - \hat{y}_{ol}(k | k)). \quad (3.21)$$

Combinando todas as respostas livres individuais,  $\mathbf{f}_l$ , obtemos vetor  $\mathbf{f}$  definido em (3.15). Observe, por fim, que o primeiro termo do lado direito de (3.21) corresponde aos termos de 1 a  $N_{2,l}$  do vetor  $\hat{\mathbf{y}}_{ol}$  já que, em geral,  $N_{2,l} < N_{ss,l}$ .

No caso de termos perturbações medidas, devemos alterar a formulação recursiva, alterando a atualização do vetor de predição de malha aberta,  $\hat{\mathbf{y}}_{ol}(k | k)$ . Caso o modelo considere perturbações mensuráveis, essa etapa do

algoritmo recursivo passa a ser dada por:

$$\begin{aligned} \hat{\mathbf{y}}_{ol}(k | k) = & \hat{\mathbf{y}}_{ol}(k | k-1) + \sum_{p=1}^m \begin{bmatrix} g_{l,p,1} \\ g_{l,p,2} \\ \vdots \\ g_{l,p,N_{ss,l}} \end{bmatrix} \Delta u_p(k-1) \\ & + \sum_{s=1}^{m_q} \begin{bmatrix} g_{q,l,s,0} \\ g_{q,l,s,1} \\ \vdots \\ g_{q,l,s,N_{sq,l}-1} \end{bmatrix} \Delta q_s(k). \end{aligned} \quad (3.22)$$

Caso os valores futuros da perturbação sejam conhecidos (ou estimados), o cálculo do vetor de resposta livre,  $\mathbf{f}_l$ , também precisa ser modificado, usando:

$$\mathbf{f}_l = \begin{bmatrix} \hat{y}_{ol}(k+1 | k) \\ \vdots \\ \hat{y}_{ol}(k+N_{2,l} | k) \end{bmatrix} + \mathbf{1}_{N_{2,l}} (y_l(k) - \hat{y}_{ol}(k | k)) + \sum_{s=1}^{m_q} \mathbf{G}_{q,l,s} \Delta \mathbf{q}_s. \quad (3.23)$$

O restante do algoritmo recursivo permanece exatamente igual.

**Exemplo 3.10.** *Este exemplo trata do controle de nível e temperatura da água contida no interior de um tanque termicamente isolado. Utiliza-se um resistor de potência para aquecimento da água dentro do tanque e para controlar as vazões de entrada e de saída utilizam-se duas válvulas, sendo uma para cada variável. Desejamos controlar simultaneamente a temperatura ( $y_1$ ) e o nível ( $y_2$ ) no interior do tanque. O controlador pode atuar na potência entregue ao resistor de aquecimento instalado no tanque ( $u_1$ ) e na abertura da válvula de alimentação ( $u_2$ ). Considera-se que a abertura da válvula de saída do tanque ( $q$ ) é a principal perturbação do sistema.*

Vamos considerar aqui que o comportamento da planta real pode ser bem representado pela seguinte matriz de funções de transferência:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1,43e^{-s}}{1+7,5s} & \frac{-4,5e^{-s}}{1+13,4s} \\ 0 & \frac{5,72}{1+14,7s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} + \begin{bmatrix} \frac{2,1e^{-s}}{1+13s} \\ \frac{-3,12}{1+14,7s} \end{bmatrix} Q(s)$$

onde observamos que, como esperado, o nível do tanque não é afetado pela potência de aquecimento.

O cenário de simulação considera um tempo total de simulação de 200 s. O sistema parte de  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$  e, em  $t = 10$  s, uma



variação tipo degrau unitário da referência de  $y_1$  é aplicada. Posteriormente, em  $t = 80$  s aplica-se uma variação tipo degrau de amplitude 0,5 na referência de  $y_2$ . Finalmente, em  $t = 130$  s aplica-se uma variação tipo degrau de amplitude 0,2 na perturbação  $q$ .

Para a formulação e o ajuste do controlador DMC, é realizado primeiro um teste de resposta ao degrau para a obtenção dos coeficientes da resposta ao degrau. Utilizando um período de amostragem de 1 segundo, foram obtidas as respostas ao degrau do sistema mostradas na Figura 3.1. A partir dessas respostas, foram escolhidos os horizontes de modelo como  $N_{ss,i,j} = N_{ssq,i,j} = 50, \forall i,j$ .

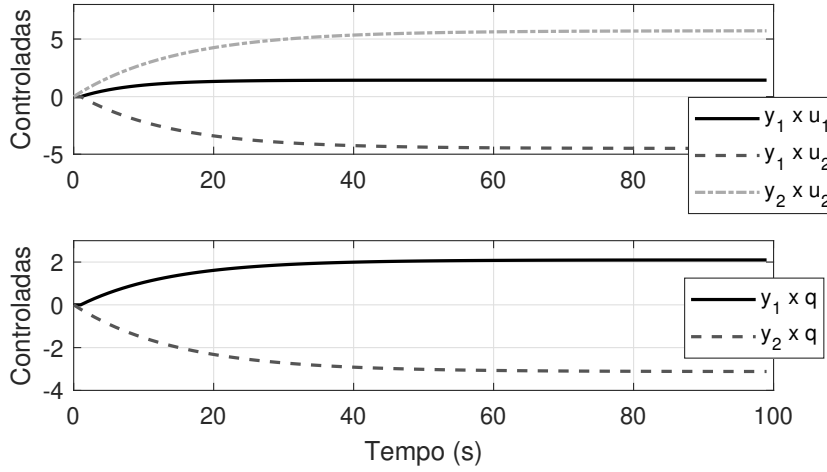


Figura 3.1: Exemplo 3.10 – Resposta ao degrau unitário nas entradas e perturbações.

A escolha dos horizontes de predição e controle foi a seguinte. Para a saída 1, dado que há uma amostra de atraso e a constante de tempo do elemento (1,1) da matriz função de transferência é de 7 amostras, escolhemos  $N_{1,1} = 2$  e  $N_{2,1} = 17$ , de forma tal a ter um horizonte de predição de duas constantes de tempo de malha aberta. Usando o mesmo critério, escolhemos  $N_{1,2} = 1$  e  $N_{2,2} = 31$  para a saída 2. O horizonte de controle foi escolhido como aproximadamente um terço do de predição, assim,  $N_{u,1} = 5$  e  $N_{u,2} = 10$ . Esta escolha foi realizada pois neste sistema espera-se que o controlador atue na potência para controlar a temperatura e na válvula para o nível, dada a característica triangular do modelo. As ponderações do erro e do controle foram consideradas constantes ao longo dos horizontes:  $\delta_1 = 1/N_{y,1}$ ,  $\delta_2 = 9/N_{y,2}$ ,  $\lambda = 1/N_{u,1}$  e  $\lambda_2 = 30/N_{u,2}$ .

Os resultados de simulação podem ser vistos na Figura 3.2. Como pode ser observado, o controlador permite desacoplar de forma satisfatória os efeitos

da variação de nível na temperatura tanto no caso de mudanças de referência como na rejeição de perturbações.

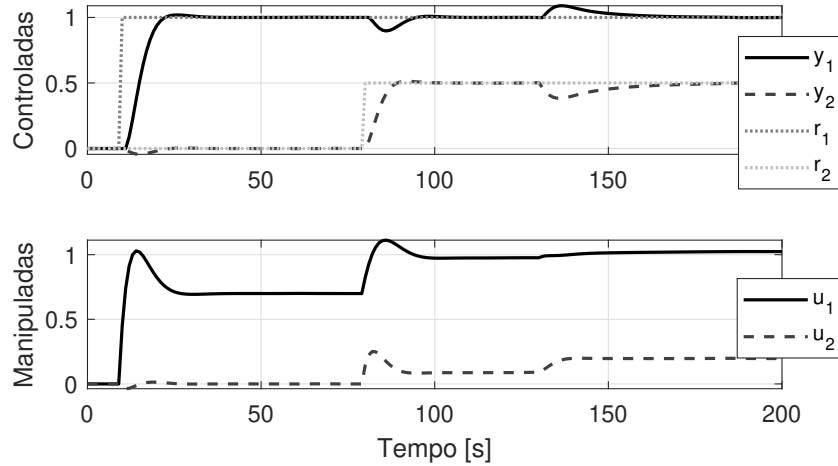


Figura 3.2: Exemplo 3.10 – resposta e sinal de controle do tanque térmico em malha fechada para seguimento de referências e rejeição de perturbações não mensuráveis.

*A resposta às perturbações poderia ser melhorada caso a abertura da válvula de saída fosse medida e usada na lei de controle antecipativa do DMC MIMO. Esta análise se estuda no Exercício 6 da lista apresentada no final do capítulo.*

### 3.2.4 DMC MIMO para plantas integradoras

Quando alguma das dinâmicas do sistema multivariável é integradora, temos que aplicar nessa variável as mesmas ideias usadas no Volume 1 para o caso monovariável, de forma a poder calcular a resposta livre com o truncamento necessário. No caso de aproximar o comportamento integrador por um modelo estável lento, nenhuma modificação do algoritmo é necessária. Caso um outro método seja usado, a parte da resposta livre correspondente a essa variável com dinâmica integradora deverá ser alterada. A modificação do DMC para plantas integradoras e instáveis será apresentada na Seção 3.2.5. Aqui apresentaremos um exemplo de aplicação do método mais simples, que aproxima o comportamento integrador por um modelo estável lento. Como veremos no exemplo, dependendo das características do processo, o ajuste do controlador não é simples, pelo erro introduzido no modelo.

**Exemplo 3.11.** *A Figura 3.3 mostra um sistema de evaporação da indústria de cana de açúcar. Neste processo, o objetivo é concentrar o caldo de cana,*

aumentando a concentração de açúcar por meio da evaporação da água contida no caldo. Para isso, são utilizados evaporadores nos quais vapor produzido em caldeiras é usado para aquecer o caldo e produzir a evaporação. Para ter um processo mais eficiente, somente o primeiro evaporador recebe vapor direto da caldeira. O segundo recebe o vapor vegetal produzido no primeiro evaporador, o terceiro recebe o vapor do segundo, e assim sucessivamente.

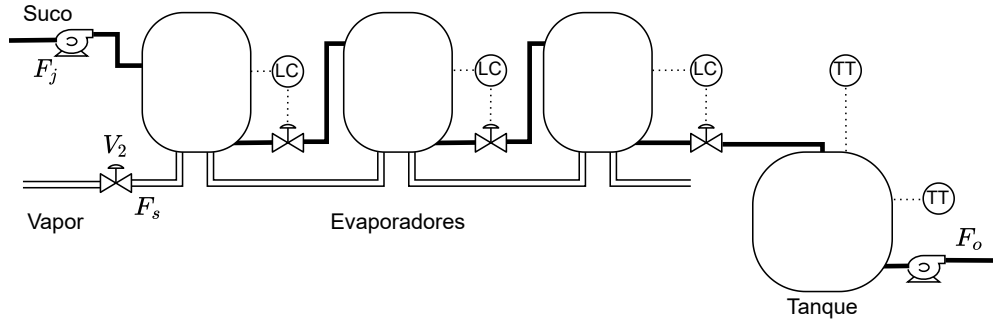


Figura 3.3: Exemplo 3.11 – Unidade de evaporação na indústria do açúcar.

Neste estudo de caso, o sistema de evaporação tem três estágios e um tanque de armazenamento. Cada estágio possui um sistema de controle de nível e o tanque armazenador alimenta o processo seguinte com caldo concentrado de vazão  $F_o$ . Neste tanque, devem ser controladas a temperatura ( $T$ ) e o nível ( $L$ ) manipulando as vazões de caldo na entrada ( $F_j$ ) e a vazão de vapor ( $F_s$ ) usando, respectivamente, a bomba de entrada e a válvula  $V_2$ . O modelo que representa o sistema no ponto de operação é dado por:

$$Y(s) = P(s)U(s) - P_q(s)Q(s),$$

onde  $Y(s) = [L(s) \ T(s)]^T$  é o vetor de saídas,  $U(s) = [F_j \ F_s]^T$  é o vetor de variáveis manipuladas e  $Q(s) = F_o$  é a perturbação (vazão de saída). Os modelos são dados por:

$$\mathbf{P}(s) = \begin{bmatrix} \frac{3,5e^{-0.1s}}{s} & \frac{-e^{-0.5s}}{1+2s} \\ \frac{-2e^{-0.7s}}{1+1,5s} & \frac{e^{-0.5s}}{1+3,2s} \end{bmatrix}, \quad \mathbf{P}_q(s) = \begin{bmatrix} \frac{3,5e^{-0.1s}}{s} \\ \frac{-4,5e^{-0.2s}}{1+2s} \end{bmatrix}.$$

Desejamos controlar este sistema para que rejeite as perturbações de tipo degrau de  $Q(s)$  tanto no nível como na temperatura. As respostas transitórias para mudanças de referência devem ser pouco oscilatórias e com o menor pico possível. A perturbação neste sistema, que não pode ser medida, deve

ser rejeitada também com uma resposta com pouca oscilação. Neste sistema, vamos projetar o controle DMC usando um modelo de primeira ordem para representar o sistema integrador.

Na Figura 3.4, observamos as respostas obtidas usando um modelo de primeira ordem com uma constante de tempo de 3 segundos para representar o sistema integrador. O ajuste desta constante de tempo depende de cada processo e pode ser ajustada por simulação. Ainda neste DMC, no caso da referência 1, para evitar mudanças bruscas nas ações de controle ao mudar as referências em forma de degraus, usamos um filtro de referência de primeira ordem com constante de tempo de 1 segundo (o polo do filtro discreto é  $a_f = 0,9672$ ). Os demais parâmetros usados foram:  $N_u = 5$  para as duas manipuladas,  $N_{1,l} = d_l + 1$  e  $N_{2,l} = d_l + 20$ ,  $\forall l$  para as saídas, horizonte de modelo  $N_{ss,1} = 90$  e  $N_{ss,2} = 220$ . As ponderações foram normalizadas pelos horizontes e se usou um peso 5 vezes maior para o erro e para o controle da variável 1.

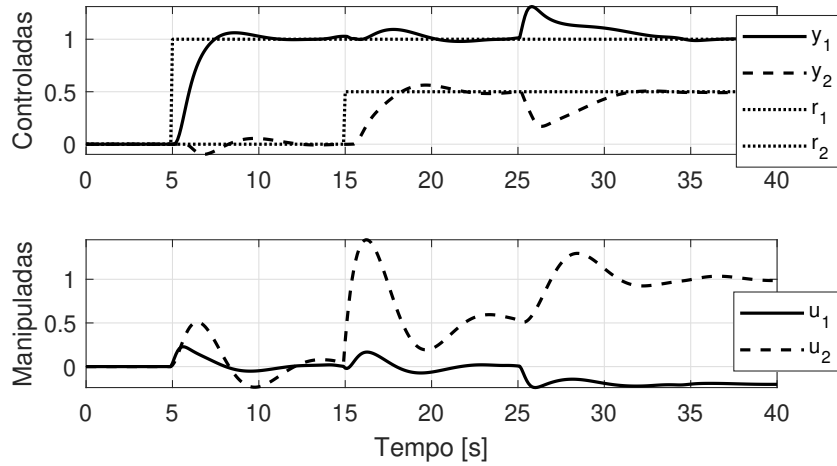


Figura 3.4: Exemplo 3.11 – Variáveis controladas e manipuladas do evaporador em malha fechada para seguimento de referências e rejeição de perturbações não mensuráveis. Caso de um modelo de primeira ordem para representar o sistema integrador.

Como pode ser observado, neste caso, o desempenho do controlador é aceitável, mesmo considerando o erro introduzido pela aproximação da dinâmica integradora. Como veremos na próxima seção, esta resposta pode ser bastante melhorada usando uma versão modificada do DMC.

Mais exemplos com o DMC multivariável serão apresentados na Seção 3.4. Na próxima seção, apresentamos a versão do DMC capaz de tratar de processos estáveis e não estáveis de forma unificada.

### 3.2.5 DMC Generalizado multivariável

O DMC generalizado (*Generalized DMC* - GDMC) apresentado no Volume 1 para o caso SISO pode ser facilmente estendido para o caso MIMO de forma tal a obter um algoritmo que trata sistemas MIMO instáveis ou integradores. A simplicidade da generalização se dá pelo fato de que o erro de predição é calculado de forma independente para cada saída  $y_l$  do processo. Assim, usando a mesma ideia do caso SISO, podemos usar um filtro  $F_{e,l,j}(z)$  em cada saída predita no horizonte  $j$  para filtrar o erro de predição:

$$\hat{n}_{f,l}(k+j | k) = F_{e,l,j}(z)(y_l(k) - \hat{y}_l(k | k))$$

e obter, então:

$$\begin{aligned} \hat{y}_l(k+j | k) = & \sum_{p=1}^m \sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) \\ & + \sum_{p=1}^m \sum_{i=j+1}^{N_{ss,l,p}} (g_{l,p,j+i} - \tilde{g}_{l,p,i}) \Delta u(k-i) \\ & + F_{e,l,j}(z) y_l(k) \end{aligned} \quad (3.24)$$

com  $\tilde{g}_{l,p,i}$  sendo os coeficientes da resposta ao degrau do modelo filtrados.

O ajuste do filtro no caso MIMO segue a mesma ideia do caso SISO. Devemos conseguir que a resposta livre seja convergente mesmo que o modelo do processo não seja estável. Isso equivale a conseguir que  $g_{l,p,j+i} - \tilde{g}_{l,p,i} \approx 0$  para todo  $i \geq N_{f,l}$ , sendo  $N_{f,l}$  o horizonte de modelo filtrado para a saída  $l$ .

No caso MIMO, o cálculo de cada  $F_{e,l,j}(z)$  deve então ser realizado considerando todos os polos indesejados (associados a dinâmicas não estáveis) das funções de transferência associadas às entradas que afetam a saída  $y_l$ . Do ponto de vista do cálculo dos filtros, estamos resolvendo  $n$  problemas independentes, um por saída, e assim, usamos um filtro por saída e por instante de predição.

Observe que o problema MIMO pode ser dividido em vários problemas do tipo MISO (*Multiple-Input Single-Output*), com mais de uma entrada e uma única saída, mas o procedimento de análise é igual ao caso SISO.

Primeiro, lembre-se que o modelo de resposta ao degrau:

$$y_l(k) = \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} \Delta u_p(k-i)$$

pode ser representado no domínio da frequência, após aplicação da transfor-

mada  $Z$ , como:

$$Y_l(z) = \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} z^{-i} (1 - z^{-1}) U_p(z) = \sum_{p=1}^m G_{l,p}(z) U_p(z).$$

Como desejamos usar a expressão compacta  $G_{l,p}(z)$  para a relação entre  $u_p$  e  $y_l$ , devemos reescrever a resposta livre de forma que os coeficientes  $g_{l,p,i}$  fiquem explícitos no somatório:

$$\begin{aligned} f_l(k+j \mid k) &= \sum_{p=1}^m \sum_{i=j+1}^{\infty} g_{l,p,i} \Delta u_p(k+j-i) + F_{e,l,j}(z) y(k) \\ &\quad - F_{e,l,j}(z) \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} \Delta u_p(k-i). \end{aligned}$$

Dado que esta última equação representa a resposta livre, que é calculada considerando os incrementos de controles futuros nulos  $\Delta u_p(k+j-i) = 0$  se  $j-i \geq 0$ , então  $u_p(k+j-i)$  é constante se  $j-i \geq 0$ . Assim, o seguinte somatório é sempre zero:

$$\sum_{i=1}^j g_{l,p,i} \Delta u_p(k+j-i) = 0$$

para todo  $p$ , e podemos adicioná-los à equação anterior e agregá-los ao seu primeiro termo, resultando em:

$$\begin{aligned} f_l(k+j \mid k) &= \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} \Delta u_p(k+j-i) + F_{e,l,j}(z) y(k) \\ &\quad - F_{e,l,j}(z) \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} \Delta u_p(k-i), \end{aligned}$$

ou de forma equivalente:

$$\begin{aligned} f_l(k+j \mid k) &= \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} z^j (1 - z^{-1}) u_p(k+j-i) + F_{e,l,j}(z) y(k) \\ &\quad - F_{e,l,j}(z) \sum_{p=1}^m \sum_{i=1}^{\infty} g_{l,p,i} (1 - z^{-1}) u_p(k-i). \end{aligned}$$

Vemos, nesta última expressão, que a resposta livre está expressada como função de dois somatórios na forma do modelo de resposta ao degrau do sistema, o que vai permitir usar o modelo compacto  $G_{l,p}(z)$  já definido.

Definindo  $y_{c,l,j}(k) = f_l(k + j \mid k)$ , para simplificar, e aplicando a transformada Z, obtemos:

$$\begin{aligned} Y_{c,l,j}(z) &= \sum_{p=1}^m (G_{l,p}(z) z^j U_p(z) - F_{e,l,j}(z) G_{l,p}(z) U_p(z)) + F_{e,l,j}(z) Y_l(z), \\ &= \sum_{p=1}^m (z^j - F_{e,l,j}(z)) G_{l,p}(z) U_p(z) + F_{e,l,j}(z) Y_l(z), \\ &= \sum_{p=1}^m S_{l,p,j}(z) U_p(z) + F_{e,l,j}(z) Y_l(z), \end{aligned}$$

sendo  $Y_{c,l,j}(z)$ ,  $Y_l(z)$  e  $U_p(z)$  as transformadas Z de  $y_{c,l,j}(k)$ ,  $y_l(k)$  e  $u_p(k)$ , respectivamente, e

$$\begin{aligned} S_{l,p,j}(z) &= (z^j - F_{e,l,j}(z)) G_{l,p}(z), \\ G_{l,p}(z) &= \sum_{i=1}^{\infty} g_{l,p,i} z^{-i} (1 - z^{-1}). \end{aligned}$$

Neste novo modelo, cada  $G_{l,p}(z)$  pode ser considerada uma função de transferência de ordem infinita, mas que pode ser aproximada por uma função de transferência de ordem finita, e dado que o sistema sob análise não é estável, pelo menos para algum  $l$  e  $p$ ,  $G_{l,p}(z)$  tem polos com  $|z| \geq 1$ , e a resposta livre correspondente, representada por  $Y_{c,l,j}(z)$ , não converge quando  $k \rightarrow \infty$ . Para conseguir uma resposta livre  $y_{c,l,j}(k)$  convergente, para cada  $l$ ,  $F_{e,l,j}(z)$  precisa ser estável e ajustado para obter todos os  $S_{l,p,j}(z)$  estáveis, e ainda garantir que  $y_{c,l,j}(k) = y_l(k)$  em regime permanente. Sob estabilidade, esta última condição é garantida se  $F_{e,l,j}(z)$  tiver ganho estático unitário, isto é,  $F_{e,l,j}(1) = 1$ , e  $S_{l,p,j}(z)$  tiver ganho estático nulo, pois, assim  $S_{l,p,j}(1) = 0$  e  $y_{c,l,j}(k) \rightarrow F_{e,l,j}(1) y_l(k) = y_l(k)$  quando  $k \rightarrow \infty$ , ou seja, a resposta livre converge para o valor da saída em regime permanente.

Já a primeira condição, que exige um  $S_{l,p,j}(z) = (z^j - F_{e,l,j}(z)) G_{l,p}(z)$  estável, é garantida se:

$$z^j - F_{e,l,j}(z) = 0 \quad \forall z = z_q, \quad (3.25)$$

sendo  $z_q$  um polo de  $G_{l,p}(z)$  com  $|z_q| \geq 1$ , para todo  $p$ . Note que aqui o filtro pode ter que cumprir várias condições simultaneamente, pois podem existir mais de uma  $G_{l,p}(z)$  com polos com  $|z_q| \geq 1$  e cada uma delas pode ter mais de um polo com  $|z_q| \geq 1$ . Assim, mesmo a condição teórica sendo a mesma do caso SISO apresentada no Volume 1, no caso MIMO a ordem

do filtro precisa em geral ser maior para conseguir que todas as  $S_{l,p,j}(z)$  com  $p = 1 \dots m$  sejam estáveis para uma dada  $l$ .

Caso alguma  $G_{l,p}(z)$  tenha polos múltiplos com  $|z_q| > 1$ ,  $F_{e,l,j}(z)$  deve satisfazer múltiplas condições simultaneamente, que podem ser resumidas na seguinte equação:

$$\frac{d^m(z^j - F_{e,l,j}(z))}{dz^m} = 0 \quad \forall z = z_q, m = 0, 1, 2, \dots, M_q - 1, \quad (3.26)$$

sendo  $M_q$  a multiplicidade da raiz  $z_q$ . Essa condição é obtida a partir das propriedades de polinômios: um polinômio  $p(x)$  com raiz múltipla em  $x = x_0$  satisfaz  $\frac{d^m p(x)}{dx^m} = 0$  para  $x = x_0$  com  $m = 0, 1, 2, \dots, M_l - 1$ . Note que esta condição contém a anterior, apresentada em (3.25), para  $M_l = 1$ .

O caso de plantas integradoras (com  $M_q$  polos em  $z = 1$ ) pode também ser considerado, mudando a condição para:

$$\frac{d^m(z^j - F_{e,l,j}(z))}{dz^m} = 0 \quad z = 1, m = 0, 1, 2, \dots, M_q, \quad (3.27)$$

de forma a garantir o ganho estático unitário do filtro e a eliminação do efeito dos  $M_q$  polos em  $z = 1$  da resposta livre.

Veja que a ordem do filtro  $F_{e,l,j}(z)$  depende do número de polos fora do círculo de todas as  $G_{l,p}(z)$  associadas à saída  $l$ , dado que precisamos de um número de coeficientes ajustáveis no numerador do filtro igual ao número de polos indesejados do modelo do processo. Uma forma simples de definir o filtro é:

$$F_{e,l,j}(z) = \frac{\alpha_{j,n_f} z^{n_f} + \alpha_{j,n_f-1} z^{n_f-1} + \dots + \alpha_{j,0}}{(z - z_f)^{n_f}},$$

onde  $n_f$  define a ordem do filtro e  $z_f$  define o tempo de resposta do filtro. Note que, no caso MIMO, a dinâmica de convergência a zero da resposta livre depende do  $z_f$  e dos polos associados às dinâmicas estáveis mais lentas de todos os modelos do processo associados à saída  $l$ . Assim, analisando todas as  $G_{l,p}(z)$  associadas à saída  $l$ , podemos determinar as dinâmicas lentas não canceladas e, combinando essas dinâmicas com as associadas ao filtro, determinar o tempo necessário para que a resposta livre convirja ( $g_{l,p,j+i} \approx \tilde{g}_{l,p,j,i}$  para  $i > N_f$ ). Observe que  $N_f$  está sempre associado ao tempo de convergência da resposta livre. Assim como sugerido no caso SISO apresentado no Volume 1, podemos ajustar um valor mínimo para  $N_f$  usando o tempo de assentamento de 1% (em amostras) de um sistema auxiliar  $G_{aux}(z)$  que tenha os polos de  $F_{e,l,j}$  e os polos mais lentos não cancelados das  $G_{l,p}(z)$ . Na prática, usar um  $N_f$  maior que esse valor mínimo diminui os erros de truncamento, mais aumenta a complexidade computacional. Assim, há um compromisso na escolha de  $N_f$ .



Na sequência apresentamos um exemplo de ajuste do filtro e de escolha de  $N_f$  para um sistema MIMO  $2 \times 2$ .

**Exemplo 3.12.** *Neste exemplo vamos considerar a aplicação do GDMC para o processo integrador do Exemplo 3.11. Mostraremos como as dificuldades observadas naquele caso para o ajuste do controle DMC não são observadas quando usamos o GDMC.*

*Dado que a matriz de transferência do sistema possui uma dinâmica integradora no elemento (1,1), é necessário projetar os filtros do erro de predição associados à saída 1 para que a resposta livre seja convergente. As condições que precisam ser satisfeitas são:*

$$\begin{aligned} F_{e,1,j}(z)|_{z=1} &= 1 \rightarrow \text{condição de ganho unitário} \\ \left. \frac{d(z^j - F_{e,1,j}(z))}{dz} \right|_{z=1} &= 0 \rightarrow \text{condição de estabilidade} \end{aligned}$$

*Como não existem polos instáveis ou integradores associados à saída 2, não é necessário filtro do erro de predição para essa saída.*

*Como duas condições precisam ser satisfeitas para  $F_{e,1,j}(z)$ , um filtro de primeira ordem tem parâmetros livres suficientes para garantir ambas as condições. Assim, a estrutura do filtro é dada por:*

$$F_{e,1,j} = \frac{a(j)z + b(j)}{z - z_f},$$

*sendo  $z_f$  um parâmetro de projeto e  $a(j), b(j)$  os parâmetros a calcular para satisfazer as condições. Com este filtro, o termo envolvendo a derivada da condição de estabilidade se torna:*

$$\begin{aligned} \frac{d(z^j - F_{e,1,j}(z))}{dz} &= jz^{j-1} - \frac{a(j)(z - z_f) - (a(j)z + b(j))}{(z - z_f)^2}, \\ &= jz^{j-1} - \frac{-a(j)z_f - b(j)}{(z - z_f)^2}, \end{aligned}$$

*Agora podemos calcular os coeficientes  $a(j)$  e  $b(j)$  de tal modo que as condições acima sejam satisfeitas para  $j = N_1 \dots N_2$ :*

$$\begin{aligned} \frac{a(j) + b(j)}{1 - z_f} &= 1 \rightarrow a(j) + b(j) = 1 - z_f \\ j - \frac{-a(j)z_f - b(j)}{(1 - z_f)^2} &= 0 \rightarrow -z_f a(j) - b(j) = j(1 - z_f)^2 \end{aligned}$$

e note que estas últimas equações podem ser rearranjadas como um conjunto de equações lineares que é de fácil solução para cada  $j$ :

$$\begin{bmatrix} 1 & 1 \\ -z_f & -1 \end{bmatrix} \begin{bmatrix} a(j) \\ b(j) \end{bmatrix} = \begin{bmatrix} (1 - z_f) \\ j(1 - z_f)^2 \end{bmatrix}.$$

A sintonia do GDMC é idêntica à do DMC do Exemplo 3.11, exceto que agora, para a saída 1, temos o filtro do erro de predição. Neste caso, escolhemos  $z_f = 0,8$ , que equivale a uma constante de tempo de aproximadamente 0,45s. Para calcular o horizonte  $N_f$ , utilizaremos a regra prática discutida anteriormente. Para a saída 1, fora a dinâmica cancelada, temos ainda uma dinâmica de primeira ordem com constante de tempo de 2s no elemento (1,2) da matriz de transferência do processo. Combinando esta constante de tempo com a do filtro, o tempo de 1 % é de 10,6s, e dado que o período de amostragem é 0,1s, então  $N_f \approx 100$ .

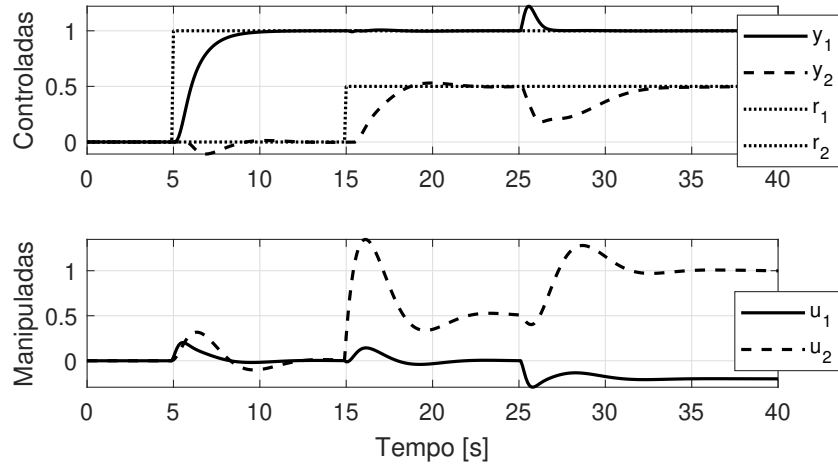


Figura 3.5: Exemplo 3.12 – Resposta e sinal de controle do evaporador em malha fechada com o GDMC.

Os resultados de simulação são apresentados na Figura 3.5. Note que a resposta do sistema é consideravelmente melhor, tem menos oscilações e tem uma rejeição de perturbação mais rápida do que no caso apresentado no Exemplo 3.11, em que a dinâmica integradora foi aproximada para poder aplicar o DMC clássico (Figura 3.4).

Como visto nesta seção, o GDMC MIMO, que é uma generalização do DMC MIMO clássico, permite realizar o projeto de controle de plantas multivariáveis estáveis e não estáveis mantendo a simplicidade de representação do processo mediante o uso de modelos de resposta ao degrau. Ainda, destacamos que a extensão para o caso MIMO do DMC e o GDMC, se realiza

de forma relativamente simples graças a estrutura de cálculo das predições de forma independente por variável controlada.

Na próxima seção realizaremos um estudo similar ao apresentado para o DMC, para tratar do GPC MIMO. Posteriormente apresentaremos alguns exemplos comparativos de plantas MIMO usando o DMC e GPC.

### 3.3 GPC multivariável

Nesta seção, apresentaremos a formulação multivariável do algoritmo GPC já considerando a inclusão de ação antecipativa. Na apresentação do caso mono-variável tratado no Volume 1 desta obra, mostramos que o modelo CARIMA era obtido realizando uma manipulação matemática dos numeradores e denominadores das funções de transferência do sistema de forma a ter todos os denominadores de todas as funções de transferência iguais, inclusive considerando também as perturbações medidas e não medidas. Aqui utilizaremos o mesmo procedimento.

Primeiro, note que um sistema multivariável com  $n$  saídas,  $m$  entradas e  $m_q$  perturbações medidas pode ser representado por um modelo do tipo

$$\mathbf{Y}(z) = \mathbf{P}(z)\mathbf{U}(z) + \mathbf{P}_q(z)\mathbf{Q}(z),$$

sendo  $\mathbf{U}(z)$  o vetor de variáveis manipuladas,  $\mathbf{Y}(z)$  o vetor de saídas,  $\mathbf{Q}(z)$  o vetor das perturbações e  $\mathbf{P}(z)$  e  $\mathbf{P}_q(z)$  matrizes de funções de transferência:

$$\mathbf{P}(z) = \begin{bmatrix} P_{1,1}(z) & P_{1,2}(z) & \dots & P_{1,m}(z) \\ P_{2,1}(z) & P_{2,2}(z) & \dots & P_{2,m}(z) \\ \vdots & \vdots & \ddots & \vdots \\ P_{n,1}(z) & P_{n,2}(z) & \dots & P_{n,m}(z) \end{bmatrix},$$

$$\mathbf{P}_q(z) = \begin{bmatrix} P_{q,1,1}(z) & P_{q,1,2}(z) & \dots & P_{q,1,m_q}(z) \\ P_{q,2,1}(z) & P_{q,2,2}(z) & \dots & P_{q,2,m_q}(z) \\ \vdots & \vdots & \ddots & \vdots \\ P_{q,n,1}(z) & P_{q,n,2}(z) & \dots & P_{q,n,m_q}(z) \end{bmatrix}.$$

Ainda, podemos colocar cada função de transferência  $P_{i,j}(z)$  explicitando numerador, denominador e atraso:

$$\mathbf{P}(z) = \begin{bmatrix} \frac{N_{1,1}(z)}{D_{1,1}(z)} z^{-d_{1,1}} & \dots & \frac{N_{1,m}(z)}{D_{1,m}(z)} z^{-d_{1,m}} \\ \vdots & \ddots & \vdots \\ \frac{N_{n,1}(z)}{D_{n,1}(z)} z^{-d_{n,1}} & \dots & \frac{N_{n,m}(z)}{D_{n,m}(z)} z^{-d_{n,m}} \end{bmatrix}, \quad (3.28)$$

onde  $N_{l,p}(z)$ ,  $D_{l,p}(z)$  e  $d_{l,p}$  representam, respectivamente, o numerador, o denominador e o atraso de transporte discreto da função de transferência que relaciona a entrada  $p$  com a saída  $l$ . O mesmo pode ser feito com os elementos de  $\mathbf{P}_q(z)$ , definindo  $P_{q,i}(z) = \frac{N_{q,i,j}(z)}{D_{q,i,j}(z)} z^{-d_{q,i,j}}$ . Observe que o modelo multivariável pode ser visto como  $n$  sistemas de uma única saída e  $m + m_q$  entradas onde, para cada saída  $Y_l(z)$ :

$$Y_l(z) = P_{l,1}(z)U_1(z) + P_{l,2}(z)U_2(z) + \dots + P_{l,m}(z)U_m(z) \\ + P_{q,l,1}(z)Q_1(z) + P_{q,l,2}(z)Q_2(z) + \dots + P_{q,l,m_q}(z)Q_{m_q}(z).$$

Assim, se calcularmos o mínimo múltiplo comum  $A_l(z)$  dos denominadores de todas as funções de transferência do modelo da saída  $l$ , podemos escrever:

$$B_{i,j}(z) = A_l(z) \frac{N_{i,j}(z)}{D_{i,j}(z)}, \quad B_{q,i,j}(z) = A_l(z) \frac{N_{q,i,j}(z)}{D_{q,i,j}(z)}. \quad (3.29)$$

Veja que, desta forma, todas as as funções de transferência de uma linha têm o mesmo denominador  $A_l(z)$  e recalculamos os numeradores da mesma forma como realizado no caso monovariável (com modelo de perturbação). Assim, utilizando agora a nomenclatura com polinômios em  $z^{-1}$ , podemos escrever um modelo para cada saída  $l$  como:

$$A_l(z^{-1})y(k) = \sum_{p=1}^m B_{l,p}(z^{-1})z^{-d_{l,p}}u_p(k-1) \\ + \sum_{p=1}^{m_q} B_{q,l,p}(z^{-1})z^{-d_{q,l,p}}q_p(k), \quad (3.30)$$

e, assim como no caso monovariável, adicionando uma perturbação não medida  $\frac{C_l(z^{-1})}{\Delta}e(k)$ , com  $e(k)$  sendo um ruído branco com média nula, chegamos ao modelo CARIMA de uma saída,  $m$  entradas e  $m_q$  perturbações medidas:

$$A_l(z^{-1})y(k) = \sum_{p=1}^m B_{l,p}(z^{-1})z^{-d_{l,p}}u_p(k-1) \\ + \sum_{p=1}^{m_q} B_{q,l,p}(z^{-1})z^{-d_{q,l,p}}q_p(k) + C_l(z^{-1})\frac{e(k)}{\Delta}, \quad (3.31)$$

onde  $A_l(z^{-1})$  e  $C_l(z^{-1})$  são polinômios de grau  $n_{a,l}$  e  $n_{c,l}$ , respectivamente,  $B_{l,p}(z^{-1})$  possui grau  $n_{b,l,p}$ ,  $d_{l,p}$  é o atraso entre a entrada  $p$  e a saída  $l$ ,

$B_{q,l,p}(z^{-1})$  possui grau  $n_{bq,l,p}$ , e  $d_{q,l,p}$  é o atraso entre a perturbação medível  $p$  e a saída  $l$ . Apesar de a notação parecer complicada, o procedimento é exatamente o mesmo que foi usado no caso monovariável já apresentado no Volume 1. Porém, no caso monovariável com perturbações medíveis são consideradas três entradas (variáveis manipulada, perturbação medível e não medível), e no caso multivariável o procedimento é generalizado para considerar as  $m$  manipuladas do processo e  $m_q$  perturbações medíveis para cada uma das  $n$  saídas.

Considerando um sistema com  $n$  saídas,  $m$  manipuladas e  $m_q$  perturbações medíveis, é possível representar o sistema completo de forma matricial da seguinte forma:

$$\begin{aligned} \mathbf{A}(z^{-1})\mathbf{y}(k) &= (\mathbf{B}(z^{-1}) \circ \mathbf{D}(z^{-1})) \mathbf{u}(k-1) \\ &+ (\mathbf{B}_q(z^{-1}) \circ \mathbf{D}_q(z^{-1})) \mathbf{q}(k) + \mathbf{C}(z^{-1}) \frac{e(k)}{\Delta}, \end{aligned} \quad (3.32)$$

com

$$\begin{aligned} \mathbf{A}(z^{-1}) &= \begin{bmatrix} A_1(z^{-1}) & & \\ & \ddots & \\ & & A_n(z^{-1}) \end{bmatrix}, \quad \mathbf{B}(z^{-1}) = \begin{bmatrix} B_{1,1} & \cdots & B_{1,m} \\ \vdots & \ddots & \vdots \\ B_{n,1} & \cdots & B_{n,m} \end{bmatrix}, \\ \mathbf{D}(z^{-1}) &= \begin{bmatrix} z^{-d_{1,1}} & \cdots & z^{-d_{1,m}} \\ \vdots & \ddots & \vdots \\ z^{-d_{n,1}} & \cdots & z^{-d_{n,m}} \end{bmatrix}, \quad \mathbf{B}_q(z^{-1}) = \begin{bmatrix} B_{q,1,1} & \cdots & B_{q,1,m_q} \\ \vdots & \ddots & \vdots \\ B_{q,n,1} & \cdots & B_{q,n,m_q} \end{bmatrix}, \\ \mathbf{D}_q(z^{-1}) &= \begin{bmatrix} z^{-d_{q,1,1}} & \cdots & z^{-d_{q,1,m_q}} \\ \vdots & \ddots & \vdots \\ z^{-d_{q,n,1}} & \cdots & z^{-d_{q,n,m_q}} \end{bmatrix}, \quad \mathbf{C}(z^{-1}) = \begin{bmatrix} C_1(z^{-1}) & & \\ & \ddots & \\ & & C_n(z^{-1}) \end{bmatrix}, \\ \mathbf{y}(k) &= \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_n(k) \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_m(k) \end{bmatrix}, \quad \mathbf{q}(k) = \begin{bmatrix} q_1(k) \\ q_2(k) \\ \vdots \\ q_{m_q}(k) \end{bmatrix}, \end{aligned}$$

e onde  $\mathbf{A}(z^{-1})$ ,  $\mathbf{B}(z^{-1})$ ,  $\mathbf{B}_q(z^{-1})$ ,  $\mathbf{D}(z^{-1})$ ,  $\mathbf{D}_q(z^{-1})$  e  $\mathbf{C}(z^{-1})$  são matrizes polinomiais com  $\mathbf{A}(z^{-1})$  e  $\mathbf{C}(z^{-1})$  diagonais (os termos fora da diagonal não representados são nulos). Os termos  $\mathbf{B}(z^{-1}) \circ \mathbf{D}(z^{-1})$  e  $\mathbf{B}_q(z^{-1}) \circ \mathbf{D}_q(z^{-1})$  fazem uso do produto matricial de Hadamard, indicado por  $\circ$ , que é obtido a partir de duas matrizes  $n \times m$  multiplicando-se cada elemento  $i,j$  das matrizes originais, ou seja,  $(\mathbf{B} \circ \mathbf{D})_{i,j} = B_{i,j} \cdot D_{i,j}$  (produto elemento a elemento).

Logo, o resultado do produto de Hadamard de duas matrizes  $n \times m$  é também uma matriz com dimensão  $n \times m$ .

Na prática, é possível obter uma representação CARIMA de um processo MIMO como em (3.32) diretamente num processo de identificação mas, em geral, é mais comum que as funções de transferência entre cada par  $l, p$  do sistema sejam identificadas individualmente, gerando uma matriz de transferência que, posteriormente, é processada para unificação dos denominadores por saída. O método aqui descrito para se obter a representação CARIMA (3.32) é conhecido como o de obtenção da descrição matricial fracionária (MFD, do inglês *Matrix Fraction Description*), de tal maneira que matrizes polinomiais são usadas para descrever a matriz de funções de transferência:

$$\begin{aligned}\mathbf{P}(z^{-1}) &= \mathbf{A}(z)^{-1} (\mathbf{B}(z) \circ \mathbf{D}(z)), \\ \mathbf{P}_q(z^{-1}) &= \mathbf{A}(z)^{-1} (\mathbf{B}_q(z) \circ \mathbf{D}_q(z)).\end{aligned}$$

Vamos ilustrar no próximo exemplo como obter a MDF do processo a partir das matrizes função de transferência originais.

**Exemplo 3.13.** *Retomando o processo do Exemplo 3.10 que tratou de um tanque termicamente isolado, considerando um período de amostragem de  $T_s = 1$  s, temos o seguinte modelo discreto:*

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} \frac{0,1785z^{-1}}{z - 0,8752} & \frac{-0,3236z^{-1}}{z - 0,9281} \\ 0 & \frac{0,3762}{z - 0,9342} \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \end{bmatrix} + \begin{bmatrix} \frac{0,1555z^{-1}}{z - 0,9260} \\ \frac{-0,2052}{z - 0,9342} \end{bmatrix} Q(z)$$

Para obtermos a representação MFD, podemos analisar as funções de transferências relacionadas a cada saída de forma independente. Para a primeira saída, temos os seguintes denominadores:

$$D_{1,1}(z) = z - 0,8752, \quad D_{1,2}(z) = z - 0,9281, \quad D_{q,1,1}(z) = z - 0,9260.$$

Veja que todos possuem raízes distintas, então o mínimo múltiplo comum destes polinômios é simplesmente a multiplicação dos mesmos, assim,

$$\begin{aligned}A_1(z) &= (z - 0,8752)(z - 0,9281)(z - 0,9260) \\ A_1(z) &= z^3 - 2,7292z^2 + 2,4820z - 0,7521\end{aligned}$$

Realizando o mesmo processo para a segunda saída, temos

$$D_{2,2}(z) = z - 0,9342, \quad D_{q,2,1} = z - 0,9342,$$

e veja que, neste caso, os polinômios possuem a mesma raiz, assim, o mínimo múltiplo comum é

$$A_2(z) = z - 0,9342.$$

Com os polinômios  $A_i(z)$ , podemos obter  $B_{i,j}(z)$  e  $B_{q,i,j}(z)$  para completar o modelo no formato MFD desejado utilizando (3.29):

$$B_{1,1}(z) = A_1(z) \frac{N_{1,1}(z)}{D_{1,1}(z)}$$

$$B_{1,1}(z) = (z - 0,8752)(z - 0,9281)(z - 0,9260) \frac{0,1785}{(z - 0,8752)}$$

$$B_{1,1}(z) = 0,1785(z - 0,9281)(z - 0,9260)$$

$$B_{1,2}(z) = A_1(z) \frac{N_{1,2}(z)}{D_{1,2}(z)}$$

$$B_{1,2}(z) = (z - 0,8752)(z - 0,9281)(z - 0,9260) \frac{-0,3236}{(z - 0,9281)}$$

$$B_{1,2}(z) = -0,3236(z - 0,8752)(z - 0,9260)$$

$$B_{q,1,1}(z) = A_1(z) \frac{N_{q,1,1}(z)}{D_{q,1,1}(z)}$$

$$B_{q,1,1}(z) = (z - 0,8752)(z - 0,9281)(z - 0,9260) \frac{0,1555}{z - 0,9260}$$

$$B_{q,1,1}(z) = 0,1555(z - 0,8752)(z - 0,9281)$$

$$B_{2,2}(z) = A_2(z) \frac{N_{2,2}(z)}{D_{2,2}(z)}$$

$$B_{2,2}(z) = (z - 0,9342) \frac{0,3762}{(z - 0,9342)}$$

$$B_{2,2}(z) = 0,3762$$

$$B_{q,2,1}(z) = A_2(z) \frac{N_{q,2,1}(z)}{D_{q,2,1}(z)}$$

$$B_{q,2,1}(z) = (z - 0,9342) \frac{-0,2052}{(z - 0,9342)}$$

$$B_{q,2,1}(z) = -0,2052$$

Com estes polinômios, podemos montar as matrizes do modelo MFD discreto do sistema:

$$\mathbf{A}(z) = \begin{bmatrix} (z - 0,8752)(z - 0,9281)(z - 0,9260) & 0 \\ 0 & z - 0,9342 \end{bmatrix},$$

$$\mathbf{B}(z) = \begin{bmatrix} 0,1785(z - 0,9281)(z - 0,9260) & -0,3236(z - 0,8752)(z - 0,9260) \\ 0 & 0,3762 \end{bmatrix},$$

$$\mathbf{B}_q(z) = \begin{bmatrix} 0,1555(z - 0,8752)(z - 0,9281) \\ -0,2052 \end{bmatrix}.$$

Uma vez obtido o modelo CARIMA MIMO, os passos discutidos para o caso SISO podem ser repetidos para cada saída  $l$ , ou seja, a partir de (3.31), supondo inicialmente  $C_l(z^{-1}) = 1$ ,  $\forall l$ , calculamos um conjunto de polinômios  $E_{l,j}(z^{-1})$  e  $F_{l,j}(z^{-1})$  por meio de equações diofantinas

$$1 = E_{l,j}(z^{-1})\Delta A_l(z^{-1}) + z^{-j}F_{l,j}(z^{-1}),$$

para obter as predições futuras do sistema em função dos dados passados e controles futuros:

$$\begin{aligned} \hat{y}_l(k+j | k) = & \sum_{p=1}^m E_{l,j}(z^{-1})B_{l,p}(z^{-1})\Delta u_p(k+j-d_{l,p}-1) \\ & + \sum_{p=1}^{m_q} E_{l,j}(z^{-1})B_{q,l,p}(z^{-1})\Delta q_p(k+j-d_{q,l,p}) \\ & + F_{l,j}(z^{-1})y_l(k). \end{aligned} \quad (3.33)$$

Os dois primeiros termos da equação anterior dependem de valores futuros e passados de  $\Delta u_p$  e  $\Delta q_p$ , respectivamente. Assim, no caso da perturbação, é interessante fazer a separação entre termos futuros e passados pois nem sempre é possível saber seus valores futuros e, no caso da entrada, a separação é importante para evidenciar as respostas livre e forçada da predição:

$$\begin{aligned} \hat{y}_l(k+j | k) = & \sum_{p=1}^m [G_{l,p,j}(z^{-1})\Delta u_p(k+j-1) + H_{l,p,j}(z^{-1})\Delta u_p(k-1)] \\ & + \sum_{p=1}^{m_q} [G_{q,l,p,j}(z^{-1})\Delta q_p(k+j) + H_{q,l,p,j}(z^{-1})\Delta q_p(k)] \\ & + F_{l,j}(z^{-1})y_l(k), \end{aligned}$$

sendo que  $G_{l,p,j}(z^{-1})$  e  $H_{l,p,j}(z^{-1})$  são obtidos a partir do polinômio  $E_{l,j}(z^{-1})B_{l,p}(z^{-1})$ , e  $G_{l,p,j}(z^{-1})$  representa os coeficientes da resposta ao degrau da saída  $l$  em relação à entrada  $p$ . Da mesma forma,  $G_{q,l,p,j}(z^{-1})$  e  $H_{q,l,p,j}(z^{-1})$  são obtidos a partir do polinômio  $E_{l,j}(z^{-1})B_{q,l,p}(z^{-1})$ , e  $G_{q,l,p,j}(z^{-1})$  representa os coeficientes da resposta ao degrau da saída  $l$  em relação à perturbação  $p$ .  $G_{l,p,j}(z^{-1})$  tem grau  $j - d_{l,p} - 1$  quando  $j > d_{l,p}$ , e se  $j \leq d_{l,p}$ ,



então  $G_{l,p,j}(z^{-1}) = 0$  por conta do efeito do atraso na resposta do sistema. Note que  $E_j(z^{-1})B_{l,p}(z^{-1})\Delta u_p(k+j-d_{l,p}-1)$  só depende de valores passados da entrada se  $j \leq d_{l,p}$ . Já  $H_{l,p,j}(z^{-1})$  é um polinômio de grau fixo  $n_{b,l,p} + d_{l,p}$ . De maneira similar,  $G_{q,l,p,j}(z^{-1})$  tem grau  $j - d_{q,l,p} - 1$  quando  $j > d_{q,l,p}$ , e se  $j \leq d_{q,l,p}$ , então  $G_{q,l,p,j}(z^{-1}) = 0$  por conta do efeito do atraso, e o termo  $E_j(z^{-1})B_{q,l,p}(z^{-1})$  só depende de valores passados da perturbação. Por fim,  $H_{q,l,p,j}(z^{-1})$  é um polinômio de grau fixo  $n_{bq,l,p} + d_{q,l,p} - 1$ .

Então, as previsões futuras da saída  $l$  (considerando o horizonte de 1 a  $N_{2,l}$ ) podem ser escritas de forma matricial da seguinte maneira:

$$\hat{\mathbf{y}}_l = \sum_{p=1}^m \mathbf{G}_{l,p} \Delta \mathbf{u}_p + \mathbf{f}_l, \quad (3.34)$$

onde  $\mathbf{G}_{l,p} \in \mathbb{R}^{N_{2,l} \times N_{u,p}}$  é a matriz de coeficientes da resposta ao degrau e  $\mathbf{f}_l \in \mathbb{R}^{N_{2,l}}$  é o vetor de resposta livre do sistema,  $N_{2,l}$  é o horizonte de previsão da saída  $l$ ,  $N_{u,p}$  é o horizonte de controle da entrada  $p$ ,  $N_{q,p}$  é o horizonte da perturbação  $p$ , e

$$\mathbf{f}_l = \begin{bmatrix} f_l(k+1 | k) \\ f_l(k+2 | k) \\ \vdots \\ f_l(k+N_{2,l} | k) \end{bmatrix} = \mathbf{F}_l \underline{\mathbf{y}}^l + \sum_{p=1}^m \mathbf{H}_{l,p} \underline{\Delta \mathbf{u}}_p + \sum_{p=1}^{m_q} \mathbf{G}_{q,l,p} \Delta \mathbf{q}_p + \sum_{p=1}^{m_q} \mathbf{H}_{q,l,p} \underline{\Delta \mathbf{q}}_p,$$

com

$$\mathbf{F}_l = \begin{bmatrix} f_{l,1,0} & f_{l,1,1} & \cdots & f_{l,1,n_{a,l}} \\ f_{l,2,0} & f_{l,2,1} & \cdots & f_{l,2,n_{a,l}} \\ \vdots & \vdots & \ddots & \vdots \\ f_{l,N_{2,0}} & f_{l,N_{2,1}} & \cdots & f_{l,N_{2,n_{a,l}}} \end{bmatrix}, \quad \underline{\mathbf{y}}^l = \begin{bmatrix} y_l(k) \\ y_l(k-1) \\ \vdots \\ y_l(k-n_{a,l}) \end{bmatrix},$$

$$\underline{\Delta \mathbf{u}}_p = \begin{bmatrix} \Delta u_p(k-1) \\ \Delta u_p(k-2) \\ \vdots \\ \Delta u_p(k-n_{b,l,p}-d_{l,p}) \end{bmatrix}, \quad \mathbf{H}_{l,p} = \begin{bmatrix} h_{l,p,1,1} & h_{l,p,1,2} & \cdots & h_{l,p,1,n_{b,l,p}+d_{l,p}} \\ h_{l,p,2,1} & h_{l,p,2,2} & \cdots & h_{l,p,2,n_{b,l,p}+d_{l,p}} \\ \vdots & \vdots & \ddots & \vdots \\ h_{l,p,N_{2,1}} & h_{l,p,N_{2,2}} & \cdots & h_{l,p,N_{2,n_{b,l,p}+d_{l,p}}} \end{bmatrix},$$

$$\mathbf{G}_{l,p} = \begin{bmatrix} g_{l,p,1} & 0 & \cdots & 0 \\ g_{l,p,2} & g_{l,p,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{l,p,N_{2,l}} & g_{l,p,N_{2,l}-1} & \cdots & g_{l,p,N_{2,l}-N_{u,p}+1} \end{bmatrix}, \quad \Delta \mathbf{u}_p = \begin{bmatrix} \Delta u_p(k) \\ \Delta u_p(k+1) \\ \vdots \\ \Delta u_p(k+N_{u,p}-1) \end{bmatrix},$$

$$\begin{aligned}
\overleftarrow{\Delta \mathbf{q}_p} &= \begin{bmatrix} \Delta q_p(k) \\ \Delta q_p(k-1) \\ \vdots \\ \Delta q_p(k - n_{bq,l,p} - d_{l,p} + 1) \end{bmatrix}, \quad \Delta \mathbf{q}_p = \begin{bmatrix} \Delta q_p(k+1) \\ \Delta q_p(k+2) \\ \vdots \\ \Delta q_p(k + N_{q,p}) \end{bmatrix}, \\
\mathbf{H}_{q,l,p} &= \begin{bmatrix} h_{q,l,p,1,0} & h_{q,l,p,1,1} & \cdots & h_{l,p,1,n_{qb,l,p}+d_{q,l,p}-1} \\ h_{q,l,p,2,0} & h_{q,l,p,2,1} & \cdots & h_{l,p,2,n_{qb,l,p}+d_{q,l,p}-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{q,l,p,N_{2,l},0} & h_{q,l,p,N_{2,l},1} & \cdots & h_{l,p,N_{2,l},n_{qb,l,p}+d_{q,l,p}-1} \end{bmatrix}, \\
\mathbf{G}_{q,l,p} &= \begin{bmatrix} g_{q,l,p,0} & 0 & \cdots & 0 \\ g_{q,l,p,1} & g_{q,l,p,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{q,l,p,N_{2,l}-1} & g_{q,l,p,N_{2,l}-2} & \cdots & g_{q,l,p,N_{2,l}-N_{q,p}} \end{bmatrix},
\end{aligned}$$

no qual  $\mathbf{H}_{l,p} \in \mathbb{R}^{N_{2,l} \times n_{b,l,p} + d_{l,p}}$ ,  $\mathbf{H}_{q,l,p} \in \mathbb{R}^{N_{2,l} \times n_{bq,l,p} + d_{l,p}}$ ,  $\mathbf{G}_{q,l,p} \in \mathbb{R}^{N_{2,l} \times N_{q,p}}$  e  $\mathbf{F} \in \mathbb{R}^{N_2 \times n_{a,l} + 1}$  são matrizes com os coeficientes dos polinômios  $H_{l,p,j}(z^{-1})$ ,  $H_{q,l,p,j}(z^{-1})$ ,  $G_{q,l,p}(z^{-1})$  e  $F_{l,j}(z^{-1})$ , respectivamente, e, da mesma forma que no caso monovariável, se o horizonte inicial de predição da saída  $l$  for maior do que um ( $N_{1,l} > 1$ ), as  $N_{1,l} - 1$  linhas iniciais de  $\mathbf{G}_{l,p}$  e  $\mathbf{f}_l$  devem ser removidas.

Combinando as predições de todas as  $n$  saídas, dadas por (3.34), obtemos o formato padrão

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f} \quad (3.35)$$

com  $\mathbf{f} = \mathbf{F} \underline{\mathbf{y}} + \mathbf{H} \overleftarrow{\Delta \mathbf{u}} + \mathbf{G}_q \Delta \mathbf{q} + \mathbf{H}_q \overleftarrow{\Delta \mathbf{q}}$ , e

$$\begin{aligned}
\mathbf{G} &= \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{1,m} \\ \mathbf{G}_{2,1} & \cdots & \mathbf{G}_{1,m} \\ \vdots & \vdots & \vdots \\ \mathbf{G}_{n,1} & \cdots & \mathbf{G}_{n,m} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}, \quad \Delta \mathbf{u} = \begin{bmatrix} \Delta \mathbf{u}_1 \\ \Delta \mathbf{u}_2 \\ \vdots \\ \Delta \mathbf{u}_m \end{bmatrix} \\
\mathbf{H} &= \begin{bmatrix} \mathbf{H}_{1,1} & \cdots & \mathbf{H}_{1,m} \\ \mathbf{H}_{2,1} & \cdots & \mathbf{H}_{1,m} \\ \vdots & \vdots & \vdots \\ \mathbf{H}_{n,1} & \cdots & \mathbf{H}_{n,m} \end{bmatrix}, \quad \overleftarrow{\Delta \mathbf{u}} = \begin{bmatrix} \overleftarrow{\Delta \mathbf{u}}_1 \\ \overleftarrow{\Delta \mathbf{u}}_2 \\ \vdots \\ \overleftarrow{\Delta \mathbf{u}}_m \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & & \\ & \ddots & \\ & & \mathbf{F}_n \end{bmatrix}, \\
\underline{\mathbf{y}} &= \begin{bmatrix} \underline{\mathbf{y}}^1 \\ \underline{\mathbf{y}}^2 \\ \vdots \\ \underline{\mathbf{y}}^n \end{bmatrix}, \quad \mathbf{G}_q = \begin{bmatrix} \mathbf{G}_{q,1,1} & \cdots & \mathbf{G}_{q,1,m_q} \\ \mathbf{G}_{q,2,1} & \cdots & \mathbf{G}_{q,1,m_q} \\ \vdots & \vdots & \vdots \\ \mathbf{G}_{q,n,1} & \cdots & \mathbf{G}_{q,n,m_q} \end{bmatrix}, \quad \Delta \mathbf{q} = \begin{bmatrix} \Delta \mathbf{q}_1 \\ \Delta \mathbf{q}_2 \\ \vdots \\ \Delta \mathbf{q}_{m_q} \end{bmatrix},
\end{aligned}$$

$$\mathbf{H}_q = \begin{bmatrix} \mathbf{H}_{q,1,1} & \cdots & \mathbf{H}_{q,1,m_q} \\ \mathbf{H}_{q,2,1} & \cdots & \mathbf{H}_{q,2,m_q} \\ \vdots & \vdots & \vdots \\ \mathbf{H}_{q,n,1} & \cdots & \mathbf{H}_{q,n,m_q} \end{bmatrix}, \quad \overleftarrow{\Delta \mathbf{q}} = \begin{bmatrix} \overleftarrow{\Delta \mathbf{q}_1} \\ \overleftarrow{\Delta \mathbf{q}_2} \\ \vdots \\ \overleftarrow{\Delta \mathbf{q}_{m_q}} \end{bmatrix},$$

com  $\mathbf{F}$  bloco diagonal. Um detalhe importante que deve ser observado é que, para uma dada entrada  $p$ , é possível ter polinômios  $\mathbf{H}_{l,p,j}(z^{-1})$  com graus distintos, fazendo com que as matrizes  $\mathbf{H}_{l,p}$  tenham diferentes números de colunas, inviabilizando a obtenção de  $\mathbf{H}$  por concatenação direta. Por isso, é importante observar qual o polinômio  $H_{l,p,j}(z^{-1})$  de maior grau, ou seja, obter o valor  $n_{H,p} = \sup\{n_{b,l,p} + d_{l,p} : \forall l\}$ . Assim, é possível definir o grau adequado de  $\overleftarrow{\Delta \mathbf{u}}_p$  e também saber o número de colunas com que cada  $\mathbf{H}_{l,p}$  deve ser preenchido para ser possível a obtenção de  $\mathbf{H}$ . Note que  $\mathbf{H}_{l,p}$  deverá ser completada à direita com colunas de zeros sempre que  $n_{b,l,p} + d_{l,p} \leq n_{H,p}$ . No exemplo a seguir, este ponto é discutido de forma mais clara.

**Exemplo 3.14.** *A partir do Exemplo 3.13, no qual calculamos a representação MFD de um sistema de tanque termicamente isolado, vamos, agora, obter a representação matricial das predições para este processo. A título de ilustração, utilizaremos horizontes curtos, assim,  $N_{1,1} = N_{2,1} = 1$ ,  $N_{2,1} = N_{2,2} = 3$ ,  $N_q = 2$ ,  $N_{u,1} = 2$  e  $N_{u,2} = 1$ . Os polinômios do modelo MFD são:*

$$\begin{aligned} A_1(z) &= 1 - 2,7292z^{-1} + 2,482z^{-2} - 0,7521z^{-3} \\ A_2(z) &= 1 - 0,9342z^{-1} \\ B_{1,1}(z) &= 0,1785 - 0,331z^{-1} + 0,1534z^{-2} \\ B_{1,2}(z) &= -0,3236 + 0,5828z^{-1} - 0,2622z^{-2} \\ B_{2,1}(z) &= 0 \\ B_{2,2}(z) &= 0,3762 \\ B_{q,1,1}(z) &= 0 + 0,1555z^{-1} - 0,2804z^{-2} + 0,1263z^{-3} \\ B_{q,2,1}(z) &= 0 - 0,2052z^{-1} \end{aligned}$$

*Calculando aos polinômios  $F_{i,j}(z^{-1})$  e  $E_{i,j}(z^{-1})$ , a partir das equações diofantinas, temos*

$$\begin{aligned} F_{1,1}(z^{-1}) &= 3,7292 - 5,2112z^{-1} + 3,2341z^{-2} - 0,7521z^{-3} \\ F_{1,2}(z^{-1}) &= 8,6959 - 16,1997z^{-1} + 11,3086z^{-2} - 2,8048z^{-3} \\ F_{1,3}(z^{-1}) &= 16,2293 - 34,0076z^{-1} + 25,3186z^{-2} - 6,5402z^{-3} \\ F_{2,1}(z^{-1}) &= 1,9342 - 0,9342z^{-1} \\ F_{2,2}(z^{-1}) &= 2,807 - 1,807z^{-1} \\ F_{2,3}(z^{-1}) &= 3,6224 - 2,6224z^{-1} \end{aligned}$$

$$\begin{aligned}
E_{1,1}(z^{-1}) &= 1 \\
E_{1,2}(z^{-1}) &= 1 + 3,7292z^{-1} \\
E_{1,3}(z^{-1}) &= 1 + 3,7292z^{-1} + 8,6959z^{-2} \\
E_{2,1}(z^{-1}) &= 1 \\
E_{2,2}(z^{-1}) &= 1 + 1,9342z^{-1} \\
E_{2,3}(z^{-1}) &= 1 + 1,9342z^{-1} + 2,807z^{-2}
\end{aligned}$$

Analisando os polinômios  $F_{i,j}(z^{-1})$ , podemos concluir que precisaremos de informações passadas de  $k$  até  $k - 3$  para a saída 1, e de  $k$  até  $k - 1$  para a saída 2, o que é condizente com a ordem os polinômios originais  $A_i(z^{-1})$  da representação MFD, que são de ordem 3 e 1, respectivamente. As matrizes  $\mathbf{F}_i$  podem ser obtidas diretamente dos polinômios  $F_{i,j}(z^{-1})$ . A matriz  $\mathbf{F}$  resultante é, então,

$$\mathbf{F} = \left[ \begin{array}{cccc|cc} 3,7292 & -5,2112 & 3,2341 & -0,7521 & 0 & 0 \\ 8,6959 & -16,1997 & 11,3086 & -2,8048 & 0 & 0 \\ 16,2293 & -34,0076 & 25,3186 & -6,5402 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1,9342 & -0,93424 \\ 0 & 0 & 0 & 0 & 2,807 & -1,807 \\ 0 & 0 & 0 & 0 & 3,6224 & -2,6224 \end{array} \right],$$

e

$$\underleftarrow{\mathbf{y}} = \left[ \begin{array}{c} \underleftarrow{\mathbf{y}}_1 \\ \underleftarrow{\mathbf{y}}_2 \end{array} \right], \quad \underleftarrow{\mathbf{y}}_1 = \left[ \begin{array}{c} y_1(k) \\ y_1(k-1) \\ y_1(k-2) \\ y_1(k-3) \end{array} \right], \quad \underleftarrow{\mathbf{y}}_2 = \left[ \begin{array}{c} y_2(k) \\ y_2(k-1) \end{array} \right].$$

As outras matrizes são obtidas a partir dos polinômios  $E_{l,j}(z^{-1})B_{l,p}(z^{-1})$  e  $E_{l,j}(z^{-1})B_{q,l,p}(z^{-1})$ , que são multiplicados por  $\Delta u_p(k+j-d_{l,p}-1)$  e  $\Delta q_p(k+j-d_{q,l,p})$ , respectivamente, então:

$$\begin{aligned}
E_{1,1}(z^{-1})B_{1,1}(z^{-1})\Delta u_1(k+1-1-1) &= 0,1785\Delta u_1(k-1) - 0,331\Delta u_1(k-2) \\
&\quad + 0,1534\Delta u_1(k-3) \\
E_{1,2}(z^{-1})B_{1,1}(z^{-1})\Delta u_1(k+2-1-1) &= 0,1785\Delta u_1(k) + 0,3347\Delta u_1(k-1) \\
&\quad - 1,0808\Delta u_1(k-2) \\
&\quad + 0,5721\Delta u_1(k-3) \\
E_{1,3}(z^{-1})B_{1,1}(z^{-1})\Delta u_1(k+3-1-1) &= 0,1785\Delta u_1(k+1) + 0,3347\Delta u_1(k) \\
&\quad + 0,4714\Delta u_1(k-1) \\
&\quad - 2,3059\Delta u_1(k-2) \\
&\quad + 1,334\Delta u_1(k-3)
\end{aligned}$$

$$\begin{aligned}
E_{1,1}(z^{-1})B_{1,2}(z^{-1})\Delta u_2(k+1-1-1) &= -0,3236\Delta u_2(k-1) + 0,5828\Delta u_2(k-2) \\
&\quad - 0,2622\Delta u_2(k-3) \\
E_{1,2}(z^{-1})B_{1,2}(z^{-1})\Delta u_2(k+2-1-1) &= -0,3236\Delta u_2(k) - 0,6239\Delta u_2(k-1) \\
&\quad + 1,9113\Delta u_2(k-2) \\
&\quad - 0,9779\Delta u_2(k-3) \\
E_{1,3}(z^{-1})B_{1,2}(z^{-1})\Delta u_2(k+3-1-1) &= -0,3236\Delta u_2(k+1) - 0,6239\Delta u_2(k) \\
&\quad - 0,9027\Delta u_2(k-1) \\
&\quad + 4,0904\Delta u_2(k-2) \\
&\quad - 2,2804\Delta u_2(k-3) \\
E_{2,j}(z^{-1})B_{2,1}(z^{-1})\Delta u_1(k+j-d_{2,1}-1) &= 0, \quad \forall j, \text{ já que } B_{2,1}(z^{-1}) = 0, \\
E_{2,1}(z^{-1})B_{2,2}(z^{-1})\Delta u_2(k+1-0-1) &= 0,3762\Delta u_2(k) \\
E_{2,2}(z^{-1})B_{2,2}(z^{-1})\Delta u_2(k+2-0-1) &= 0,3762\Delta u_2(k+1) + 0,7276\Delta u_2(k) \\
E_{2,3}(z^{-1})B_{2,2}(z^{-1})\Delta u_2(k+3-0-1) &= 0,3762\Delta u_2(k+2) + 0,7276\Delta u_2(k+1) \\
&\quad + 1,0559\Delta u_2(k) \\
E_{1,1}(z^{-1})B_{q,1,1}(z^{-1})\Delta q_1(k+1-1) &= 0\Delta q_1(k) + 0,1555\Delta q_1(k-1) \\
&\quad - 0,2804\Delta q_1(k-2) \\
&\quad + 0,1263\Delta q_1(k-3) \\
E_{1,2}(z^{-1})B_{q,1,1}(z^{-1})\Delta q_1(k+2-1) &= 0\Delta q_1(k+1) + 0,1555\Delta q_1(k) \\
&\quad + 0,2995\Delta q_1(k-1) \\
&\quad - 0,9193\Delta q_1(k-2) \\
&\quad + 0,471\Delta q_1(k-3) \\
E_{1,3}(z^{-1})B_{q,1,1}(z^{-1})\Delta q_1(k+3-1) &= 0\Delta q_1(k+2) + 0,1555\Delta q_1(k+1) \\
&\quad + 0,2995\Delta q_1(k) \\
&\quad + 0,4328\Delta q_1(k-1) \\
&\quad - 1,9671\Delta q_1(k-2) \\
&\quad + 1,0982\Delta q_1(k-3) \\
E_{2,1}(z^{-1})B_{q,2,1}(z^{-1})\Delta q_1(k+1-0) &= 0\Delta q_1(k+1) - 0,2052\Delta q_1(k) \\
E_{2,2}(z^{-1})B_{q,2,1}(z^{-1})\Delta q_1(k+2-0) &= 0\Delta q_1(k+2) - 0,2052\Delta q_1(k+1) \\
&\quad - 0,3969\Delta q_1(k) \\
E_{2,3}(z^{-1})B_{q,2,1}(z^{-1})\Delta q_1(k+3-0) &= 0\Delta q_1(k+3) - 0,2052\Delta q_1(k+2) \\
&\quad - 0,3969\Delta q_1(k+1) - 0,576\Delta q_1(k)
\end{aligned}$$

A partir das equações anteriores, podemos facilmente montar todas as matrizes  $\mathbf{H}_{l,p}$ ,  $\mathbf{G}_{l,p}$ ,  $\mathbf{G}_{q,l,p}$  e  $\mathbf{H}_{q,l,p}$ , e lembrando de desprezar os termos em

função de  $\Delta u_p(k+j)$  e  $\Delta q_p(k+j)$  se  $j > N_{u,p}-1$  e  $j > N_q$ , respectivamente, por conta dos horizontes, temos:

$$\begin{aligned} \mathbf{G}_{1,1} &= \begin{bmatrix} 0 & 0 \\ 0,1785 & 0 \\ 0,33472 & 0,1785 \end{bmatrix}, \quad \mathbf{G}_{1,2} = \begin{bmatrix} 0 \\ -0,3236 \\ -0,62392 \end{bmatrix}, \quad \mathbf{G}_{2,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \mathbf{G}_{2,2} &= \begin{bmatrix} 0,37618 \\ 0,72761 \\ 1,0559 \end{bmatrix}, \quad \mathbf{H}_{1,1} = \begin{bmatrix} 0,1785 & -0,33095 & 0,1534 \\ 0,33472 & -1,0808 & 0,57206 \\ 0,47144 & -2,3059 & 1,334 \end{bmatrix}, \\ \mathbf{H}_{1,2} &= \begin{bmatrix} -0,3236 & 0,58284 & -0,26223 \\ -0,62392 & 1,9113 & -0,97793 \\ -0,90265 & 4,0904 & -2,2804 \end{bmatrix}, \quad \mathbf{H}_{2,1} = \mathbf{H}_{2,2} = [ \quad ] \\ \mathbf{G}_{q,1,1} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0,15548 & 0 \end{bmatrix}, \quad \mathbf{G}_{q,2,1} = \begin{bmatrix} 0 & 0 \\ -0,20519 & 0 \\ -0,39688 & -0,20519 \end{bmatrix}, \\ \mathbf{H}_{q,1,1} &= \begin{bmatrix} 0 & 0,15548 & -0,28037 & 0,12629 \\ 0,15548 & 0,29945 & -0,91929 & 0,47096 \\ 0,29945 & 0,43276 & -1,9671 & 1,0982 \end{bmatrix}, \quad \mathbf{H}_{q,2,1} = \begin{bmatrix} -0,20519 \\ -0,39688 \\ -0,57597 \end{bmatrix}, \end{aligned}$$

e note que as matrizes  $\mathbf{H}_{q,l,p}$  e  $\mathbf{H}_{l,p}$  para um  $p$  fixo não possuem o mesmo número de colunas. Isso acontece por conta da ordem dos polinômios e atrasos envolvidos, que são diferentes. Particularmente no caso de  $\mathbf{H}_{2,1}$  e  $\mathbf{H}_{2,2}$ , veja que são vazias, ou seja, as previsões de  $y_2$  não dependem de valores passados das entradas. A obtenção de  $\mathbf{G}$  e  $\mathbf{G}_q$  se dá sem grandes problemas, pois as matrizes  $\mathbf{G}_{l,p}$  e  $\mathbf{G}_{q,l,p}$  sempre possuem dimensões adequadas. Desta forma,

$$\mathbf{G} = \left[ \begin{array}{cc|c} 0 & 0 & 0 \\ 0,1785 & 0 & -0,3236 \\ 0,33472 & 0,1785 & -0,62392 \\ \hline 0 & 0 & 0,37618 \\ 0 & 0 & 0,72761 \\ 0 & 0 & 1,0559 \end{array} \right], \quad \mathbf{G}_q = \left[ \begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0,15548 & 0 & 0 \\ \hline 0 & 0 & 0 \\ -0,20519 & 0 & 0 \\ -0,39688 & -0,20519 & 0 \end{array} \right],$$

com

$$\Delta \mathbf{u}_1 = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_1(k+1) \end{bmatrix}, \quad \Delta \mathbf{u}_2 = [\Delta u_2(k)], \quad \Delta \mathbf{q}_1 = \begin{bmatrix} \Delta q_1(k+1) \\ \Delta q_1(k+2) \end{bmatrix}.$$

Já para a concatenação das matrizes  $\mathbf{H}_{l,p}$ , para cada  $p$ , devemos descobrir o maior número de colunas existente, que está relacionado com os atrasos e ordens dos polinômios  $B_{l,p}(z^{-1})$ , e completar as matrizes  $\mathbf{H}_{l,p}$  com menos

colunas, com colunas de zeros à direita. Por exemplo, para a entrada  $p = 1$ ,  $\mathbf{H}_{1,1}$  possui 3 colunas, e  $\mathbf{H}_{2,1}$  não possui nenhuma. Então devemos completar  $\mathbf{H}_{2,1}$  com 3 colunas de zeros. Neste caso particular, o mesmo deve ser feito para  $\mathbf{H}_{2,2}$ , já que esta matriz é vazia e  $\mathbf{H}_{2,1}$  possui três colunas. Assim,

$$\mathbf{H} = \begin{bmatrix} 0,1785 & -0,33095 & 0,1534 & -0,3236 & 0,58284 & -0,26223 \\ 0,33472 & -1,0808 & 0,57206 & -0,62392 & 1,9113 & -0,97793 \\ 0,47144 & -2,3059 & 1,334 & -0,90265 & 4,0904 & -2,2804 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

com

$$\underline{\Delta \mathbf{u}}_1 = \begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_1(k-2) \\ \Delta u_1(k-3) \end{bmatrix}, \quad \underline{\Delta \mathbf{u}}_2 = \begin{bmatrix} \Delta u_2(k-1) \\ \Delta u_2(k-2) \\ \Delta u_2(k-3) \end{bmatrix}$$

O mesmo princípio deve ser aplicado para o caso das matrizes  $\mathbf{H}_{q,l,p}$  da perturbação.  $\mathbf{H}_{q,1,1}$  possui 4 colunas, enquanto que  $\mathbf{H}_{q,2,1}$  possui apenas 1. Então esta última deve ser completada com três colunas de zeros à direita para obter a matriz  $\mathbf{H}_q$ , obtendo, assim,

$$\mathbf{H}_q = \begin{bmatrix} 0 & 0,15548 & -0,28037 & 0,12629 \\ 0,15548 & 0,29945 & -0,91929 & 0,47096 \\ 0,29945 & 0,43276 & -1,9671 & 1,0982 \\ \hline -0,20519 & 0 & 0 & 0 \\ -0,39688 & 0 & 0 & 0 \\ -0,57597 & 0 & 0 & 0 \end{bmatrix},$$

com

$$\underline{\Delta \mathbf{q}}_1 = \begin{bmatrix} \Delta q_1(k) \\ \Delta q_1(k-1) \\ \Delta q_1(k-2) \\ \Delta q_1(k-3) \end{bmatrix}.$$

Por fim, a seguir exemplificaremos a aplicação do GPC no processo do tanque térmico, usando agora horizontes mais longos e simulando o resultado d controle, considerando assim todos os passos estudados.

**Exemplo 3.15.** Neste exemplo utilizaremos o processo que trata do controle de nível e temperatura da água contida no interior de um tanque termicamente isolado, apresentado inicialmente no Exemplo 3.10, que é um sistema com duas entradas, duas saídas e uma perturbação. Para este sistema,  $y_1$  e  $y_2$  são respectivamente a temperatura e o nível no interior do tanque,  $u_1$

é a potência do resistor para aquecimento,  $u_2$  é a abertura da válvula de alimentação e  $q$  é a abertura da válvula de saída do tanque.

O cenário de simulação é o mesmo do Exemplo 3.10, que considera um tempo total de simulação de 160s. O sistema parte de  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$  e, em  $t = 10$ s, uma variação tipo degrau unitário da referência de  $y_1$  é aplicada. Posteriormente, em  $t = 80$ s aplica-se uma variação tipo degrau de amplitude 0,5 na referência de  $y_2$ . Finalmente, em  $t = 130$ s aplica-se uma variação tipo degrau de amplitude 0,2 na perturbação  $q$ .

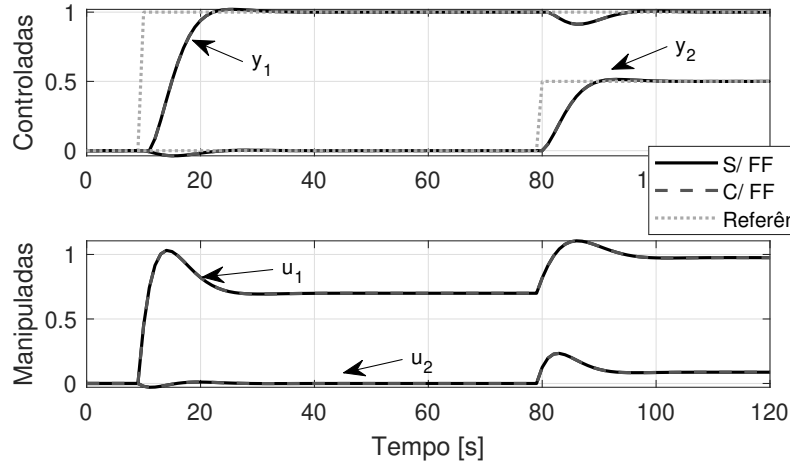


Figura 3.6: Exemplo 3.15 – resposta e sinal de controle do tanque térmico em malha fechada para seguimento de referências sem e com ação antecipativa.

A representação MFD do modelo do sistema, necessária para a aplicação do GPC, foi apresentada no Exemplo 3.13, e a sintonia do controlador GPC será a mesma do controlador DMC do Exemplo 3.10, ou seja,  $N_{1,1} = 2$  e  $N_{2,1} = 17$ , de forma tal a ter um horizonte de predição de duas constantes de tempo de malha aberta. Usando o mesmo critério,  $N_{1,2} = 1$  e  $N_{2,2} = 31$ . O horizonte de controle foi escolhido como aproximadamente um terço do de predição, assim,  $N_{u,1} = 5$  e  $N_{u,2} = 10$ . Com relação às ponderações,  $\delta_1 = 1/N_{y,1}$ ,  $\delta_2 = 9/N_{y,2}$ ,  $\lambda = 1/N_{u,1}$  e  $\lambda_2 = 30/N_{u,2}$ , e todas são constantes ao longo do horizonte. Dois casos de simulação foram considerados, um considerando a perturbação  $q$  não medida e outro considerando-a medida, com  $N_q = 0$ .

Na Figura 3.6, é apresentada a etapa do cenário de simulação que trata do seguimento de referências. Neste caso, os controladores com e sem ação antecipativa possuem o mesmo desempenho, já que a ação antecipativa só afeta a rejeição de perturbações. Veja, também, que o controlador GPC consegue minimizar o acoplamento entre as variáveis, de forma similar ao DMC.



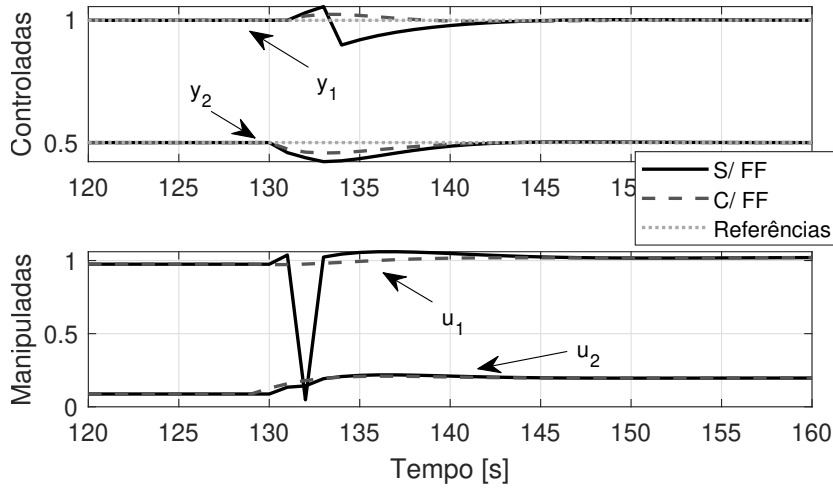


Figura 3.7: Exemplo 3.15 – resposta e sinal de controle do tanque térmico em malha fechada para rejeição de perturbação sem e com ação antecipativa.

Já no caso da rejeição de perturbações, apresentado na Figura 3.7, o GPC, mesmo sem ação antecipativa, tem uma rejeição de perturbações adequada. No entanto, no caso com ação antecipativa, veja que o controlador consegue uma rejeição mais rápida, com menos desvio em relação à referência e com menos esforço de controle do que a sintonia sem ação antecipativa. Isto mostra, como o conhecimento da perturbação dentro do modelo de predição permite uma ação de controle mais eficiente.

### 3.3.1 Cálculo das predições (caso $C(z^{-1}) \neq 1$ )

Assim como no caso monovariável analisado no Volume 1, a utilização do polinômio  $C_l(z^{-1})$  no modelo de predição da saída  $y_l$  do sistema MIMO, altera unicamente a resposta livre e, se ajustado adequadamente, pode melhorar a robustez ou a rejeição de perturbações do sistema em malha fechada. Lembre-se que, na literatura de GPC, opta-se por utilizar um polinômio  $C_l(z^{-1})$  mônico, com o seu primeiro coeficiente unitário,  $c_{l,0} = 1$ . Veja que isto não causa problemas na análise, dado que o efeito da perturbação na saída do processo pode ser atribuído a uma perturbação de amplitude diferente, de forma a obter o mesmo resultado que considerando  $c_{l,0} \neq 1$ . Para isso bastaria multiplicar e dividir  $C_l(z^{-1})e(k)$  por  $c_{l,0}$ , obtendo  $\frac{C_l(z^{-1})}{c_{l,0}}c_{l,0}e(k)$ , de forma tal que o novo polinômio  $\frac{C_l(z^{-1})}{c_{l,0}}$  tem o seu primeiro coeficiente unitário e a nova perturbação  $c_{l,0}e(k)$  tem sua amplitude ajustada com  $c_{l,0}$ . Assim, no que segue, o polinômio  $C_l(z^{-1})$  é considerado mônico.

Com  $C_l(z^{-1}) \neq 1$ , a equação diofantina a ser resolvida para obter os polinômios  $E_{l,j}(z^{-1})$  e  $F_{l,j}(z^{-1})$  passa a ser

$$C_l(z^{-1}) = E_{l,j}(z^{-1})\Delta A_l(z^{-1}) + z^{-j}F_{l,j}(z^{-1}), \quad (3.36)$$

que pode ser considerada como a divisão de  $C_l(z^{-1})$  por  $\Delta A_l(z^{-1})$ , com quociente  $E_{l,j}(z^{-1})$  e resto  $z^{-j}F_{l,j}(z^{-1})$  (note que estes polinômios não são os mesmos encontrados para o caso com  $C_l(z^{-1}) = 1$ ). Neste caso, o polinômio  $E_{l,j}(z^{-1})$  continua tendo grau  $j - 1$  e o grau de  $F_{l,j}(z^{-1})$  depende do grau de  $C_l(z^{-1})$  e pode até ser variável. Se  $n_{c,l} \leq n_{a,l} + 1$ , a ordem de  $F_{l,j}(z^{-1})$  continua tendo grau  $n_{a,l}$ , como no caso de  $C_l(z^{-1}) = 1$ . Se  $n_{c,l} > n_{a,l} + 1$ , para  $j \leq n_{c,l} - n_{a,l}$ , o grau de  $F_{l,j}(z^{-1})$  é  $n_{c,l} - j$ , e se  $j > n_{c,l} - n_{a,l}$  o grau é sempre  $n_{a,l}$ .

Multiplicando (3.31) por  $\Delta E_{l,j}(z^{-1})/C_l(z^{-1})$ , e dado que

$$C_l(z^{-1}) - z^{-j}F_{l,j}(z^{-1}) = E_{l,j}(z^{-1})\Delta A_l(z^{-1}),$$

os valores futuros da saída  $l$  são dados por:

$$\begin{aligned} y_l(k+j) = & \sum_{p=1}^m \frac{E_{l,j}(z^{-1})\tilde{B}_{l,p}(z^{-1})}{C_l(z^{-1})} \Delta u_p(k+j-1) \\ & + \sum_{p=1}^{m_q} \frac{E_{l,j}(z^{-1})\tilde{B}_{q,l,p}(z^{-1})}{C_l(z^{-1})} \Delta q_p(k+j) \\ & + \frac{F_{l,j}(z^{-1})}{C_l(z^{-1})} y_l(k) + E_{l,j}(z^{-1})e(k+j), \end{aligned} \quad (3.37)$$

sendo

$$\tilde{B}_{l,p}(z^{-1}) = z^{-d_{l,p}} B_{l,p}(z^{-1}), \quad \tilde{B}_{q,l,p}(z^{-1}) = z^{-d_{q,l,p}} B_{q,l,p}(z^{-1}),$$

ou seja, os atrasos agora ficam implícitos nos novos polinômios. Isto foi feito para facilitar a notação e simplificar a explicação. Isto também poderia ter sido usado no caso de  $C_l(z^{-1}) = 1$ , mas nesta seção é especialmente interessante pois os cálculos aqui são um pouco mais elaborados e o uso de  $\tilde{B}_{l,p}(z^{-1})$  e  $\tilde{B}_{q,l,p}(z^{-1})$  facilita a discussão.

Note que esta última equação é similar à encontrada no caso de  $C_l(z^{-1}) = 1$ . A única diferença é que  $y_l(k)$ ,  $\Delta u_p(k+j-1)$  e  $\Delta q_p(k+j)$  aparecem multiplicados por quocientes de polinômios que dependem de  $C_l(z^{-1})$  e não mais por polinômios que dependem apenas do modelo do processo. Finalmente, assim como no caso de  $C_l(z^{-1}) = 1$ , o termo dependente da perturbação:

$$E_{l,j}(z^{-1})e(k+j)$$

corresponde a valores de  $e$  no futuro, dado que  $E_{l,j}(z^{-1})$  tem grau  $j - 1$ . Como o valor esperado das perturbações futuras é zero, podemos obter a predição de  $y_l(k + j)$  como:

$$\begin{aligned}\hat{y}_l(k + j | k) = & \sum_{p=1}^m \frac{E_{l,j}(z^{-1})\tilde{B}_{l,p}(z^{-1})}{C(z^{-1})} \Delta u_p(k + j - 1) \\ & + \sum_{p=1}^{m_q} \frac{E_{l,j}(z^{-1})\tilde{B}_{q,l,p}(z^{-1})}{C(z^{-1})} \Delta q_p(k + j) \\ & + \frac{F_{l,j}(z^{-1})}{C(z^{-1})} y_l(k),\end{aligned}\quad (3.38)$$

Observe que as divisões dos polinômios por  $C(z^{-1})$  podem ser consideradas como polinômios em  $z^{-1}$  de grau infinito. Assim, o termo  $\frac{F_{l,j}(z^{-1})}{C(z^{-1})} y_l(k)$  corresponde a uma combinação linear de valores passados de  $y(k)$ . Já os termos  $\frac{E_{l,j}(z^{-1})\tilde{B}_{l,p}(z^{-1})}{C(z^{-1})} \Delta u(k + j - 1)$  e  $\frac{E_{l,j}(z^{-1})\tilde{B}_{q,l,p}(z^{-1})}{C(z^{-1})} \Delta q(k + j)$  podem ser separados em duas partes, uma correspondente aos valores passados da ação de controle e da perturbação (para os elementos do polinômio com expoente em  $z^{-1}$  maior que  $j - 1$ ), e outra correspondente aos valores futuros da ação de controle e da perturbação (para os elementos do polinômio com expoente em  $z^{-1}$  menor ou igual a  $j - 1$ ). Assim como acontece no caso de  $C_l(z^{-1}) = 1$ , podemos compor a expressão que calcula as predições com uma componente de resposta livre e outra de forçada.

Apesar da simplicidade da análise anterior, para poder chegar à formulação final do controle, é interessante o uso de sinais auxiliares, como veremos na continuação. Como o último termo do lado direito de (3.38) só depende de termos passados de  $y_l$ , podemos utilizar

$$y_{c,l}(k) = \frac{1}{C_l(z^{-1})} y_l(k), \quad (3.39)$$

ou seja, utilizamos uma variável auxiliar  $y_{c,l}$  que nada mais é do que a saída  $y_l$  filtrada por  $1/C_l(z^{-1})$ , com isso simplificamos este termo da equação. E podemos replicar o procedimento para a perturbação medida:

$$q_{c,p}(k) = \frac{1}{C_l(z^{-1})} q_p(k). \quad (3.40)$$

Se valores futuros da perturbação forem conhecidos ou estimados, essa filtragem pode ser realizada sem nenhuma dificuldade adicional. O mesmo não pode ser feito diretamente com os termos dependentes de  $\Delta u_p$  pois há termos

que dependem de valores futuros de  $\Delta u_p$ , e estes são desconhecidos. Assim, considere uma segunda equação diofantina:

$$E_{l,j}(z^{-1})\tilde{B}_{l,p}(z^{-1}) = G_{l,p,j}(z^{-1})C_l(z^{-1}) + z^{-j}H_{l,p,j}(z^{-1}) \quad (3.41)$$

que pode ser considerada associada à divisão do polinômio  $E_{l,j}(z^{-1})\tilde{B}_p(z^{-1})$  pelo polinômio  $C_l(z^{-1})$ . Note que, assim como em (3.36), o grau de  $G_{l,p,j}(z^{-1})$  é  $j-1$  e  $H_{l,p,j}(z^{-1})$  pode ter grau variável, dependendo do grau do polinômio  $C_l(z^{-1})$ . Substituindo (3.41) em (3.38), temos que

$$\begin{aligned} \hat{y}_l(k+j | k) = & \sum_{p=1}^m [G_{l,p,j}(z^{-1})\Delta u_p(k+j-1) + H_{l,p,j}(z^{-1})\Delta u_{c,p}(k-1)] \\ & + \sum_{p=1}^{m_q} [G_{q,l,p,j}(z^{-1})\Delta q_{c,p}(k+j) + H_{q,l,p,j}(z^{-1})\Delta q_{c,p}(k)] \\ & + F_{l,j}(z^{-1})y_{c,l}(k), \end{aligned}$$

com  $G_{q,l,p,j}(z^{-1})$  e  $H_{q,l,p,j}(z^{-1})$  os polinômios que separam os termos dependentes de valores futuros e passados, respectivamente, da perturbação medida, e

$$\Delta u_{c,l,p}(k) = \frac{1}{C_l(z^{-1})}\Delta u_p(k),$$

define os sinais de controle filtrados por  $C_l(z^{-1})$ . Note que, nesta última equação, os termos  $G_{l,p,j}(z^{-1})\Delta u_p(k+j-1)$  só possuem termos dependentes de valores futuros de  $\Delta u_p$ , pois a ordem de  $G_{l,p,j}(z^{-1})$  é igual a  $j-1$ , e que estes termos não dependem do polinômio  $C_l(z^{-1})$ . E, também, que para cada saída  $l$ , devemos filtrar os incrementos de controle passados pelo respectivo filtro  $C_l(z^{-1})$ .

Ainda podemos reordenar a equação de forma a evidenciar a resposta livre e forçada da predição da saída  $y_l$ :

$$\hat{y}_l(k+j | k) = \sum_{p=1}^m G_{l,p,j}(z^{-1})\Delta u_p(k+j-1) + f_l(k+j | k),$$

com

$$\begin{aligned} f_l(k+j | k) = & \sum_{p=1}^m H_{l,p,j}(z^{-1})\Delta u_{c,l,p}(k-1) \\ & + \sum_{p=1}^{m_q} [G_{q,l,p,j}(z^{-1})\Delta q_{c,l,p}(k+j) + H_{q,l,p,j}(z^{-1})\Delta q_{c,l,p}(k)] \\ & + F_{l,j}(z^{-1})y_{c,l}(k). \end{aligned}$$

Essa equação tem a mesma forma que no caso de  $C_l(z^{-1}) = 1$ , fazendo com que o restante do algoritmo GPC não precise ser alterado. Em termos práticos, podemos calcular os polinômios  $F_{l,j}(z^{-1})$ ,  $G_{q,l,p,j}(z^{-1})$ ,  $H_{q,l,p,j}(z^{-1})$  e  $H_{l,j}(z^{-1})$ , filtrar as perturbações, e os sinais de controle e saídas passadas com um filtro  $1/C(z^{-1})$  e, em seguida, calcular a resposta livre do sistema. A resposta forçada não muda com a inclusão do polinômio  $C_l(z^{-1}) \neq 1$ , dado que ela representa apenas o efeito do controle futuro na predição, sem levar em conta as perturbações.

Então, concluindo, o uso do polinômio  $C_l(z^{-1})$  não unitário adiciona complexidade na obtenção das predições, porém, caso se conheça o modelo das perturbações, o modelo de predição será melhor que considerando  $C_l(z^{-1}) = 1$ .

Na prática, é em geral difícil obter bons modelos das perturbações não medidas. Assim, o GPC é formulado diretamente com  $C_l(z^{-1}) = 1$  ou se considera esse polinômio como mais um parâmetro de projeto. Esta última opção dá um grau de liberdade adicional que pode ser utilizado para, por exemplo, aumentar a robustez do sistema em malha fechada ou mudar as respostas às perturbações. Por ora, podemos apenas realizar uma análise mais intuitiva: observe que se o sistema está sujeito a perturbações, ruído ou erro de modelagem, ao filtrarmos os sinais passados da saída e do controle para calcular a resposta livre, estamos modificando a informação enviada ao algoritmo de otimização, podendo, por exemplo, evitar que sejam calculadas ações de controle violentas por causa de variações bruscas da saída e do controle causadas pelas perturbações não medidas. É interessante observar que, no caso nominal, isto é, sem erros de modelagem nem perturbações, a inclusão de um  $C_l(z^{-1}) \neq 1$  não tem efeito no sistema em malha fechada, assim como no caso monovariável discutido no Volume 1.

Na sequência, vamos mostrar com um exemplo como obter as predições com o polinômio  $C_l(z^{-1}) \neq 1$  e também um exemplo completo mostrando o efeito deste polinômio na rejeição de perturbações e robustez.

**Exemplo 3.16.** *A partir do Exemplo 3.13, no qual calculamos a representação MFD de um sistema de tanque termicamente isolado, vamos, agora, obter a representação matricial das predições para este processo para o caso com polinômio  $C_l(z^{-1}) \neq 1$ . A título de ilustração, utilizaremos horizontes curtos e não consideraremos a perturbação medida, assim,  $N_{1,1} = N_{2,1} = 1$ ,  $N_{2,1} = N_{2,2} = 3$ ,  $N_{u,1} = 2$  e  $N_{u,2} = 1$ . Os polinômios do modelo MFD, já*

incorporando os atrasos nos polinômios  $B$  são:

$$\begin{aligned}A_1(z) &= 1 - 2,7292z^{-1} + 2,482z^{-2} - 0,7521z^{-3} \\A_2(z) &= 1 - 0,9342z^{-1} \\ \tilde{B}_{1,1}(z) &= 0,1785z^{-1} - 0,331z^{-2} + 0,1534z^{-3} \\ \tilde{B}_{1,2}(z) &= -0,3236z^{-1} + 0,5828z^{-2} - 0,2622z^{-3} \\ \tilde{B}_{2,1}(z) &= 0 \\ \tilde{B}_{2,2}(z) &= 0,3762\end{aligned}$$

e os polinômios  $C_l(z^{-1})$  são

$$\begin{aligned}C_1(z^{-1}) &= 1 - 0,8z^{-1} \\ C_2(z^{-1}) &= 1 - 0,9z^{-1}\end{aligned}$$

Calculando aos polinômios  $F_{i,j}(z^{-1})$  e  $E_{i,j}(z^{-1})$ , a partir das equações diofantinas, temos

$$\begin{aligned}F_{1,1}(z^{-1}) &= 2,9292 - 5,2112z^{-1} + 3,2341z^{-2} - 0,7521z^{-3} \\ F_{1,2}(z^{-1}) &= 5,7125 - 12,0307z^{-1} + 8,7213z^{-2} - 2,2031z^{-3} \\ F_{1,3}(z^{-1}) &= 9,2725 - 21,0479z^{-1} + 16,2717z^{-2} - 4,2964z^{-3} \\ F_{2,1}(z^{-1}) &= 1,0342 - 0,9342z^{-1} \\ F_{2,2}(z^{-1}) &= 1,0662 - 0,9662z^{-1} \\ F_{2,3}(z^{-1}) &= 1,0961 - 0,9961z^{-1}\end{aligned}$$

$$\begin{aligned}E_{1,1}(z^{-1}) &= 1 \\ E_{1,2}(z^{-1}) &= 1 + 2,9292z^{-1} \\ E_{1,3}(z^{-1}) &= 1 + 2,9292z^{-1} + 5,7125z^{-2} \\ E_{2,1}(z^{-1}) &= 1 \\ E_{2,2}(z^{-1}) &= 1 + 1,0342z^{-1} \\ E_{2,3}(z^{-1}) &= 1 + 1,0342z^{-1} + 1,0662z^{-2}\end{aligned}$$

Analisando os polinômios  $F_{i,j}(z^{-1})$ , podemos concluir que precisaremos de informações passadas de  $k$  até  $k - 3$  para a saída 1, e de  $k$  até  $k - 1$  para a saída 2, o que é condizente com a ordem os polinômios originais  $A_i(z^{-1})$  da representação MFD, que são de ordem 3 e 1, respectivamente.

As matrizes  $\mathbf{F}_i$  podem ser obtidas diretamente dos polinômios  $F_{i,j}(z^{-1})$ . A matriz  $\mathbf{F}$  resultante é, então,

$$\mathbf{F} = \left[ \begin{array}{cccc|cc} 2,9292 & -5,2112 & 3,2341 & -0,7521 & 0 & 0 \\ 5,7125 & -12,0307 & 8,7213 & -2,2031 & 0 & 0 \\ 9,2725 & -21,0479 & 16,2717 & -4,2964 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1,0342 & -0,93424 \\ 0 & 0 & 0 & 0 & 1,0662 & -0,96622 \\ 0 & 0 & 0 & 0 & 1,0961 & -0,9961 \end{array} \right],$$

e

$$\underline{\mathbf{y}}^c = \begin{bmatrix} \underline{\mathbf{y}}_{c,1} \\ \underline{\mathbf{y}}_{c,2} \end{bmatrix}, \quad \underline{\mathbf{y}}^{c,1} = \begin{bmatrix} y_{c,1}(k) \\ y_{c,1}(k-1) \\ y_{c,1}(k-2) \\ y_{c,1}(k-3) \end{bmatrix}, \quad \underline{\mathbf{y}}^2 = \begin{bmatrix} y_{c,2}(k) \\ y_{c,2}(k-1) \end{bmatrix}.$$

As outras matrizes são obtidas a partir dos polinômios  $E_{l,j}(z^{-1})\tilde{B}_{l,p}(z^{-1})$  e das novas equações diofantinas, que resultam em:

$$\begin{aligned} G_{1,1,1}(z^{-1}) &= 0 \\ G_{1,1,2}(z^{-1}) &= 0 + 0,1785z^{-1} \\ G_{1,1,3}(z^{-1}) &= 0 + 0,1785z^{-1} + 0,3347z^{-2} \\ G_{1,2,1}(z^{-1}) &= 0 \\ G_{1,2,2}(z^{-1}) &= 0 - 0,3236z^{-1} \\ G_{1,2,3}(z^{-1}) &= 0 - 0,3236z^{-1} - 0,6239z^{-2} \\ G_{2,2,1}(z^{-1}) &= 0,3762 \\ G_{2,2,2}(z^{-1}) &= 0,3762 + 0,7276z^{-1} \\ G_{2,2,3}(z^{-1}) &= 0,3762 + 0,7276z^{-1} + 1,0559z^{-2} \end{aligned}$$

$$\begin{aligned} H_{1,1,1}(z^{-1}) &= 0,1785 - 0,331z^{-1} + 0,1534z^{-2} \\ H_{1,1,2}(z^{-1}) &= 0,3347 - 0,816z^{-1} + 0,4493z^{-2} \\ H_{1,1,3}(z^{-1}) &= 0,4714 - 1,4412z^{-1} + 0,8763z^{-2} \\ H_{1,2,1}(z^{-1}) &= -0,3236 + 0,5828z^{-1} - 0,2622z^{-2} \\ H_{1,2,2}(z^{-1}) &= -0,6239 + 1,445z^{-1} - 0,7681z^{-2} \\ H_{1,2,3}(z^{-1}) &= -0,9027 + 2,5613z^{-1} - 1,498z^{-2} \\ H_{2,2,1}(z^{-1}) &= 0,3386 \\ H_{2,2,2}(z^{-1}) &= 0,6549 \\ H_{2,2,3}(z^{-1}) &= 0,9503 \end{aligned}$$

e veja que os polinômios do par (2,1) não foram calculados já que a função de transferência deste par é zero. Eliminando os termos de  $G_{l,p,j}(z^{-1})$  fora

do horizonte de controle, podemos escrever as predições futuras no formato matricial:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{H}\underline{\Delta\mathbf{u}}_c + \mathbf{F}\underline{\mathbf{y}}_c,$$

com

$$\mathbf{G} = \left[ \begin{array}{cc|c} 0 & 0 & 0 \\ 0,1785 & 0 & -0,3236 \\ 0,33472 & 0,1785 & -0,62392 \\ \hline 0 & 0 & 0,37618 \\ 0 & 0 & 0,72761 \\ 0 & 0 & 1,0559 \end{array} \right], \quad \mathbf{u} = \left[ \begin{array}{c} u_1(k) \\ u_1(k+1) \\ u_2(k) \end{array} \right],$$

$$\mathbf{H} = \left[ \begin{array}{ccc|ccc} 0,1785 & -0,33095 & 0,1534 & -0,3236 & 0,58284 & -0,26223 \\ 0,33472 & -0,81603 & 0,44934 & -0,62392 & 1,445 & -0,76814 \\ 0,47144 & -1,4412 & 0,8763 & -0,90265 & 2,5613 & -1,498 \\ \hline 0 & 0 & 0 & 0,33856 & 0 & 0 \\ 0 & 0 & 0 & 0,65485 & 0 & 0 \\ 0 & 0 & 0 & 0,95034 & 0 & 0 \end{array} \right]$$

$$\underline{\Delta\mathbf{u}}_c = \left[ \begin{array}{c} \underline{\Delta\mathbf{u}}_{c,1} \\ \underline{\Delta\mathbf{u}}_{c,2} \end{array} \right], \underline{\Delta\mathbf{u}}_{c,1} = \left[ \begin{array}{c} \Delta u_{c,1}(k-1) \\ \Delta u_{c,1}(k-2) \\ \Delta u_{c,1}(k-3) \end{array} \right], \underline{\Delta\mathbf{u}}_{c,2} = \left[ \begin{array}{c} \Delta u_{c,2}(k-1) \\ \Delta u_{c,2}(k-2) \\ \Delta u_{c,2}(k-3) \end{array} \right],$$

e note que as repartições das matrizes indicadas por linhas tracejadas definem as submatrizes  $\mathbf{G}_{l,p}$ ,  $\mathbf{H}_{l,p}$  e  $\mathbf{F}_l$ .

**Exemplo 3.17.** Ainda utilizando o processo que trata do controle de nível e temperatura, tratado no exemplo anterior, e apresentado inicialmente no Exemplo 3.10, apresentaremos os resultado quando utilizamos  $C_l(z^{-1}) \neq 1$ . O cenário de simulação é o mesmo do Exemplo 3.10, que considera um tempo total de simulação de 160 s. O sistema parte de  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$  e, em  $t = 10$  s, uma variação tipo degrau unitário da referência de  $y_1$  é aplicada. Posteriormente, em  $t = 80$  s aplica-se uma variação tipo degrau de amplitude 0,5 na referência de  $y_2$ . Finalmente, em  $t = 130$  s aplica-se uma variação tipo degrau de amplitude 0,2 na perturbação  $q$ .

A representação MFD do modelo do sistema, necessária para a aplicação do GPC, foi apresentada no Exemplo 3.13, e a sintonia do controlador GPC será a mesma do controlador DMC do Exemplo 3.10, ou seja,  $N_{1,1} = 2$  e  $N_{2,1} = 17$ , de forma tal a ter um horizonte de predição de duas constantes de tempo de malha aberta. Usando o mesmo critério,  $N_{1,2} = 1$  e  $N_{2,2} = 31$ . O horizonte de controle foi escolhido como aproximadamente um terço do de predição, assim,  $N_{u,1} = 5$  e  $N_{u,2} = 10$ . Com relação às ponderações,



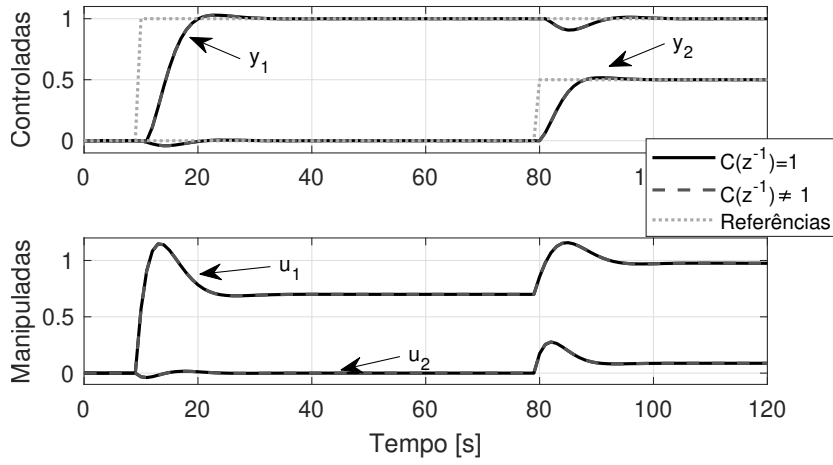


Figura 3.8: Exemplo 3.15 – resposta e sinal de controle do tanque térmico em malha fechada para seguimento de referências com e sem polinômio  $C(z^{-1})$ .

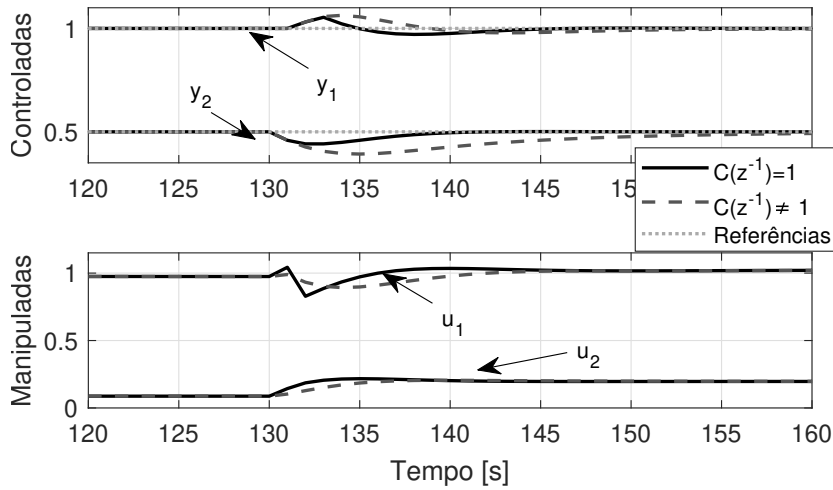


Figura 3.9: Exemplo 3.15 – resposta e sinal de controle do tanque térmico em malha fechada para rejeição de perturbação com e sem polinômio  $C(z^{-1})$ .

$\delta_1 = 1/N_{y,1}$ ,  $\delta_2 = 9/N_{y,2}$ ,  $\lambda = 1/N_{u,1}$  e  $\lambda_2 = 30/N_{u,2}$ , e todas são constante ao longo do horizonte. Não foi considerada, neste caso, que a perturbação é medida. Dois casos serão analisados, um com  $C_l(z^{-1}) = 1, \forall l$ , e um segundo caso com

$$C_l(z^{-1}) = 1 - 0,92z^{-1}, \forall l.$$

Na Figura 3.8, é apresentada a etapa do cenário de simulação que trata do seguimento de referências. Neste caso, ambas as sintonias, com e sem polinômio  $C(z^{-1})$ , possuem o mesmo desempenho, já que este polinômio só

afeta o efeito de perturbações não medidas. Já no caso da rejeição de perturbações, apresentado na Figura 3.9, veja que o caso com  $C(z^{-1}) \neq 1$  resultou numa rejeição de perturbações distinta, e mais lenta do que no outro caso.

É interessante salientar que o caso  $C(z^{-1}) \neq 1$  também tem efeito na robustez do sistema. Considere a mesma sintonia e cenário de simulação, com a diferença de que agora há um erro de modelagem, e o processo tem, na verdade, sua dinâmica representada pela seguinte matriz de transferência:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{2,5e^{-s}}{1+9,0s} & \frac{-5,5e^{-s}}{1+11,0s} \\ 0 & \frac{4}{1+17,0s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} + \begin{bmatrix} \frac{2,1e^{-s}}{1+13s} \\ \frac{-3,12}{1+14,7s} \end{bmatrix} Q(s)$$

ou seja, há um erro de modelagem nos ganhos e nas constantes de tempo das funções de transferência que relacionam as saídas com as entradas.

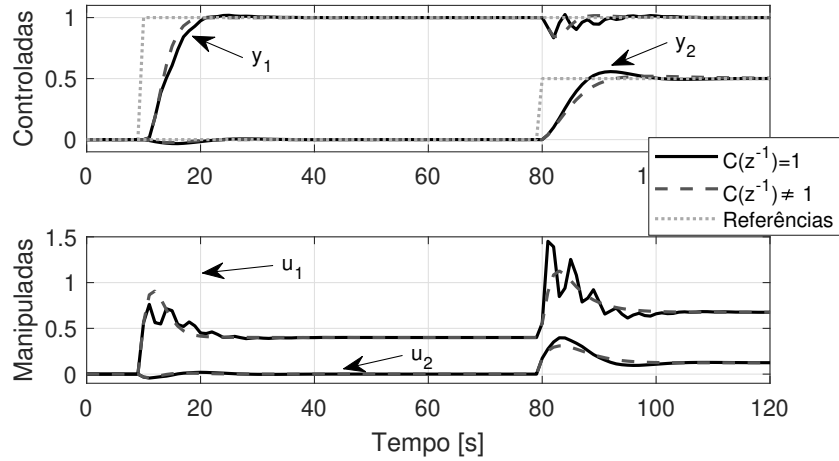


Figura 3.10: Exemplo 3.15 – resposta e sinal de controle do tanque térmico em malha fechada para seguimento de referências com e sem polinômio  $C(z^{-1})$  com erro de modelagem.

Os resultados de simulação para este caso com erro de modelagem são apresentados nas figuras 3.10 e 3.11. Veja que, agora, a resposta com  $C(z^{-1}) = 1$  apresenta oscilações, especialmente no transitório de mudança de referência da saída 2, próximo do tempo de simulação 80 s. Esse comportamento oscilatório é mais evidente no sinal de controle. Já no caso  $C(z^{-1}) \neq 1$ , a ação de controle é menos agressiva o que, por consequência, deixa as respostas das saídas também menos oscilatórias. Isto acontece pelo efeito do filtro  $C(z^{-1})$ , que modifica como perturbações não medidas são consideradas no modelo de predição do controlador GPC.

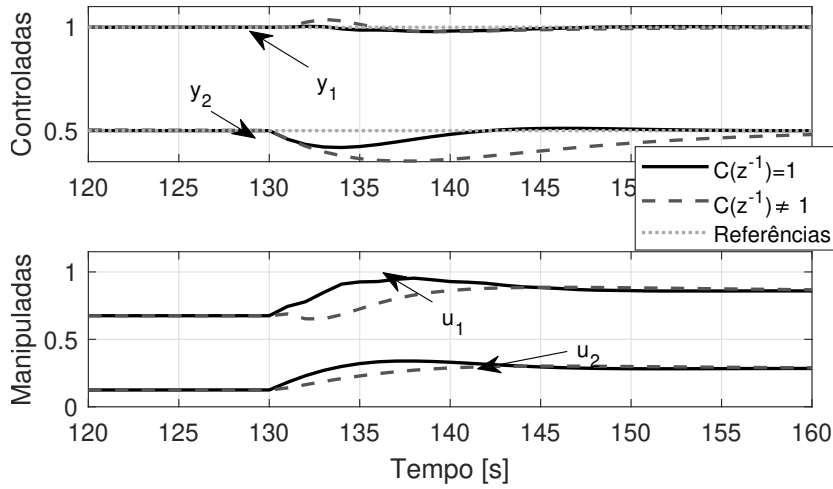


Figura 3.11: Exemplo 3.15 – resposta e sinal de controle do tanque térmico em malha fechada para rejeição de perturbação com e sem polinômio  $C(z^{-1})$  com erro de modelagem.

### 3.4 Estudos de caso

Nesta seção, apresentaremos alguns estudos de caso completos e faremos comparações entre os algoritmos GPC e DMC.

#### 3.4.1 Coluna de destilação água-metanol

Considere o modelo do comportamento dinâmico aproximado de uma coluna de destilação água-metanol funcionando nas vizinhanças de um ponto de operação dado por:

$$\mathbf{P}(s) = \begin{bmatrix} \frac{12,8e^{-s}}{1 + 16,7s} & \frac{-18,9e^{-3s}}{1 + 21s} \\ \frac{6,6e^{-7s}}{1 + 10,9s} & \frac{-19,4e^{-3s}}{1 + 14,4s} \end{bmatrix} \quad \mathbf{P}_q(s) = \begin{bmatrix} \frac{3,8e^{-8s}}{1 + 14,9s} \\ \frac{4,9e^{-4s}}{1 + 13,2s} \end{bmatrix}$$

com  $\mathbf{Y}(s) = \mathbf{P}(s)\mathbf{U} + \mathbf{P}_q(s)Q(s)$ ,  $\mathbf{y} = [y_1 \ y_2]^T$  e  $\mathbf{u} = [u_1 \ u_2]^T$ . Neste processo, a vazão de refluxo ( $u_1$ ) e a vazão de vapor no refeedor ( $u_2$ ) são manipuladas para controlar as concentrações de metanol no topo ( $y_1$ ) e na base ( $y_2$ ) da coluna. A principal perturbação do sistema é a vazão de alimentação ( $q$ ). Trata-se de um modelo incremental, assim, podemos considerar que o ponto de equilíbrio inicial é  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$ .

Para este processo, vamos utilizar um controlador DMC e um GPC com duas sintonias cada, uma considerando a perturbação  $q$  como sendo não mensurável e outra considerando que ela pode ser medida, mas desconsiderando seus valores futuros  $N_q = 0$ . Para efeitos de simulação, vamos considerar o modelo de matriz função de transferência como se fosse o processo real. O cenário de simulação considera um tempo total de simulação de 180 segundos. O sistema parte de  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$  e em  $t = 10$  s uma variação tipo degrau unitário da referência de  $y_1$  é aplicada. Posteriormente, em  $t = 60$  s se aplica uma variação tipo degrau de amplitude 0,5 na referência de  $y_2$ . Finalmente, em  $t = 110$  s se aplica uma perturbação tipo degrau de amplitude  $-0,2$  em  $q$ .

Inicialmente, vamos obter o modelo de resposta ao degrau aplicando degraus em  $u_1$ ,  $u_2$  e  $q$  no processo, e considerando um período de amostragem de  $T_s = 1$  s. Assim, obtemos:

$$\begin{aligned} \mathbf{g}_{1,1} &= [0 \ 0,7440 \ 1,447 \ 2,1047 \ 2,7263 \ \dots \ 12,4 \ 12,4 \ \dots], \\ \mathbf{g}_{2,1} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0,5786 \ 1,1064 \ 1,5880 \ \dots \ 6,6 \ 6,6 \ 6,6 \ \dots], \\ \mathbf{g}_{1,2} &= [0 \ 0 \ 0 \ -0,8789 \ -1,7169 \ -2,5160 \ \dots \ -18,9 \ -18,9 \ -18,9 \ \dots], \\ \mathbf{g}_{2,2} &= [0 \ 0 \ 0 \ -1,3015 \ -2,5157 \ -3,6484 \ \dots \ -19,4 \ -19,4 \ -19,4 \ \dots], \\ \mathbf{g}_{q,1,1} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0,2467 \ 0,4773 \ 0,6930 \ \dots \ 3,8 \ 3,8 \ 3,8 \ \dots], \\ \mathbf{g}_{q,2,1} &= [0 \ 0 \ 0 \ 0 \ 0,3575 \ 0,6889 \ 0,9962 \ \dots \ 4,9 \ 4,9 \ 4,9 \ \dots]. \end{aligned}$$

A partir destes dados, definimos os horizontes de modelo do DMC como  $N_{ss,l,p} = N_{ss,q,l,p} = 100$ ,  $\forall l,p$ , ou seja, todos os horizontes de modelo são iguais, por simplicidade.

Para o caso do GPC, basta discretizar as matrizes de transferência. Considerando o mesmo período de amostragem de  $T_s = 1$  s, temos

$$\mathbf{P}(s) = \begin{bmatrix} \frac{0,74397z^{-1}}{z - 0,9419} & \frac{-0,87891z^{-3}}{z - 0,9535} \\ \frac{0,57856z^{-7}}{z - 0,9123} & \frac{-1,3015z^{-3}}{z - 0,9329} \end{bmatrix} \quad \mathbf{P}_q(s) = \begin{bmatrix} \frac{0,24666z^{-8}}{z - 0,9351} \\ \frac{0,3575z^{-4}}{z - 0,927} \end{bmatrix},$$

e, a partir deste modelo, obtemos a representação MFD do processo:

$$\begin{aligned}
A_1(z^{-1}) &= 1 - 2,8305z^{-1} + 2,6704z^{-2} - 0,8398z^{-3} \\
A_2(z^{-1}) &= 1 - 2,7723z^{-1} + 2,5618z^{-2} - 0,789z^{-3} \\
B_{1,1}(z^{-1}) &= 0,744 - 1,4051z^{-1} + 0,6633z^{-2} \\
B_{1,2}(z^{-1}) &= -0,8789 + 1,6497z^{-1} - 0,7741z^{-2} \\
B_{2,1}(z^{-1}) &= 0,5786 - 1,0761z^{-1} + 0,5004z^{-2} \\
B_{2,2}(z^{-1}) &= -1,3015 + 2,394z^{-1} - 1,1008z^{-2} \\
B_{q,1,1}(z^{-1}) &= 0 + 0,2467z^{-1} - 0,4675z^{-2} + 0,2215z^{-3} \\
B_{q,2,1}(z^{-1}) &= 0 + 0,3575z^{-1} - 0,6597z^{-2} + 0,3043z^{-3}
\end{aligned}$$

Como mencionado, serão consideradas duas sintonias, uma com ação antecipativa  $N_q = 0$  e outra sem. Em ambos os casos os horizontes são  $N_{1,1} = 2$ ,  $N_{1,2} = 3$ ,  $N_{2,1} = 40$ ,  $N_{2,2} = 30$  e  $N_{u,1} = N_{u,2} = 5$ . Já as ponderações são normalizadas pelos horizontes e são dadas por  $\delta_l = \frac{1}{N_l}$ ,  $\lambda_p = \frac{1}{N_{u,p}}$ ,  $\forall l, p$ .

Para a Sintonia 1, sem ação antecipativa, os resultados para seguimento de referência e rejeição de perturbação são apresentados nas figuras 3.12 e 3.13, respectivamente. Note que, por conta do acoplamento, quando há uma mudança de referência em uma das saídas, há um transitório nas demais saídas também, que pode ser visto em torno dos tempos 10s e 60s da Figura 3.12. No entanto, os controladores tentam minimizar este efeito da melhor maneira possível. Além disso, veja que para o seguimento de referências as respostas são exatamente as mesmas tanto para o DMC quanto para o GPC quando as sintonias são idênticas pois, apesar dos modelos serem diferentes, ambos os modelos representam adequadamente o comportamento do sistema, assim, as predições são iguais. No entanto, isso já não é verdade para a rejeição de perturbação não medida, como pode ser visto na Figura 3.13. Esta diferença de respostas se dá por conta da forma como cada algoritmo MPC modela perturbações não medidas. O GPC, de forma geral, possui uma resposta mais agressiva do que o DMC a este tipo de perturbações. Veja que o sinal de controle no caso GPC assume valores muito mais elevados e oscilatórios o que, por consequência, deixa a resposta de rejeição de perturbações mais oscilatória, mas mais rápida também. Esta agressividade do GPC a perturbações não medidas pode ser minimizada escolhendo adequadamente os polinômios  $C(z^{-1})$  do modelo, como já explicado no Volume 1.

Já para a Sintonia 2, com ação antecipativa e  $N_q = 0$ , a resposta de rejeição de perturbação pode ser vista na Figura 3.14. Veja que agora não há mais diferenças entre o GPC e o DMC, pois o efeito da perturbação foi definido no modelo, assim, as predições são idênticas, o que gera repostas idênticas já que a sintonia é a mesma.

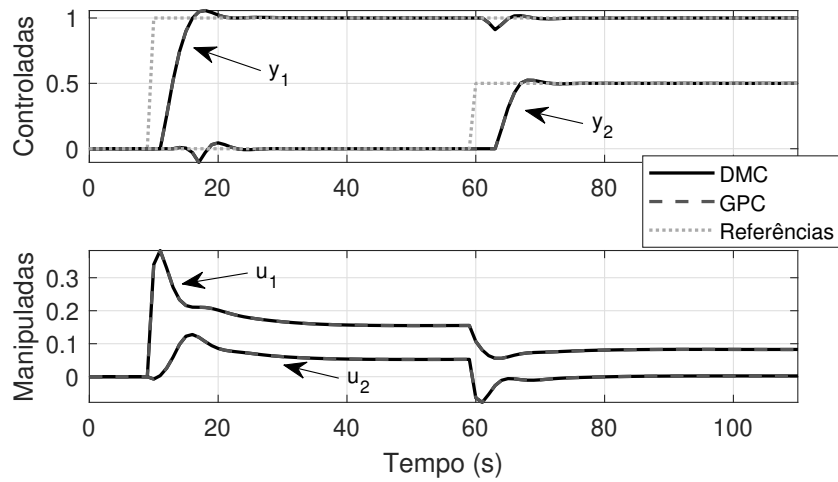


Figura 3.12: Estudo de caso 3.4.1 – Coluna de destilação água-metanol: Resposta e sinal de controle em malha fechada para seguimento de referências para a Sintonia 1.

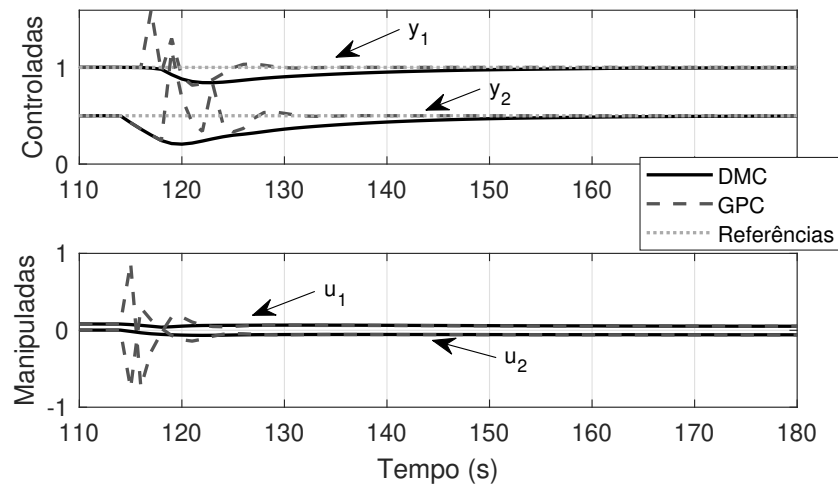


Figura 3.13: Estudo de caso 3.4.1 – Coluna de destilação água-metanol: Resposta e sinal de controle em malha fechada para rejeição de perturbações para a Sintonia 1.

### 3.4.2 Fracionador de óleo pesado

Considere o modelo de um fracionador de óleo pesado, que tem 3 variáveis que devem ser controladas: a composição de topo ( $Y_1$ ), a composição da extração lateral ( $Y_2$ ) e a temperatura da base ( $Y_3$ ). Para isso se manipulam as vazões de extração de topo ( $U_1$ ) e lateral ( $U_2$ ) e o refluxo da base ( $U_3$ ). O

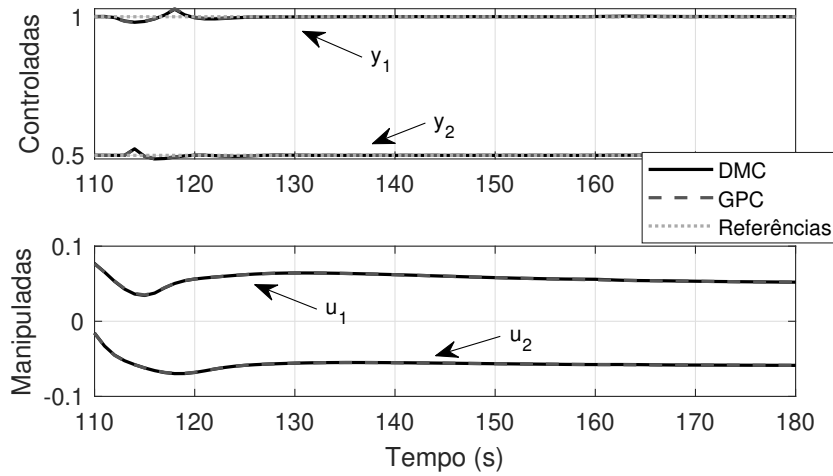


Figura 3.14: Estudo de caso 3.4.1 – Coluna de destilação água-metanol: Resposta e sinal de controle em malha fechada para rejeição de perturbações para a Sintonia 2.

modelo do processo, funcionando nas vizinhanças de um ponto de operação, é dado por:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} = \begin{bmatrix} \frac{4,05e^{-27s}}{1 + 50s} & \frac{1,77e^{-28s}}{1 + 60s} & \frac{5,88e^{-27s}}{1 + 50s} \\ \frac{5,39e^{-18s}}{1 + 50s} & \frac{5,72e^{-14s}}{1 + 60s} & \frac{6,9e^{-15s}}{1 + 40s} \\ \frac{4,38e^{-20s}}{1 + 33s} & \frac{4,42e^{-22s}}{1 + 44s} & \frac{7,2}{1 + 19s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{bmatrix}.$$

O modelo é incremental, dessa forma, podemos considerar que o ponto de equilíbrio inicial é zero para todas as variáveis. Note que se trata de um processo muito acoplado e com importantes atrasos de transporte. Vamos analisar diversos ajustes de horizontes de controle e ponderações para estudar os efeitos na resposta e na robustez, isto é, a sensibilidade do sistema em malha fechada aos possíveis erros de modelagem. Faremos este estudo de forma intuitiva, por simulação de cenários, para motivar estudos posteriores. Além disso, neste caso vamos considerar restrições no problema de otimização, tanto para as variáveis de controle como para os incrementos das mesmas.

Para efeitos de simulação, vamos considerar o modelo de matriz função de transferência como se fosse o processo real. O cenário de simulação considera um tempo total de simulação de 300 s. O sistema parte de  $u_1 = u_2 = u_3 = 0$

e  $y_1 = y_2 = y_3 = 0$  e em  $t = 10$  s uma variação do tipo degrau unitário da referência de  $y_1$  é aplicada. Posteriormente, em  $t = 100$  s se aplica uma variação do tipo degrau de amplitude 0,5 na referência de  $y_2$ . Finalmente, em  $t = 170$  s se aplica uma variação do tipo degrau de amplitude 0,2 na referência de  $y_3$ .

O controlador a ser utilizado é um GPC com período de amostragem de  $T_s = 1,0$  s. O modelo MFD do processo pode ser obtido facilmente através dos passos já explicados. Em termos de sintonia, os horizontes iniciais foram definidos de acordo com os menores atrasos de cada saída, assim,  $N_{1,1} = 28$ ,  $N_{1,2} = 15$  e  $N_{1,3} = 1$ . Para o primeiro cenário, o final dos horizontes de predição foram definidos de modo a se obter um horizonte de predição do tamanho de uma constante de tempo. Dessa forma,  $N_{2,1} = 88$ ,  $N_{2,2} = 75$  e  $N_{2,3} = 45$ . Os horizontes de controle foram definidos como  $N_{u,1} = N_{u,2} = N_{u,3} = 5$ . As ponderações foram normalizadas com os horizontes e definidas como  $\delta_i = 1/N_{y,i}$ ,  $\lambda_i = 1/N_{u,i}$ ,  $\forall i$ . Os resultados para este caso (Sintonia 1) podem ser vistos na Figura 3.15. Podemos perceber que, mesmo sendo uma planta bastante acoplada, o controle consegue levar as saídas para as referências desejadas de forma independente, apesar de haver uma perturbação nas outras saídas quando há uma troca de referência.

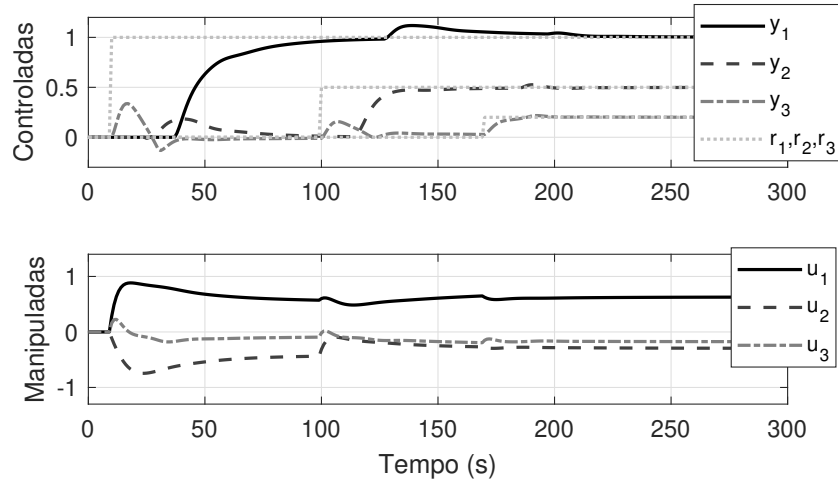


Figura 3.15: Estudo de caso 3.4.2 – Fracionador de óleo pesado: Resposta e sinal de controle do fracionador de óleo em malha fechada para seguimento de referências para a Sintonia 1.

Para avaliarmos o efeito da escolha do horizonte de predição, vamos considerar a metade dos valores atuais e definir  $N_{2,1} = 44$ ,  $N_{2,2} = 38$  e  $N_{2,3} = 23$ . Os resultados para este caso (Sintonia 2), podem ser observados na Figura 3.16. Podemos perceber que com o horizonte de predição menor



houve uma maior ênfase do controle no regime transitório, apresentando uma resposta mais rápida e sinais de controle de amplitude mais elevada. Também observamos que, neste caso, o acoplamento entre as saídas é menor, assim temos menos variações em cada uma das saídas quando se mudar as referências das outras saídas.

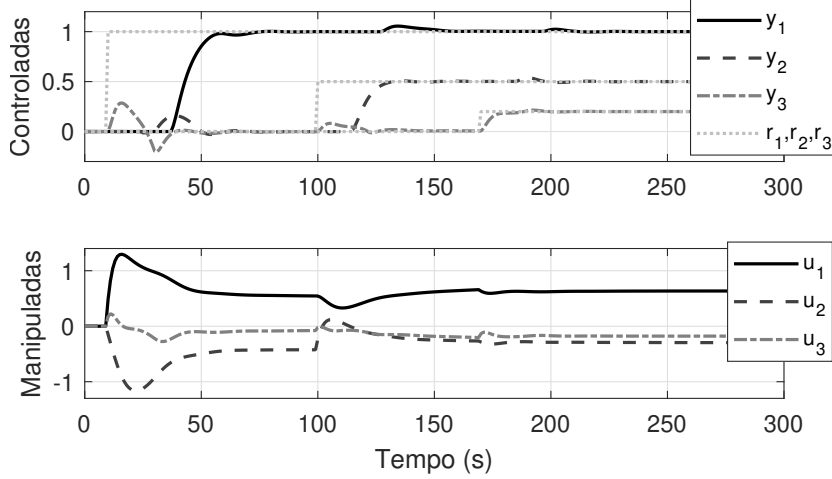


Figura 3.16: Estudo de caso 3.4.2 – Fracionador de óleo pesado: Resposta e sinal de controle do fracionador de óleo em malha fechada para seguimento de referências para a Sintonia 2.

Por outro lado, podemos avaliar o resultado da variação nas ponderações dos erros ou esforços de controle. Os horizontes de predição foram definidos com os valores iniciais e as ponderações no esforço de controle foram considerados constantes no horizonte e definidos com valores 10 vezes menores, ou seja,  $\lambda_i = 0,1/N_{u,i}, \forall i$ . Os resultados da simulação para este caso (Sintonia 3), podem ser constatados na Figura 3.17. Podemos perceber as respostas com dinâmicas mais rápidas e sinais de controle maiores. É interessante ressaltar que os valores poderiam ser atribuídos independentemente e acelerar a resposta de apenas uma das variáveis como pode ser visto na Figura 3.18. Nesse caso (Sintonia 4), apenas a variável  $U_2$  foi ponderada diferente,  $\lambda_1 = 1/N_{u,1}$ ,  $\lambda_2 = 0,1/N_{u,2}$  e  $\lambda_3 = 1/N_{u,3}$ . Vale reforçar que estas considerações sobre a sintonia também continuam válidas para outros algoritmos MPC, tal como o DMC.

Por fim, podemos avaliar a utilização de restrições no sinal e no incremento de controle. Dessa forma, podemos restringir o sinal de controle e seus incrementos com valores máximos. Nesse cenário, utilizamos as mesmas ponderações da Sintonia 3, porém, foram inseridas as restrições  $-0.8 \leq u_1, u_2, u_3 \leq 0.8$  e  $-0.1 \leq \Delta u_1, \Delta u_2, \Delta u_3 \leq 0.1$ . Os resultados da

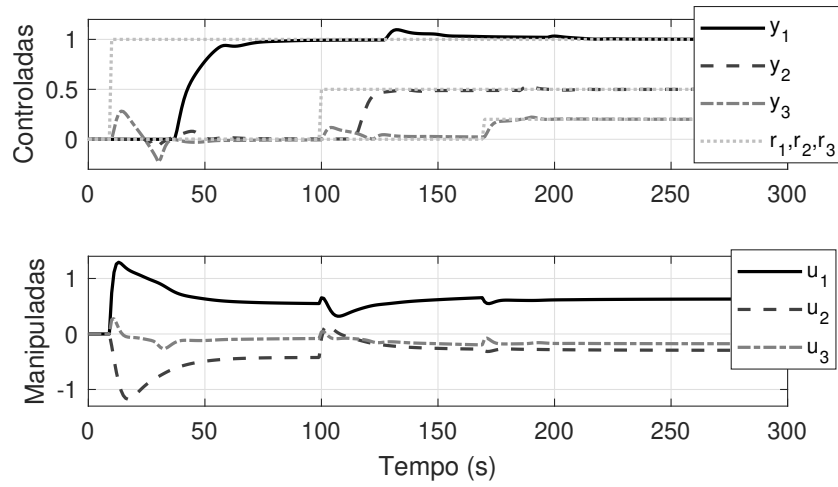


Figura 3.17: Estudo de caso 3.4.2 – Fracionador de óleo pesado: Resposta e sinal de controle do fracionador de óleo em malha fechada para seguimento de referências para a Sintonia 3.

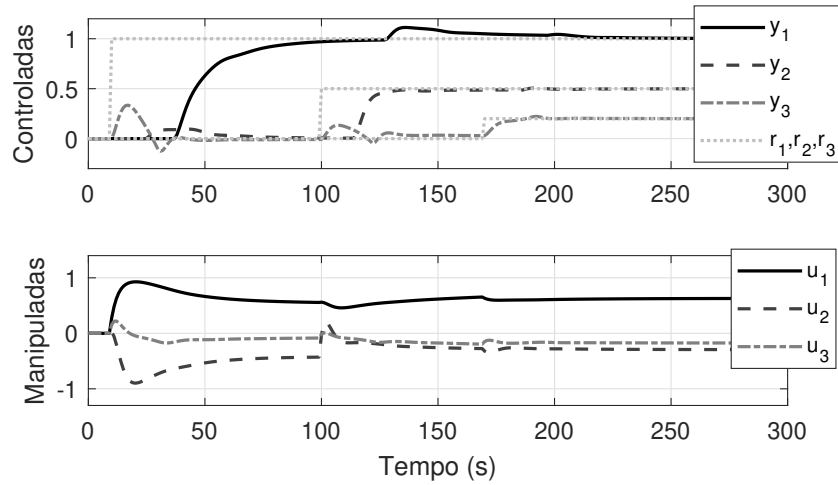


Figura 3.18: Estudo de caso 3.4.2 – Fracionador de óleo pesado: Resposta e sinal de controle do fracionador de óleo em malha fechada para seguimento de referências para a Sintonia 4.

simulação são apresentados na Figura 3.19. Podemos perceber os sinais de controle sendo limitados, com os valores pré-estabelecidos de 0.8, no começo da simulação e também uma taxa de variação do sinal de controle menor e limitada em 0.1.

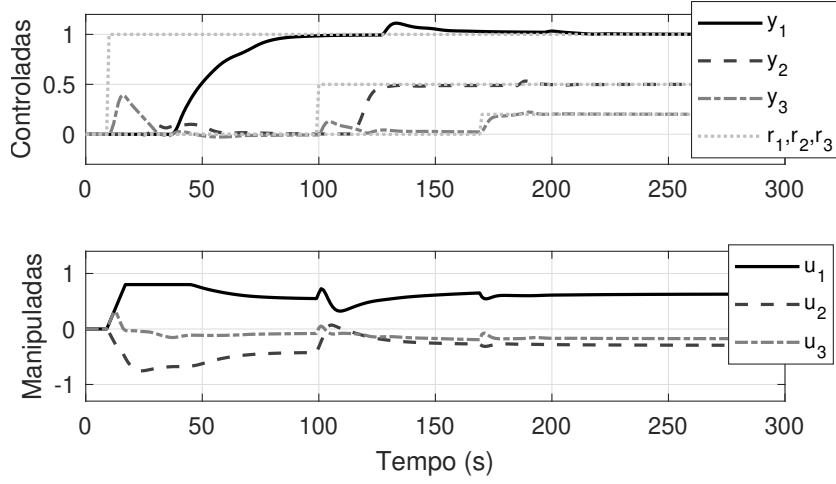


Figura 3.19: Estudo de caso 3.4.2 – Fracionador de óleo pesado: Resposta e sinal de controle do fracionador de óleo em malha fechada para seguimento de referências para a Sintonia 5, com restrições.

### 3.4.3 Sistema de quatro tanques

Neste estudo de caso, trabalharemos com um sistema de quatro tanques acoplados. Este é um sistema não-quadrado com mais saídas do que entradas, então o ajuste do controlador precisa ser feito de maneira adequada para que as especificações sejam atendidas. O diagrama esquemático do processo é apresentado na Figura 3.20. As saídas do processo são os níveis dos tanques ( $h_1, h_2, h_3, h_4$ ), e as manipuladas são as tensões aplicadas nas bombas 1 e 2 ( $v_1$  e  $v_2$ ). Veja que a vazão da Bomba 1 é dividida entre os tanques 1 e 4, e a vazão da Bomba 2 é dividida entre os tanques 2 e 3. Essa divisão é definida pelas aberturas das válvulas 1 e 2,  $\gamma_1, \gamma_2 \in [0,1]$ . Este sistema possui um comportamento não-linear dado pelas equações abaixo:

$$\begin{cases} \frac{dh_1}{dt} = -\frac{a_1}{A_1} \sqrt{2gh_1} + \frac{a_3}{A_1} \sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1 \\ \frac{dh_2}{dt} = -\frac{a_2}{A_2} \sqrt{2gh_2} + \frac{a_4}{A_2} \sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2 \\ \frac{dh_3}{dt} = -\frac{a_3}{A_3} \sqrt{2gh_3} + \frac{(1-\gamma_2) k_2}{A_3} v_2 \\ \frac{dh_4}{dt} = -\frac{a_4}{A_4} \sqrt{2gh_4} + \frac{(1-\gamma_1) k_1}{A_4} v_1 \end{cases}$$

sendo  $A_i$  as áreas das seções transversais de cada tanque,  $a_i$  o diâmetro do tubo de saída de cada tanque,  $g$  é a aceleração da gravidade, e  $k_i$  são

constantes relacionadas às bombas. Para este estudo de caso, consideraremos  $A_1 = A_3 = 28 \text{ cm}^2$ ,  $A_2 = A_4 = 32 \text{ cm}^2$ ,  $a_1 = a_3 = 0,071 \text{ cm}^2$ ,  $a_2 = a_4 = 0,057 \text{ cm}^2$ ,  $k_1 = 3,33 \text{ cm}^3 \text{ V}^{-1} \text{ s}^{-1}$ ,  $k_2 = 3,35 \text{ cm}^3 \text{ V}^{-1} \text{ s}^{-1}$ ,  $g = 981 \text{ cm s}^{-2}$ ,  $\gamma_1 = 0,7$  e  $\gamma_2 = 0,6$ . A altura máxima de cada tanque é 20 cm e a tensão das bombas está limitada à faixa  $[0 \ 12] \text{ V}$ .

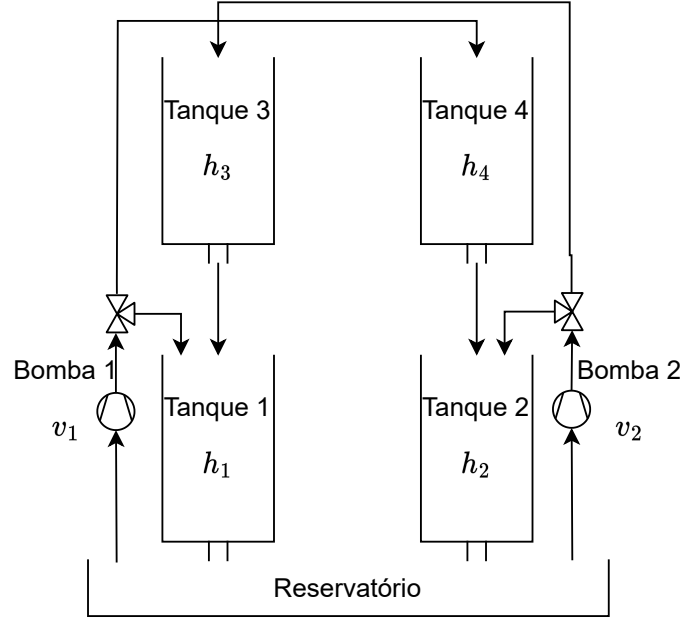


Figura 3.20: Estudo de caso 3.4.3 – Quatro Tanques: diagrama esquemático do processo.

Como o processo é não linear, consideraremos que este operará em torno de um ponto de operação, que é dado por

$$\bar{h} = [12,2630 \ 12,7831 \ 1,6339 \ 1,4090]^T, \\ \bar{v} = [3,0 \ 3,0]^T.$$

Um modelo por funções de transferência pode ser obtido linearizando o modelo do processo no ponto de operação escolhido. Considerando um período de amostragem de  $T_s = 2 \text{ s}$ , a matriz de transferência do sistema é:

$$\mathbf{G}(z^{-1}) = \begin{bmatrix} \frac{0,16386}{(z-0,9684)} & \frac{0,004041(z+0,9608)}{(z-0,9684)(z-0,9159)} \\ \frac{0,0020149(z+0,9709)}{(z-0,9782)(z-0,9357)} & \frac{0,12425}{(z-0,9782)} \\ 0 & \frac{0,09163}{(z-0,9159)} \\ \frac{0,060408}{(z-0,9357)} & 0 \end{bmatrix}.$$

Agora, vamos analisar o comportamento do sistema para diferentes sintonias. A Sintonia 1 é definida da seguinte forma: os horizontes de predição são iguais para todas as saídas  $N_1 = 1$  e  $N_2 = 50$ ; os horizontes de controle são iguais para todas as manipuladas  $N_u = 5$ ; restrições nos valores máximos das manipuladas  $0 \leq v_p \leq 12, \forall p$ ; as ponderações do esforço de controle são  $\lambda = 1/N_{u,p}, \forall p$ ; e as ponderações das saídas são  $\delta_l = 1/N_{y,l}, \forall l$ . Como as ponderações das saídas são todas iguais, nenhuma tem prioridade em relação à outra.

O cenário de simulação inicia com o sistema no ponto de operação previamente definido. Em  $t = 5$  s, há uma mudança de referência para o tanque 1 que passa de 12,2630 cm para 14,7156 cm, e em  $t = 60$  s, a referência para o tanque 2 muda de 12,7831 cm para 10,2265 cm, e em  $t = 120$  s as referências de cada um dos tanques muda para 8,7664 cm, 9,3579 cm, 1,2273 cm e 0,9785 cm, respectivamente.

O resultado de simulação para a Sintonia 1 é mostrado na Figura 3.21. Veja que o sistema inicia no ponto de operação e, como as referências são iguais aos valores dos níveis no ponto de operação, o erro de rastreamento é zero. Porém, quando ocorre a primeira e a segunda mudanças de referência, veja que existe um transitório inicial e o sistema converge para um novo ponto de operação, no entanto, o erro não é nulo. Como discutido anteriormente, como o sistema possui mais saídas do que entradas, não é possível escolher as referências das saídas de forma arbitrária e esperar que o controlador garanta erro nulo em regime permanente, pois não há graus de liberdade suficientes para o controlador garantir isso. Neste caso, o controlador irá para um ponto de operação factível que minimize a função custo do problema de otimização do MPC, e este ponto dependerá da sintonia e das próprias características do processo. Ainda nesta mesma simulação, veja que as referências definidas a partir de  $t = 120$  s são rastreadas com erro nulo. Isto acontece pois os valores das referências foram adequadamente escolhidos para ser um ponto factível de acordo com as características estáticas do processo, sendo esta a única maneira de garantir erro nulo para todas as saídas caso o sistema possua mais saídas do que entradas.

Considere, agora, uma Sintonia 2, igual à Sintonia 1, exceto que a ponderação do nível do tanque 2 foi aumentada para  $\delta_2 = 10/N_{y,2}$ , ou seja, desejamos que o controlador priorize o seguimento da saída 2 em detrimento das outras. O resultado de simulação com esta sintonia é apresentado na Figura 3.22. Veja que o erro de rastreamento de referência para a saída 2 diminuiu consideravelmente, porém, ainda não vai a zero, e os erros das outras saídas aumentaram. Assim, o projetista, no caso de sistemas não-quadrados, pode utilizar as ponderações da sintonia para priorizar determinadas saídas de acordo com a relevância delas no processo.

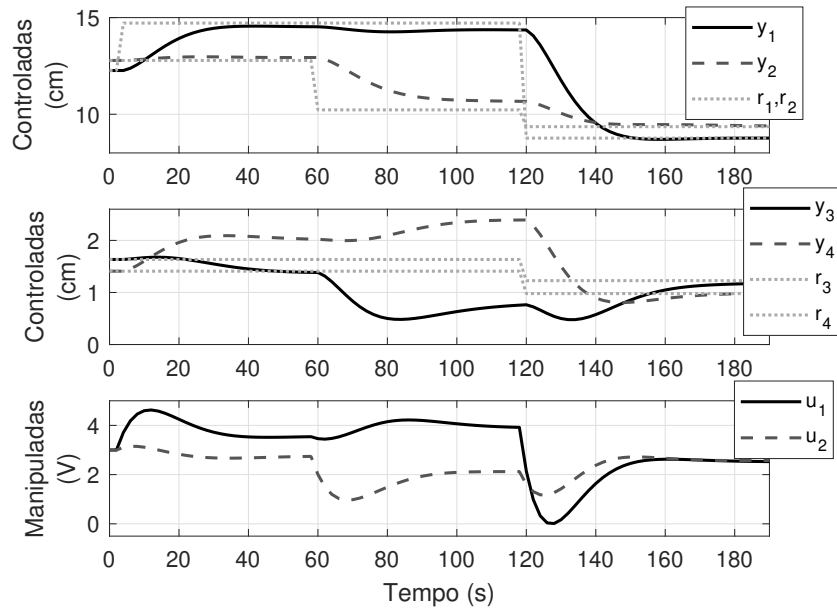


Figura 3.21: Estudo de caso 3.4.3 – Quatro Tanques: Resposta e sinal de controle em malha fechada para seguimento de referências para a Sintonia 1.

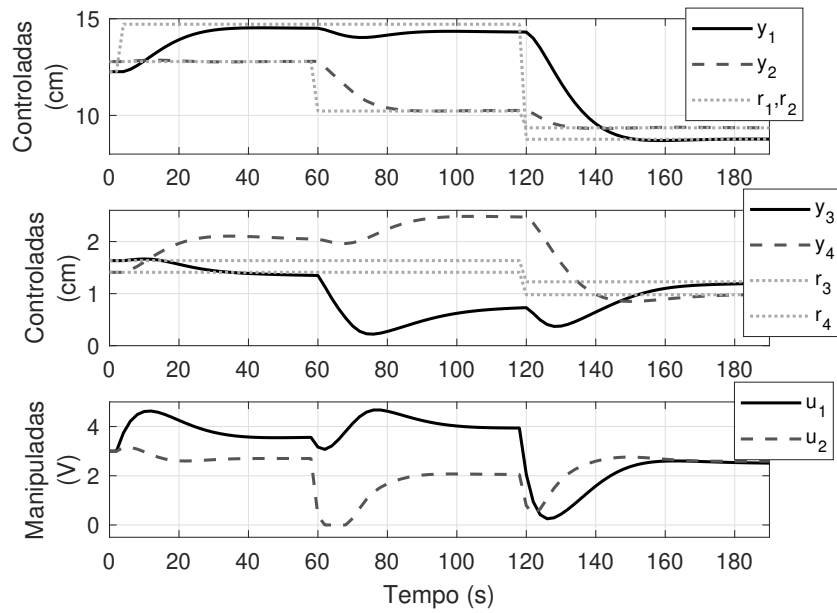


Figura 3.22: Estudo de caso 3.4.3 – Quatro Tanques: Resposta e sinal de controle em malha fechada para seguimento de referências para a Sintonia 2.

Por fim, considere uma Sintonia 3, igual à Sintonia 1, exceto que  $\delta_3 = \delta_4 =$

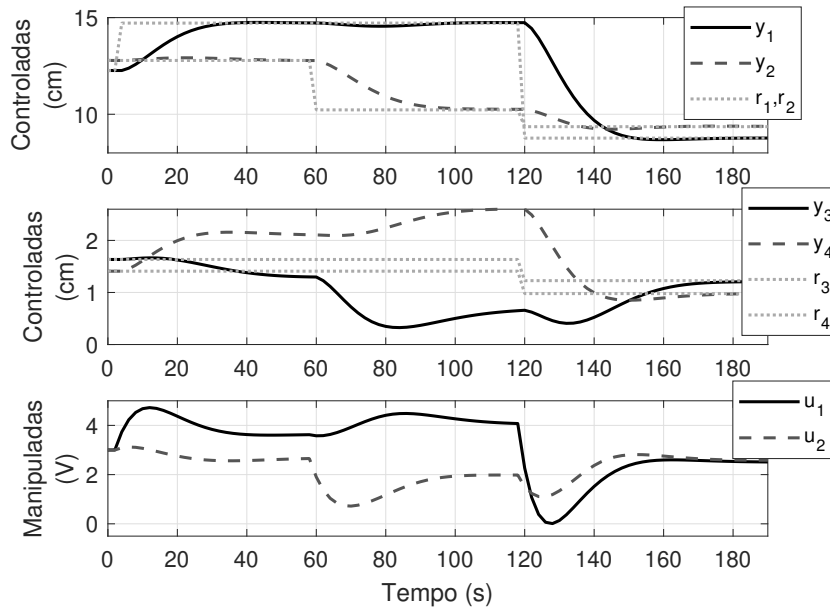


Figura 3.23: Estudo de caso 3.4.3 – Quatro Tanques: Resposta e sinal de controle em malha fechada para seguimento de referências para a Sintonia 3.

0, ou seja, desejamos que o controlador ignore o rastreamento de referência das saídas 3 e 4. O resultado de simulação é apresentado na Figura 3.23. Veja que agora o controlador consegue garantir erro nulo para as saídas 1 e 2 pois indicamos que as saídas 3 e 4 não tem relevância alguma no problema de controle. Assim, do ponto de vista do controlador, há apenas duas saídas e duas entradas, ou seja, o sistema se tornou quadrado, e agora ele é capaz de garantir erro nulo para as duas saídas independente dos valores das referências. Vale ressaltar que, neste caso, as saídas 3 e 4 ficam totalmente livres e isso, na prática, pode não ser desejável. **No Capítulo ??, discutiremos uma abordagem mais avançada, chamada controle por bandas, que resolve este problema.**

### 3.4.4 Reator Continuamente Agitado (CSTR)

Neste problema, consideramos um reator continuamente agitado (*Continuous Stirred Tank Reactor* - CSTR) não-isotérmico descrito em [Gup98], onde uma reação química exotérmica  $A \rightarrow B$  acontece, e cujo diagrama esquemático é apresentado na Figura 3.24. A dinâmica do processo é não linear, e são dadas pelas equações abaixo, que dizem respeito ao balanço de massa, dos

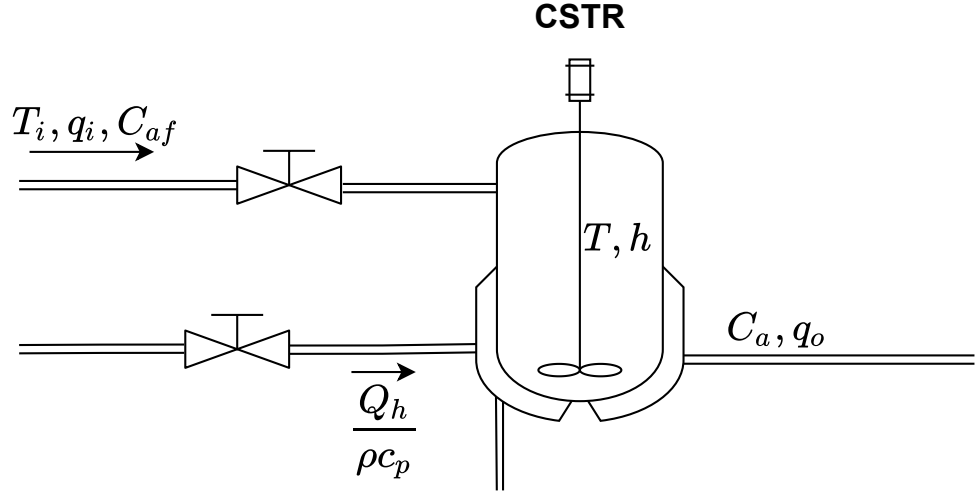


Figura 3.24: Estudo de caso 3.4.4 – CSTR: diagrama esquemático.

componentes e de energia:

$$\begin{aligned}\frac{dh(t)}{dt} &= \frac{q_i(t) - q_o(t)}{A_c} \\ \frac{dC_a(t)}{dt} &= q_i(t) \frac{C_{af}(t) - C_a(t)}{V} - R(t)C_a(t) \\ \frac{dT(t)}{dt} &= q_i(t) \frac{T_i(t) - T(t)}{V} - \frac{\Delta H}{\rho c_p} R(t)C_a(t) - \frac{Q_h(t)}{\rho c_p V} \\ R(t) &= k_0 e^{-\frac{E}{RT(t)}}$$

sendo que as variáveis controladas são o nível dentro do tanque  $h$ , a concentração de saída do produto  $A$   $C_a$ , a temperatura dentro do reator  $T$ . As variáveis manipuladas são a vazão de saída  $q_o$ , a concentração do produto  $A$  na alimentação do tanque  $C_{af}$ , e a taxa de remoção de calor normalizada  $\frac{Q_h}{\rho c_p}$ . As constantes presentes na equação e o ponto de operação do processo são apresentados na Tabela 3.1.

Como a intenção é aplicar um algoritmo DMC, necessitamos dos modelos de resposta ao degrau do sistema. Para isso, foram aplicados degraus nas entradas do processo de 2 % em torno do ponto de operação. Os resultados são apresentados na Figura 3.25, veja que todas as respostas são estáveis exceto a que relaciona  $q_o$  com  $h$ , que possui uma dinâmica integradora. Por conta disto, não é possível aplicar o algoritmo DMC original, sendo necessário o GDMC para estabilizar o cálculo da resposta livre.



Tabela 3.1: Parâmetros e ponto de operação do tanque reator

Nível dentro do reator	$h^*$	1 m
Concentração medida do produto A	$C_a^*$	1 kmol m <sup>-3</sup>
Temperatura dentro do reator	$T^*$	400 K
Vazão de saída	$q_o^*$	0,005 m <sup>3</sup> s <sup>-1</sup>
Concentração de A na alimentação	$C_{af}^*$	5 kmol m <sup>-3</sup>
Taxa de remoção de calor	$\frac{Q_h^*}{\rho c_p}$	0,75 K m <sup>3</sup> s <sup>-1</sup>
Vazão de entrada	$q_i^*$	0,005 m <sup>3</sup> s <sup>-1</sup>
Temperatura de entrada do fluido	$T_i^*$	350 K
Termo de energia de ativação	$E/R$	1000 K
Área do reator	$A_c$	0,05 m <sup>2</sup>
Volume do reator	$V$	0,05 m <sup>3</sup>
Calor da reação	$\Delta H$	-50 000 000 cal/kmol
Densidades dos líquidos	$\rho$	1 000 000 g m <sup>-3</sup>
Calor específico	$c_p$	1 cal/gK
Constante da taxa de reação	$k_0$	4,972 997 584 s <sup>-1</sup>
Período de amostragem	$T_s$	3 s

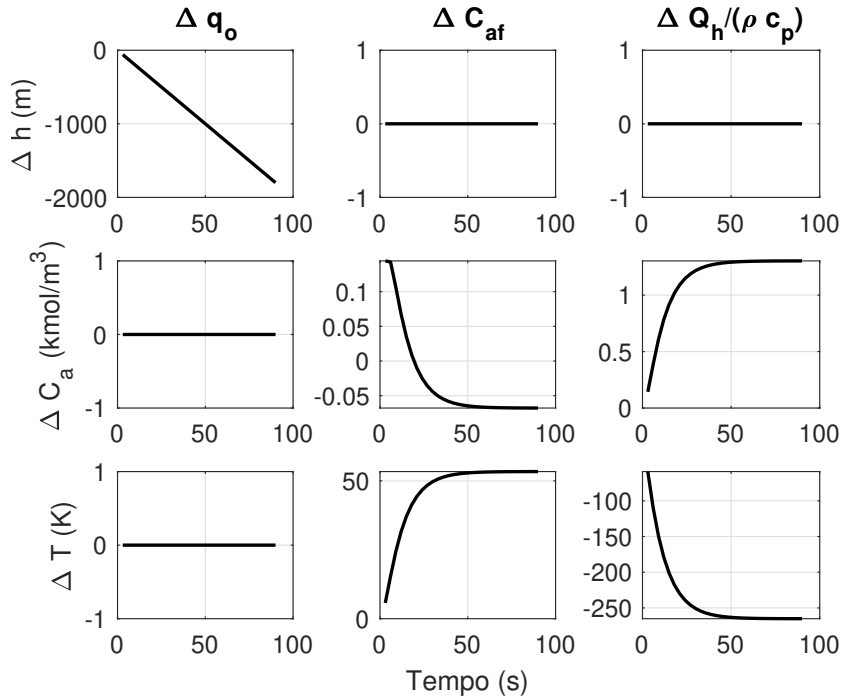


Figura 3.25: Estudo de caso 3.4.4 – CSTR: modelos de resposta ao degrau.

Então, um primeiro passo é definir os filtros dos erros de predição para a primeira saída de forma a estabilizar a resposta livre. Como a única raiz indesejada é a integradora, as condições do filtro são:

$$\begin{aligned} F_{e,1,j}(z)|_{z=1} &= 1 \rightarrow \text{condição de ganho unitário} \\ \left. \frac{d(z^j - F_{e,1,j}(z))}{dz} \right|_{z=1} &= 0 \rightarrow \text{condição de estabilidade} \end{aligned}$$

Para as outras saídas  $F_{e,i,j} = 1$ , ou seja, não são necessário filtros. Pelo número de condições a serem satisfeitas, um filtro de primeira ordem seria suficiente. Mas, para minimizar o efeito de ruídos e ajudar na robustez, um filtro de segunda ordem será utilizado, e sua estrutura é da forma:

$$F_{e,1,j} = \frac{(a(j)z^2 + b(j)z)}{(z - z_f)^2}.$$

Com a estrutura do filtro definida, a derivada que aparece na condição de estabilidade pode ser escrita como:

$$\begin{aligned} \frac{d(z^j - F_{e,j}(z))}{dz} &= jz^{j-1} - \frac{(2a(j)z + b(j))(z - z_f)^2 - 2(a(j)z^2 + b(j)z)(z - z_f)}{(z - z_f)^4}, \\ &= jz^{j-1} - \frac{(2a(j)z + b(j))(z - z_f) - 2(a(j)z^2 + b(j)z)}{(z - z_f)^3}, \\ &= jz^{j-1} - \frac{a(j)(-2zz_f) + b(j)(-z - z_f)}{(z - z_f)^3}. \end{aligned}$$

Agora podemos calcular os coeficientes  $a(j)$  e  $b(j)$  de tal modo que as condições acima sejam satisfeitas para  $j = N_1 \dots N_2$ :

$$\begin{aligned} \frac{a(j) + b(j)}{(1 - z_f)^2} &= 1 \rightarrow a(j) + b(j) = (1 - z_f)^2 \\ j - \frac{a(j)(-2z_f) + b(j)(-1 - z_f)}{(1 - z_f)^3} &= 0 \rightarrow (-2z_f)a(j) + (-1 - z_f)b(j) = j(1 - z_f)^3 \end{aligned}$$

e note que estas últimas equações podem ser rearranjadas como um conjunto de equações lineares que é de fácil solução:

$$\begin{bmatrix} 1 & 1 \\ -2z_f & -1 - z_f \end{bmatrix} \begin{bmatrix} a(j) \\ b(j) \end{bmatrix} = \begin{bmatrix} (1 - z_f)^2 \\ j(1 - z_f)^3 \end{bmatrix}.$$

Os demais parâmetros de sintonia são:  $N_{1,i} = 1$  e  $N_{2,i} = 30$ ,  $\forall i$ ;  $N_{u,p} = 5$ ,  $\forall p$ ;  $N_{f,1} = N_{ss,2} = N_{ss,3} = 30$ , e as ponderações são todas iguais a um mas foram normalizadas pelos horizontes e pelo quadrado dos

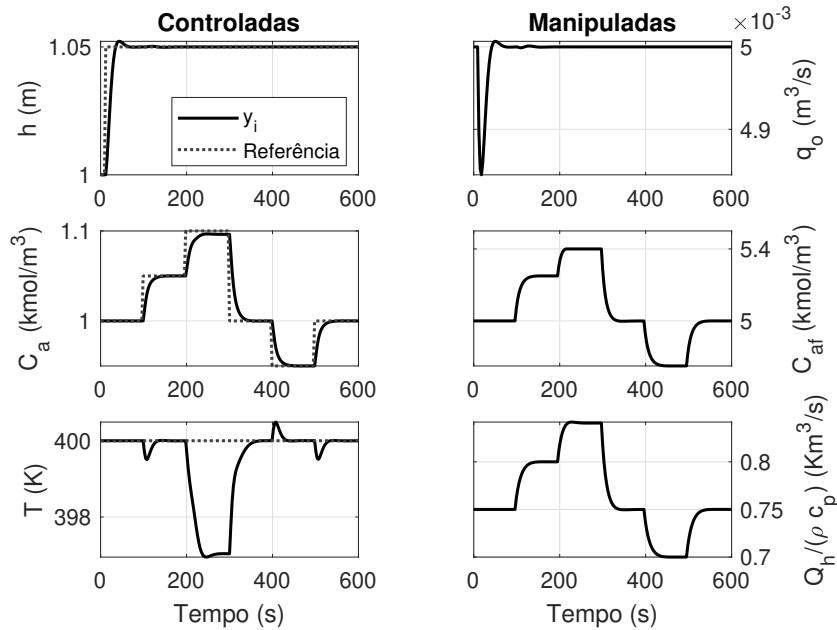


Figura 3.26: Estudo de caso 3.4.4 – CSTR: resposta em malha fechada com o GDMC para seguimento de referência.

pontos de operação. Além disso, são consideradas restrições para as manipuladas:  $q_0 \in [0; 0,01] \text{ m}^3 \text{ s}^{-1}$ ,  $C_{af} \in [4,6; 5,4] \text{ kmol m}^{-3}$ ,  $Q_h/(\rho c_p) \in [0,45; 1,05] \text{ K m}^3 \text{ s}^{-1}$ .

Os resultados de simulação para diversas mudanças de referências são mostrados na Figura 3.26. Veja que o GDMC permanece estável e consegue garantir o seguimento de referências perfeitamente, exceto quando as manipuladas estão saturadas o que ocorre entre 200 e 300s da simulação. A resposta para rejeição de perturbações pode ser vista na Figura 3.27, e está identificada como GDMC1. Foram aplicados degraus nas duas perturbações,  $q_i$  passa de 0,005 K para 0,005 25 K em  $t = 700 \text{ s}$  e  $T_i$  passa de 350 K para 332,5 K em  $t = 800 \text{ s}$ . Neste caso, veja que o controlador consegue rejeitar o efeito das perturbações adequadamente.

Um segundo ajuste do GDMC foi feito para melhorar a resposta de rejeição de perturbações. Analisando as respostas ao degrau próximas do ponto de operação, apresentadas na Figura 3.28, é possível identificar que as dinâmicas das perturbações para  $C_a$  e  $T_i$  possuem constantes de tempo próximas de 15s. Então, é possível utilizar esta informação para reprojetar os filtros do erro de predição para obtermos melhores resultados.

Considerando a constante de tempo encontrada, o polo indesejado discreto está localizado em  $p_z = 0,8187$ . Assim, para as saídas  $C_a$  e  $T_i$ , os filtros

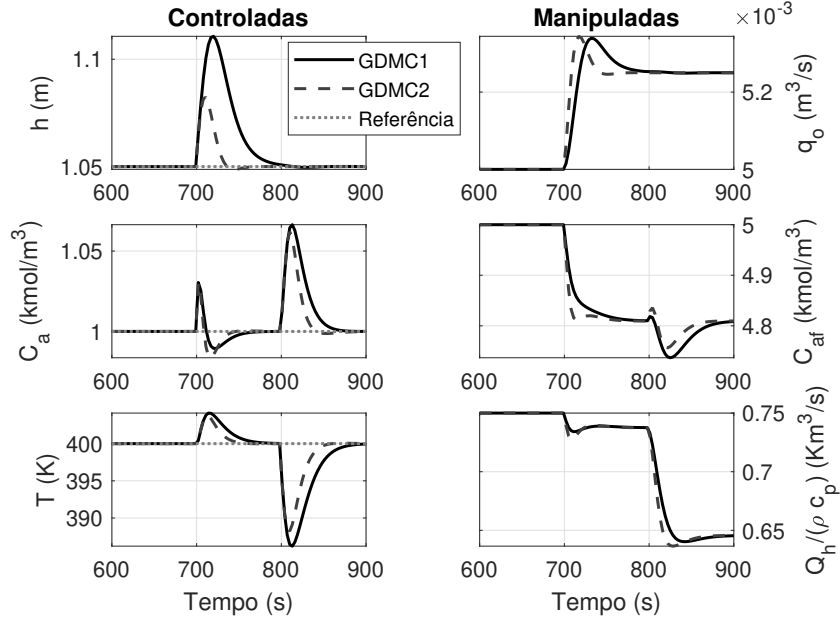


Figura 3.27: Estudo de caso 3.4.4 – CSTR: resposta em malha fechada com o GDMC para rejeição de perturbações.

$F_{e,i,j}(z)$  devem satisfazer as seguintes condições:

$$\begin{aligned} F_{e,1,j}(z)|_{z=1} &= 1 \rightarrow \text{condição de ganho unitário} \\ z^j - F_{e,1,j}(z)|_{z=0,8187} &= 0 \rightarrow \text{condição para polo indesejado} \end{aligned}$$

que podem ser escritas no seguinte formato matricial:

$$\begin{bmatrix} 1 & 1 \\ p_z^2 & p_z \end{bmatrix} \begin{bmatrix} a(j) \\ b(j) \end{bmatrix} = \begin{bmatrix} (1 - z_f)^2 \\ pz^j(p_z - z_f)^2 \end{bmatrix}.$$

Para esta segunda sintonia, consideramos  $z_f = 0,4$ , ou seja, o tempo de assentamento do filtro é mais rápido, o que deve ajudar na rejeição de perturbações. Os resultados para este caso (GDMC2) podem ser vistos na Figura 3.27, e note que a rejeição de perturbações melhorou, como esperado. No entanto, vale lembrar que, em geral, acelerar a rejeição de perturbações piora a robustez do sistema em malha fechada, assim, é sempre necessário balancear a rejeição de perturbações com a robustez do sistema.

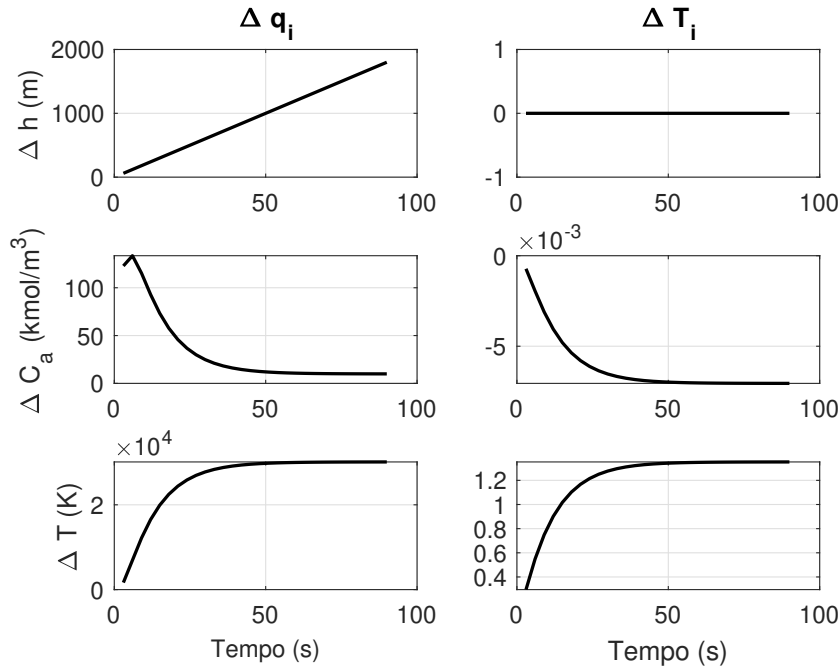


Figura 3.28: Estudo de caso 3.4.4 – CSTR: modelos de resposta ao degrau para perturbações.

### 3.5 Exercícios propostos

1. Realize uma análise equivalente a apresentada no Exemplo 3.1 para o processo multivariável dado pelo modelo:

$$6 \frac{d^2 y_1(t)}{dt^2} + 5 \frac{dy_1(t)}{dt} + y_1(t) = 3u_1(t) - u_2(t),$$

$$8 \frac{dy_2(t)}{dt} + y_2(t) = 2,5u_1(t) + 3u_2(t),$$

considerando horizontes de predição de 4 amostras para as duas saídas e horizonte de controle de 2 amostras para as duas manipuladas.

2. No processo do exercício anterior se deseja aplicar um controle MPC MIMO de forma tal a conseguir levar o sistema em malha fechada para um ponto de equilíbrio dado por  $y_1 = 3$  e  $y_2 = 2$ . Assumindo que o MPC ajustado mantém a estabilidade em malha fechada, e que o controle tem ação integral, o ponto de equilíbrio será alcançado? Caso o modelo do processo fosse dado por:

$$6 \frac{d^2 y_1(t)}{dt^2} + 5 \frac{dy_1(t)}{dt} + y_1(t) = 3u_1(t) - u_2(t),$$

$$8\frac{dy_2(t)}{dt} + y_2(t) = 4,5u_1(t) - 1,5u_2(t),$$

o equilíbrio seria alcançado? Qual seria neste caso a condição para o ponto de equilíbrio ser alcançável?

3. No problema do Exercício 1 vai ser aplicado um DMC MIMO. Se a variável  $y_1$  está no intervalo  $[0, 100]$ ,  $y_2$  está no intervalo  $[0, 10]$ ,  $\Delta u_1$  está no intervalo  $[-2, 2]$  e  $\Delta u_2$  está no intervalo  $[-5, 5]$ , defina uma normalização adequada para os pesos a serem usados na função custo do DMC.
4. Ainda para o problema do Exercício 1, considerando que os intervalos das variáveis do exercício anterior são colocados como restrições do problema de otimização: (a) determine as dimensões das matrizes a serem usadas na formulação do problema QP a ser resolvido a cada amostragem, (b) Encontre as matrizes  $\bar{\mathbf{R}}$  e  $\bar{\mathbf{r}}$ .
5. Considere o sistema dos quatro tanques apresentado neste capítulo. Suponha que se deseja controlar o nível dos 4 tanques com as seguintes especificações: o nível dos tanques 1 e 2 deve ser mantido no valor de referência desejado, já os dos tanques 3 e 4 apenas dentro da faixa de operação  $[1, 19]$ cm, ou seja, evitar que trasborde ou esvazie. Considere um ponto de operação com  $h_1 = h_2 = 10$ cm. Obtenha um modelo de resposta ao degrau nesse ponto de operação usando a metodologia do DMC e sintonize um DMC MIMO. Analise as respostas em malha fechada do sistema para variações da referência de nível, considerando diferentes amplitudes dos degraus e discuta. Analise o efeito dos parâmetros de ajuste do DMC MIMO nas respostas em malha fechada. Analise também a robustez do seu controle, simulando situações onde o processo é simulado com parâmetros diferentes dos usados no ajuste do DMC MIMO.
6. Considere o problema do tanque térmico do Exemplo 3.10. Implemente a ação antecipativa do controlador supondo que a perturbação é medida. Compare as respostas do novo controlador com as obtidas sem a ação antecipativa. Analise o caso sem erro de modelo e o caso com erro de modelo. Para este segundo caso assuma que o processo é simulado com:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1,1e^{-1,2s}}{1+7s} & \frac{-4e^{-1,1s}}{1+12s} \\ 0 & \frac{5}{1+17s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} + \begin{bmatrix} \frac{2e^{-0,9s}}{1+11s} \\ \frac{-2,5}{1+14s} \end{bmatrix} Q(s)$$

e o modelo é o mesmo usado no Exemplo 3.10.

7. Considere o sistema de evaporação estudado no Exemplo 3.11. Utilize um GPC MIMO para controlar esse processo e compare os resultados com os obtidos usando o DMC MIMO. Estude os mesmos dois casos do exemplo, com atraso menor e com atraso maior. Analise a facilidade/-dificuldade de sintonia neste caso particular com dinâmica integradora. Para completar o estudo compare os dois controles com um terceiro caso com atrasos sendo o dobro dos usados no caso de maior atraso do exemplo.
8. Pensar em mais um ex para GPC





## Capítulo 4

# MPC em Variáveis de Estado

A representação por espaço de estados é um modelo no domínio do tempo que descreve diretamente as equações diferenciais (em tempo contínuo), ou as equações a diferenças (em tempo discreto), de um sistema usando uma equação diferencial (ou a diferenças) vetorial de primeira ordem. É um modelo naturalmente multivariável e, por esse motivo, neste livro, apresentamos o caso monovariável como um caso particular do multivariável. Neste capítulo, apenas consideramos modelos lineares discretos, determinísticos e invariantes no tempo, assim, a equação que modela um processo de  $m$  entradas de controle,  $m_q$  perturbações,  $n$  saídas controladas e  $n_s$  estados, na forma denominada variáveis de estado é:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_q\mathbf{q}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{C}_q\mathbf{q}(k) \end{cases} \quad (4.1a)$$

$$(4.1b)$$

onde  $\mathbf{x} \in \mathbb{R}^{n_s}$  representa o vetor de variáveis de estado,  $\mathbf{u} \in \mathbb{R}^m$  é o vetor de sinais de controle,  $\mathbf{y} \in \mathbb{R}^n$  é o vetor de saídas,  $\mathbf{A} \in \mathbb{R}^{n_s \times n_s}$  é a matriz de estados,  $\mathbf{B} \in \mathbb{R}^{n_s \times m}$  é a matriz de entrada,  $\mathbf{B}_q \in \mathbb{R}^{n_s \times m_q}$  é a matriz de perturbação,  $\mathbf{C} \in \mathbb{R}^{n \times n_s}$  é a matriz de saída e  $\mathbf{C}_q \in \mathbb{R}^{n \times m_q}$  é a matriz de saída da perturbação. Este modelo, assim como no caso do DMC e GPC, considera que não há resposta instantânea no sinal de controle. Desta forma, a saída do sistema em  $k$  somente depende dos controles até  $k-1$ . Observe que se a perturbação tem efeito instantâneo na saída a matriz  $\mathbf{C}_q$  será não nula.

Veja que a estrutura do modelo no espaço de estados não sofre modificações independente do número de saídas ou entradas consideradas, o que é uma vantagem em relação a outros tipos de modelos. Apenas as dimensões dos vetores e matrizes se alteram, sendo que no caso monovariável  $n = m = 1$ , e a dimensão do estado  $n_s$  é geralmente menor que no caso multivariável.

Os modelos em variáveis de estado podem ser obtidos a partir das equações diferenciais que representam fenomenologicamente o processo ou a partir de modelos matriz função de transferência ou de resposta ao degrau. Ainda, caso os estados sejam medidos, o modelo pode ser obtido usando métodos de identificação de parâmetros. Esta situação, onde os estados são todos associados a variáveis físicas do processo, não é a mais comum na prática. De fato, na maioria das aplicações, o modelo é construído com muitos estados não tendo qualquer sentido físico. Não é objetivo deste livro estudar com detalhes este tipo de modelos, mas considerar a sua utilização no contexto de MPC.

**Exemplo 4.1.** *Considere o sistema de controle de nível de dois tanques acoplados. O tanque 1 recebe água de uma bomba acionada pelo sinal de controle  $u_1$ . A saída do tanque 1 flui livremente para o tanque 2, o qual recebe água também de uma segunda bomba, com acionamento  $u_2$ . Finalmente, uma válvula é usada para controlar a saída de água do tanque 2. Esta válvula é controlada pelo processo a jusante e, portanto, causa perturbações no nível dos tanques. Considerando que a vazão das bombas 1 e 2 é proporcional ao seu sinal de controle  $u_1$  e  $u_2$ , e que o comportamento do nível na região de operação pode ser considerado linear, podemos escrever um modelo simples:*

$$\begin{cases} h_1(k+1) = h_1(k) + \alpha_1 u_1(k) - \alpha_2 h_1(k) \\ h_2(k+1) = h_2(k) + \alpha_2 h_1(k) + \alpha_3 u_2(k) - \alpha_4 a(k) \end{cases}$$

com  $h_1$  e  $h_2$  os níveis dos tanques,  $a$  a abertura da válvula de saída, e  $\alpha_i$  coeficientes que dependem das características das bombas, do período de amostragem e da gravidade. Este modelo pode ser escrito de forma compacta, considerando os níveis  $h_1$  e  $h_2$  como estados do sistema, os sinais de acionamento  $u_1$  e  $u_2$  como entradas de controle e a abertura  $a$  como perturbação. O modelo resultante neste caso é:

$$\begin{cases} \begin{bmatrix} h_1(k+1) \\ h_2(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} + \mathbf{B} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + \mathbf{B}_q a(k) \\ \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \mathbf{C} \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} \end{cases}$$

com o estado definido como:  $\mathbf{x} = [h_1 \ h_2]^T$  e

$$\mathbf{A} = \begin{bmatrix} 1 - \alpha_2 & 0 \\ \alpha_2 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_3 \end{bmatrix}, \quad \mathbf{B}_q = \begin{bmatrix} 0 \\ -\alpha_4 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Neste exemplo, que é um sistema com dois estados, duas saídas e duas entradas, os níveis podem ser medidos.

**Exemplo 4.2.** Considere uma turbina a vapor da indústria que teve seu modelo determinado com experimentos de identificação, e foi obtida uma função de transferência relacionando as variações da abertura da válvula de vapor  $u(k)$  e as variações de velocidade  $y(k)$  nas vizinhanças de um ponto de operação:

$$Y(z) = -0,2 \frac{(z - 1,1)}{(z - 0,9)^2} U(z),$$

mostrando um comportamento de resposta inversa ao degrau. A equação a diferenças deste sistema pode ser escrita como:

$$y(k + 2) + 1,8y(k + 1) - 0,81y(k) = -0,2u(k + 1) + 0,22u(k),$$

e, neste caso, trata-se de um sistema monovariável. Uma possível representação deste processo em variáveis de estado é:

$$\begin{cases} \begin{bmatrix} x_1(k + 1) \\ x_2(k + 1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \mathbf{B} [u(k)] \\ y(k) = \mathbf{C} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \end{cases}$$

com:  $\mathbf{x} = [x_1 \ x_2]^T$  e

$$\mathbf{A} = \begin{bmatrix} 1,8 & -0,81 \\ 1 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{C} = [-0,2 \ 0,22].$$

Neste sistema, o modelo em espaço de estados é obtido utilizando a forma canônica de controlabilidade, um método para se obter a representação em espaço de estados a partir da função de transferência<sup>1</sup>. No entanto, os estados perdem sentido físico e, por consequência, não são mensuráveis.

Apesar de sua simplicidade de representação, a formulação do MPC com modelo por espaço de estados requer que os estados sejam medidos ou, no mínimo, estimados. Como comentado, na prática, raramente é possível medir todos os estados do sistema devido ao alto custo associado aos sensores, ou devido à impossibilidade prática de se medir diretamente certas variáveis. Por exemplo, em colunas de destilação, as concentrações dos diferentes produtos extraídos da coluna são os estados do sistema mas, na prática, devido ao alto custo dos equipamentos, estes estados não são medidos. Em um sistema de aquecimento de água com energia solar, a temperatura da tubulação metálica

---

<sup>1</sup>Não existe uma única representação de estados para uma dada função de transferência. É possível mostrar que há uma infinidade de modelos em espaço de estados diferentes que resultam na mesma função de transferência [Lat06, FPEN13].

e da água são estados do sistema mas, normalmente, o equipamento não tem medidor de temperatura na parede metálica, apenas medidor para a temperatura da água.

Para solucionar esse problema, é comum o uso de observadores de estados, estruturas que estimam os estados do sistema de tal forma que seja possível a implementação de sistemas de monitoramento e controle. No entanto, esses observadores adicionam dinâmicas na realimentação, de forma similar à adição de um filtro na saída, dificultando a sintonia do controlador e afetando sua robustez e capacidade de rejeitar perturbações.

A literatura de observadores de estados é bastante extensa e não será tratada nesse livro, sugerimos a consulta de [HJS08, Che13]. Dessa forma, nesse livro assumiremos que as medições dos estados estão disponíveis de alguma maneira, seja por medição direta, seja através de observadores. A formulação do MPC em espaço de estados (*State-Space MPC* - SSMPC) será apresentada a seguir.

## 4.1 Formulação do SSMPC

Na literatura de MPC, SSMPC na verdade faz referência a todos os algoritmos de MPC que utilizam um modelo em espaço de estados, não indicando precisamente um algoritmo específico. Por exemplo, diferentes SSMPC podem usar tipos de função custo ou de modelos de perturbações distintos, assim, o algoritmo de controle e, por consequência, as características do sistema em malha fechada são diferentes. Desta forma, ao leitor é sugerido sempre analisar com cuidado o algoritmo SSMPC sendo estudado nas diferentes fontes da literatura. Neste capítulo, utilizaremos o mesmo tipo de função custo do DMC e GPC. Posteriormente, estudaremos o uso de diferentes funções custo, de forma genérica para todos os algoritmos MPC multivariáveis.

Assim como no caso do GPC, podemos alterar o modelo em espaço de estados (4.1) para considerar o efeito de perturbações não medidas. Inicialmente, não consideramos a perturbação  $\mathbf{q}(k)$  medida, assim ela não é considerada nas predições.

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \frac{1}{\Delta}\mathbf{e}(k+1) & (4.5a) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) & (4.5b) \end{cases}$$

onde  $\mathbf{e}(k) \in \mathbb{R}^{n_s}$  é um ruído branco de média zero que representa perturbações externas ou erros de modelagem em cada estado. Veja que esse modelo é muito similar ao utilizado no algoritmo GPC, apresentado na Seção 3.3.

A partir de (4.5), as predições futuras do sistema em  $k + j$  podem ser escritas como

$$\begin{cases} \mathbf{x}(k + j + 1) = \mathbf{A}\mathbf{x}(k + j) + \mathbf{B}\mathbf{u}(k + j) + \frac{1}{\Delta}\mathbf{e}(k + j + 1) & (4.6a) \\ \mathbf{y}(k + j) = \mathbf{C}\mathbf{x}(k + j) & (4.6b) \end{cases}$$

Rearranjando (4.6a),

$$\Delta\mathbf{x}(k + j + 1) = \mathbf{A}\Delta\mathbf{x}(k + j) + \mathbf{B}\Delta\mathbf{u}(k + j) + \mathbf{e}(k + j + 1), \quad (4.7)$$

e como o valor esperado de  $\mathbf{e}(k + j)$  é zero para  $j > 0$ , então,

$$\begin{aligned} \hat{\mathbf{x}}(k + j + 1 | k) - \hat{\mathbf{x}}(k + j | k) &= \mathbf{A} (\hat{\mathbf{x}}(k + j | k) - \hat{\mathbf{x}}(k + j - 1 | k)) + \mathbf{B}\Delta\mathbf{u}(k + j) \\ \hat{\mathbf{x}}(k + j + 1 | k) &= (\mathbf{I} + \mathbf{A}) \hat{\mathbf{x}}(k + j | k) - \mathbf{A}\hat{\mathbf{x}}(k + j - 1 | k) + \mathbf{B}\Delta\mathbf{u}(k + j). \end{aligned}$$

A partir desta última equação, para a predição em  $k + 1$ , temos

$$\hat{\mathbf{x}}(k + 1 | k) = (\mathbf{I} + \mathbf{A}) \hat{\mathbf{x}}(k | k) - \mathbf{A}\hat{\mathbf{x}}(k - 1 | k) + \mathbf{B}\Delta\mathbf{u}(k), \quad (4.8)$$

sendo que  $\hat{\mathbf{x}}(k | k) = \mathbf{x}(k)$  e  $\hat{\mathbf{x}}(k - 1 | k) = \mathbf{x}(k - 1)$ , pois são conhecidos. Para  $k + 2$ ,

$$\hat{\mathbf{x}}(k + 2 | k) = (\mathbf{I} + \mathbf{A}) \hat{\mathbf{x}}(k + 1 | k) - \mathbf{A}\hat{\mathbf{x}}(k | k) + \mathbf{B}\Delta\mathbf{u}(k + 1), \quad (4.9)$$

e, substituindo (4.8),

$$\begin{aligned} \hat{\mathbf{x}}(k + 2 | k) &= (\mathbf{I} + \mathbf{A}) [(\mathbf{I} + \mathbf{A}) \mathbf{x}(k) - \mathbf{A}\mathbf{x}(k - 1) + \mathbf{B}\Delta\mathbf{u}(k)] \\ &\quad - \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k + 1) \\ \hat{\mathbf{x}}(k + 2 | k) &= [(\mathbf{I} + \mathbf{A})^2 - \mathbf{A}] \mathbf{x}(k) - (\mathbf{I} + \mathbf{A}) \mathbf{A}\mathbf{x}(k - 1) \\ &\quad + (\mathbf{I} + \mathbf{A}) \mathbf{B}\Delta\mathbf{u}(k) + \mathbf{B}\Delta\mathbf{u}(k + 1) \\ \hat{\mathbf{x}}(k + 2 | k) &= [\mathbf{I} + \mathbf{A} + \mathbf{A}^2] \mathbf{x}(k) - (\mathbf{A} + \mathbf{A}^2) \mathbf{x}(k - 1) \\ &\quad + (\mathbf{I} + \mathbf{A}) \mathbf{B}\Delta\mathbf{u}(k) + \mathbf{B}\Delta\mathbf{u}(k + 1). \end{aligned}$$

Repetindo esses passos, é possível demonstrar que

$$\hat{\mathbf{x}}(k + j | k) = \sum_{i=1}^j \left( \sum_{l=0}^{i-1} \mathbf{A}^l \right) \mathbf{B}\Delta\mathbf{u}(k + j - i) + \sum_{i=0}^j \mathbf{A}^i \mathbf{x}(k) - \sum_{i=1}^j \mathbf{A}^i \mathbf{x}(k - 1), \quad (4.10)$$

e note que o primeiro termo só depende de valores futuros da entrada e os dois últimos dependem da valores passados dos estados, assim, eles definem, respectivamente, a resposta forçada e a livre do sistema.

Para encontrar as predições da saída basta multiplicar as predições por  $\mathbf{C}$ . Então, a partir de (4.6),

$$\begin{aligned}\hat{\mathbf{y}}(k+j | k) &= \mathbf{C}\hat{\mathbf{x}}(k+j | k) \\ &= \mathbf{C} \sum_{i=1}^j \left( \sum_{l=0}^{i-1} \mathbf{A}^l \right) \mathbf{B} \Delta \mathbf{u}(k+j-i) \\ &\quad + \mathbf{C} \sum_{i=0}^j \mathbf{A}^i \mathbf{x}(k) - \mathbf{C} \sum_{i=1}^j \mathbf{A}^i \mathbf{x}(k-1).\end{aligned}\quad (4.11)$$

Com isso, é possível reescrever as predições futuras do sistema no padrão formato já apresentado:

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f} \quad (4.12)$$

com  $\mathbf{f} = \mathbf{F} \underline{\mathbf{x}}$  e

$$\begin{aligned}\hat{\mathbf{y}} &= \begin{bmatrix} \hat{\mathbf{y}}(k+1 | k) \\ \hat{\mathbf{y}}(k+2 | k) \\ \vdots \\ \hat{\mathbf{y}}(k+N_2 | k) \end{bmatrix}, \quad \hat{\mathbf{y}}(k+j | k) = \begin{bmatrix} \hat{y}_1(k+j | k) \\ \hat{y}_2(k+j | k) \\ \vdots \\ \hat{y}_n(k+j | k) \end{bmatrix} \\ \Delta \mathbf{u} &= \begin{bmatrix} \Delta \mathbf{u}(k) \\ \Delta \mathbf{u}(k+1) \\ \vdots \\ \Delta \mathbf{u}(k+N_u-1) \end{bmatrix}, \quad \Delta \mathbf{u}(k+j) = \begin{bmatrix} \Delta u_1(k+j) \\ \Delta u_2(k+j) \\ \vdots \\ \Delta u_m(k+j) \end{bmatrix}, \\ \mathbf{G} &= \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B} & \mathbf{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C} \left( \sum_{l=0}^{N_2-1} \mathbf{A}^l \right) \mathbf{B} & \cdots & \cdots & \mathbf{C} \left( \sum_{l=0}^{N_2-N_u} \mathbf{A}^l \right) \mathbf{B} \end{bmatrix}, \\ \mathbf{F} &= \begin{bmatrix} \mathbf{C}(\mathbf{I} + \mathbf{A}) & -\mathbf{CA} \\ \mathbf{C}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2) & -\mathbf{C}(\mathbf{A} + \mathbf{A}^2) \\ \vdots & \vdots \\ \mathbf{C} \left( \sum_{i=0}^{N_2} \mathbf{A}^i \right) & -\mathbf{C} \left( \sum_{i=1}^{N_2} \mathbf{A}^i \right) \end{bmatrix}, \quad \underline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k-1) \end{bmatrix}.\end{aligned}$$

Caso se utilize um horizonte inicial  $N_1 > 1$ , é necessário remover as  $n(N_1 - 1)$  linhas iniciais de  $\mathbf{G}$  e  $\mathbf{F}$  para se obter as predições corretas. Note, também,

que as definições de  $\hat{\mathbf{y}}$  e  $\Delta \mathbf{u}$  são diferentes das apresentadas na Seção 3.1, e isso impacta na obtenção da lei de controle, como será discutido a seguir num exemplo.

**Exemplo 4.3.** Considere novamente o exemplo do sistema de dois tanques apresentado no Exemplo 4.1, e suponha  $\alpha_1 = 0,1$ ,  $\alpha_2 = 0,4$ ,  $\alpha_3 = \alpha_4 = 0,5$ . Assim, o sistema é representado da seguinte forma:

$$\begin{cases} \begin{bmatrix} h_1(k+1) \\ h_2(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} + \mathbf{B} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + \mathbf{B}_q a(k) \\ \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \mathbf{C} \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} \end{cases}$$

com o estado definido como:  $\mathbf{x} = [h_1 \ h_2]^T$  e

$$\mathbf{A} = \begin{bmatrix} 0,6 & 0 \\ 0,4 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0,1 & 0 \\ 0 & 0,5 \end{bmatrix}, \quad \mathbf{B}_q = \begin{bmatrix} 0 \\ -0,5 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Neste exemplo, temos dois estados, duas saídas e duas entradas, e uma perturbação. Consideraremos que apenas os níveis podem ser medidos. Em termos de sintonia do controlador, os horizontes de predição são  $N_1 = 1$ ,  $N_2 = 3$  para ambas as saídas, e  $N_u = 2$  para ambas as manipuladas. De acordo com o exposto anteriormente, as matrizes para se obter as predições são:

$$\mathbf{G} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} \\ \mathbf{C(I+A)B} & \mathbf{CB} \\ \mathbf{C(I+A+A^2)B} & \mathbf{C(I+A)B} \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{C(I+A)} & -\mathbf{CA} \\ \mathbf{C(I+A+A^2)} & -\mathbf{C(A+A^2)} \\ \mathbf{C(I+A+A^2+A^3)} & -\mathbf{C(A+A^2+A^3)} \end{bmatrix},$$

e fazendo as substituições adequadas, chegamos nas seguintes matrizes para os cálculos das predições:

$$\mathbf{G} = \begin{bmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 \\ 0,16 & 0 & 0,1 & 0 \\ 0,04 & 1 & 0 & 0,5 \\ 0,196 & 0 & 0,1 & 0 \\ 0,104 & 1,5 & 0 & 0,5 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 1,6 & 0 & -0,6 & 0 \\ 0,4 & 2 & -0,4 & -1 \\ 1,96 & 0 & -0,96 & 0 \\ 1,04 & 3 & -1,04 & -2 \\ 2,176 & 0 & -1,176 & 0 \\ 1,824 & 4 & -1,824 & -3 \end{bmatrix},$$

com os vetores das variáveis definidos da seguinte forma:

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \end{bmatrix}, \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1(k+1|k) \\ \hat{y}_2(k+1|k) \\ \hat{y}_1(k+2|k) \\ \hat{y}_2(k+2|k) \\ \hat{y}_1(k+3|k) \\ \hat{y}_2(k+3|k) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} h_1(k) \\ h_2(k) \\ h_1(k-1) \\ h_2(k-1) \end{bmatrix}.$$

É possível escolher horizontes distintos para cada variável de saída e manipulada no SSMPC. Porém, é preciso cuidado nas definições das matrizes por conta da maneira como o modelo é descrito. Por exemplo, vamos supor agora que  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = 2$ ,  $N_{2,2} = 3$ , e  $N_{u,1} = 2$  e  $N_{u,2} = 1$ , de tal forma que os vetores dos incrementos de controle e predição agora são dados por:

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_1(k+1) \end{bmatrix}, \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1(k+1|k) \\ \hat{y}_2(k+1|k) \\ \hat{y}_1(k+2|k) \\ \hat{y}_2(k+2|k) \\ \hat{y}_2(k+3|k) \end{bmatrix}.$$

Com esta sintonia, as matrizes para o cálculo das predições precisam ser adaptadas:

$$\mathbf{G} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B} & \mathbf{CB}(:,1) \\ \mathbf{C}(2,:)(\mathbf{I} + \mathbf{A} + \mathbf{A}^2)\mathbf{B} & \mathbf{C}(2,:)(\mathbf{I} + \mathbf{A})\mathbf{B}(:,1) \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{C}(\mathbf{I} + \mathbf{A}) & -\mathbf{CA} \\ \mathbf{C}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2) & -\mathbf{C}(\mathbf{A} + \mathbf{A}^2) \\ \mathbf{C}(2,:)(\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3) & -\mathbf{C}(2,:)(\mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3) \end{bmatrix},$$

onde  $\mathbf{X}(:,i)$  indica uma matriz-coluna obtida a partir  $i$ -ésima coluna de  $\mathbf{X}$ , e  $\mathbf{X}(i,:)$  indica uma matriz-linha obtida a partir  $i$ -ésima linha de  $\mathbf{X}$ . A partir do horizonte 2, veja que desejamos apenas as informações das predições de  $y_2$ , assim, utilizamos a matriz  $\mathbf{C}(2,:) = [0, 1]$  para obter apenas as informações referentes a  $y_2$ . De forma similar, para as manipuladas, como desejamos considerar apenas  $\Delta u_1$  após o instante  $k$ , devemos adaptar as matrizes para considerar apenas o efeito de  $\Delta u_1$ , e fazemos isso eliminando as colunas de  $\mathbf{B}$  das variáveis desprezadas, assim,  $\mathbf{B}(:,1)^T = [0, 1; 0]$ . Então, para esta





A partir destas últimas equações, para a predição em  $k + 1$ , temos

$$\begin{cases} \hat{\mathbf{x}}(k+1) = (\mathbf{I} + \mathbf{A}) \hat{\mathbf{x}}(k | k) - \mathbf{A} \hat{\mathbf{x}}(k-1 | k) \\ \quad + \mathbf{B} \Delta \mathbf{u}(k) + \mathbf{B}_q \Delta \mathbf{q}(k) \end{cases} \quad (4.17a)$$

$$\hat{\mathbf{y}}(k+1) = \mathbf{C} \hat{\mathbf{x}}(k+1 | k) + \mathbf{C}_q \mathbf{q}(k+1) \quad (4.17b)$$

sendo que  $\hat{\mathbf{x}}(k | k) = \mathbf{x}(k)$  e  $\hat{\mathbf{x}}(k-1 | k) = \mathbf{x}(k-1)$ , pois são conhecidos. Para  $k+2$ ,

$$\begin{cases} \hat{\mathbf{x}}(k+2) = (\mathbf{I} + \mathbf{A}) \hat{\mathbf{x}}(k+1 | k) - \mathbf{A} \hat{\mathbf{x}}(k | k) \\ \quad + \mathbf{B} \Delta \mathbf{u}(k+1) + \mathbf{B}_q \Delta \mathbf{q}(k+1) \\ \hat{\mathbf{y}}(k+2) = \mathbf{C} \hat{\mathbf{x}}(k+2 | k) + \mathbf{C}_q \mathbf{q}(k+2) \end{cases}$$

e, substituindo (4.17) nesta última equação e rearranjando,

$$\begin{cases} \hat{\mathbf{x}}(k+2) = [\mathbf{I} + \mathbf{A} + \mathbf{A}^2] \mathbf{x}(k) - (\mathbf{A} + \mathbf{A}^2) \mathbf{x}(k-1) \\ \quad + (\mathbf{I} + \mathbf{A}) \mathbf{B} \Delta \mathbf{u}(k) + \mathbf{B} \Delta \mathbf{u}(k+1) \\ \quad + (\mathbf{I} + \mathbf{A}) \mathbf{B}_q \Delta \mathbf{q}(k) + \mathbf{B}_q \Delta \mathbf{q}(k+1) \\ \hat{\mathbf{y}}(k+2) = \mathbf{C} \hat{\mathbf{x}}(k+2 | k) + \mathbf{C}_q \mathbf{q}(k+2) \end{cases}$$

Repetindo esses passos, é possível demonstrar que

$$\begin{cases} \hat{\mathbf{x}}(k+j) = \sum_{i=1}^j \left( \sum_{l=0}^{i-1} \mathbf{A}^l \right) \mathbf{B} \Delta \mathbf{u}(k+j-i) \\ \quad + \sum_{i=1}^j \left( \sum_{l=0}^{i-1} \mathbf{A}^l \right) \mathbf{B}_q \Delta \mathbf{q}(k+j-i) \\ \quad + \sum_{i=0}^j \mathbf{A}^i \mathbf{x}(k) - \sum_{i=1}^j \mathbf{A}^i \mathbf{x}(k-1) \\ \hat{\mathbf{y}}(k+j) = \mathbf{C} \hat{\mathbf{x}}(k+j | k) + \mathbf{C}_q \mathbf{q}(k+j) \end{cases}$$

Como estamos interessados apenas nas predições das saídas do sistema, podemos combinar estas duas equações em uma só. Antes disso, considere a igualdade abaixo

$$\mathbf{q}(k+j) = \mathbf{q}(k) + \sum_{i=1}^j \mathbf{I}_{m_q} \Delta \mathbf{q}(k+i), \quad \forall j > 0,$$

assim, as predições futuras das saídas são dadas por

$$\begin{aligned}
\hat{\mathbf{y}}(k+j | k) = & \mathbf{C} \sum_{i=1}^j \left( \sum_{l=0}^{i-1} \mathbf{A}^l \right) \mathbf{B} \Delta \mathbf{u}(k+j-i) \\
& + \mathbf{C} \sum_{i=1}^j \left( \sum_{l=0}^{i-1} \mathbf{A}^l \right) \mathbf{B}_q \Delta \mathbf{q}(k+j-i) \\
& + \mathbf{C}_q \left( \mathbf{q}(k) + \sum_{i=1}^j \mathbf{I}_{m_q} \Delta \mathbf{q}(k+i) \right) \\
& + \mathbf{C} \sum_{i=0}^j \mathbf{A}^i \mathbf{x}(k) - \mathbf{C} \sum_{i=1}^j \mathbf{A}^i \mathbf{x}(k-1). \tag{4.21}
\end{aligned}$$

e note que as mesmas considerações no caso sem ação antecipativa ainda são válidas. O primeiro termo só depende de valores futuros da entrada definindo, assim, a resposta forçada. Os outros termos, que dependem dos estados passados e das perturbações futuras e passadas, definem a resposta livre. Os termos das perturbações ainda podem ser separados em futuros e passados para o caso em que os valores futuros são desconhecidos, assim, simplificando a expressão.

Com isso, é possível reescrever as predições futuras do sistema no formato padrão já apresentado,

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f} \tag{4.22}$$

com  $\mathbf{f} = \mathbf{F} \underline{\mathbf{x}} + \mathbf{G}_q \Delta \mathbf{q} + \mathbf{F}_q \underline{\mathbf{q}}$  e

$$\begin{aligned}
\hat{\mathbf{y}} &= \begin{bmatrix} \hat{\mathbf{y}}(k+1 | k) \\ \hat{\mathbf{y}}(k+2 | k) \\ \vdots \\ \hat{\mathbf{y}}(k+N_2 | k) \end{bmatrix}, \quad \hat{\mathbf{y}}(k+j | k) = \begin{bmatrix} \hat{y}_1(k+j | k) \\ \hat{y}_2(k+j | k) \\ \vdots \\ \hat{y}_n(k+j | k) \end{bmatrix} \\
\Delta \mathbf{u} &= \begin{bmatrix} \Delta \mathbf{u}(k) \\ \Delta \mathbf{u}(k+1) \\ \vdots \\ \Delta \mathbf{u}(k+N_u-1) \end{bmatrix}, \quad \Delta \mathbf{u}(k+j) = \begin{bmatrix} \Delta u_1(k+j) \\ \Delta u_2(k+j) \\ \vdots \\ \Delta u_m(k+j) \end{bmatrix}, \\
\Delta \mathbf{q} &= \begin{bmatrix} \Delta \mathbf{q}(k+1) \\ \Delta \mathbf{q}(k+2) \\ \vdots \\ \Delta \mathbf{q}(k+N_q) \end{bmatrix}, \quad \Delta \mathbf{q}(k+j) = \begin{bmatrix} \Delta q_1(k+j) \\ \Delta q_2(k+j) \\ \vdots \\ \Delta q_{m_q}(k+j) \end{bmatrix},
\end{aligned}$$

$$\begin{aligned}
\mathbf{G} &= \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B} & \mathbf{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C} \left( \sum_{l=0}^{N_2-1} \mathbf{A}^l \right) \mathbf{B} & \cdots & \cdots & \mathbf{C} \left( \sum_{l=0}^{N_2-N_u} \mathbf{A}^l \right) \mathbf{B} \end{bmatrix}, \\
\mathbf{F} &= \begin{bmatrix} \mathbf{C}(\mathbf{I} + \mathbf{A}) & -\mathbf{CA} \\ \mathbf{C}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2) & -\mathbf{C}(\mathbf{A} + \mathbf{A}^2) \\ \vdots & \vdots \\ \mathbf{C} \left( \sum_{i=0}^{N_2} \mathbf{A}^i \right) & -\mathbf{C} \left( \sum_{i=1}^{N_2} \mathbf{A}^i \right) \end{bmatrix}, \quad \underline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k-1) \end{bmatrix}, \\
\mathbf{G}_q &= \begin{bmatrix} \mathbf{C}_q & 0 & \cdots & 0 \\ \mathbf{CB}_q + \mathbf{C}_q & \mathbf{C}_q & \cdots & 0 \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B}_q & \mathbf{CB}_q + \mathbf{C}_q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C} \left( \sum_{l=0}^{N_2-2} \mathbf{A}^l \right) \mathbf{B}_q + \mathbf{C}_q & \cdots & \cdots & \mathbf{C} \left( \sum_{l=0}^{N_2-N_q-1} \mathbf{A}^l \right) \mathbf{B}_q + \mathbf{C}_q \end{bmatrix}, \\
\mathbf{F}_q &= \begin{bmatrix} \mathbf{CB}_q + \mathbf{C}_q & -\mathbf{CB}_q \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B}_q + \mathbf{C}_q & -\mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B}_q \\ \vdots & \vdots \\ \mathbf{C} \left( \sum_{i=0}^{N_2-1} \mathbf{A}^i \right) \mathbf{B}_q + \mathbf{C}_q & -\mathbf{C} \left( \sum_{i=0}^{N_2-1} \mathbf{A}^i \right) \mathbf{B}_q \end{bmatrix}, \quad \underline{\mathbf{q}} = \begin{bmatrix} \mathbf{q}(k) \\ \mathbf{q}(k-1) \end{bmatrix},
\end{aligned}$$

sendo  $N_q$  o horizonte de valores futuros da perturbação  $\mathbf{q}$ . Caso os horizontes para cada uma das variáveis for definido com valores diferentes, é preciso adaptar as matrizes anteriores, de forma análoga ao explicado no Exemplo 4.3.

**Exemplo 4.4.** *Vamos retomar o sistema de dois tanques apresentado no Exemplo 4.3 mas, agora, suponha que as perturbações são medidas e que  $N_q = 3$ . Os outros horizontes são  $N_1 = 1$ ,  $N_2 = 3$  para ambas as saídas, e  $N_u = 2$  para ambas as manipuladas. As matrizes  $\mathbf{G}$  e  $\mathbf{F}$  já foram calculadas no Exemplo 4.3 restando, então, apenas o cálculo de  $\mathbf{F}_q$  e  $\mathbf{G}_q$*

$$\begin{aligned}
\mathbf{G}_q &= \begin{bmatrix} \mathbf{C}_q & 0 & 0 \\ \mathbf{CB}_q + \mathbf{C}_q & \mathbf{C}_q & 0 \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B}_q + \mathbf{C}_q & \mathbf{CB}_q + \mathbf{C}_q & \mathbf{C}_q \end{bmatrix}, \\
\mathbf{F}_q &= \begin{bmatrix} \mathbf{CB}_q + \mathbf{C}_q & -\mathbf{CB}_q \\ \mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B}_q + \mathbf{C}_q & -\mathbf{C}(\mathbf{I} + \mathbf{A})\mathbf{B}_q \\ \mathbf{C}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2)\mathbf{B}_q + \mathbf{C}_q & -\mathbf{C}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2)\mathbf{B}_q \end{bmatrix},
\end{aligned}$$

dado que

$$\Delta \mathbf{q} = \begin{bmatrix} \Delta a(k+1) \\ \Delta a(k+2) \\ \Delta a(k+3) \end{bmatrix}, \quad \underline{\mathbf{q}} = \begin{bmatrix} \Delta a(k) \\ \Delta a(k-1) \end{bmatrix},$$

e fazendo as substituições adequadas, chegamos nas seguintes matrizes para os cálculos das predições com  $\mathbf{C}_q = 0$ :

$$\mathbf{G}_q = \begin{bmatrix} 0 & 0 & 0 \\ -0,5 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & 0 \end{bmatrix}, \quad \mathbf{F}_q = \begin{bmatrix} 0 & 0 \\ -0,5 & 0,5 \\ 0 & 0 \\ -1 & 1 \\ 0 & 0 \\ -1,5 & 1,5 \end{bmatrix}.$$

Caso as perturbações futuras forem desconhecidas, pode-se desprezar a matriz  $\mathbf{G}_q$  e, caso o horizonte  $N_q$  seja diferente, basta remover as colunas de  $\mathbf{G}_q$  adequadamente, assim como mostrado no Exemplo 4.3.

### 4.3 Obtenção da lei de controle

A obtenção da lei de controle é feita de forma similar à apresentada na Seção 3.1. Porém, devido a algumas particularidades da representação utilizada para o SSMPC, vale a pena rever este ponto.

A função custo é a mesma apresentada na Seção 3.1, que é repetida aqui por facilidade:

$$J = \sum_{l=1}^n \sum_{i=N_{1,l}}^{N_{2,l}} \delta_l(i) (r_l(k+i) - \hat{y}_l(k+i | k))^2 + \sum_{p=1}^m \sum_{i=0}^{N_{u,p}-1} \lambda_p(i) \Delta u_p(k+i)^2. \quad (4.23)$$

Com o MPC em espaço de estados, é bastante usual a definição de apenas um único horizonte de predição inicial e final, e horizonte de controle para todas as saídas e entradas, respectivamente. Ou seja,  $N_{1,l} = N_1$ ,  $N_{2,l} = N_2$ , e  $N_{u,p} = N_u$ ,  $\forall l, p$ . Horizontes diferentes podem ser utilizados, mas isso complica as matrizes de predição definidas como mostrado no Exemplo 4.3.

A função custo (3.2) pode ser reordenada para evidenciar a formulação

apresentada do SSMPC:

$$J = \sum_{i=N_1}^{N_2} [\mathbf{r}(k+i) - \hat{\mathbf{y}}(k+i | k)]^T \tilde{\mathbf{Q}}_y(i) [\mathbf{r}(k+i) - \hat{\mathbf{y}}(k+i | k)] + \sum_{i=0}^{N_u-1} \Delta \mathbf{u}(k+i)^T \tilde{\mathbf{Q}}_u(i) \Delta \mathbf{u}(k+i), \quad (4.24)$$

com as definições apresentadas na seção anterior e

$$\mathbf{r}(k+i) = \begin{bmatrix} r_1(k+i) \\ r_2(k+i) \\ \vdots \\ r_n(k+i) \end{bmatrix}, \quad \tilde{\mathbf{Q}}_y(i) = \begin{bmatrix} \delta_1(i) & & 0 \\ & \ddots & \\ 0 & & \delta_n(i) \end{bmatrix},$$

$$\tilde{\mathbf{Q}}_u(i) = \begin{bmatrix} \lambda_1(i) & & 0 \\ & \ddots & \\ 0 & & \lambda_m(i) \end{bmatrix},$$

o que permite reescrever a função custo na forma matricial

$$J = [\mathbf{R} - \hat{\mathbf{y}}]^T \mathbf{Q}_y [\mathbf{R} - \hat{\mathbf{y}}] + \Delta \mathbf{u}^T \mathbf{Q}_u \Delta \mathbf{u}, \quad (4.25)$$

com as definições apresentadas na seção anterior e

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}(k+N_1) \\ \mathbf{r}(k+N_1+1) \\ \vdots \\ \mathbf{r}(k+N_2) \end{bmatrix}, \quad \mathbf{Q}_y = \begin{bmatrix} \tilde{\mathbf{Q}}_y(N_1) & & 0 \\ & \ddots & \\ 0 & & \tilde{\mathbf{Q}}_y(N_2) \end{bmatrix},$$

$$\mathbf{Q}_u = \begin{bmatrix} \tilde{\mathbf{Q}}_u(0) & & 0 \\ & \ddots & \\ 0 & & \tilde{\mathbf{Q}}_u(N_u-1) \end{bmatrix}.$$

Por sua vez, essa função custo pode ser reorganizada na forma padrão, como explicado na Seção 3.1 e, após ser minimizada, nos dá o vetor de incrementos futuros a ser aplicado no processo

$$\Delta \mathbf{u} = \mathbf{K} [\mathbf{R} - \mathbf{f}], \quad (4.26)$$

com  $\mathbf{K} = (\mathbf{G}^T \mathbf{Q}_y \mathbf{G} + \mathbf{Q}_u)^{-1} \mathbf{G}^T \mathbf{Q}_y$ . Note que  $\Delta \mathbf{u}$  define um vetor de incrementos futuros de controle, no entanto, apenas  $\Delta u_p(k)$  é aplicado no instante

$k$ , pois  $\Delta \mathbf{u}$  deve ser recalculado a cada instante de amostragem, seguindo o horizonte deslizante. Assim, é necessário obter apenas o valor de  $\Delta u_p(k)$  ótimo, dado por:

$$\Delta \mathbf{u}(k) = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \vdots \\ \Delta u_p(k) \end{bmatrix} = \mathbf{K}_1 [\mathbf{R} - \mathbf{f}] \quad (4.27)$$

onde  $\mathbf{K}_1$  define uma matriz com as  $m$  primeiras linhas de  $\mathbf{K}$ , que é uma definição diferente da dos algoritmos GPC e DMC MIMO por causa das definições distintas dos vetores de controle futuros. A ação de controle a ser aplicada no processo é então

$$u_p(k) = u_p(k-1) + \Delta u_p(k), \forall p. \quad (4.28)$$

**Exemplo 4.5.** A partir das matrizes obtidas no Exemplo 4.3, considerando a primeira sintonia,  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = N_{2,2} = 3$ ,  $N_{u,1} = N_{u,2} = 2$ , e definindo ponderações fixas ao longo do horizonte tanto para as manipuladas quanto para as controladas e  $\delta_1 = 0,8$  e  $\delta_2 = 1$ , e  $\lambda_1 = 1$  e  $\lambda_2 = 0,6$ , assim:

$$\tilde{\mathbf{Q}}_y(i) = \begin{bmatrix} 0,8 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tilde{\mathbf{Q}}_u(i) = \begin{bmatrix} 1 & 0 \\ 0 & 0,6 \end{bmatrix}, \quad \forall i.$$

Então, a partir de (4.26), a matriz de ganhos do controlador é:

$$\mathbf{K}^T = \begin{bmatrix} 0,075383 & -0,0032122 & -0,0021131 & -0,0012839 \\ -0,020076 & 0,18745 & 0,00056277 & -0,2117 \\ 0,1185 & -0,0050495 & 0,075418 & -0,0020182 \\ -0,010486 & 0,1616 & 0,00029393 & 0,2716 \\ 0,14564 & -0,0062059 & 0,074658 & -0,0024804 \\ 0,029744 & 0,34648 & -0,00083376 & 0,058866 \end{bmatrix}^T,$$

e  $\mathbf{K}_1$  seria obtido a partir das duas primeiras linhas da matriz anterior:

$$\mathbf{K}_1^T = \begin{bmatrix} 0,075383 & -0,0032122 \\ -0,020076 & 0,18745 \\ 0,1185 & -0,0050495 \\ -0,010486 & 0,1616 \\ 0,14564 & -0,0062059 \\ 0,029744 & 0,34648 \end{bmatrix}^T$$

Já para a segunda sintonia, que considerava os horizontes  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = 2$ ,  $N_{2,2} = 3$ , e  $N_{u,1} = 2$  e  $N_{u,2} = 1$ , precisamos adaptar as matrizes das ponderações por conta das mudanças de horizontes, assim,

$$\mathbf{Q}_y = \begin{bmatrix} 0,8 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0,8 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0,6 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

e a matriz de ganho do controlador é:

$$\mathbf{K} = \begin{bmatrix} 0,077567 & -0,023176 & 0,12312 & -0,0075675 & 0,031311 \\ -0,0037081 & 0,12306 & -0,0058859 & 0,24426 & 0,36436 \\ -0,00098498 & 0,00029429 & 0,077802 & 9,6096e-05 & -0,0003976 \end{bmatrix},$$

e  $\mathbf{K}_1$  seria obtido, novamente, a partir das duas primeiras linhas da matriz anterior:

$$\mathbf{K}_1 = \begin{bmatrix} 0,077567 & -0,023176 & 0,12312 & -0,0075675 & 0,031311 \\ -0,0037081 & 0,12306 & -0,0058859 & 0,24426 & 0,36436 \end{bmatrix}$$

### 4.3.1 Inclusão de restrições

A adição de restrições no SSMPC segue passos similares aos dos outros algoritmos, tal como o MIMO DMC e GPC. Porém, como a definição do vetor de variáveis de decisão  $\Delta \mathbf{u}$  é feito de forma diferente, a montagem das matrizes de restrições muda ligeiramente, e é necessário uma certa adaptação. De qualquer maneira, é possível reescrever o problema de controle apresentado anteriormente na forma padrão do problema quadrático:

$$J = \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H}_{qp} \Delta \mathbf{u} + \mathbf{f}_{qp}^T \Delta \mathbf{u} + c_{qp}, \quad (4.29)$$

com

$$\begin{aligned} \mathbf{H}_{qp} &= 2 (\mathbf{G}^T \mathbf{Q}_y \mathbf{G} + \mathbf{Q}_u) \\ \mathbf{f}_{qp} &= -2 \mathbf{G}^T \mathbf{Q}_y [\mathbf{r} - \mathbf{f}] \\ c_{qp} &= [\mathbf{r} - \mathbf{f}]^T \mathbf{Q}_y [\mathbf{r} - \mathbf{f}]. \end{aligned}$$

sujeito a restrições do tipo

$$\bar{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}.$$

Agora, vamos apresentar como se obter as restrições no formato padrão.



## Restrição no incremento de controle

Neste caso, desejamos que

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u}(k+j) \leq \Delta \mathbf{u}_{\max}, \quad \forall j \geq 0$$

com vetores  $\Delta \mathbf{u}_{\min}, \Delta \mathbf{u}_{\max} \in \mathbb{R}^m$ . Então, considerando um horizonte de predição  $N_u$  igual para todas as manipuladas,

$$\begin{bmatrix} \mathbf{I}_m & 0 & \cdots & 0 \\ 0 & \mathbf{I}_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{I}_m \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \Delta \mathbf{u}_{\max} \\ \Delta \mathbf{u}_{\max} \\ \vdots \\ \Delta \mathbf{u}_{\max} \end{bmatrix}, \quad \Delta \mathbf{u}_{\max} = \begin{bmatrix} \Delta u_{1, \max} \\ \Delta u_{2, \max} \\ \vdots \\ \Delta u_{m, \max} \end{bmatrix}$$

veja que, em comparação com o MIMO DMC (ou GPC), a estrutura destas matrizes é diferente, por conta da definição distinta de  $\Delta \mathbf{u}$ . De forma similar, para o limite inferior

$$\begin{bmatrix} -\mathbf{I}_m & 0 & \cdots & 0 \\ 0 & -\mathbf{I}_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\mathbf{I}_m \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} -\Delta \mathbf{u}_{\min} \\ -\Delta \mathbf{u}_{\min} \\ \vdots \\ -\Delta \mathbf{u}_{\min} \end{bmatrix}, \quad \Delta \mathbf{u}_{\min} = \begin{bmatrix} \Delta u_{1, \min} \\ \Delta u_{2, \min} \\ \vdots \\ \Delta u_{m, \min} \end{bmatrix}$$

ou, de forma compacta,

$$\begin{bmatrix} \mathbf{I}_{mN_u} \\ -\mathbf{I}_{mN_u} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \bar{\mathbf{r}}_{\Delta u, \max} \\ \bar{\mathbf{r}}_{\Delta u, \min} \end{bmatrix},$$

com

$$\bar{\mathbf{r}}_{\Delta u, \max} = \begin{bmatrix} \Delta \mathbf{u}_{\max} \\ \Delta \mathbf{u}_{\max} \\ \vdots \\ \Delta \mathbf{u}_{\max} \end{bmatrix}, \quad \bar{\mathbf{r}}_{\Delta u, \min} = \begin{bmatrix} -\Delta \mathbf{u}_{\min} \\ -\Delta \mathbf{u}_{\min} \\ \vdots \\ -\Delta \mathbf{u}_{\min} \end{bmatrix},$$

cabe salientar que nos vetores  $\bar{\mathbf{r}}_{\Delta u, \min}$  e  $\bar{\mathbf{r}}_{\Delta u, \max}$ , os vetores  $\Delta \mathbf{u}_{\min}$  e  $\Delta \mathbf{u}_{\max}$  são repetidos  $N_u$  vezes.

## Restrição no sinal de controle

Neste caso, desejamos que

$$\mathbf{u}_{\min} \leq \mathbf{u}(k+j) \leq \mathbf{u}_{\max}, \quad \forall j \geq 0$$

com vetores  $\mathbf{u}_{\min}, \mathbf{u}_{\max} \in \mathbb{R}^m$ . Inicialmente, é preciso obter essas restrições em função do vetor de variáveis de decisão  $\Delta \mathbf{u}$ . Para isso, lembre-se que os sinais de controle futuros podem ser escritos da seguinte forma:

$$\mathbf{u}(k+j) = \mathbf{u}(k-1) + \sum_{i=0}^j \mathbf{I}_m \Delta \mathbf{u}(k+i), \quad \forall j \geq 0,$$

assim,

$$\begin{bmatrix} \mathbf{I}_m & 0 & \cdots & 0 \\ \mathbf{I}_m & \mathbf{I}_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_m & \mathbf{I}_m & \cdots & \mathbf{I}_m \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ \vdots \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \end{bmatrix}, \quad \mathbf{u}_{\max} = \begin{bmatrix} u_{1,\max} \\ u_{2,\max} \\ \vdots \\ u_{m,\max} \end{bmatrix}$$

$$\begin{bmatrix} -\mathbf{I}_m & 0 & \cdots & 0 \\ -\mathbf{I}_m & -\mathbf{I}_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{I}_m & -\mathbf{I}_m & \cdots & -\mathbf{I}_m \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ \vdots \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \end{bmatrix}, \quad \mathbf{u}_{\min} = \begin{bmatrix} u_{1,\min} \\ u_{2,\min} \\ \vdots \\ u_{m,\min} \end{bmatrix}$$

ou, de forma compacta,

$$\begin{bmatrix} \mathbb{T}_{m,N_u} \\ -\mathbb{T}_{m,N_u} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \bar{\mathbf{r}}_{u,\max} \\ \bar{\mathbf{r}}_{u,\min} \end{bmatrix},$$

onde  $\mathbb{T}_{m,N_u}$  pode ser interpretada como uma matriz bloco triangular inferior, onde cada bloco de elementos é uma matriz identidade de dimensão  $m$ , e

$$\bar{\mathbf{r}}_{u,\max} = \begin{bmatrix} \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ \vdots \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \end{bmatrix}, \quad \bar{\mathbf{r}}_{u,\min} = \begin{bmatrix} -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ \vdots \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \end{bmatrix}.$$

Nestes últimos vetores, os termos  $[\mathbf{u}_{\max} - \mathbf{u}(k-1)]$  e  $[-\mathbf{u}_{\min} + \mathbf{u}(k-1)]$  são repetidos  $N_u$  vezes.

### Restrições na saída

Neste caso, desejamos que

$$\mathbf{y}_{\min} \leq \hat{\mathbf{y}}(k+j) \leq \mathbf{y}_{\max}, \quad \forall j \in [N_1, N_2]$$

com vetores  $\mathbf{y}_{\min}, \mathbf{y}_{\max} \in \mathbb{R}^n$  e supondo que os horizontes  $N_1$  e  $N_2$  são iguais para todas as saídas. Inicialmente, é preciso obter essas restrições em função do vetor de variáveis de decisão  $\Delta \mathbf{u}$ . Para isso, lembre-se que

$$\hat{\mathbf{y}} = \mathbf{G}\Delta \mathbf{u} + \mathbf{f}.$$

Assim,

$$\begin{bmatrix} \mathbf{y}_{\min} \\ \mathbf{y}_{\min} \\ \vdots \\ \mathbf{y}_{\min} \end{bmatrix} \leq \mathbf{G}\Delta \mathbf{u} + \mathbf{f} \leq \begin{bmatrix} \mathbf{y}_{\max} \\ \mathbf{y}_{\max} \\ \vdots \\ \mathbf{y}_{\max} \end{bmatrix}$$

$$\mathbf{y}_{\max} = \begin{bmatrix} y_{1,\max} \\ y_{2,\max} \\ \vdots \\ y_{n,\max} \end{bmatrix}, \mathbf{y}_{\min} = \begin{bmatrix} y_{1,\min} \\ y_{2,\min} \\ \vdots \\ y_{n,\min} \end{bmatrix}$$

onde  $\mathbf{y}_{\min}$  e  $\mathbf{y}_{\max}$  são repetidos  $N_y = N_2 - N_1 + 1$  vezes. Então, as restrições são descritas no formato padrão como:

$$\begin{bmatrix} \mathbf{G} \\ -\mathbf{G} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \bar{\mathbf{r}}_{y,\max} \\ \bar{\mathbf{r}}_{y,\min} \end{bmatrix},$$

com

$$\bar{\mathbf{r}}_{y,\max} = \begin{bmatrix} \mathbf{y}_{\max} \\ \mathbf{y}_{\max} \\ \vdots \\ \mathbf{y}_{\max} \end{bmatrix} - \mathbf{f}, \quad \bar{\mathbf{r}}_{y,\min} = \begin{bmatrix} -\mathbf{y}_{\min} \\ -\mathbf{y}_{\min} \\ \vdots \\ -\mathbf{y}_{\min} \end{bmatrix} + \mathbf{f}.$$

**Exemplo 4.6.** Ainda considerando o sistema de dois tanques do Exemplo 4.3, utilizando a primeira sintonia,  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = N_{2,2} = 3$ ,  $N_{u,1} = N_{u,2} = 2$ , vamos obter as matrizes de restrições do problema de otimização supondo restrições nos incremento e no sinal de controle, e na saída. Os limites são definidos como

$$\Delta \mathbf{u}_{\max} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad \Delta \mathbf{u}_{\min} = -\Delta \mathbf{u}_{\max},$$

$$\mathbf{u}_{\max} = \begin{bmatrix} 50 \\ 60 \end{bmatrix}, \quad \mathbf{u}_{\min} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\mathbf{y}_{\max} = \begin{bmatrix} 40 \\ 30 \end{bmatrix}, \quad \mathbf{y}_{\min} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

As restrições, no formato padrão  $\overline{\mathbf{R}}\Delta\mathbf{u} \leq \overline{\mathbf{r}}$ , são dadas por,

$$\begin{bmatrix} \mathbf{I}_4 \\ -\mathbf{I}_4 \\ \overline{\mathbf{T}}_{2,2} \\ -\overline{\mathbf{T}}_{2,2} \\ \mathbf{G} \\ -\mathbf{G} \end{bmatrix} \Delta\mathbf{u} \leq \begin{bmatrix} \Delta\mathbf{u}_{\max} \\ \Delta\mathbf{u}_{\max} \\ \Delta\mathbf{u}_{\min} \\ \Delta\mathbf{u}_{\min} \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ \mathbf{u}_{\max} - \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ -\mathbf{u}_{\min} + \mathbf{u}(k-1) \\ \begin{bmatrix} \mathbf{y}_{\max} \\ \mathbf{y}_{\max} \\ \mathbf{y}_{\max} \end{bmatrix} - \mathbf{f} \\ \begin{bmatrix} -\mathbf{y}_{\min} \\ -\mathbf{y}_{\min} \\ -\mathbf{y}_{\min} \end{bmatrix} + \mathbf{f} \end{bmatrix},$$

ou, numericamente,

$$\Delta \mathbf{u} \leq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 0,1 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 \\ 0,16 & 0 & 0,1 & 0 \\ 0,04 & 1 & 0 & 0,5 \\ 0,196 & 0 & 0,1 & 0 \\ 0,104 & 1,5 & 0 & 0,5 \\ -0,1 & 0 & 0 & 0 \\ 0 & -0,5 & 0 & 0 \\ -0,16 & 0 & -0,1 & 0 \\ -0,04 & -1 & 0 & -0,5 \\ -0,196 & 0 & -0,1 & 0 \\ -0,104 & -1,5 & 0 & -0,5 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \\ -5 \\ -5 \\ -5 \\ -5 \\ 50 - u_1(k-1) \\ 60 - u_2(k-1) \\ 50 - u_1(k-1) \\ 60 - u_2(k-1) \\ 0 + u_1(k-1) \\ 0 + u_2(k-1) \\ 0 + u_1(k-1) \\ 0 + u_2(k-1) \\ 40 - \mathbf{f}(1,1) \\ 30 - \mathbf{f}(2,1) \\ 40 - \mathbf{f}(3,1) \\ 30 - \mathbf{f}(4,1) \\ 40 - \mathbf{f}(5,1) \\ 30 - \mathbf{f}(6,1) \\ 0 + \mathbf{f}(1,1) \\ 0 + \mathbf{f}(2,1) \\ 0 + \mathbf{f}(3,1) \\ 0 + \mathbf{f}(4,1) \\ 0 + \mathbf{f}(5,1) \\ 0 + \mathbf{f}(6,1) \end{bmatrix},$$

*mudar essa notação de matriz para  $f_1(k+1)$ ,  $f_2(k+1)$ , etc? Talvez seja melhor...*

sendo que  $\mathbf{X}(i,j)$  indica o elemento  $i,j$  da matriz  $\mathbf{X}$ .

Se considerarmos a segunda sintonia do Exemplo 4.3, onde os horizontes são escolhidos de forma distinta para cada variável, as matrizes de restrição precisam refletir essa mudança. Então, dado que  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = 2$ ,  $N_{2,2} = 3$ , e  $N_{u,1} = 2$  e  $N_{u,2} = 1$ , as novas matrizes de restrição

seriam

$$\Delta \mathbf{u} \leq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0,1 & 0 & 0 \\ 0 & 0,5 & 0 \\ 0,16 & 0 & 0,1 \\ 0,04 & 1 & 0 \\ 0,104 & 1,5 & 0 \\ -0,1 & 0 & 0 \\ 0 & -0,5 & 0 \\ -0,16 & 0 & -0,1 \\ -0,04 & -1 & 0 \\ -0,104 & -1,5 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \\ 5 \\ -5 \\ -5 \\ -5 \\ 50 - u_1(k-1) \\ 60 - u_2(k-1) \\ 50 - u_1(k-1) \\ 0 + u_1(k-1) \\ 0 + u_2(k-1) \\ 0 + u_1(k-1) \\ 40 - \mathbf{f}(1,1) \\ 30 - \mathbf{f}(2,1) \\ 40 - \mathbf{f}(3,1) \\ 30 - \mathbf{f}(4,1) \\ 30 - \mathbf{f}(5,1) \\ 0 + \mathbf{f}(1,1) \\ 0 + \mathbf{f}(2,1) \\ 0 + \mathbf{f}(3,1) \\ 0 + \mathbf{f}(4,1) \\ 0 + \mathbf{f}(5,1) \end{bmatrix}.$$

## 4.4 Exemplos

**Exemplo 4.7.** Considere o controle de posição de um motor de corrente contínua alimentado em tensão, apresentado inicialmente no Capítulo 4 do Volume 1. Neste sistema, o objetivo é controlar a posição do eixo do motor dentro de um intervalo, atuando na tensão de alimentação do motor. A principal perturbação do sistema é o torque de carga aplicado no eixo do motor que consideraremos representada por sinais do tipo degrau.

Um modelo simplificado em função de transferência discreta desse sistema, com variáveis normalizadas de 0% a 100% é dado por:

$$Y(z) = \frac{0,09675z + 0,09358}{z^2 - 1,905z + 0,905} [4U(z) - Q(z)],$$

sendo  $U$  o sinal de tensão aplicado,  $Y$  a posição do eixo do motor e  $Q$  o torque de carga. Uma representação em espaço de estados do sistema obtida

a partir destas funções de transferência é dada a seguir:

$$\begin{cases} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \mathbf{B}u(k) + \mathbf{B}_q q(k) \\ y(k) = \mathbf{C} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \end{cases}$$

com

$$\mathbf{A} = \begin{bmatrix} 0 & -0,905 \\ 1 & 1,905 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0,74864 \\ 0,774 \end{bmatrix}, \quad \mathbf{B}_q = \begin{bmatrix} -0,18716 \\ -0,1935 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 0,5 \end{bmatrix}.$$

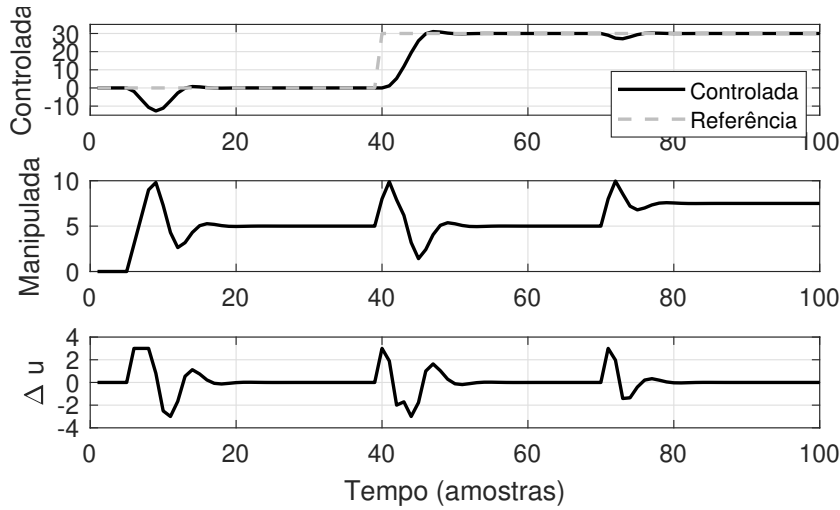


Figura 4.1: Exemplo 4.7 – respostas do motor DC com o SSMPC com  $N_1 = 1$ ,  $N_2 = 30$ ,  $\lambda = 1$  e  $N_u = 5$ , e com restrições.

O cenário de simulação em malha fechada considera uma mudança de referência de tipo degrau de amplitude 30% no tempo 40 amostras, e uma perturbação de amplitude 20% no tempo 5 amostras e 30% no tempo 70 amostras. Um controle SSMPC foi sintonizado com  $N_u = 5$ ,  $N_1 = 1$ ,  $N_2 = 30$ ,  $\delta = 1$ ,  $\lambda = 1$ , e restrições no sinal de controle, que deve estar entre  $[0, 100]\%$ , e no incremento de controle que não pode variar mais do que 3% a cada amostra. Os resultados obtidos são mostrados na Figura 4.1, e são muito similares ao caso GPC apresentado originalmente no Volume 1. O sistema consegue seguir a referência, rejeitar as perturbações e atender as restrições impostas. Na Figura 4.2, são mostrados os resultados de rejeição de perturbação sem e com ação antecipativa, e ainda avaliando sem conhecimento futuro da perturbação e com  $N_q = 3$ . Veja que a adição da ação antecipativa melhora

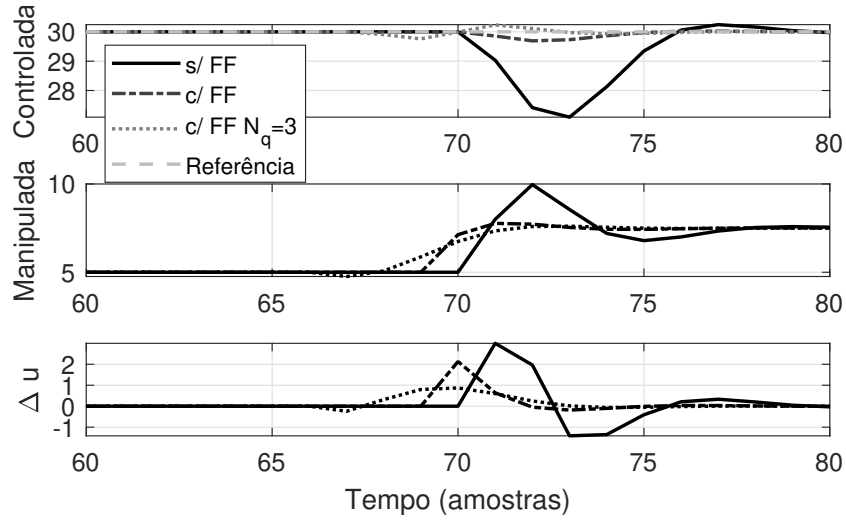


Figura 4.2: Exemplo 4.7 – respostas do motor DC considerando ação antecipativa.

consideravelmente a rejeição de perturbação, especialmente no caso em que seu valor futuro é conhecido. Neste caso, o controlador preditivo se antecipa ao efeito da perturbação e reage de tal forma a diminuir os erros futuros.

**Exemplo 4.8.** Neste exemplo, o SSMPC será aplicado ao processo de uma coluna de destilação água-metanol apresentado inicialmente no caso de estudo da Seção 3.4.1. Como vimos, próximo do ponto de operação o sistema pode ser representado pelas seguintes matrizes de transferência:

$$\mathbf{P}(s) = \begin{bmatrix} \frac{12,8e^{-s}}{1 + 16,7s} & \frac{-18,9e^{-3s}}{1 + 21s} \\ \frac{6,6e^{-7s}}{1 + 10,9s} & \frac{-19,4e^{-3s}}{1 + 14,4s} \end{bmatrix} \quad \mathbf{P}_q(s) = \begin{bmatrix} \frac{3,8e^{-8s}}{1 + 14,9s} \\ \frac{4,9e^{-4s}}{1 + 13,2s} \end{bmatrix}$$

com  $\mathbf{Y}(s) = \mathbf{P}(s)\mathbf{U} + \mathbf{P}_q(s)Q(s)$ ,  $\mathbf{y} = [y_1 \ y_2]^T$  e  $\mathbf{u} = [u_1 \ u_2]^T$ . Neste processo, a vazão de refluxo ( $u_1$ ) e a vazão de vapor no refeedor ( $u_2$ ) são manipuladas para controlar as concentrações de metanol no topo ( $y_1$ ) e na base ( $y_2$ ) da coluna. A principal perturbação do sistema é a vazão de alimentação ( $q$ ). Trata-se de um modelo incremental, assim, podemos considerar que o ponto de equilíbrio inicial é  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$ .

A presença do atraso no sistema dificulta a representação de estados em tempo contínuo, porém, o atraso em tempo discreto nada mais é do que polos na origem. Assim, a mudança de representação de função de transferência



discreta com atraso para espaço de estados é bastante simples, especialmente se considerarmos um modelo não-mínimo (com mais estados do que o estritamente necessário). Para este processo, a matriz de transferência discreta com  $T_s = 1\text{ s}$  é

$$\mathbf{P}(s) = \begin{bmatrix} \frac{0,74397z^{-1}}{z - 0,9419} & \frac{-0,87891z^{-3}}{z - 0,9535} \\ \frac{0,57856z^{-7}}{z - 0,9123} & \frac{-1,3015z^{-3}}{z - 0,9329} \end{bmatrix} \quad \mathbf{P}_q(s) = \begin{bmatrix} \frac{0,24666z^{-8}}{z - 0,9351} \\ \frac{0,3575z^{-4}}{z - 0,927} \end{bmatrix}.$$

Através do Matlab, obtemos uma representação mínima com 21 estados, e vale ressaltar que esta representação não é única mas, em geral, busca-se uma representação com o menor número de estados possível. O modelo não será replicado aqui por questão de espaço, mas o leitor pode verificar o modelo através do código Matlab disponibilizado.

Para efeitos de simulação, vamos considerar o modelo em espaço de estados como se fosse o processo real. O cenário de simulação considera um tempo total de simulação de 150s. O sistema parte de  $u_1 = u_2 = 0$  e  $y_1 = y_2 = 0$  e em  $t = 10\text{ s}$  uma variação tipo degrau unitário da referência de  $y_1$  é aplicada. Posteriormente, em  $t = 50\text{ s}$  se aplica uma variação tipo degrau de amplitude 0,5 na referência de  $y_2$ . Finalmente, em  $t = 100\text{ s}$  se aplica uma perturbação tipo degrau de amplitude  $-0,4$  em  $q$ .

Para este processo, vamos projetar SSMPC considerando a perturbação  $q$  como sendo não mensurável. Os parâmetros de sintonia são:  $N_{1,1} = 2$  e  $N_{1,2} = 3$ . O final dos horizontes de predição foram definidos como  $N_{2,1} = 40$  e  $N_{2,2} = 30$  e os horizontes de controle foram definidos como  $N_{u,1} = N_{u,2} = 5$ , as ponderações foram normalizadas pelos horizontes e definidas como  $\delta_l = \frac{1}{N_{y,l}}$  e  $\lambda_p = \frac{10}{N_{u,p}}$ , e foram consideradas restrições no sinal de controle  $u_p(k) \in [-1; 1]$ ,  $\forall p, k$ , e no incremento de controle  $\Delta u_p(k) \in [-0,1; 0,1]$ ,  $\forall p, k$ .

O resultado da simulação podem ser visto na Figura 4.3. Veja que o SSMPC consegue fazer as saídas seguirem as referências ao mesmo tempo em que o acoplamento entre as variáveis é minimizado. Isto fica evidente nas trocas de referências de ambas as saídas. O SSMPC também garante rejeição de perturbações do tipo degrau, como visto na Figura 4.3 a partir do instante 200 amostras. Assim como nos outros algoritmos, é possível obter uma resposta mais rápida diminuindo a ponderação das ações de controle. Na Figura 4.4, é apresentado o resultado de simulação quando se altera a ponderação de controle para  $\lambda_p = \frac{1}{N_{u,p}}$ , e note como a resposta do sistema se torna mais agressiva.

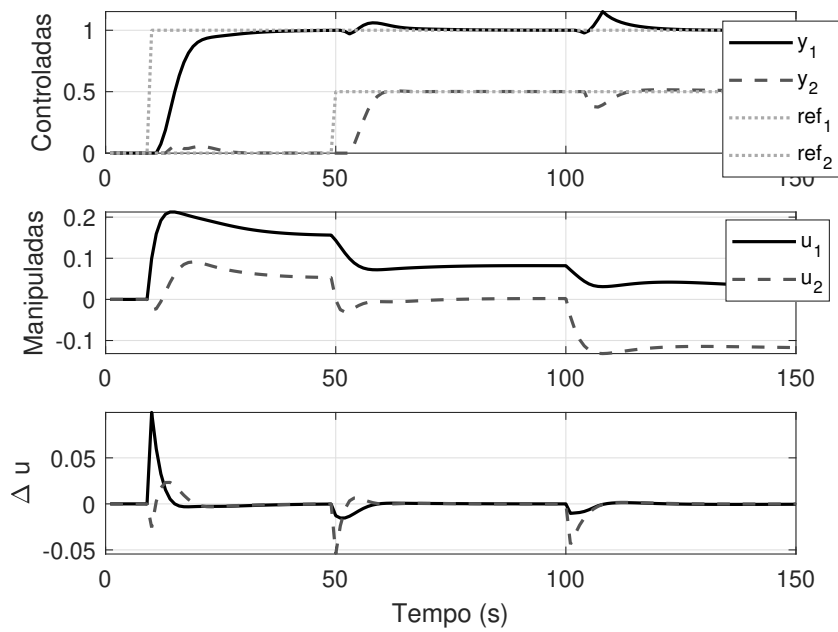


Figura 4.3: Exemplo 4.8 – resultados de simulação para a coluna de destilação de metanol.

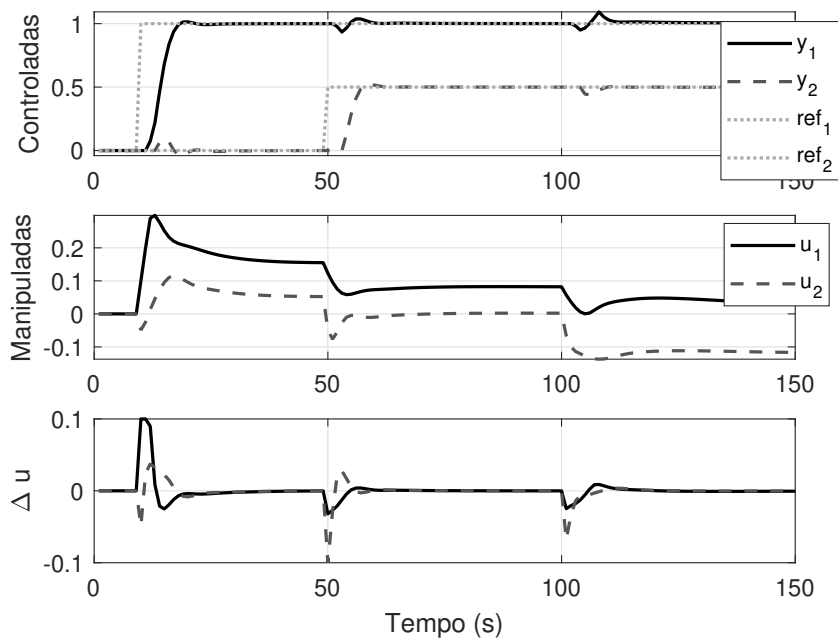


Figura 4.4: Exemplo 4.8 – resultados de simulação para a coluna de destilação de metanol com uma sintonia mais agressiva.

## 4.5 Controlador equivalente

É possível obter um sistema em espaço de estado que representa o controlador equivalente implementado pelo algoritmo SSMPC apresentado. Com esse controlador equivalente, é possível analisar, no caso irrestrito, as propriedades do controlador e da malha fechada resultantes. Utilizando (4.22) e (4.27), e desconsiderando os valores futuros das perturbações,

$$\Delta \mathbf{u}(k) = \mathbf{K}_1 \mathbf{R} - \mathbf{K}_1 \mathbf{F} \underline{\mathbf{x}} - \mathbf{K}_1 \mathbf{F}_q \underline{\mathbf{q}} \quad (4.31)$$

supondo que as referências futuras são constantes,  $\mathbf{r}(k+j) = \mathbf{r}(k), \forall j$ , então

$$\mathbf{R} = \begin{bmatrix} \mathbf{I}_n \\ \vdots \\ \mathbf{I}_n \end{bmatrix} \mathbf{r}(k),$$

e

$$\begin{aligned} \mathbf{u}(k) - \mathbf{u}(k-1) = & \mathbf{K}_r \mathbf{r}(k) - \mathbf{K}_{f0} \mathbf{x}(k) - \mathbf{K}_{f1} \mathbf{x}(k-1) \\ & - \mathbf{K}_{fq0} \mathbf{q}(k) - \mathbf{K}_{fq1} \mathbf{q}(k-1) \end{aligned} \quad (4.32)$$

$$\begin{aligned} \mathbf{u}(k) = & \mathbf{u}(k-1) + \mathbf{K}_r \mathbf{r}(k) - \mathbf{K}_{f0} \mathbf{x}(k) \\ & - \mathbf{K}_{f1} \mathbf{x}(k-1) - \mathbf{K}_{fq0} \mathbf{q}(k) - \mathbf{K}_{fq1} \mathbf{q}(k-1), \end{aligned} \quad (4.33)$$

com  $\mathbf{K}_1 \mathbf{F} = [\mathbf{K}_{f0}, \mathbf{K}_{f1}]$  e  $\mathbf{K}_1 \mathbf{F}_q = [\mathbf{K}_{fq0}, \mathbf{K}_{fq1}]$ .

Para se obter a representação equivalente do SSMPC em espaço de estados, é necessário descrever a equação do sistema no formato padrão do espaço de estados (4.1). Para isso, definimos um conjunto de estados auxiliares,

$$\mathbf{x}_c(k) = \begin{bmatrix} \mathbf{u}(k-1) \\ \mathbf{x}(k-1) \\ \mathbf{q}(k-1) \end{bmatrix} \quad (4.34)$$

e rearranjamos (4.33), obtendo controlador equivalente

$$\left\{ \begin{aligned} \mathbf{x}_c(k+1) = & \begin{bmatrix} \mathbf{I}_m & -\mathbf{K}_{f1} & -\mathbf{K}_{fq1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}_c(k) \\ & + \begin{bmatrix} \mathbf{K}_r & -\mathbf{K}_{f0} & -\mathbf{K}_{fq0} \\ 0 & \mathbf{I}_{n_s} & 0 \\ 0 & 0 & \mathbf{I}_{m_q} \end{bmatrix} \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \\ \mathbf{u}(k) = & [\mathbf{I}_m \quad -\mathbf{K}_{f1} \quad -\mathbf{K}_{fq1}] \mathbf{x}_c(k) \\ & + [\mathbf{K}_r \quad -\mathbf{K}_{f0} \quad -\mathbf{K}_{fq0}] \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \end{aligned} \right. \quad (4.35a)$$

$$(4.35b)$$

ou, de forma compacta,

$$\begin{cases} \mathbf{x}_c(k+1) = \mathbf{A}_c \mathbf{x}_c(k) + \mathbf{B}_c \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \\ \mathbf{u}(k) = \mathbf{C}_c \mathbf{x}_c(k) + \mathbf{D}_c \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \end{cases} \quad (4.36a)$$

$$\quad (4.36b)$$

Com o controlador equivalente, é possível obter a representação em espaço de estados da malha fechada completa. Para isso, considere a representação (4.1), replicada aqui por conveniência:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_q\mathbf{q}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{C}_q\mathbf{q}(k) \end{cases} \quad (4.37a)$$

$$\quad (4.37b)$$

Definindo saída auxiliares do erro de seguimento  $\mathbf{e}(k) = \mathbf{r}(k) - \mathbf{y}(k)$ , e do sinal de controle  $\mathbf{u}(k)$ , e combinando (4.35) com a equação anterior através da junção dos estados em um vetor de estados aumentado, a malha fechada completa é dada por

$$\begin{cases} \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_c(k+1) \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{K}_{f0}) & \mathbf{B}\mathbf{C}_c \\ \begin{bmatrix} -\mathbf{K}_{f0} \\ \mathbf{I}_{n_s} \\ 0 \end{bmatrix} & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_c(k) \end{bmatrix} \\ \quad + \begin{bmatrix} \mathbf{B}\mathbf{K}_r & (\mathbf{B}_q - \mathbf{B}\mathbf{K}_{fq0}) \\ \mathbf{K}_r & -\mathbf{K}_{fq0} \\ 0 & 0 \\ 0 & \mathbf{I}_{m_q} \end{bmatrix} \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{q}(k) \end{bmatrix} \end{cases} \quad (4.38a)$$

$$\begin{cases} \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{u}(k) \\ \mathbf{e}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{C} & 0 \\ -\mathbf{K}_{f0} & \mathbf{C}_c \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_c(k) \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{C}_q \\ \mathbf{K}_r & -\mathbf{K}_{fq0} \\ \mathbf{I} & -\mathbf{C}_q \end{bmatrix} \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{q}(k) \end{bmatrix} \end{cases} \quad (4.38b)$$

ou, de forma compacta:

$$\begin{cases} \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_c(k+1) \end{bmatrix} = \mathbf{A}_{mf} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_c(k) \end{bmatrix} + \mathbf{B}_{mf} \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{q}(k) \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{u}(k) \\ \mathbf{e}(k) \end{bmatrix} = \mathbf{C}_{mf} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_c(k) \end{bmatrix} + \mathbf{D}_{mf} \begin{bmatrix} \mathbf{r}(k) \\ \mathbf{q}(k) \end{bmatrix} \end{cases} \quad (4.39a)$$

$$\quad (4.39b)$$

**Exemplo 4.9.** *Calcularemos os controladores equivalentes para o sistema de dois tanques apresentado no Exemplo 4.3 levando em conta a primeira sintonia,  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = N_{2,2} = 3$ ,  $N_{u,1} = N_{u,2} = 2$ ,  $\delta_1 = 0,8$ ,  $\delta_2 = 1$ ,  $\lambda_1 = 1$  e  $\lambda_2 = 0,6$ , e com medição da perturbação mas sem levar em conta seus valores futuros ( $N_q = 0$ ). Como mostrado no Exemplo 4.5, a matriz de ganhos do controlador é:*

$$\mathbf{K}_1^T = \begin{bmatrix} 0,075383 & -0,0032122 \\ -0,020076 & 0,18745 \\ 0,1185 & -0,0050495 \\ -0,010486 & 0,1616 \\ 0,14564 & -0,0062059 \\ 0,029744 & 0,34648 \end{bmatrix}^T,$$

e as matrizes  $\mathbf{F}$  e  $\mathbf{F}_q$ , calculadas nos exemplos 4.3 e 4.5, respectivamente, são:

$$\mathbf{F} = \begin{bmatrix} 1,6 & 0 & -0,6 & 0 \\ 0,4 & 2 & -0,4 & -1 \\ 1,96 & 0 & -0,96 & 0 \\ 1,04 & 3 & -1,04 & -2 \\ 2,176 & 0 & -1,176 & 0 \\ 1,824 & 4 & -1,824 & -3 \end{bmatrix}, \quad \mathbf{F}_q = \begin{bmatrix} 0 & 0 \\ -0,5 & 0,5 \\ 0 & 0 \\ -1 & 1 \\ 0 & 0 \\ -1,5 & 1,5 \end{bmatrix}.$$

Assim, podemos obter,

$$\mathbf{K}_1 \mathbf{F} = \begin{bmatrix} 0,70509 & 0,047365 & -0,36557 & -0,048184 \\ 0,84649 & 2,2456 & -0,86096 & -1,5501 \end{bmatrix},$$

$$\mathbf{K}_r = \begin{bmatrix} 0,33952 & -0,0008184 \\ -0,014468 & 0,69554 \end{bmatrix}, \quad \mathbf{K}_1 \mathbf{F}_q = \begin{bmatrix} -0,024092 & 0,024092 \\ -0,77505 & 0,77505 \end{bmatrix},$$

que são suficientes para calcular as matrizes do controlador equivalente de acordo com (4.36)

$$\mathbf{A}_c = \begin{bmatrix} 1 & 0 & 0,36557 & 0,048184 & -0,024092 \\ 0 & 1 & 0,86096 & 1,5501 & -0,77505 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B}_c = \begin{bmatrix} 0,33952 & -0,0008184 & -0,70509 & -0,047365 & 0,024092 \\ -0,014468 & 0,69554 & -0,84649 & -2,2456 & 0,77505 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{C}_c = \begin{bmatrix} 1 & 0 & 0,36557 & 0,048184 & -0,024092 \\ 0 & 1 & 0,86096 & 1,5501 & -0,77505 \end{bmatrix},$$

$$\mathbf{D}_c = \begin{bmatrix} 0,33952 & -0,0008184 & -0,70509 & -0,047365 & 0,024092 \\ -0,014468 & 0,69554 & -0,84649 & -2,2456 & 0,77505 \end{bmatrix}.$$

Por fim, com o modelo em espaço de estados do controlador equivalente, é possível combiná-lo com o modelo do sistema, resultando nas seguintes matrizes do modelo completo de malha fechada de acordo com (4.39)

$$\mathbf{A}_{mf} = \begin{bmatrix} 0,52949 & -0,0047365 & 0,1 & 0 & 0,036557 & 0,0048184 & -0,0024092 \\ -0,023247 & -0,12282 & 0 & 0,5 & 0,43048 & 0,77505 & -0,38753 \\ -0,70509 & -0,047365 & 1 & 0 & 0,36557 & 0,048184 & -0,024092 \\ -0,84649 & -2,2456 & 0 & 1 & 0,86096 & 1,5501 & -0,77505 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B}_{mf} = \begin{bmatrix} 0,033952 & -8,184e^{-05} & 0,0024092 \\ -0,0072339 & 0,34777 & -0,11247 \\ 0,33952 & -0,0008184 & 0,024092 \\ -0,014468 & 0,69554 & 0,77505 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{C}_{mf} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0,70509 & -0,047365 & 1 & 0 & 0,36557 & 0,048184 & -0,024092 \\ -0,84649 & -2,2456 & 0 & 1 & 0,86096 & 1,5501 & -0,77505 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{D}_{mf} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0,33952 & -0,0008184 & 0,024092 \\ -0,014468 & 0,69554 & 0,77505 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

O modelo completo em malha fechada pode ser utilizado para verificar estabilidade, robustez, e como o sistema se comporta para sinais em diferentes frequências (diagrama de Bode).

Colocar estudo de caso aqui? Podemos repetir algum e comparar com o DMC/GPC

## 4.6 Comentários finais

Este capítulo apresentou uma formulação em espaço de estados de controle preditivo denominada aqui de SSMPC. Vale ressaltar que esta é uma formulação particular e outras fontes da literatura podem definir um MPC em espaço de estados de forma diferente. De qualquer maneira, as premissas básicas aqui apresentadas ainda são válidas.

A representação em espaço de estados possui como vantagem ser intrinsecamente multivariável, não exigindo nenhuma alteração em relação ao caso monovariável, facilitando a aplicação do SSMPC a esse tipo de sistema. Além disso, o SSMPC pode lidar com qualquer tipo de dinâmica tal como, por exemplo, dinâmicas instáveis e de fase não-mínima, sem a necessidade de qualquer adaptação. Como desvantagem, podemos citar a necessidade de se medir todos os estados do sistema, o que nem sempre é viável na prática, ou a necessidade de se utilizar estimadores de estado, que podem complicar o projeto do controlador, pois influenciam o comportamento de malha fechada.

Além disso, mostramos como obter o controlador SSMPC com e sem ação antecipativa (*feed-forward*), e considerando ou não restrições no processo. Nos demais aspectos, tal como sintonia, o SSMPC é idêntico aos outros algoritmos MPC multivariáveis.





## Capítulo 5

# Controle Preditivo Não Linear

Como mostramos no Capítulo 1, a filosofia dos controladores MPC pode ser aplicada a qualquer tipo de processo sempre que o modelo usado no controlador represente adequadamente as relações entre as principais variáveis do sistema: manipuladas, controladas e perturbações. Sabemos que a maioria dos sistemas reais têm dinâmica não linear, assim, é natural pensar em utilizar uma estratégia de controle MPC com modelos de predição não lineares. Porém, como é comum os processos operarem numa pequena faixa ao redor de um ponto de operação, as dinâmicas tipicamente podem ser aproximadas de modo satisfatório usando modelos lineares, o que motiva o desenvolvimento de técnicas MPC que utilizam modelos lineares, como visto nos capítulos anteriores. Assim, do ponto de vista prático, fica claro que se um modelo linear for suficiente para obter um bom desempenho do controlador, não parece justificável complicar o projeto usando modelos mais complexos.

Porém, quando os processos têm dinâmica muito não linear ou quando o processo trabalha em várias faixas de operação, o emprego de modelos não lineares no projeto do controle tende a permitir manter o desempenho desejado para o sistema em malha fechada. Do ponto de vista conceitual, o controle preditivo não linear (NMPC, do inglês *Nonlinear MPC*) não apresenta dificuldades. O NMPC considera a mesma função custo que no caso linear, um modelo não linear para o cálculo das predições e o objetivo consiste, novamente, em encontrar o controle que minimize a função custo num algoritmo de horizonte deslizante. Já do ponto de vista prático, são várias as dificuldades encontradas neste problema: (a) a determinação de um modelo não linear de um processo a partir de dados por métodos de identificação é mais complexa; (b) a obtenção do sinal de controle requer, em geral, a solução de um problema otimização não convexo; (c) toda a análise teórica do sistema em malha fechada, como estudos de estabilidade e robustez, tornam-se mais difíceis de serem feitos; (d) a simulação interna usada pelo MPC na

própria implementação do controlador para o cálculo das previsões torna-se mais complexa. Por esses motivos, o NPMC é um campo aberto para pesquisas tanto na área de determinação de modelos como nos procedimentos e algoritmos de otimização para o cálculo do controle.

Neste capítulo, estudaremos algumas das técnicas disponíveis na literatura para a solução do problema NMPC, focando naquelas que podem ser mais simples de usar na prática, tanto com o objetivo de uso no ambiente acadêmico quanto no industrial. Além disso, na próxima seção apresentamos uma visão geral do conjunto de técnicas existentes na literatura.

## 5.1 Controle preditivo não linear: Uma visão geral

Existe uma grande quantidade de técnicas MPC que direta ou indiretamente tratam sistemas não lineares. Uma possível classificação destas técnicas as divide em três grupos:

- (a) as que utilizam diretamente um modelo não linear genérico e um algoritmo de otimização não linear de propósito geral para calcular a ação de controle;
- (b) as que utilizam modelos não lineares particulares e métodos de otimização específicos para o modelo escolhido;
- (c) as que utilizam um ou vários modelos lineares para representar parte do (ou todo o) comportamento do processo e, portanto, resolvem vários problemas de controle preditivo com modelos lineares para calcular a ação de controle.

Evidentemente, as vantagens de uma ou outra alternativa estão diretamente relacionadas ao compromisso desempenho-custo computacional. Espera-se que técnicas que utilizem modelos mais complexos apresentem melhor desempenho em termos de resposta do sistema em malha fechada, mas sejam mais custosas computacionalmente. Além disso, cabe destacar que pode ser necessário aumentar o período de amostragem para que uma abordagem mais complexa possa convergir, o que pode prejudicar o desempenho dessa estratégia de controle e fazer com que abordagens mais simples acabem apresentando respostas em malha fechada melhores do que as apresentadas por técnicas mais complexas. Por fim, merece destaque o fato de que a formulação direta com modelos não lineares pode levar a problemas de otimização

que não convergem ou que convergem a mínimos locais. Nesta seção, as vantagens e desvantagens de algumas dessas técnicas serão discutidas, com uma visão prática, para fornecer subsídios ao estudante ou projetista na hora de escolher uma ou outra metodologia.

Uma abordagem natural é a utilização das predições na função custo empregando o modelo não linear do processo, que chamaremos de NMPC direto. Neste caso, a minimização da função custo é realizada com programas de otimização especializados na solução de funções não lineares, como Programação Não Linear ou Programação Quadrática Sequencial, por exemplo. Comparando com o problema QP já apresentado, aqui são preservadas as características de horizonte deslizante e a repetição da otimização a cada período de amostragem, mas a formulação não linear apresenta a vantagem de poder considerar restrições não lineares no problema. Do ponto de vista prático, a principal vantagem deste método reside no uso de um modelo não linear que permite, em geral, obter predições muito próximas das saídas reais da planta, o que tipicamente permite melhorar o desempenho do sistema em malha fechada. Como desvantagens podemos citar:

- (i) o elevado tempo de cômputo quando comparado ao MPC linear, geralmente devido ao grande número de iterações usadas nos métodos de otimização não linear;
- (ii) maior incerteza no tempo que o algoritmo leva para entregar uma solução do problema em cada amostragem, quando comparado a um problema QP;
- (iii) os problemas de convergência dos algoritmos, que podem, por exemplo, chegar a soluções que sejam mínimos locais.

Todos esses fatores fazem com que a utilização do NMPC direto seja ainda pouco difundida nas aplicações práticas de controle em tempo real.

Uma forma de obter uma solução mais simples que a correspondente ao modelo não linear genérico consiste em utilizar classes de modelos não lineares particulares como, por exemplo, Séries de Volterra ou modelos de Hammersstein. Essas alternativas são interessantes na prática quando o processo que está sendo estudado enquadra-se bem na aproximação selecionada. O compromisso entre complexidade de cálculo e desempenho é novamente o fator determinante da escolha. A otimização nesses casos pode ser mais simples que no caso geral. Do ponto de vista prático, é importante notar que o uso de um modelo mais complexo que represente melhor a dinâmica do processo real somente trará benefício quando o erro de modelagem for pequeno. Em outros casos, a dessintonia do controlador necessária para manter a robustez pode eliminar a vantagem do uso de um melhor modelo.

Uma terceira abordagem consiste em utilizar uma versão linearizada do modelo para a síntese do controlador. As formulações desta classe utilizam diferentes técnicas, como procedimentos de linearização em vários pontos de operação e uma técnica de escolha do modelo a ser usado (ou de ponderação entre modelos), ou linearizações sobre a trajetória, por exemplo. Nas técnicas de ponderação de múltiplos modelos lineares, calculam-se vários modelos lineares em diferentes pontos de equilíbrio do sistema, e emprega-se uma estratégia que decide que modelo deve ser usado em cada amostra. Essa estratégia usa algum critério, que pode ser, por exemplo, a distância do ponto atual de operação em relação aos diferentes pontos usados na linearização. Esse mesmo critério pode ser usado também para obter um modelo calculado como uma combinação linear dos modelos lineares encontrados nos diferentes pontos de operação. O custo computacional e o desempenho desses métodos está diretamente relacionado ao número de modelos necessários e aos procedimentos de ponderação que, em geral, não são simples de generalizar. Nas técnicas de linearização sobre trajetória, muito usadas, por exemplo, em robótica ou em veículos autônomos, calcula-se um modelo linearizado genérico de forma que os parâmetros do modelo linear são escritos como funções do ponto de operação do sistema (por exemplo, em um veículo eles são escritos como função da posição, da velocidade e da orientação do mesmo). Com esse modelo linear a parâmetros variáveis (LPV, do inglês *Linear Parameter Varying*), aplica-se qualquer algoritmo de MPC linear, que, neste caso, terá uma matriz dinâmica variável no tempo, de acordo com o ponto sobre a trajetória. Observe que, ao usar estes modelos LPV, devemos congelar os parâmetros quando calculamos as previsões, dado que não conhecemos a trajetória futura para poder calcular exatamente os parâmetros do modelo linear ao longo do horizonte.

Uma outra forma de usar modelos lineares é utilizar linearizações do modelo não linear a cada instante de amostragem, o que ajusta melhor a resposta do modelo ao comportamento não linear do sistema, diminuindo o erro de aproximação ao custo de um aumento da carga computacional. Neste caso, também se resolve, de forma aproximada, o problema de otimização não linear, usando um problema do tipo QP. Uma dessas técnicas, denominada NEPSAC (*Nonlinear Extended Predictive Self-Adaptive Controller*), proposta pelo Prof. Robin De Keyser [KL03], usa um modelo linear para o cálculo da resposta do modelo, separando em duas partes essa resposta, denominadas resposta base e ótima. No algoritmo, ambas as respostas são recalculadas usando um procedimento iterativo, buscando que a solução achada seja muito próxima da não linear. Assim, o problema de otimização não linear pode ser transformado numa sequência de problemas de otimização quadrática. Essa metodologia funciona bem em muitos casos práticos sem

necessidade de muitas iterações, apesar de não ter garantia de obtenção do ótimo nem da convergência do processo iterativo.

Uma solução simples que também pode ser usada na prática é considerar dois modelos para o sistema, um modelo não linear completo e um modelo linearizado em algum ponto de operação que o projetista considera o mais adequado para as condições de operação do processo (por exemplo, no meio da faixa de operação ou no ponto de operação nominal do processo). Com base nesses dois modelos, as previsões são calculadas de forma aproximada, usando o modelo não linear para o cálculo da resposta livre, e o modelo linear para a resposta forçada. Essa aproximação vai conter um erro devido a dois fatores: por usar o princípio de superposição para um sistema não linear, e por calcular a resposta forçada com um modelo aproximado. Por outro lado, a grande vantagem dessa abordagem é que pode ser usado um algoritmo de otimização do tipo QP, como no controle MPC de modelo linear. Observe que como a função custo pode ser colocada em função apenas dos controles futuros, todo o procedimento de minimização pode ser mantido como no caso linear, apenas calculando a resposta livre usando o modelo não linear. Veja que, neste caso, praticamente manteríamos todo o código do MPC linear, apenas mudando a parte que calcula a resposta livre, o que pode ser feito usando um código que simula a resposta livre no horizonte de previsão a partir dos valores passados do controle e saídas ou estados medidos.

Uma proposta que melhora o método anterior, chamada de PNPMC (do inglês *Practical NMPC*), proposta pelo Prof. Agostinho Plucênio [PPBNR07], mantém o cálculo da resposta livre com o modelo não linear, mas calcula a resposta forçada usando um modelo linearizado em cada passo de amostragem, e não um modelo fixo. Desta forma, o modelo de previsão se ajusta ao comportamento não linear do processo, diminuindo o erro de previsão.

Note que, em todos os casos anteriores nos quais são usadas aproximações e iterações, as técnicas tratam de compensar as perdas no desempenho (por causa da aproximação do modelo) com a maior velocidade de processamento e simplicidade dos algoritmos. Assim, pode-se considerar como melhor aquele algoritmo que consiga o melhor compromisso entre desempenho, simplicidade e velocidade de execução.

Nas próximas seções, estudaremos três algoritmos de NPMC, o de cálculo direto, usando as poderosas ferramentas hoje disponíveis para otimização não linear, e duas de aproximação linear, o NEPSAC e o PNPMC.

## 5.2 Controle NMPC direto

Do ponto de vista da sua formulação, o NMPC direto é muito simples. Vamos usar aqui um modelo genérico não linear em variáveis de estado, mas poderíamos usar qualquer outra representação. Assim como no caso linear, a vantagem dessa formulação é que vale para sistemas SISO e MIMO.

Assim, considere que o processo pode ser representado pelo modelo de tempo discreto:

$$\begin{cases} \mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{q}(k)), \\ \mathbf{y}(k) = g(\mathbf{x}(k), \mathbf{u}(k), \mathbf{q}(k)), \end{cases} \quad (5.1a)$$

$$(5.1b)$$

sendo  $\mathbf{x} \in \mathbb{R}^{n_s}$ ,  $\mathbf{u} \in \mathbb{R}^m$ ,  $\mathbf{q} \in \mathbb{R}^{m_q}$  e  $\mathbf{y} \in \mathbb{R}^n$  o estado de dimensão  $n_s$ , o controle de dimensão  $m$ , a perturbação de dimensão  $m_q$  e a saída de dimensão  $n$  do sistema, respectivamente. Ainda  $f$  e  $g$  são duas funções não lineares contínuas e diferenciáveis. Cabe destacar que se as condições anteriores para  $f$  e  $g$  não forem cumpridas, ainda é possível aplicar o método, porém torna-se mais difícil encontrar a solução ótima do problema.

Considerando a mesma função custo utilizada nos métodos de MPC linear e, para simplificar, um horizonte de predição de 1 a  $N$  e horizonte de controle  $N_u$ , temos:

$$J = (\hat{\mathbf{y}} - \mathbf{r})^T \mathbf{Q}_y (\hat{\mathbf{y}} - \mathbf{r}) + \mathbf{u}^T \mathbf{Q}_u \mathbf{u},$$

com as mesmas definições já vistas para as variáveis e parâmetros usados em  $J$ . O problema a ser resolvido para a formulação NMPC é dado por:

$$\min J \quad (5.2)$$

s.a.

$$\begin{aligned} \hat{\mathbf{x}}(k+j+1) &= f(\hat{\mathbf{x}}(k+j), \mathbf{u}(k+j), \mathbf{q}(k+j)), \quad j = 1, \dots, N \\ \hat{\mathbf{y}}(k+j) &= g(\hat{\mathbf{x}}(k+j), \mathbf{u}(k+j), \mathbf{q}(k+j)), \quad j = 1, \dots, N \\ h(\hat{\mathbf{x}}(k), \mathbf{u}(k), \hat{\mathbf{y}}(k)) &\leq 0, \end{aligned}$$

onde  $h$  é uma função que representa as restrições colocadas no controle, estados e saídas do processo. Essa função pode incluir restrições comuns de limites das variáveis e também restrições não lineares em geral. Observa que, além das restrições definidas por  $h$ , o problema tem como restrição o modelo do processo usado para o cálculo das predições.

Em termos práticos, este problema pode ser resolvido diretamente com o uso de *solvers* não lineares de uso geral (como por exemplo *fmincon*, *ipopt*) ou por ferramentas que disponibilizam interfaces para *solvers* e variáveis simbólicas como, por exemplo, CasADi, AMPL, CVX, CVXPY ou GRAMPC.

Para exemplificar, apresentamos, na sequência, o procedimento para colocar o problema do NMPC no CasADi.

### 5.2.1 CasADi – Generalidades

CasADi (do inglês *Computer-algebra software for Automatic Differentiation*) é uma ferramenta de código aberto para otimização não linear e diferenciação algorítmica [AGH<sup>+</sup>19]. Essa ferramenta facilita a implementação rápida e eficiente de diferentes métodos para controle numérico ótimo, tanto em um contexto *off-line* quanto para controle preditivo de modelo não linear (NMPC). CasADi é implementado em C++, que é conhecido por seu bom desempenho, porém possui interfaces para linguagens de mais alto nível, como Python, Julia e MATLAB. Esta última foi utilizada neste livro.

O CasADi possui uma estrutura simbólica independente, o que permite ao usuário construir expressões usando uma sintaxe inspirada em MATLAB. Assim, por exemplo, vetores são tratados como matrizes  $n \times 1$  e escalares como matrizes  $1 \times 1$ .

#### Simbologias

Após instalar o CasADi, é possível importar as simbologias no ambiente de programação, para a linguagem MATLAB, da seguinte maneira:

```
import casadi.*
```

Após importar a ferramenta CasADi no MATLAB, é possível trabalhar no software de maneira análoga ao que fazemos normalmente em MATLAB, utilizando os comandos conhecidos. Por exemplo, para concatenar matrizes horizontalmente e verticalmente usamos:

```
Y = [Y1,Y2,Y3,Y4]
Z = [Z1;Z2;Z3;Z4]
```

Essas classes são úteis na solução de um problema de NMPC via CasADi. Por exemplo, em um caso multivariável, a concatenação vertical permite de maneira simples juntar as predições das diferentes saídas em um único vetor.

A principal classe utilizada para implementar o NMPC é a Opti Stack, detalhada na sequência.

#### Opti Stack

A Opti Stack é um tipo de interface que simplifica a escrita do problema de otimização, deixando o código mais limpo e legível. Nela, é utilizada uma notação matemática de um problema não linear de uma forma similar ao que se coloca em uma formulação teórica. Um objeto da classe Opti Stack é instanciado da seguinte forma:

```
opti = casadi.Opti()
```

Após definir a ferramenta de otimização, podemos utilizá-la para criar variáveis que serão o argumento do problema de otimização. Esta variável pode ser escalar, por exemplo para um sistema SISO com  $N_u = 1$ , ou vetorial, para um sistema SISO com  $N_u > 1$  ou um sistema MIMO qualquer. Exemplos desta criação de variáveis são:

```
% Escalar
X = opti.variable()
% Vetor Coluna de tamanho N
X = opti.variable(N)
% Matriz de tamanho N por M
X = opti.variable(N,M)
```

No problema de otimização, precisamos fornecer valores iniciais para as variáveis de otimização, usando a seguinte sentença, colocada aqui para um escalar  $x$ :

```
% Atribui 2 como valor inicial de x[1]
opti.set_initial(X(1), 2)
```

Para declarar restrições de igualdade ou desigualdade temos três possibilidades de comandos, que são novamente exemplificadas aqui para um escalar  $x$ :

```
% Restrição de desigualdade: x tem que ser maior ou igual a 2
opti.subject_to(x>=2)
% Restrição de igualdade: x tem que ser igual a 2
opti.subject_to(x==2)
% Restrição de desigualdade dupla: -2<=x<=2
opti.subject_to( -2<=X<=2)
```

Para declarar o objetivo do problema, uma vez definido a função custo  $J$ , usamos o método:

```
% Objetivo: minimizar a função J
opti.minimize(J)
```

Ainda precisamos escolher um *solver*, usando o comando:

```
opti.solver('nome do solver')
```

Existem diversas opções de *solvers* que podem ser utilizados. Por exemplo, *ipopt* é um *solver* livre do grupo de pesquisa do Prof. Biegler [WB06], e é baseado no método do ponto interior, sendo bem robusto. Outra opção é o *sqpmethod*, que é um QP sequencial básico, tal como a formulação que se encontra em livros de otimização, e é implementado diretamente em CasADi. Já o *bonmin* é um *solver* experimental que permite usar variáveis mistas inteiras. Também foi desenvolvido pelo grupo do Prof. Biegler e utiliza *ipopt*



junto com técnicas de *branch and bound*.

Uma vez configurado o problema, pode-se chamar o método `solve`, que constrói o problema na forma não linear e o soluciona. Essa chamada é feita da seguinte maneira:

```
solution=opti.solve()
```

### 5.2.2 Montando o NMPC em CasADi

Vamos considerar um exemplo genérico multivariável de um processo com  $n$  saídas e  $m$  entradas, horizonte de predição de 1 a  $N$  e  $N_u = N$ . O NMPC tem como variáveis de decisão ou otimização o vetor de estados e o vetor de controle  $U$ , que devem ser mantidas entre 0 e 100. Assim, para configurar o algoritmo, precisamos de uma matriz de dimensão  $m \times N$  para o controle e de  $n \times N + 1$  para a variável de estado  $X$ . Observe que, como explicado, o problema de otimização vai considerar a equação de estados como uma das restrições, e como esta calcula  $X(k + 1)$  em cada passo, precisamos  $N + 1$  colunas em  $X$ , pois o primeiro elemento será  $X(k)$ , que é medido, e no último elemento do horizonte teremos  $X(N + 1)$ .

Para montar o problema de otimização, montamos o custo iterativamente de 1 a  $N$ . As restrições também são formuladas nesse mesmo laço. Assim, pode-se resolver o NMPC da seguinte maneira:

```
% Problema NMPC com horizonte de predição e de controle iguais a N
% Definindo a ferramenta de Otimização
opti = casadi.Opti()

% Variáveis de Otimização
% Variável de estado: matriz de tamanho n por N+1
X = opti.variable(n,N+1)
% Variável de entrada: matriz de tamanho m por N
U = opti.variable(m,N)

% Inicializar o objetivo
obj = 0

% Definir condição inicial para X
opti.set_initial(X(:,1),X_inicial)

% Definir condição inicial para U (necessário para quando usamos
warm-start)
opti.set_initial(U(:,1),U_inicial)
```

```

% Iteração para montagem de J(X,U) (função custo genérica) a cada
    passo dentro do horizonte considerando as restrições estáticas e
    dinâmicas.

for k=1:N
    obj = obj + J(X(:,k+1),U(:,k))
    opti.subject_to(X(:,k+1) == f(X(:,k), U(:,k)))
    opti.subject_to(0<=X(:,k+1)<=100)
    opti.subject_to(0<=U(:,k+1)<=100)
end

% Declarando o objetivo do problema: minimizar obj
opti.minimize(obj)

% Utilizando o solver ipopt
opti.solver('ipopt')

% Solucionando o problema não linear
solution = opti.solve()

% Retorna valor do objetivo
J = solution.value(obj)

% Retorna valor do controle solução
u=solution.value(U)

```

### 5.2.3 Um exemplo completo

Vamos apresentar, na sequência, um exemplo simples, empregado para discutir alguns detalhes da formulação do problema NMPC e a sua resolução com CasADi.

**Exemplo 5.1.** *Vamos considerar o modelo de controle de nível  $h(k)$  de um tanque com acionamento pela abertura da válvula de saída  $u(k)$  e considerando a abertura da válvula de entrada  $q(k)$  como uma perturbação mensurável. O modelo deste tanque é:*

$$h(k) = h(k-1) + K_1 q(k-1) - K_2 u(k-1) \sqrt{2gh(k-1)},$$

com  $K_1$  e  $K_2$  constantes reais.

Neste sistema simples de primeira ordem, o estado coincide com a saída  $e$ , ainda, somente temos escalares nas variáveis da função não linear que

descreve o modelo, assim  $y(k) = x(k) = h(k)$  e o modelo na forma geral sugerida é:

$$\begin{cases} x(k+1) = f(x(k), u(k), q(k)), \\ y(k) = x(k), \end{cases}$$

com  $f(x(k), u(k), q(k)) = x(k) + K_1 q(k) - K_2 u(k) \sqrt{2gx(k)}$ .

Vamos considerar, também, que as restrições do problema são apenas as bandas de cada variável, dadas em valores percentuais:  $x \in [0\%, 100\%]$ ,  $u \in [0\%, 100\%]$  e  $q \in [0\%, 100\%]$ . Para este problema, vamos considerar uma função custo simples, que penaliza a ação de controle e não o incremento, e usaremos a seguinte sintonia:  $N_1 = 1$ ,  $N_2 = 20$ ,  $N_u = 5$  e  $\lambda = 0,01$ . Assim:

$$J = \sum_{j=1}^{20} (r(k+j) - x(k+j))^2 + \sum_{j=0}^4 \lambda u(k+j)^2.$$

Ainda, como a perturbação futura não é conhecida, usamos  $q(k+j) = q(k)$  no modelo. Assim, o problema a resolver é dado por:

$$\min J$$

s.a.

$$\begin{aligned} x(k+j+1) &= x(k+j) + K_1 q(k) - K_2 u(k+j) \sqrt{2gx(k+j)}, \\ x(k+j) &\in [0\%, 100\%], & j &= 1, \dots, 20 \\ u(k+j) &\in [0\%, 100\%] & j &= 0, \dots, 4, \end{aligned}$$

que usando CasADi se coloca como:

```
import casadi.*

% Define parâmetros do problema

K1=2;
K2=.1;
g=9.8;           % Aceleração da gravidade [m/s^2]
tsim=100;        % Tempo de simulação [s]
ts=1;            % Período de amostragem [s]
nit=round(tsim/ts); % número de iterações

ns=1;            % Numero de estados
m=1;             % Numero de entradas
n=1;             % Numero de saídas
N=20;            % Horizonte de predição
Nu=5;            % Horizonte de controle
```

```

Qx = eye(n)           % Ponderação estado
Qu = 0.1              % Ponderação controle

xs = 50*ones(n,nit);  % Estados
us = 31.9438*ones(m,nit); % Controle
qs = 50*ones(n,nit);  % Perturbação
xr = 50*ones(n,nit);  % Referência
J = zeros(1,nit);     % Objetivo

Umax = 100;           % Valor máximo da entrada
Umin = 0;             % Valor mínimo da entrada

Xmax = 100;           % Valor máximo do estado
Xmin = 0;             % Valor mínimo do estado

% valor da referência ao longo da simulação
xr(1,round(tsim*0.1/ts):end) = 45;

% Valor da perturbação ao longo da simulação
qs(1,round(nit/2):end) = 45;

% Laço de simulação em i
for k=2:nit
    %% simulação do modelo
    xs(k) = xs(k-1) + K1*qs(k-1) - K2*us(k-1)*sqrt(2*g*xs(k-1));

    %% Controlador NMPC com CasADi
    %% Variáveis de minimização NLP
    opti = casadi.Opti();
    X = opti.variable(n,N+1);
    U = opti.variable(n,Nu);
    obj = 0;

    % Inicialização X e U com os valores atuais (warm start)
    opti.set_initial(U,us(k-1));
    opti.set_initial(X,xs(k-1));
    % Montando as restrições e o objetivo
    for j=1:N
        obj = obj + (xr(:,k)-X(:,j+1))'*Qx*(xr(:,k)-X(:,j+1));
        % Restrição dinâmica do modelo para j<=Nu
        if j<=Nu

```

```

        % Caso j+1<Nu soma parcela de controle e adiciona
        % restrição
        obj = obj + U(:,j)'*Qu*U(:,j);
        opti.subject_to(X(:,j+1) == X(:,j) +K1*qs(k)
            -K2*U(:,j)*sqrt(2*g*max(X(:,j),1e-3)));
        % Restrições de operação
        opti.subject_to(Xmin<=X(:,j+1)<=Xmax);
        opti.subject_to(Umin<=U(:,j)<=Umax);
    else
        % Se j+1>=Nu, sinal de controle é fixo
        opti.subject_to(X(:,j+1) == X(:,j) +K1*qs(k)
            -K2*U(:,Nu)*sqrt(2*g*max(X(:,j),1e-3)));
        % Restrições de operação
        opti.subject_to(Xmin<=X(:,j+1)<=Xmax);
    end
end

% Declarando o objetivo do problema: minimizar obj
opti.minimize(obj);
% Utilizando o solver ipopt
opti.solver('ipopt');
% Solucionando o problema não linear
sol = opti.solve();
% Vetor com os valores do objetivo
J(k) = sol.value(obj);
% Vetor com os valores da entrada
us(:,k) = sol.value(U(:,1));
end

```

A resposta do sistema é apresentada na Figura 5.1, para um cenário em que o sistema inicia no ponto de operação  $h^* = 50\%$ ,  $u^* = 31,9438\%$  e  $q^* = 50\%$ . Em  $t = 10$  s, a referência muda para  $45\%$ , e em  $t = 50$  s a perturbação muda para  $45\%$ . Um aspecto interessante deste NMPC implementado é que o sistema não garante seguinte de referência ou rejeição de perturbação com erro nulo. Isto é devido ao modelo utilizado que não leva em conta uma ação integral do controlador e a estimação de perturbações não medidas. Isto pode ser remediado definindo o modelo de predição de maneira adequada, mas, por simplicidade, isto não será apresentado para o NMPC genérico.

**Nota 5.1.** O código utilizando CasADi apresentado no Exemplo 5.1 não está otimizado para diminuir o custo computacional da solução do problema de otimização, pois este é redefinido completamente a cada iteração. Isto foi

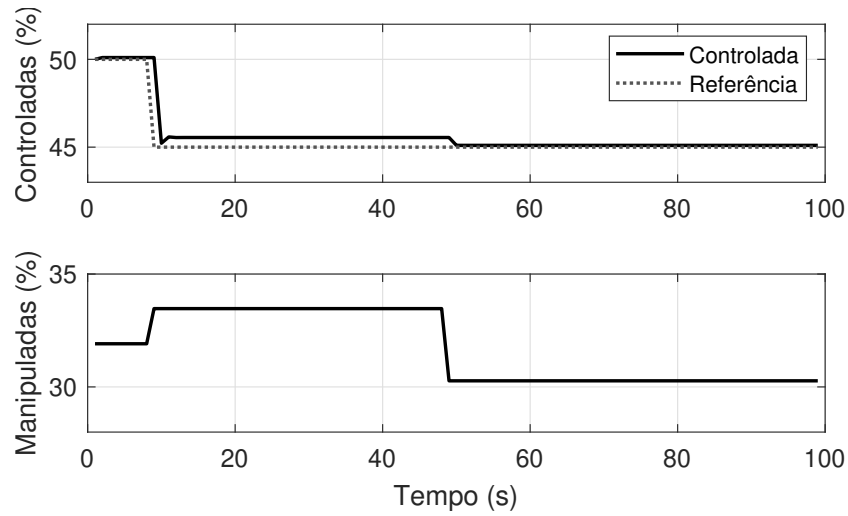


Figura 5.1: Exemplo 5.1 – resposta e sinal de controle para o controle de nível de um tanque utilizando NMPC.

*feito para simplificar a apresentação. Para diminuir o custo computacional, é possível definir o problema de otimização apenas uma única vez de forma paramétrica, ou seja, o problema de otimização é fixo, o que muda são os parâmetros que definem a condição inicial. Implementar desta maneira acelera de maneira considerável a execução do código.*

**Nota 5.2.** *Caso se deseje utilizar diretamente o modelo em equações diferenciais do modelo para o cálculo das predições, é possível usar diretamente no código a solução dessa equação de forma discreta. Ainda é possível escolher o método de integração numérica, considerando que em problemas complexos essa escolha pode ser fundamental para obter um resultado adequado. CasADi possui funções de integração e diferenciação numéricas, como Runge Kutta ou collocation. Mais informações podem ser obtidas diretamente na documentação do CasADi<sup>1</sup>.*

### 5.3 O algoritmo NEPSAC

O algoritmo NEPSAC é uma alternativa simples para a implementação de um controle NMPC de forma aproximada, usando conceitos similares aos do GPC em termos de modelo do processo e de perturbações. Além disso, ele usa a mesma função custo e a mesma formulação do problema de otimização.

<sup>1</sup><https://web.casadi.org/docs/>

Uma das principais ideias usadas no MPC linear é a decomposição das predições em resposta livre e forçada, sendo esta última dependente apenas dos controles futuros que se deseja calcular e que afetam a função custo. Embora isso não seja mais válido para modelos não lineares, dado que o princípio da superposição não pode ser aplicado nesses sistemas, o NEPSAC propõe manter o uso de uma decomposição da sequência de controle e da separação das predições em duas partes para encontrar a ação de controle ótima sem necessidade de recorrer a otimização não linear. Vamos iniciar estudando a forma de cálculo das predições.

### 5.3.1 Modelo de predição

No NEPSAC, que foi derivado da sua versão linear (EPSAC), o modelo do processo é dado por:

$$y(k) = x(k) + \eta(k),$$

sendo  $\eta(k)$  uma perturbação aditiva não mensurável e  $x(k)$  a saída do modelo sem as perturbações não medidas. Em geral, o modelo recursivo para  $x$  será:

$$x(k) = f\left(x(k-1), \dots, x(k-n_x), u(k-1), \dots, u(k-n_u), q(k), \dots, q(k-n_q)\right), \quad (5.4)$$

com  $f$  uma função não linear qualquer e  $n_x$ ,  $n_u$  e  $n_q$  as ordens que definem a dependência com os valores passados de  $x$ ,  $u$  e  $q$ , respectivamente. Neste modelo,  $q$  representa as perturbações medidas do processo. Na sequência, usaremos a notação compacta

$$x(k) = f(\mathbf{x}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k))$$

para representar a relação entre  $x(k)$  e os vetores

$$\begin{aligned} \mathbf{x}_p(k-1) &= [x(k-1) \ x(k-2) \ \dots \ x(k-n_x)]^T, \\ \mathbf{u}_p(k-1) &= [u(k-1) \ u(k-2) \ \dots \ u(k-n_u)]^T, \\ \mathbf{q}_p(k) &= [q(k) \ q(k-1) \ \dots \ q(k-n_q)]^T, \end{aligned}$$

que armazenam, respectivamente, os valores passados de  $x$ ,  $u$  e  $q$  (neste último caso considerando  $q(k)$  dentro desse vetor, já que no instante  $k$  se tem informação sobre a perturbação atual). Ressaltamos que este modelo pode ser de vários tipos, como uma equação em espaço de estados, um modelo polinomial recursivo ou uma rede neural, por exemplo.

O modelo das perturbações é o mesmo usado no GPC:

$$\eta(k) = \frac{C(z^{-1})}{D(z^{-1})}e(k), \quad (5.5)$$

com  $e(k)$  um ruído branco e  $C(z^{-1})$  e  $D(z^{-1})$  polinômios em  $z^{-1}$  com graus  $n_c$  e  $n_d$ , respectivamente. A estrutura dos polinômios  $C(z^{-1})$  e  $D(z^{-1})$  é a mesma usada no modelo de perturbações do GPC, com coeficientes  $c_i$  e  $d_i$  multiplicando as potências  $z^{-i}$ . Assim como no GPC,  $D(z^{-1})$  geralmente inclui um integrador, para representar as perturbações que têm valor estático não nulo. Observe que, para o caso linear, se o modelo do processo é dado por  $x(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k-1)$  e  $D(z^{-1}) = \Delta A(z^{-1})$  o modelo resultante para processo mais perturbações é o clássico CARIMA usado no GPC.

No caso não linear geral, temos duas alternativas para compor os dois modelos, do processo e das perturbações. Um consiste em encontrar primeiro as predições ótimas da perturbação  $e$ , posteriormente, adicionar os valores calculados às predições dadas pelo modelo não linear do processo. A segunda forma é escrever o modelo completo do processo mais perturbações e calcular as predições de forma recursiva, considerando também as características definidas para a perturbação. Em ambos os casos, a hipótese usada é que o ruído é branco e que seu valor esperado no futuro é nulo ( $e(k+j | k) = 0$  para  $j > 0$ ). Analisaremos as duas alternativas na sequência.

No primeiro método, a predição de  $\eta$  pode ser calculada a partir do seguinte procedimento. Primeiro, observamos que os valores de  $\eta(k-j)$  são todos calculados usando  $\eta(k-j) = y(k-j) - x(k-j)$  para  $j \leq 0$ , isto é, no passado, a perturbação é simplesmente a diferença entre o valor da saída do processo,  $y$ , e do modelo,  $x$ . Logo, como:

$$\eta(k-j) = \frac{C(z^{-1})}{D(z^{-1})}e(k-j),$$

podemos escrever

$$e(k-j) = \frac{D(z^{-1})}{C(z^{-1})}\eta(k-j)$$

e, recursivamente, calcular os valores passados de  $e(k-j)$  que serão necessários para iniciar o procedimento recursivo de cálculo das predições. Para os valores futuros ou predições da perturbação, escrevemos de forma recursiva o modelo de  $\eta(k+j)$  como:

$$\begin{aligned} \eta(k+j | k) = & -d_1\eta(k+j-1 | k) - d_2\eta(k+j-2 | k) - \dots \\ & + e(k+j | k) + c_1e(k+j-1 | k) + \dots \end{aligned}$$



e, como temos conhecimento dos valores passados de  $\eta$  e  $e$ , podemos achar os valores de  $\eta(k+j | k)$  para o todo o horizonte, de  $j = N_1$  a  $j = N_2$ .

Um caso particular que vale a pena considerar é o de  $D(z^{-1}) = \Delta$ , que é muito usado na prática para considerar as perturbações constantes no modelo. Neste caso, a equação resulta em:

$$\begin{aligned}\eta(k+j | k) = & \eta(k+j-1 | k) + e(k+j | k) + c_1 e(k+j-1 | k) \\ & + c_2 e(k+j-2 | k) + \dots + c_{n_c} e(k+j-n_c | k)\end{aligned}$$

Conhecidos os valores futuros estimados da perturbação não medida  $\eta$ , podemos usar duas formas de adicionar esses valores no modelo da saída:

- simplesmente adicionando os valores de  $\eta(k+j | k)$  aos valores de  $x(k+j | k)$  calculados com o modelo não linear usando como valores anteriores os  $x$  calculados sem considerar a perturbação. Este modelo é chamado de modelo paralelo e é similar ao procedimento usado no DMC, no qual a perturbação é adicionada à resposta ao degrau. Note que este modelo somente pode ser usado com processos estáveis. O modelo de cálculo das predições da saída neste caso seria:

$$\hat{y}(k+j+1 | k) = f(\mathbf{x}_p(k+j | k), \mathbf{u}_p(k+j), \mathbf{q}_p(k+j+1)) + \eta(k+j+1 | k).$$

Sendo que

$$\mathbf{x}_p(k+j | k) = [x(k+j | k) \ x(k+j-1 | k) \ \dots \ x(k+j-n_x | k)]^T;$$

- calculando os valores de  $\hat{y}(k+j | k)$  de forma recursiva, de forma que o valor de  $\hat{\mathbf{y}}_p(k+j | k)$  é usado no lugar de  $\mathbf{x}_p(k+j | k)$  para o cálculo de  $x(k+j+1 | k)$  em cada passo. Este modelo é chamado de série-paralelo, e usa a mesma ideia do GPC e pode ser usado com todo tipo de processo, estável ou instável. O modelo de cálculo neste caso seria:

$$\hat{y}(k+j+1 | k) = f(\hat{\mathbf{y}}_p(k+j | k), \mathbf{u}_p(k+j), \mathbf{q}_p(k+j+1)) + \eta(k+j+1 | k)$$

Com

$$\hat{\mathbf{y}}_p(k+j | k) = [\hat{y}(k+j | k) \ \hat{y}(k+j-1 | k) \ \dots \ \hat{y}(k+j-n_x | k)]^T,$$

e considerando  $\hat{y}(k+j | k) = y(k+j)$ , se  $j \leq 0$ .

Na sequência, apresentamos um exemplo de cálculo das predições usando este método, nas duas formulações.

**Exemplo 5.2.** Vamos considerar o mesmo modelo de controle de nível  $h(k)$  de um tanque usado no caso do NMPC direto mas, agora, com acionamento pela abertura da válvula de entrada  $a(k)$  e considerando a abertura da válvula de saída  $q(k)$  como uma perturbação mensurável. Usaremos os seguintes horizontes  $N_1 = 1, N_2 = N_u = 2$ . O modelo sem as perturbações não medidas é:

$$h(k) = h(k-1) + K_1 a(k-1) - K_2 q(k-1) \sqrt{2gh(k-1)},$$

com  $K_1$  e  $K_2$  constantes reais. A saída medida é:

$$y(k) = h(k) + \eta(k).$$

As perturbações não medidas são modeladas por:

$$\eta(k) = \frac{1 + 0,5z^{-1}}{1 - z^{-1}} e(k).$$

Usando o procedimento explicado, primeiro calculamos os valores de  $e$  para inicializar o cálculo das predições de  $\eta$ . Observe que precisaremos do valor de  $e(k)$  no primeiro passo de cálculo recursivo (para calcular  $\eta(k+1 | k)$ ). Usando o modelo, temos:

$$e(k) = \frac{1 - z^{-1}}{1 + 0,5z^{-1}} \eta(k),$$

e assim obtemos:

$$e(k) + 0,5e(k-1) = \eta(k) - \eta(k-1),$$

$$e(k) = -0,5e(k-1) + \eta(k) - \eta(k-1),$$

que permite calcular  $e(k)$  a partir de

$$\eta(k) = y(k) - h(k)$$

e

$$\eta(k-1) = y(k-1) - h(k-1).$$

Inicializando  $e(k-1) = 0$ :

$$e(k) = \eta(k) - \eta(k-1).$$

A predição  $\eta(k+1 | k)$  pode ser obtida sabendo que  $e(k) = \eta(k) - \eta(k-1)$ , como:

$$\begin{aligned} \eta(k+1 | k) &= \eta(k) + e(k) + 0,5e(k-1) \\ &= \eta(k) + \eta(k) - \eta(k-1) \\ &= 2\eta(k) - \eta(k-1), \end{aligned}$$

e a predição  $\eta(k+2 | k)$  é obtida substituindo o valor de  $\eta(k+1 | k)$  já calculado, e sabendo que  $e(k+1 | k) = 0$ :

$$\begin{aligned}\eta(k+2 | k) &= \eta(k+1 | k) + e(k+1 | k) + 0,5e(k) \\ &= 2,5\eta(k) - 1,5\eta(k-1).\end{aligned}$$

Vamos, agora, considerar o modelo paralelo para as predições de  $y$ :

$$\hat{y}(k+1 | k) = \hat{h}(k+1 | k) + \eta(k+1 | k).$$

Neste caso, como o sistema é de primeira ordem, basta conhecer o valor de  $h(k)$  para o primeiro passo ( $\mathbf{h}_p(k) = h(k)$ ):

$$\hat{y}(k+1 | k) = f(h(k), \mathbf{u}_p(k), \mathbf{q}_p(k+1)) + 2\eta(k) - \eta(k-1).$$

A predição seguinte é:

$$\begin{aligned}\hat{y}(k+2 | k) &= f(h(k+1 | k), \mathbf{u}_p(k+1), \mathbf{q}_p(k+2)) + \eta(k+2 | k) \\ &= f(h(k+1 | k), \mathbf{u}_p(k+1), \mathbf{q}_p(k+2)) + 2,5\eta(k) - 1,5\eta(k-1).\end{aligned}$$

Usando o modelo de  $h(k)$  do tanque temos:

$$\begin{aligned}\hat{y}(k+1 | k) &= h(k) + K_1a(k) - K_2q(k)\sqrt{2gh(k)} + 2\eta(k) - \eta(k-1), \\ \hat{y}(k+2 | k) &= h(k+1 | k) + K_1a(k+1) - K_2q(k+1)\sqrt{2gh(k+1 | k)} \\ &\quad + 2,5\eta(k) - 1,5\eta(k-1), \\ \hat{y}(k+2 | k) &= h(k) + K_1a(k) - K_2q(k)\sqrt{2gh(k)} + K_1a(k+1) \\ &\quad - K_2q(k+1)\sqrt{2g(h(k) + K_1a(k) - K_2q(k)\sqrt{2gh(k)})} \\ &\quad + 2,5\eta(k) - 1,5\eta(k-1).\end{aligned}$$

Se usamos o modelo série-paralelo, que também é de primeira ordem, obtemos:

$$\begin{aligned}\hat{y}(k+1 | k) &= f(y(k), \mathbf{u}_p(k), \mathbf{q}_p(k+1)) + 2\eta(k) - \eta(k-1), \\ \hat{y}(k+2 | k) &= f(\hat{y}(k+1 | k), \mathbf{u}_p(k+1), \mathbf{q}_p(k+2)) + 2,5\eta(k) - 1,5\eta(k-1),\end{aligned}$$

que, para o caso do tanque, resulta em:

$$\begin{aligned}\hat{y}(k+1 | k) &= y(k) + K_1a(k) - K_2q(k)\sqrt{2gy(k)} + 2\eta(k) - \eta(k-1), \\ \hat{y}(k+2 | k) &= \hat{y}(k+1 | k) + K_1a(k+1) - K_2q(k+1)\sqrt{2g\hat{y}(k+1 | k)} \\ &\quad + 2,5\eta(k) - 1,5\eta(k-1), \\ \hat{y}(k+2 | k) &= y(k) + K_1a(k) - K_2q(k)\sqrt{2gy(k)} + K_1a(k+1) \\ &\quad - K_2q(k+1)\sqrt{2g(y(k) + K_1a(k) - K_2q(k)\sqrt{2gy(k)} + 2\eta(k) - \eta(k-1))} \\ &\quad + 2,5\eta(k) - 1,5\eta(k-1).\end{aligned}$$

No segundo método, juntamos os modelos propostos para a perturbação e processo na configuração série-paralelo, e depois realizamos o cálculo das predições, como é o procedimento do GPC, mas sem uso de equações diofantinas, apenas usando recursividade e a condição de  $e$  ser um ruído branco. Considerando, por simplicidade nesta explicação, que  $D(z^{-1}) = \Delta$ , temos:

$$y(k) = f(\mathbf{x}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) + \frac{C(z^{-1})}{(1-z^{-1})}e(k),$$

e usando a ideia do modelo série-paralelo com  $\mathbf{y}_p(k-1)$  na função  $f$ :

$$y(k) = f(\mathbf{y}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) + \frac{C(z^{-1})}{(1-z^{-1})}e(k),$$

o que leva a calcular:

$$(1-z^{-1})y(k) = (1-z^{-1})f(\mathbf{y}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) + C(z^{-1})e(k),$$

ou de forma equivalente:

$$\begin{aligned} y(k) = & y(k-1) + f(\mathbf{y}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) \\ & - f(\mathbf{y}_p(k-2), \mathbf{u}_p(k-2), \mathbf{q}_p(k-1)) + C(z^{-1})e(k). \end{aligned}$$

As predições, então, são calculadas passo a passo:

$$\begin{aligned} y(k+1) = & y(k) + f(\mathbf{y}_p(k), \mathbf{u}_p(k), \mathbf{q}_p(k+1)) - f(\mathbf{y}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) \\ & + e(k+1) + c_1e(k) + \dots + c_{n_c}e(k-n_c) \end{aligned}$$

e considerando as predições e que  $e(k+1 | k) = 0$ . Assim:

$$\begin{aligned} \hat{y}(k+1 | k) = & y(k) + f(\mathbf{y}_p(k), \mathbf{u}_p(k), \mathbf{q}_p(k+1)) - f(\mathbf{y}_p(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) \\ & + c_1e(k) + \dots + c_{n_c}e(k-n_c), \end{aligned}$$

que pode ser calculada com as medições das variáveis disponíveis e considerando os  $e(k-i)$  já calculados usando o modelo das perturbações não medidas. Note que nesta equação (e nas seguintes), se os valores das perturbações medidas futuras são desconhecidos, temos  $q(k+j) = q(k)$  para todo  $j$  no horizonte.

No próximo passo, a predição em  $k+2$  é dada por:

$$\begin{aligned} y(k+2) = & y(k+1) + f(\mathbf{x}_p(k+1), \mathbf{u}_p(k+1), \mathbf{q}_p(k+2)) \\ & - f(\mathbf{x}_p(k), \mathbf{u}_p(k), \mathbf{q}_p(k+1)) \\ & + c_1e(k+1) + \dots + c_{n_c}e(k-n_c+1), \end{aligned}$$

e usamos os valores das predições anteriormente calculadas para obter a predição em  $k + 2$ :

$$\begin{aligned}\hat{y}(k + 2 | k) = & \hat{y}(k + 1 | k) + f(\hat{\mathbf{x}}_p(k + 1 | k), \mathbf{u}_p(k + 1), \mathbf{q}_p(k + 2)) \\ & - f(\mathbf{x}_p(k), \mathbf{u}_p(k), \mathbf{q}_p(k + 1)) \\ & + c_2 e(k) + \dots + c_{n_c} e(k - n_c + 1),\end{aligned}$$

Este procedimento pode ser realizado para todo o horizonte de predição. Antes de prosseguir com a análise teórica, vamos ilustrar com um exemplo esta forma de cálculo das predições para o caso do tanque do exemplo anterior.

**Exemplo 5.3.** *Para o modelo série-paralelo de controle de nível do tanque do Exemplo 5.2, com o mesmo modelo de perturbações não medidas, obtemos:*

$$y(k) = y(k - 1) + K_1 a(k - 1) - K_2 q(k - 1) \sqrt{2gy(k - 1)} + \frac{1 + 0,5z^{-1}}{1 - z^{-1}} e(k),$$

ou, de forma equivalente:

$$\begin{aligned}y(k)(1 - z^{-1}) = & y(k - 1)(1 - z^{-1}) + K_1 a(k - 1)(1 - z^{-1}) \\ & - (1 - z^{-1})[K_2 q(k - 1) \sqrt{2gy(k - 1)}] + (1 + 0,5z^{-1})e(k), \\ y(k) = & 2y(k - 1) - y(k - 2) + K_1 \Delta a(k - 1) - K_2 q(k - 1) \sqrt{2gy(k - 1)} \\ & + K_2 q(k - 2) \sqrt{2gy(k - 2)} + e(k) + 0,5e(k - 1),\end{aligned}$$

para a qual introduzimos a variável incremento de abertura,  $\Delta a$ .

Vamos calcular as predições nos primeiros dois elementos do horizonte, considerando  $N_1 = 1, N_2 = N_u = 2$ . Para  $j = 1$ , temos:

$$\begin{aligned}\hat{y}(k + 1 | k) = & 2y(k) - y(k - 1) + K_1 \Delta a(k) - K_2 q(k) \sqrt{2gy(k)} \\ & + K_2 q(k - 1) \sqrt{2gy(k - 1)} + e(k + 1 | k) + 0,5e(k),\end{aligned}$$

e como  $e(k + 1 | k) = 0$  e  $e(k)$  foi calculado ( $e(k) = \eta(k) - \eta(k - 1)$ ):

$$\begin{aligned}\hat{y}(k + 1 | k) = & 2y(k) - y(k - 1) + K_1 \Delta a(k) - K_2 q(k) \sqrt{2gy(k)} \\ & + K_2 q(k - 1) \sqrt{2gy(k - 1)} + 0,5(\eta(k) - \eta(k - 1)).\end{aligned}$$

Para  $j = 2$ , temos:

$$\begin{aligned}\hat{y}(k + 2 | k) = & 2\hat{y}(k + 1 | k) - y(k) + K_1 \Delta a(k + 1) \\ & - K_2 q(k + 1) \sqrt{2g\hat{y}(k + 1 | k)} \\ & + K_2 q(k) \sqrt{2gy(k)} + \hat{e}(k + 2 | k) + 0,5e(k + 1 | k),\end{aligned}$$

e, como  $e(k + j | k) = 0$  para  $j > 0$ :

$$\begin{aligned}\hat{y}(k + 2 | k) = & 2\hat{y}(k + 1 | k) - y(k) \\ & + K_1\Delta a(k + 1) - K_2q(k + 1)\sqrt{2g\hat{y}(k + 1 | k)} \\ & + K_2q(k)\sqrt{2gy(k)},\end{aligned}$$

que ainda pode ser reduzida substituindo o valor de  $\hat{y}(k + 1 | k)$  já calculado. Entretanto, como não temos aqui intenções de realizar um cálculo analítico das predições, mas apenas mostrar a forma de cálculo recursiva, não precisamos realizar esta substituição.

É importante observar que, para o cálculo das predições de  $y$ , usamos, como esperado, valores passados e futuros de  $a$  e  $q$ , assim como valores passados de  $y$ .

A partir da análise realizada e dos exemplos observamos que, nas equações que calculam as predições, temos uma dependência nos valores dos controles futuros que ainda não calculamos, assim como ocorre em outros métodos de controle preditivo. Ainda, observamos que, mesmo para casos simples, como o do tanque, não é trivial separar a parte da predição que depende do controle futuro, pois as expressões obtidas são complexas. Mas, como veremos na sequência, o NEPSAC propõe uma forma simples para separar os termos da predição que dependem de valores passados ou valores futuros do controle, que vão permitir usar os procedimentos já vistos no GPC para o cálculo do controle ótimo.

### 5.3.2 Decomposição da sequência de controle

Considera-se que a sequência de controle  $u(k + j)$  a ser calculada pelo algoritmo é composta por duas partes:

$$u(k + j) = u_b(k + j) + \delta u(k + j), \quad j = 0, \dots, N_u - 1,$$

sendo  $u_b(k + j)$  o chamado controle base, e  $\delta u(k + j)$  o incremento de controle com relação ao controle base (pulsos de amplitude  $\delta u(k + j)$  aplicados em cada  $j$ ) e  $N_u$  é o horizonte de controle. Na Figura 5.2, esta decomposição da sequência de controle ótimo é apresentada para um exemplo com horizonte de controle  $N_u = 4$ . O controle base pode ser escolhido de diferentes maneiras, como será discutido nesta seção, e a tarefa do controlador é encontrar os valores dos incrementos de controle dentro do horizonte que levam à resposta ótima.

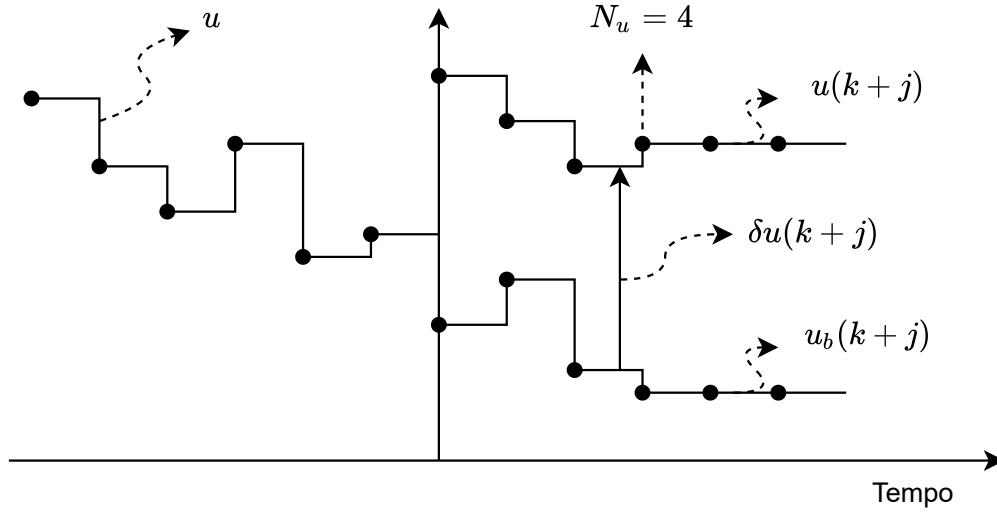


Figura 5.2: Cálculo de  $u_b$  e  $\delta u$  no algoritmo NEPSAC.

A saída futura do modelo é calculada como a soma da resposta base do modelo,  $y_b$ , que corresponde ao efeito de  $u_b$  e das perturbações, e  $y_{ot}$ , que corresponde apenas ao efeito de  $\delta u$ :

$$y(k+j) = y_b(k+j) + y_{ot}(k+j), \quad j = N_1, \dots, N_2.$$

Se o modelo é linear, claramente esta forma de separar a resposta futura é válida, e qualquer escolha de  $u_b$  pode ser usada. Note que, em particular, se  $u_b(k+j) = u(k-1)$  para todo  $j$ , temos a mesma formulação do GPC, e  $y_b$  corresponde à resposta livre do sistema. Neste caso  $\delta u(k) = \Delta u(k)$ ,  $\delta u(k+1) = \Delta u(k) + \Delta u(k+1)$  e  $\delta u(k+j) = \Delta u(k) + \Delta u(k+1) + \dots + \Delta u(k+j)$  para  $j > 1$ .

No caso não linear, a saída  $y_b$  (que depende apenas de valores de controle conhecidos e do modelo das perturbações) é associada ao modelo não linear e  $y_{ot}$  a um modelo linear que é determinado a partir da resposta do modelo não linear em cada passo.

Considerando que um modelo linear pode descrever a relação entre os valores de  $y_{ot}$  no horizonte de predição e os valores de  $\delta u$  no horizonte de controle, é possível escrever (de forma similar ao caso do GPC):

$$\hat{\mathbf{y}}_{ot} = \mathbf{G}_e \delta \mathbf{u}, \quad (5.6)$$

sendo  $\hat{\mathbf{y}}_{ot}$  e  $\delta \mathbf{u}$  dados por:

$$\hat{\mathbf{y}}_{ot} = \begin{bmatrix} \hat{y}_{ot}(k + N_1 | k) \\ \hat{y}_{ot}(k + N_1 + 1 | k) \\ \vdots \\ \hat{y}_{ot}(k + N_2 | k) \end{bmatrix}, \quad \delta \mathbf{u} = \begin{bmatrix} \delta u(k) \\ \delta u(k + 1) \\ \vdots \\ \delta u(k + N_u - 1) \end{bmatrix}$$

Para calcular  $\mathbf{G}_e$ , que tem dimensão  $N \times N_u$ , usando o modelo não linear, é importante notar que  $\hat{\mathbf{y}}_{ot}$  pode ser considerado como o resultado acumulativo de uma série de *impulsos* na entrada ao longo do horizonte de controle  $N_u$  e de uma entrada do tipo *degrau* a partir do tempo  $k + N_u$  (observe a forma do sinal de controle mostrado na Figura 5.2).

Portanto,  $\mathbf{G}_e$  tem a forma:

$$\mathbf{G}_e = \begin{bmatrix} h_{N_1} & h_{N_1-1} & \dots & h_{N_1-N_u+2} & | & g_{N_1-N_u+1} \\ h_{N_1+1} & h_{N_1} & \dots & h_{N_1-N_u+3} & | & g_{N_1-N_u+2} \\ \dots & \dots & \dots & \dots & | & \dots \\ h_{N_2} & h_{N_2-1} & \dots & h_{N_2-N_u+2} & | & g_{N_2-N_u+1} \end{bmatrix}$$

em que  $h_i$  ( $i = N_1, \dots, N_2$ ) são os coeficientes da resposta ao impulso unitário e os parâmetros  $g_i$  ( $i = N_1 - N_u + 1, \dots, N_2 - N_u + 1$ ) são os coeficientes da resposta ao degrau unitário. Valores de  $h_i$  e  $g_i$  para  $i \leq 0$  são nulos. Os coeficientes da resposta ao impulso ou degrau podem ser determinados simplesmente simulando o modelo não linear com entradas de tipo impulso ou degrau e determinando a variação da saída do modelo com relação ao valor de  $y_b$ . Este não é um processo de linearização convencional, pois não é necessário ter acesso às equações do modelo para realizar o processo, mas o modelo resultante que descreve a relação entre  $\delta u$  e  $y_{ot}$  é linear.

Observe que a matriz  $\mathbf{G}_e$  pode ser expressa totalmente em função de  $h_i$  ou totalmente em função de  $g_i$  por meio da relação  $h_i = g_i - g_{i-1}$ . **É importante saber que o impulso ou degrau aplicado no processo não precisa ter amplitude unitária. Como o sistema é não linear, amplitudes diferentes podem resultar em coeficientes diferentes, mesmo depois de ser feita a correção para que eles representem o equivalente à resposta ao impulso (ou degrau) unitário. O valor a ser utilizado depende da faixa de trabalho dos sinais de entrada e é importante obter coeficientes que sejam representativos do que se espera que aconteça com o processo.**

Se esta modelagem for aceita como válida, observamos que as predições são escritas como:

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_b + \mathbf{G}_e \delta \mathbf{u},$$



com:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k + N_1 | k) \\ \hat{y}(k + N_1 + 1 | k) \\ \vdots \\ \hat{y}(k + N_2 | k) \end{bmatrix}, \quad \hat{\mathbf{y}}_b = \begin{bmatrix} \hat{y}_b(k + N_1 | k) \\ \hat{y}_b(k + N_1 + 1 | k) \\ \vdots \\ \hat{y}_b(k + N_2 | k) \end{bmatrix},$$

que tem uma expressão similar àquela usada no GPC com a resposta livre  $\mathbf{f}$  e a matriz dinâmica  $\mathbf{G}$ . Assim como naquele caso, o problema de otimização do NEPSAC consiste em calcular o vetor  $\delta\mathbf{u}$  que minimiza a função custo. Para isso, basta escrever o custo em função do vetor  $\delta\mathbf{u}$  e transformar o problema em um QP a cada amostra. Dessa forma, o problema aqui estudado seria equivalente ao do GPC apenas trocando  $\hat{\mathbf{y}}_b$  pela resposta livre  $\mathbf{f}$ , e o vetor de incrementos  $\Delta\mathbf{u}$  pelo novo  $\delta\mathbf{u}$ . Observe que se a função custo for a mesma do GPC, que depende do vetor de incrementos de controle  $\Delta\mathbf{u}$ , é necessário escrever  $\Delta\mathbf{u}$  como função de  $\delta\mathbf{u}$  considerando que, para cada  $j$ ,  $\delta u(k + j) = u(k - 1) + \Delta u(k) + \Delta u(k + 1) + \dots + \Delta u(k + j) - u_b(k + j)$ . Usando a expressão anterior para todo o horizonte temos:

$$\begin{aligned} \begin{bmatrix} \delta u(k) \\ \delta u(k + 1) \\ \vdots \\ \delta u(k + N_u - 1) \end{bmatrix} &= \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u(k - 1) + \mathbf{T} \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \\ \vdots \\ \Delta u(k + N_u - 1) \end{bmatrix} \\ &\quad - \begin{bmatrix} u_b(k) \\ u_b(k + 1) \\ \vdots \\ u_b(k + N_u - 1) \end{bmatrix} \\ &= \mathbf{1}u(k - 1) + \mathbf{T}\Delta\mathbf{u} - \mathbf{u}_b \end{aligned}$$

Assim, podemos escrever:

$$\Delta\mathbf{u} = \mathbf{T}^{-1}[\delta\mathbf{u} - \mathbf{1}u(k - 1) + \mathbf{u}_b]$$

e substituir esta expressão na função custo  $J$  para deixá-la em função de  $\delta\mathbf{u}$ . Partindo de  $J$ :

$$J = (\mathbf{w} - \hat{\mathbf{y}})^T \mathbf{Q}_y (\mathbf{w} - \hat{\mathbf{y}}) + \Delta\mathbf{u}^T \mathbf{Q}_u \Delta\mathbf{u},$$

e usando as expressões de  $\hat{\mathbf{y}}$  e  $\Delta\mathbf{u}$  temos:

$$\begin{aligned} J &= (\mathbf{w} - \mathbf{G}_e \delta\mathbf{u} - \hat{\mathbf{y}}_b)^T \mathbf{Q}_y (\mathbf{w} - \mathbf{G}_e \delta\mathbf{u} - \hat{\mathbf{y}}_b) \\ &\quad + (\mathbf{T}^{-1}[\delta\mathbf{u} - \mathbf{1}u(k - 1) + \mathbf{u}_b])^T \mathbf{Q}_u (\mathbf{T}^{-1}[\delta\mathbf{u} - \mathbf{1}u(k - 1) + \mathbf{u}_b]). \end{aligned}$$

Essa expressão de  $J$  pode ser manipulada para uma forma quadrática padrão, igual à usada no GPC e DMC. Da mesma forma, as relações do controle e a saída com a resposta base e com os  $\delta u$  devem ser usadas nas equações que definem as restrições do problema para chegar ao problema QP tradicional (o Exercício 2 da lista trata desse problema).

Porém, é claro que esta solução, que calcula as previsões separando-as em duas partes, não é válida para o caso não linear, dado que a não linearidade do modelo invalida o princípio de superposição. Uma forma de torná-la uma boa aproximação é escolher a sequência  $u_b(k+j)$  de tal forma que seja muito próxima da solução ótima  $u(k+j)$  que estamos buscando, o que implica ter uma sequência  $\delta u(k+j) \approx 0$ . Desta forma, a solução seria quase ótima, pois a parcela linear adicionada seria muito pequena. A ideia central do NEPSAC é justamente conseguir essa condição usando um processo iterativo dentro de cada período de amostragem, seguindo os seguintes passos:

- escolher, no passo  $k$ , a sequência base como sendo igual ao controle ótimo encontrado no passo anterior:  $u_b(k+j) = u(k+j | k-1)$  para todo  $j$  no horizonte de controle, supondo que este esteja próximo da sequência de controle ótimo buscada;
- usando o modelo não linear, calcular  $\hat{\mathbf{y}}_b$  e a matriz  $\mathbf{G}_e$  correspondente, e resolver o problema QP associado para encontrar os incrementos  $\delta u(k+j)$ . Adicionar estes incrementos a  $u_b(k+j)$ , e calcular uma nova sequência  $u(k+j)$ , supondo que esta estará mais próxima da solução ótima que a  $u_b$  usada na iteração anterior;
- ainda estando no mesmo período de amostragem, repetir o procedimento usando como  $u_b(k+j)$  a nova  $u(k+j)$ , de tal forma que a cada iteração seja obtido uma solução do problema QP com uma sequência de incrementos  $\delta u(k+j)$  de amplitudes menores. O procedimento é interrompido ao atingir incrementos de controle  $\delta u(k+j)$  menores que um valor pré-especificado. Sendo os incrementos de controle quase iguais a zero,  $\hat{\mathbf{y}}_{ot}$  será quase zero e o resultado aproximado será quase igual ao ótimo.

Apesar de o procedimento proposto não ter prova de convergência, ele tem sido testado em muitas aplicações com sucesso. As condições de convergência do algoritmo são complexas de analisar e dependem das características da não linearidade do modelo, entradas e saídas passadas do processo, referência futura e perturbações.

Na sequência, vamos apresentar dois exemplos. No primeiro, usaremos o modelo do tanque para explicar, analiticamente, o procedimento para o

cálculo de  $\mathbf{G}_e$  e de  $\hat{\mathbf{y}}_b$ . No segundo, vamos ilustrar o uso do NEPSAC no exemplo do tanque via simulação.

**Exemplo 5.4.** *Considerando o modelo paralelo do tanque do Exemplo 5.2 com  $K_1 = 2$  e  $K_2 = 0,2$ , temos as seguintes equações.*

$$\begin{aligned}\hat{y}(k+1 | k) &= h(k) + 2a(k) - 0,2q(k)\sqrt{2gh(k)} + 2\eta(k) - \eta(k-1) \\ \hat{y}(k+2 | k) &= h(k) + 2a(k) - 0,2q(k)\sqrt{2gh(k)} + 2a(k+1) \\ &\quad - K_2q(k+1)\sqrt{2g(h(k) + 2a(k) - 0,2q(k)\sqrt{2gh(k)})} \\ &\quad + 2,5\eta(k) - 1,5\eta(k-1)\end{aligned}$$

Para este exemplo, vamos supor que o controle base é escolhido igual a  $u(k-1)$  nos dois elementos do horizonte de controle, assumindo que o sistema se encontra em regime permanente. Considere que no instante  $k$  medimos  $h(k) = h_0$ ,  $u(k-1) = u_0$  e  $q(k) = q_0$  (valores genéricos) e que não conhecemos os valores futuros de  $q$ . Ainda, suponha que  $\eta(k) = 0,1$  e  $\eta(k-1) = 0$ . Assim, todos os controles futuros  $u_b(k+j)$  serão iguais a  $u_0$ . Usaremos, então,  $a(k) = a(k+1) = u_0$  e  $q(k) = q(k+1) = q_0$  para o cálculo de  $\hat{y}_b(k+1 | k)$  e  $\hat{y}_b(k+2 | k)$  na primeira iteração, obtendo:

$$\begin{aligned}\hat{y}_b(k+1 | k) &= h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0} + 2(0,1), \\ \hat{y}_b(k+2 | k) &= h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0} + 2u_0 \\ &\quad - 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})} + 2,5(0,1).\end{aligned}$$

Assim temos:

$$\begin{aligned}\hat{y}_b(k+1 | k) &= h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0} + 0,2, \\ \hat{y}_b(k+2 | k) &= h_0 + 4u_0 - 0,2q_0\sqrt{2gh_0} \\ &\quad - 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})} + 0,25.\end{aligned}$$

Para o cálculo de  $\mathbf{G}_e$ , que neste caso é uma matriz  $2 \times 2$  (os horizontes são  $N_1 = 1$ ,  $N_2 = 2$  e  $N_u = 2$ ), precisamos calcular a resposta do tanque para um impulso unitário em  $a$  e guardar os dois primeiros coeficientes,  $h_1$  e  $h_2$ , para formar a primeira coluna de  $\mathbf{G}_e$ . A sequência de controle a usar aqui é  $a(k) = u_0 + 1$ ,  $a(k+1) = u_0$ . Posteriormente, aplicamos um degrau em  $a(k+1)$  para obter os coeficientes da resposta e compor a segunda coluna de  $\mathbf{G}_e$ . Neste caso, aplicamos  $a(k) = u_0$ ,  $a(k+1) = u_0 + 1$ . Desta resposta apenas precisamos de um coeficiente,  $g_1$ , dado que a matriz é neste caso triangular inferior (o elemento  $(2,1)$  é nulo). Devemos lembrar que estas

respostas são calculadas a partir da resposta base. Assim, os coeficientes são obtidos diminuindo o valor da resposta base dos valores obtidos na simulação. Para  $a(k) = u_0 + 1$ ,  $a(k + 1) = u_0$  temos:

$$\begin{aligned}\hat{y}_{ot}(k + 1 | k) &= h_0 + 2(u_0 + 1) - 0,2q_0\sqrt{2gh_0} + 0,2 - \hat{y}_b(k + 1 | k), \\ \hat{y}_{ot}(k + 1 | k) &= h_0 + 2(u_0 + 1) - 0,2q_0\sqrt{2gh_0} + 0,2 - h_0 - 2(u_0 + 1) \\ &\quad + 0,2q_0\sqrt{2gh_0} - 0,2 = 2.\end{aligned}$$

Assim,  $h_1 = 2$ . Para o segundo elemento temos:

$$\begin{aligned}\hat{y}_{ot}(k + 2 | k) &= h_0 + 2(u_0 + 1) - 0,2q_0\sqrt{2gh_0} + 2u_0 \\ &\quad - 0,2q_0\sqrt{2g(h_0 + 2(u_0 + 1) - 0,2q_0\sqrt{2gh_0})} \\ &\quad + 0,25 - \hat{y}_b(k + 2 | k). \\ \hat{y}_{ot}(k + 2 | k) &= h_0 + 2(u_0 + 1) - 0,2q_0\sqrt{2gh_0} + 2u_0 \\ &\quad - 0,2q_0\sqrt{2g(h_0 + 2(u_0 + 1) - 0,2q_0\sqrt{2gh_0})} \\ &\quad + 0,25 - h_0 - 2u_0 + 0,2q_0\sqrt{2gh_0} - 2u_0 \\ &\quad + 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})} - 0,25. \\ \hat{y}_{ot}(k + 2 | k) &= 2 - 0,2q_0\sqrt{2g(h_0 + 2(u_0 + 1) + 0,2q_0\sqrt{2gh_0})} \\ &\quad + 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})}.\end{aligned}$$

Assim:

$$\begin{aligned}h_2 &= 2 - 0,2q_0\sqrt{2g(h_0 + 2(u_0 + 1) + 0,2q_0\sqrt{2gh_0})} \\ &\quad + 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})}.\end{aligned}$$

Agora, calculamos apenas  $\hat{y}_{ot}(k + 2 | k)$  para a entrada degrau para obter  $g_1$ . Neste caso, com  $a(k) = u_0$  e  $a(k + 1) = u_0 + 1$ , temos:

$$\begin{aligned}\hat{y}_{ot}(k + 2 | k) &= h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0} + 2(u_0 + 1) \\ &\quad - 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})} + 0,25 - \hat{y}_b(k + 2 | k).\end{aligned}$$

Assim:

$$\begin{aligned}\hat{y}_{ot}(k + 2 | k) &= h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0} + 2(u_0 + 1) \\ &\quad - 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})} + 0,25 - h_0 - 2u_0 \\ &\quad + 0,2q_0\sqrt{2gh_0} - 2u_0 \\ &\quad + 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})} - 0,25 = 2\end{aligned}$$

Finalmente, a matriz resulta:

$$\mathbf{G}_e = \begin{bmatrix} 2 & 0 \\ h_2 & 2 \end{bmatrix},$$

com:

$$h_2 = 2 - 0,2q_0\sqrt{2g(h_0 + 2(u_0 + 1) + 0,2q_0\sqrt{2gh_0})} \\ + 0,2q_0\sqrt{2g(h_0 + 2u_0 - 0,2q_0\sqrt{2gh_0})}.$$

Com esta matriz, montamos o problema de otimização e calculamos  $\delta u(k)$  e  $\delta u(k+1)$  e, se não são menores que o valor especificado no ajuste do algoritmo, o procedimento é repetido fazendo  $a(k) = u_0 + \delta u(k)$  e  $a(k+1) = u_0 + \delta u(k+1)$  para serem usados como controle base. Calculamos novamente a saída base e a nova  $\mathbf{G}_e$  correspondente à segunda iteração deste passo de amostragem. Veja que os valores  $q_0$  e  $h_0$  não mudam, apenas o controle considerado como base para a nova iteração muda.

**Exemplo 5.5.** Neste exemplo vamos comparar o NEPSAC com duas versões do algoritmo GPC, uma que usa um modelo linear obtido a partir da linearização do sistema no ponto de equilíbrio, e outra com modelo misto, que calcula a resposta livre com o modelo não linear e a resposta forçada com a mesma matriz dinâmica  $\mathbf{G}$  da versão com modelo linear. Para este exemplo, considere  $g = 9,81$ ,  $K_1 = 2$  e  $K_2 = 0,1$ , e que o ponto de operação é dado por  $h^* = 50$ ,  $u^* = 31,3209$  e  $q^* = 20$ . Em todos os casos o algoritmo de controle é sintonizado com  $N_1 = 1$ ,  $N_2 = 30$ ,  $N_u = 5$ ,  $\lambda = 1$ , e a perturbação  $q$  é medida, porém,  $N_q = 0$ . No cenário de simulação proposto, usaremos diversos tipos de mudanças de referência de nível para movimentar o sistema em toda a faixa de atuação, e na amostra 50 da simulação a perturbação  $q$  assume o valor de 30.

Os resultados para todos os casos são apresentados na Figura 5.3. Note que o algoritmo do GPC linear possui respostas mais agressivas, mesmo considerando a mesma sintonia do que os outros algoritmos. Essa diferença fica mais evidente na resposta à última mudança de referência, que é mais oscilatória no caso do GPC linear, e na rejeição de perturbação, para a qual o GPC linear apresenta um transitório mais rápido e mais oscilatório que o dos outros controladores. Observe, ainda, que as diferenças mais notáveis estão na ação de controle do GPC linear no momento da rejeição da perturbação. Em parte, essas diferenças podem ser atribuídas ao erro de modelagem causado pela linearização do modelo. Veja que o GPC com modelo não linear e o NEPSAC possuem respostas muito similares neste problema em toda a simulação. Este é um aspecto importante a se considerar, pois nem sempre o uso de um algoritmo MPC não linear trará vantagens significativas para o

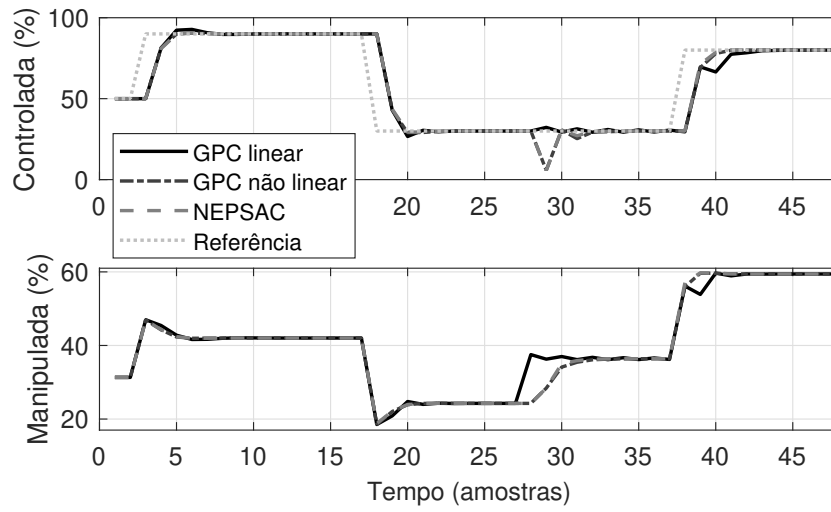


Figura 5.3: Exemplo 5.5 – comparação entre os algoritmos GPC linear, GPC com resposta livre não linear e NEPSAC para o controle de nível em um tanque.

*problema em questão. Então, o projetista deve, com o uso de simulações, avaliar se vale a pena a implementação de algoritmos mais complexos tal como o NEPSAC, que trazem um custo computacional maior. Neste exemplo em particular, o GPC não linear é uma opção bem interessante, considerando o compromisso entre desempenho e carga computacional.*

Vamos apresentar agora mais um exemplo, considerando um sistema com uma não linearidade mais forte, para ilustrar melhor as diferenças entre o controle linear e o não linear.

**Exemplo 5.6.** *Este exemplo considera o controle de concentração em um reator isotérmico continuamente agitado, onde ocorrem as reações de Van der Vusse. O reator é alimentado com o produto A e o produto B é o resultado da reação, como se mostra no esquema da Figura 5.4. Um analisador é usado em linha para obter o valor de concentração de B ( $C_b$ ). As equações que descrevem o comportamento das concentrações são:*

$$\begin{cases} \frac{dC_a(t)}{dt} = -k_1C_a(t) - k_3C_a(t)^2 + (C_{af}(t) - C_a(t))\frac{F(t)}{V} \\ \frac{dC_b(t)}{dt} = k_1C_a(t) - k_2C_b(t) - C_b(t)\frac{F(t)}{V} \end{cases}$$

$C_a$  ( $\text{mol L}^{-1}$ ) e  $C_{af}$  ( $\text{mol L}^{-1}$ ) são as concentrações de A dentro do reator e na alimentação, respectivamente. A variável manipulada é a taxa de diluição

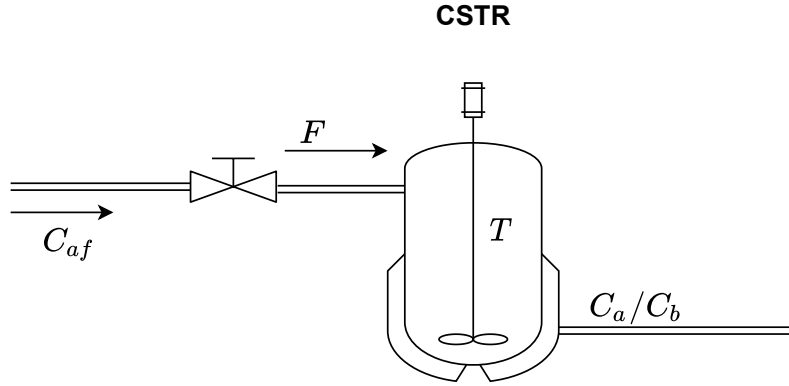


Figura 5.4: Exemplo 5.6 – reator com controle de concentração usando analisador em linha.

normalizada  $F$  ( $\text{L min}^{-1}$ ),  $V$  é o volume do reator ( $\text{L}$ ), e  $k_1 = 5/6 \text{ min}^{-1}$ ,  $k_2 = 5/3 \text{ min}^{-1}$ ,  $k_3 = 1/6 \text{ mol L}^{-1} \text{ min}^{-1}$  [Beq98]. O período de amostragem utilizado é  $T_s = 0,18 \text{ min}$ . Do ponto de vista de controle, consideraremos como variável manipulada a taxa de diluição normalizada  $F/V$ . Este sistema apresenta não linearidades consideráveis, em especial no ganho do processo. As características estáticas são apresentadas na Figura 5.5, onde podemos ver os pontos de equilíbrio tanto para  $C_a$  quanto para  $C_b$  para diferentes valores da manipulada  $F/V$  e da perturbação  $C_{af}$ . Veja que há até uma troca do sinal do ganho estático próximo de  $F/V = 0,8$  (de positivo para negativo), no entanto, a região de operação tende a ser com  $F/V$  em valores mais baixos, pois temos concentrações relativamente elevadas de  $C_b$  (e baixas de  $C_a$ ) a uma vazão mais baixa, otimizando a produção. Além disso, por conta da troca do sinal do ganho, aumentar a vazão além de um certo ponto faz a concentração de  $C_b$  diminuir, e não aumentar.

Assim como no exemplo anterior, vamos comparar o desempenho do GPC com o GPC não linear e o NEPSAC. O GPC linear foi calculado usando um modelo linearizado no meio da faixa de operação, com  $C_b = 0,8 \text{ mol L}^{-1}$ ,  $F/V = 0,2432 \text{ min}^{-1}$ ,  $C_{af} = 5,1 \text{ mol L}^{-1}$  e  $C_a = 1,032 \text{ mol L}^{-1}$ , resultando na seguinte função de transferência linearizada:

$$G(s) = \frac{1,6588(-0,3367s + 1)}{(0,7587s + 1)(0,9208s + 1)}.$$

Neste exemplo, para simplificar, consideraremos que a concentração de  $C_a$  dentro do reator é medida, porém, ressaltamos que esse não é o caso na prática, onde um modelo nominal entrada-saída seria utilizado, ou então um estimador de estados seria necessário em conjunto com o controlador. Assim,

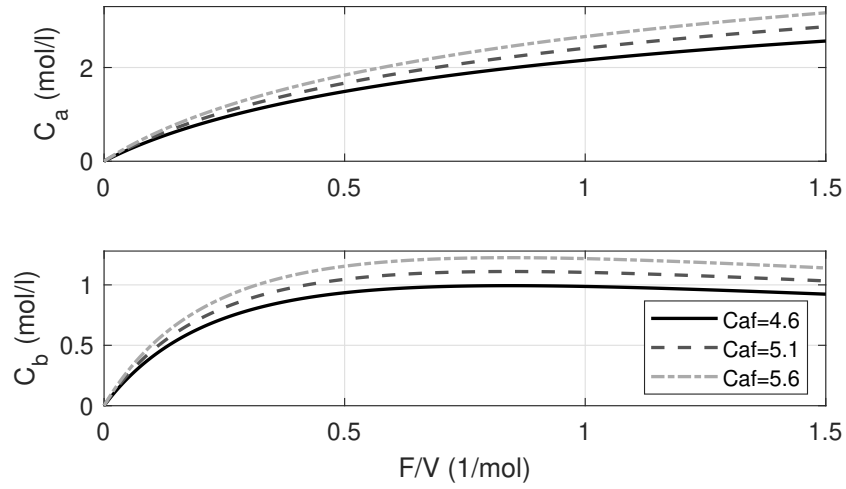


Figura 5.5: Exemplo 5.6 – características estáticas do CSTR isotérmico.

o modelo nominal do processo é obtido a partir da discretização por Euler do sistema de equações diferenciais que descreve o sistema, o que resulta em:

$$\begin{cases} C_a(k+1) = C_a(k) \\ \quad + T_s \left( -k_1 C_a(k) - k_3 C_a(k)^2 + (C_{af}(k) - C_a(k)) \frac{F(k)}{V} \right) \\ C_b(k+1) = C_b(k) \\ \quad + T_s \left( k_1 C_a(k) - k_2 C_b(k) - C_b(k) \frac{F(k)}{V} \right). \end{cases}$$

Em todos os casos, os algoritmos de controle são sintonizados com  $N_1 = 1$ ,  $N_2 = 40$ ,  $N_u = 5$ ,  $\lambda = 5$ , e a perturbação  $C_{af}$  não é medida. Para o cenário de simulação, inicialmente a referência de  $C_b$  está em  $0,9 \text{ mol L}^{-1}$ , que muda para  $0,6 \text{ mol L}^{-1}$  em  $t = 3,5 \text{ min}$  e para  $0,8 \text{ mol L}^{-1}$  em  $t = 14,6 \text{ min}$ . A perturbação  $C_{af}$  inicia em  $5,1 \text{ mol L}^{-1}$  e muda para  $4,6 \text{ mol L}^{-1}$  em  $t = 26,8 \text{ min}$ . O ponto inicial da simulação é  $u_0 = 0,3138$ ,  $C_{a0} = 1,235 \text{ mol L}^{-1}$  e  $C_{b0} = 0,9 \text{ mol L}^{-1}$ .

Os resultados das simulações são apresentados na Figura 5.6. Veja que todos os controladores conseguiram obter respostas similares, e mesmo um controlador mais simples como o GPC linear conseguiu bons resultados, apesar de ser mais agressivo em termos de sinal de controle na primeira mudança de referência. Assim, este exemplo mostra que nem sempre um controlador mais complexo resultará, necessariamente, em um desempenho melhor, e cabe ao projetista avaliar as vantagens práticas de um controlador mais complexo de acordo com o problema em questão.



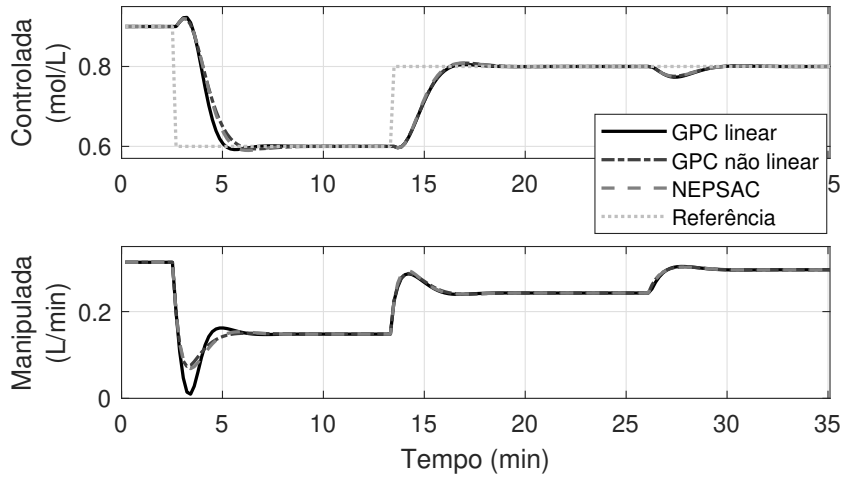


Figura 5.6: Exemplo 5.6 – concentração de saída e vazão de manipulação do reator para os três controladores aplicados.

### 5.3.3 Extensão para o caso multivariável

A extensão do NEPSAC para o caso multivariável (sistema com  $m$  entradas e  $n$  saídas) é direta, dado que podemos usar a mesma ideia empregada no GPC e DMC multivariáveis. Do ponto de vista da função custo e restrições, nada muda neste problema com modelos não lineares, e todo o desenvolvimento já estudado se aplica neste caso. Assim, teremos a possibilidade de escolher horizontes e ponderações de forma individualizada para cada variável e poderemos considerar restrições de vários tipos. Como o problema a ser resolvido em cada amostra é um QP (dada a formulação do NEPSAC), basta que modifiquemos o algoritmo de controle para encontrar a relação entre o modelo não linear do processo e as matrizes usadas na formulação do problema de otimização. Então, nesta seção, vamos detalhar a forma de calcular as previsões não lineares no caso multivariável.

Assim como no GPC e DMC multivariáveis, no NEPSAC vamos considerar que temos um modelo que permite calcular cada uma das saídas do processo em função das variáveis manipuladas e das perturbações que a afetam. Considerando o modelo completo, cada saída  $y_i(k)$  será representada por um modelo genérico da forma:

$$y_i(k) = f_i(\mathbf{y}_{p,i}(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) + \eta_i(k), \quad i = 1 \dots n,$$

onde  $\mathbf{y}_{p,i}(k-1)$ ,  $\mathbf{u}_p$  e  $\mathbf{q}_p$  são, respectivamente, os vetores contendo as informações passadas das  $n$  saídas, dos  $m$  controles e das  $m_q$  perturbações medidas que afetam a saída  $y_i$ , e  $\eta_i(k)$  é a perturbação não medida que afeta a mesma saída, com  $i = 1 \dots n$ .

Assim como no caso SISO, toda a formulação para decompor a ação de controle em duas partes se aplica a cada um dos sinais de controle  $u_l$ , separando-os em  $u_{b,l}$  e  $\delta u_l$  com  $l = 1 \dots m$ . Aplicando no sistema multivariável as entradas  $u_{b,l}$  e as perturbações medidas  $q_s$ , calculamos todas as  $y_{b,i}$  apenas simulando o modelo não linear multivariável no horizonte de predição de cada saída.

Por outro lado, para encontrar  $\mathbf{y}_{ot,i}$  usaremos a mesma metodologia do caso SISO para cada uma das entradas, obtendo uma matriz  $\mathbf{G}_{e,i,l}$  para modelar o efeito da entrada  $l$  na saída  $i$ . Dessa forma:

$$\hat{\mathbf{y}}_{ot,i} = \mathbf{G}_{e,i,1}\delta\mathbf{u}_1 + \mathbf{G}_{e,i,2}\delta\mathbf{u}_2 + \dots + \mathbf{G}_{e,i,m}\delta\mathbf{u}_m,$$

com:

$$\hat{\mathbf{y}}_{ot,i} = \begin{bmatrix} \hat{y}_{ot,i}(k + N_{1,i} | k) \\ \hat{y}_{ot,i}(k + N_{1,i} + 1 | k) \\ \vdots \\ \hat{y}_{ot,i}(k + N_{2,i} | k) \end{bmatrix}, \quad \delta\mathbf{u}_l = \begin{bmatrix} \delta u_l(k) \\ \delta u_l(k + 1) \\ \vdots \\ \delta u_l(k + N_{u,l} - 1) \end{bmatrix}$$

sendo  $N_{1,i}$  e  $N_{2,i}$  os instantes iniciais e finais dos horizontes da saída  $i$ , e  $N_{u,l}$  o horizonte de controle de  $u_l$ .

A forma de calcular cada  $\mathbf{G}_{e,i,l}$  segue o mesmo procedimento do caso SISO, excitando o modelo não linear com impulsos e degraus no controle  $l$  para obter os coeficientes de  $\mathbf{G}_{e,i,l}$ . Finalmente, a matriz completa terá  $n \times m$  blocos:

$$\mathbf{G}_e = \begin{bmatrix} \mathbf{G}_{e,1,1} & \cdots & \mathbf{G}_{e,1,m} \\ \mathbf{G}_{e,2,1} & \cdots & \mathbf{G}_{e,2,m} \\ \vdots & \ddots & \vdots \\ \mathbf{G}_{e,n,1} & \cdots & \mathbf{G}_{e,n,m} \end{bmatrix}$$

e o modelo de predição será dado por:

$$\hat{\mathbf{y}} = \mathbf{G}_e\delta\mathbf{u} + \hat{\mathbf{y}}_b, \tag{5.9}$$

com:

$$\hat{\mathbf{y}}_b = \begin{bmatrix} \hat{\mathbf{y}}_{b,1} \\ \hat{\mathbf{y}}_{b,2} \\ \vdots \\ \hat{\mathbf{y}}_{b,n} \end{bmatrix}, \quad \delta\mathbf{u} = \begin{bmatrix} \delta\mathbf{u}_1 \\ \delta\mathbf{u}_2 \\ \vdots \\ \delta\mathbf{u}_m \end{bmatrix}.$$

O procedimento iterativo usado no caso SISO para obter a melhor aproximação do controle ótimo é também usado aqui. Dentro de um período

de amostragem, o problema de otimização é resolvido repetidas vezes para calcular o vetor  $\delta \mathbf{u}$ , e em cada iteração ele é somado ao vetor de controle base  $\mathbf{u}_b$  para calcular uma nova  $\hat{\mathbf{y}}_b$  e uma nova  $\mathbf{G}_e$ , que permitem calcular um novo  $\delta \mathbf{u}$ . O procedimento para quando todos os elementos de  $\delta \mathbf{u}$  são, em módulo, menores que um limiar pré-estabelecido. Quando isso acontece, o sinal de controle calculado é aplicado ao processo e inicia-se um novo passo de amostragem.

Vamos ilustrar o uso do NEPSAC em um processo multivariável ainda neste capítulo, na Seção 5.5, mas antes apresentaremos o algoritmo PNMPC.

## 5.4 Practical Nonlinear Model Predictive Control - PNMPC

O algoritmo de controle preditivo não linear prático (PNMPC, do inglês *Practical Nonlinear Model Predictive Control*) foi proposto por Plucênio e colaboradores [PPBNR07, Plu10]. Esse algoritmo foi desenvolvido com o objetivo de apresentar uma forma simples, prática e eficiente de aplicar os conceitos de controle preditivo a sistemas não lineares, em especial, evitando múltiplas otimizações por iteração (como acontece com o NEPSAC) e, também, evitando otimizações não lineares complexas. Uma abordagem semelhante foi apresentada em [Law07], com foco no uso de redes neurais artificiais como modelos de predição.

Como já discutido, nos algoritmos lineares a predição do sistema pode ser dividida em resposta livre e forçada. No entanto, no caso de sistemas não lineares o princípio da superposição não se aplica e não é possível obter essa separação entre as respostas. No entanto, o PNMPC consegue fazer uma aproximação de uma forma engenhosa realizando uma linearização especial do modelo não linear do sistema em cada instante de amostragem.

### 5.4.1 Obtendo a resposta livre

Para o cálculo da resposta livre, o PNMPC considera exatamente o mesmo procedimento do NEPSAC série-paralelo apresentado na Seção 5.3.1. O modelo do processo para o caso SISO é dado por

$$y(k) = x(k) + \eta(k)$$

e as predições são dadas por

$$\hat{y}(k+j+1 | k) = f(\hat{\mathbf{y}}_p(k+j | k), \mathbf{u}_p(k+j), \mathbf{q}_p(k+j+1)) + \eta(k+j+1 | k), \quad (5.10)$$

com

$$\eta(k) = \frac{C(z^{-1})}{D(z^{-1})}e(k), \quad (5.11)$$

onde

$$\begin{aligned} \mathbf{u}_p(k-1) &= [u(k-1) \ u(k-2) \ \dots \ u(k-n_u)]^T, \\ \mathbf{q}_p(k) &= [q(k) \ q(k-1) \ \dots \ q(k-n_q)]^T, \\ \hat{\mathbf{y}}_p(k+j \mid k) &= [\hat{y}(k+j \mid k) \ \hat{y}(k+j-1 \mid k) \ \dots \ \hat{y}(k+j-n_x \mid k)]^T \end{aligned}$$

armazenam, respectivamente, os valores passados de  $u$ ,  $q$  e das predições, e considerando  $\hat{y}(k+j \mid k) = y(k+j)$ , se  $j \leq 0$ . Note que, para o cálculo das predições, os próprios valores das saídas preditas são utilizados de forma recursiva. O leitor pode consultar o Exemplo 5.3, que apresenta a forma de se calcular as predições utilizando o método série-paralelo. Lembrando que como a resposta livre é a resposta predita do sistema caso o sinal de controle não sofra mais alterações, então  $u(k+j) = u(k-1)$ ,  $\forall j \geq 0$  ou, de forma equivalente,  $\Delta u(k+j) = 0$ ,  $\forall j \geq 0$ .

### 5.4.2 Obtendo a resposta forçada

Para obter a resposta forçada, o PNMPC necessita das predições de malha aberta, isto é, as predições obtidas a partir de (5.10) sem considerar o efeito das perturbações não medidas, ou seja,  $\eta(k+j) = 0$ ,  $\forall j \geq 0$ . Por simplicidade, mas sem perda de generalidade, considere que  $n_u = 1$ ,  $n_x = 0$  e  $n_q = 0$ , então a predição de malha aberta em  $k+1$  se torna

$$\bar{y}(k+1 \mid k) = f(\bar{y}(k), u(k), q(k+1)). \quad (5.12)$$

Como  $u(k) = u(k-1) + \Delta u(k)$ , então a predição pode ser reescrita como

$$\begin{aligned} \bar{y}(k+1 \mid k) &= f(y(k), u(k-1) + \Delta u(k), q(k+1)) \\ \bar{y}(k+1 \mid k) &= f_0(y(k), u(k-1), \Delta u(k), q(k+1)), \end{aligned}$$

sendo que a função  $f_0(\cdot)$  é idêntica a  $f(\cdot)$ , com a exceção de que os parâmetros de entrada da função são separados. A necessidade da notação  $f_0$  se tornará clara a seguir. As predições de malha aberta futuras seguintes podem ser calculadas recursivamente com o uso de (5.12). Por exemplo, a predição em  $k+2$  é dada por:

$$\bar{y}(k+2 \mid k) = f(f_0(y(k), u(k-1), \Delta u(k), q(k+1)), u(k+1), q(k+2))$$

e, dado que  $u(k+1) = u(k-1) + \Delta u(k) + \Delta u(k+1)$ , então:

$$\bar{y}(k+2 \mid k) = f_1(y(k), u(k-1), \Delta u(k), \Delta u(k+1), q(k+1), q(k+2)),$$

sendo que  $f_1(\cdot)$  é resultado de uma composição de funções. Por inspeção, é fácil perceber que o vetor de predições de malha aberta futuras:

$$\bar{\mathbf{y}} = \begin{bmatrix} \bar{y}(k+1 | k) \\ \bar{y}(k+2 | k) \\ \vdots \\ \bar{y}(k+N_2 | k) \end{bmatrix},$$

é dado por

$$\bar{\mathbf{y}} = \begin{bmatrix} f_0(y(k), u(k-1), \Delta u(k), q(k+1)) \\ f_1(y(k), u(k-1), \Delta u(k), \Delta u(k+1), q(k+1), q(k+2)) \\ \vdots \\ f_{N_2-1}(y(k), u(k-1), \Delta u(k), \dots, \Delta u(k+N_2-1), q(k+1), \dots, q(k+N_2)) \end{bmatrix},$$

onde  $N_2$  é o horizonte final de predição e, por simplicidade,  $N_u = N_2$ . Para  $N_u < N_2$ , basta considerar os incrementos de controle nulos depois de  $N_u - 1$ . Também, se  $N_1 > 1$  apenas desconsideramos as primeiras  $N_1 - 1$  linhas de  $\bar{\mathbf{y}}$ .

Esta última equação pode ser rearranjada como:

$$\bar{\mathbf{y}} = \mathbf{F}(y(k), u(k-1), \Delta \mathbf{u}(k), \mathbf{q}(k+1)), \quad (5.13)$$

onde

$$\begin{aligned} \Delta \mathbf{u}(k) &= [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_2-1)]^T \\ \mathbf{q}(k+1) &= [q(k+1), q(k+2), \dots, q(k+N_2)]^T. \end{aligned}$$

A ideia geral do PNMPC é empregar o modelo não linear para obter a resposta livre e calcular uma relação aproximada linear da resposta forçada do sistema que permita obter um modelo similar ao do GPC ou DMC para relacionar os valores futuros do incremento de controle com os valores futuros da saída predita. Com isso, o problema de otimização, que busca definir os valores atuais e futuros do sinal de controle, é bastante simplificado e pode ser escrito na forma um QP, apesar de o algoritmo como um todo considerar pelo menos parte da não linearidade do modelo (para cálculo da resposta livre).

No PNMPC, a linearização do modelo é feita com base em uma série de Taylor de primeira ordem. É possível utilizar a série de Taylor de primeira ordem para aproximar uma função não linear em torno de um ponto de operação  $\mathbf{x}^*$  como

$$\mathbf{g}(\mathbf{x}^* + \delta \mathbf{x}) \cong \mathbf{g}(\mathbf{x}^*) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*} \delta \mathbf{x},$$

sendo que  $\delta \mathbf{x} = \mathbf{x} - \mathbf{x}^*$ , ou seja, uma variação de  $\mathbf{x}$  em torno do ponto de operação. Aplicando este conceito a (5.13),

$$\begin{aligned} \bar{\mathbf{y}} \cong & \mathbf{F}(\mathbf{x}^*) + \left. \frac{\partial \mathbf{F}}{\partial \Delta \mathbf{u}(k)} \right|_{\mathbf{x}=\mathbf{x}^*} \delta \Delta \mathbf{u}(k) + \left. \frac{\partial \mathbf{F}}{\partial \mathbf{q}(k+1)} \right|_{\mathbf{x}=\mathbf{x}^*} \delta \mathbf{q}(k+1) \\ & + \left. \frac{\partial \mathbf{F}}{\partial y(k)} \right|_{\mathbf{x}=\mathbf{x}^*} \delta y(k) + \left. \frac{\partial \mathbf{F}}{\partial u(k-1)} \right|_{\mathbf{x}=\mathbf{x}^*} \delta u(k-1), \end{aligned} \quad (5.14)$$

onde  $\mathbf{x} = [y(k), u(k-1), \Delta \mathbf{u}(k)^T, \mathbf{q}(k+1)^T]^T$ , e o ponto de operação é  $\mathbf{x}^* = [y(k), u(k-1), \mathbf{0}^T, \mathbf{q}(k+1)^T]^T$ , sendo que  $\mathbf{0}$  é um vetor de zeros de dimensão apropriada. Como buscamos uma relação entre a saída futura e a variação do controle, os três últimos termos de (5.14) são nulos, dado que  $y(k)$ ,  $u(k-1)$  e  $\mathbf{q}(k+1)$  são constantes e conhecidos no instante de amostragem  $k$ , e não é necessário saber como  $\mathbf{F}(\cdot)$  varia em função destas variáveis. Assim,  $\delta y(k) = \delta u(k-1) = 0$  e  $\mathbf{q}(k+1) = \mathbf{0}$ . E, como o ponto de operação de  $\Delta \mathbf{u}(k)$  é igual a  $\mathbf{0}$ , então  $\delta \Delta \mathbf{u}(k) = \Delta \mathbf{u}(k) - \mathbf{0} = \Delta \mathbf{u}(k)$ . Assim, podemos aproximar a equação que relaciona as predições com os incrementos de controle futuro como

$$\bar{\mathbf{y}} \cong \mathbf{F}(\mathbf{x}^*) + \left. \frac{\partial \mathbf{F}}{\partial \Delta \mathbf{u}(k)} \right|_{\mathbf{x}=\mathbf{x}^*} \Delta \mathbf{u}(k).$$

Vamos analisar os termos separadamente. O primeiro termo é o valor de  $\mathbf{F}(\mathbf{x}^*)$  para o ponto de operação, que é

$$\mathbf{x}^* = [y(k), u(k-1), (\Delta \mathbf{u} = \mathbf{0})^T, \mathbf{q}(k+1)^T]^T.$$

Note que todos os incrementos futuros de controle são zero, assim, esse termo representa a resposta livre do processo, isto é, a resposta futura do sistema se a ação de controle futura não mudar de valor (incrementos futuros nulos). No entanto, é importante frisar que esta é a resposta livre sem correção, ou seja, não leva em conta o efeito de perturbações não medidas. A resposta livre corrigida deve ser calculada de acordo com o apresentado na seção anterior.

O segundo termo representa como a função  $\mathbf{F}(\cdot)$  varia em torno do ponto de operação quando os incrementos de controle variam em instantes futuros de tempo, que é uma aproximação da resposta forçada do sistema não linear. Desta forma, o algoritmo PNMPC calcula a predição não linear do processo da seguinte maneira

$$\bar{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u}(k) + \mathbf{f},$$

onde  $\mathbf{G}$  é a matriz jacobiana de  $\mathbf{F}$  em relação a  $\Delta \mathbf{u}(k)$  em torno do ponto de

operação  $\mathbf{x}^*$ , isto é,

$$\mathbf{G} = \frac{\partial \mathbf{F}}{\partial \Delta \mathbf{u}(k)} = \begin{bmatrix} \frac{\partial \bar{y}(k+1|k)}{\partial \Delta u(k)} & 0 & \cdots & 0 \\ \frac{\partial \bar{y}(k+2|k)}{\partial \Delta u(k)} & \frac{\partial \bar{y}(k+2|k)}{\partial \Delta u(k+1)} & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ \frac{\partial \bar{y}(k+N_2|k)}{\partial \Delta u(k)} & \frac{\partial \bar{y}(k+N_2|k)}{\partial \Delta u(k+1)} & \cdots & \frac{\partial \bar{y}(k+N_2|k)}{\partial \Delta u(k+N_2-1)} \end{bmatrix}.$$

Note que essa matriz é triangular inferior, e a razão disso é intuitiva. Como o sistema é causal, a predição no tempo  $k+j$  só depende de informação atuais ou passadas.

Note ainda que nesta abordagem não linear, como o modelo de resposta livre já considera os valores da perturbação medida (tanto passados como futuros), a ação antecipativa do controle MPC é implementada diretamente, sem necessidade de calcular separadamente a relação entre a perturbação e a saída do modelo.

O cômputo analítico de  $\mathbf{G}$  pode ser extremamente complexo dependendo das não linearidades presentes nas equações do sistema. Ainda, pode acontecer, na prática, de não termos um modelo analítico do processo. Nesses casos, é preciso utilizar uma maneira alternativa para o cálculo de  $\mathbf{G}$ . Usando o conceito de limites, a derivada parcial de uma função é definida como:

$$\frac{\partial g(x)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{g(x + \Delta x) - g(x)}{\Delta x}.$$

Esta definição pode ser utilizada para calcular numericamente o jacobiano de  $\mathbf{F}$ :

1. Calcule a resposta livre sem correção do sistema  $\bar{\mathbf{y}}_0 = \mathbf{F}(\mathbf{x}^*)$ .
2. Calcule a primeira coluna de  $\mathbf{G}$ . Faça

$$\Delta \mathbf{u}(k) = [\epsilon, 0, \dots, 0]^T,$$

onde  $\epsilon$  é um valor pequeno, por exemplo,  $u(k-1)/1000$ , e calcule  $\bar{\mathbf{y}}_1 = \mathbf{F}(y(k), u(k-1), \Delta \mathbf{u}(k), q(k), \mathbf{0})$ . Então, a primeira coluna de  $\mathbf{G}$  é dada por:

$$\frac{\bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_0}{\epsilon}.$$

3. Calcule a segunda coluna de  $\mathbf{G}$ . Faça

$$\Delta \mathbf{u}(k) = [0, \epsilon, 0, \dots, 0]^T,$$

e calcule  $\bar{\mathbf{y}}_2 = \mathbf{F}(y(k), u(k-1), \Delta \mathbf{u}(k), q(k), \mathbf{0})$ . Então, a segunda coluna de  $\mathbf{G}$  é dada por:

$$\frac{\bar{\mathbf{y}}_2 - \bar{\mathbf{y}}_0}{\epsilon}.$$

4. Repita o procedimento para as outras colunas de  $\mathbf{G}$  até a matriz estar completa.

É interessante notar que a matriz  $\mathbf{G}$  não verifica a propriedade vista no caso linear (GPC ou DMC), onde cada coluna poderia ser obtida deslocando a anterior em uma posição e adicionando um zero no primeiro elemento. Isto se deve por conta da característica não linear do processo.

Vamos apresentar, na sequência, um exemplo simples de aplicação destas ideias no cálculo da matriz  $\mathbf{G}$  do PNMPC.

**Exemplo 5.7.** *Considerando novamente o modelo do tanque Exemplo 5.2 com  $g = 9,81$ ,  $K_1 = 2$  e  $K_2 = 0,1$ ,  $N_1 = 1$ ,  $N_2 = N_u = 3$ , vamos calcular a matriz  $\mathbf{G}$ . Inicialmente, para deixar mais claro o procedimento, vamos obter a função  $\mathbf{F}$  do sistema de maneira recursiva:*

$$\begin{aligned}\bar{y}(k+1 | k) &= h(k) + 2(a(k-1) + \Delta a(k)) - 0,1q(k)\sqrt{2gh(k)} \\ &= f_0(h(k), a(k-1), \Delta a(k), \mathbf{q}(k+1))\end{aligned}$$

$$\begin{aligned}\bar{y}(k+2 | k) &= \bar{y}(k+1 | k) + 2(a(k-1) + \Delta a(k) + \Delta a(k+1)) \\ &\quad - 0,1q(k+1)\sqrt{2g\bar{y}(k+1 | k)} \\ &= f_1(h(k), a(k-1), \Delta a(k), \mathbf{q}(k+2))\end{aligned}$$

$$\begin{aligned}\bar{y}(k+3 | k) &= \bar{y}(k+2 | k) + 2(a(k-1) + \Delta a(k) + \Delta a(k+1) + \Delta a(k+2)) \\ &\quad - 0,1q(k+2)\sqrt{2g\bar{y}(k+2 | k)} \\ &= f_2(h(k), a(k-1), \Delta a(k), \mathbf{q}(k+3)).\end{aligned}$$

Agora, é necessário calcular a resposta livre sem correção do sistema,  $\bar{\mathbf{y}}_0 = \mathbf{F}(\mathbf{x}^*)$ , ou seja, considerando  $\Delta \mathbf{a}(k) = \mathbf{0}$ . Desta forma, supondo  $a(k-1) = 50$ ,  $q(k+j) = 20 \forall j > 0$ , e  $h(k) = 50$ ,

$$\bar{\mathbf{y}}_0^T = [87,3582 \ 104,5580 \ 113,9726].$$

Para o cálculo de  $\mathbf{G}$ , devemos calcular  $\bar{\mathbf{y}}_1$ ,  $\bar{\mathbf{y}}_2$  e  $\bar{\mathbf{y}}_3$ , que são obtidos a partir de  $\mathbf{F}$  utilizando  $\Delta \mathbf{a}(k) = [\epsilon, 0, 0]$ ,  $\Delta \mathbf{a}(k) = [0, \epsilon, 0]$ ,  $\Delta \mathbf{a}(k) = [0, 0, \epsilon]$ , respectivamente. Neste exemplo, assumiremos  $\epsilon = a(k-1)/1000 = 0,05$ :

$$\begin{aligned}\bar{\mathbf{y}}_1^T &= [87,4582 \ 104,7106 \ 114,1591], \\ \bar{\mathbf{y}}_2^T &= [87,3582 \ 104,6580 \ 114,1293], \\ \bar{\mathbf{y}}_3^T &= [87,3582 \ 104,5580 \ 114,0726].\end{aligned}$$



Então,  $\mathbf{G}$  pode ser calculada como

$$\mathbf{G} = \begin{bmatrix} \frac{\bar{y}_1 - \bar{y}_0}{\epsilon} & \frac{\bar{y}_2 - \bar{y}_0}{\epsilon} & \frac{\bar{y}_3 - \bar{y}_0}{\epsilon} \end{bmatrix} = \begin{bmatrix} 2,0000 & 0 & 0 \\ 3,0524 & 2,0000 & 0 \\ 3,7307 & 3,1338 & 2,0000 \end{bmatrix}.$$

### 5.4.3 Algoritmo PNMPC

Com a resposta livre e forçada definidas, as previsões do sistema podem ser escritas como

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u}(k) + \mathbf{f}, \quad (5.15)$$

que é o formato padrão já apresentado para os casos lineares. Assim, supondo que a função custo utilizada seja a padrão apresentada anteriormente, para obter a sequência de controle ótima a ser aplicada no processo basta executar o procedimento de minimização da função custo apresentado para o caso linear. Como mencionado na apresentação do PNMPC, esta é uma das vantagens deste algoritmo. Enquanto muitos algoritmos NMPC necessitam, a cada instante de amostragem, da resolução de múltiplos problemas de otimização para a obtenção do controle ótimo, o PNMPC só requer uma otimização por amostragem, gerando um ganho em desempenho computacional significativo. Assim, se comparado a MPC lineares, a complexidade do PNMPC está apenas no cômputo de  $\mathbf{G}$ , que precisa ser feito a cada amostra, e no uso de um modelo não linear para o cálculo da resposta livre.

A execução do algoritmo PNMPC pode ser resumida nos seguintes passos:

1. obtenha o valor atual da saída do sistema  $y(k)$ , da perturbação medida  $q(k)$  e, possivelmente, dos valores futuros de  $q$ ;
2. calcule  $\mathbf{G}$  utilizando o procedimento apresentado na Seção 5.4.2;
3. calcule a resposta livre corrigida do sistema de acordo com o apresentado na Seção 5.4.1;
4. minimize a função custo quadrática  $J$  e obtenha o controle ótimo a ser aplicado no processo;
5. aplique a ação de controle no processo;
6. espere uma amostra e repita o procedimento.

Na prática, podem ser usadas algumas estratégias para diminuir o tempo de cômputo [Yan16]. Dependendo do sistema que está sendo controlado, pode não ser necessário calcular a matriz  $\mathbf{G}$  em todos os passos de amostragem,

mas apenas a cada  $N_o$  amostras. Outra forma de simplificar o procedimento é calculando apenas a primeira coluna da matriz e usando o procedimento de deslocamento para as demais colunas. Esta abordagem é uma simplificação pois mesmo que o sistema não linear seja invariante no tempo, as condições iniciais no momento da aplicação dos diversos  $\Delta u(k + j)$  para o cálculo dos vetores  $\bar{y}_i$  são, no caso geral, diferentes, o que implica uma resposta distinta devido à não linearidade do sistema.

**Exemplo 5.8.** *Neste exemplo, apresentaremos uma simulação completa com o PNMPC, e uma comparação com o NEPSAC e com o PNMPC alternativo, onde calcularemos a matriz  $\mathbf{G}$  a cada 5 amostras (caso PNMPC2). O sistema a ser controlado é o tanque apresentado no Exemplo 5.5, e o cenário de simulação e sintonia de ambos os controladores são os apresentados anteriormente:  $g = 9,81$ ,  $K_1 = 2$ ,  $K_2 = 0,1$ , o ponto de operação é dado por  $h^* = 50$ ,  $u^* = 31,3209$  e  $q^* = 20$ , e a sintonia é  $N_1 = 1$ ,  $N_2 = 30$ ,  $N_u = 5$ ,  $\lambda = 1$ , e a perturbação  $q$  é medida, porém,  $N_q = 0$ .*

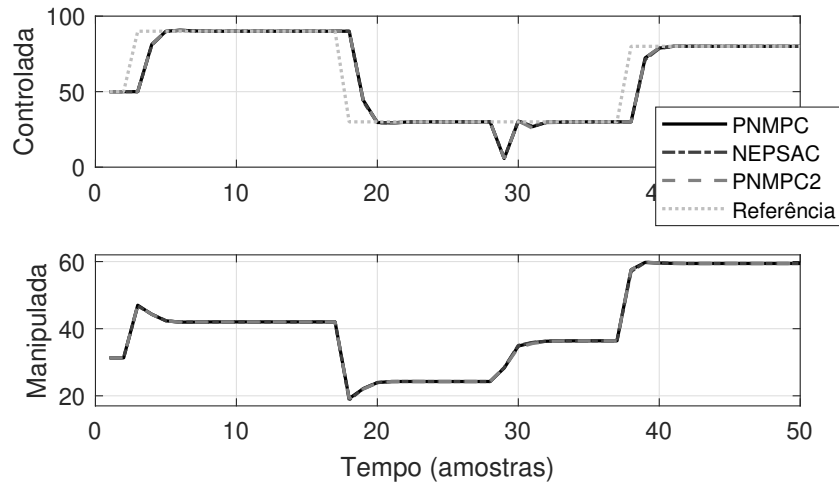


Figura 5.7: Exemplo 5.8 – comparação entre os algoritmos PNMPC e NEPSAC para o controle de nível em um tanque. PNMPC2 é o algoritmo PNMPC com cálculo da matriz  $\mathbf{G}$  a cada 5 amostras.

*Os resultados das simulações são apresentados na Figura 5.7. Veja que, para este caso particular, o desempenho de todos os algoritmos é virtualmente idêntico. No entanto, como mencionado anteriormente, o algoritmo PNMPC resolve o mesmo problema com desempenho similar ao NEPSAC mas tendo um custo computacional reduzido. E este é ainda menor se a matriz  $\mathbf{G}$  não for calculada a todo instante, como no caso do PNMPC2. Porém, vale ressaltar que não se pode generalizar e dizer que o PNMPC sempre terá o*

mesmo desempenho do NEPSAC (ou do PNMPC com o cálculo de  $\mathbf{G}$  com  $N_o > 1$ ), já que isto depende muito do sistema a ser controlado e de suas não linearidades.

#### 5.4.4 Extensão para o caso multivariável

Assim como no NEPSAC, a extensão do PNMPC para o caso multivariável (sistema com  $m$  entradas e  $n$  saídas) é direta, e podemos usar o mesmo procedimento de otimização e definição da função custo e restrições do caso linear e escolher horizontes e ponderações de forma individualizada para cada variável. Assim como fizemos com o NEPSAC, mostraremos a relação entre o modelo não linear do processo e as matrizes usadas na formulação do problema de otimização.

No modelo, cada saída  $y_i(k)$  será representada por um modelo genérico da forma:

$$y_i(k) = f_i(\mathbf{y}_{p,i}(k-1), \mathbf{u}_p(k-1), \mathbf{q}_p(k)) + \eta_i(k), \quad i = 1 \dots n,$$

onde  $\mathbf{y}_{p,i}(k-1)$ ,  $\mathbf{u}_p$  e  $\mathbf{q}_p$  são, respectivamente, os vetores contendo as informações passadas das  $n$  saídas passadas, dos  $m$  controles e das  $m_q$  perturbações medidas que afetam a saída  $y_i$ , e  $\eta_i(k)$  é a perturbação não medida que afeta a mesma saída, com  $i = 1 \dots n$ .

Assim, mantendo o controle constante no sistema multivariável e considerando as perturbações medidas  $q_s$ , calculam-se todas as saídas  $y_i$ , apenas simulando o modelo não linear multivariável no horizonte de predição de cada saída.

Por outro lado, para encontrar as respostas forçadas usaremos a mesma metodologia do caso SISO para cada uma das entradas, obtendo uma matriz  $\mathbf{G}_{i,l}$  para modelar o efeito na saída  $i$  da entrada  $l$ . Assim, a predição resulta:

$$\hat{\mathbf{y}}_i = \mathbf{G}_{i,1}\Delta\mathbf{u}_1 + \mathbf{G}_{i,2}\Delta\mathbf{u}_2 + \dots + \mathbf{G}_{i,m}\Delta\mathbf{u}_m + \mathbf{f}_i,$$

com  $\mathbf{f}_i$  a resposta livre e:

$$\hat{\mathbf{y}}_i = \begin{bmatrix} \hat{y}_i(k + N_{1,i} | k) \\ \hat{y}_i(k + N_{1,i} + 1 | k) \\ \vdots \\ \hat{y}_i(k + N_{2,i} | k) \end{bmatrix}, \Delta\mathbf{u}_l = \begin{bmatrix} \Delta u_l(k) \\ \Delta u_l(k + 1) \\ \vdots \\ \Delta u_l(k + N_{u,l} - 1) \end{bmatrix}$$

sendo  $N_{1,i}$  e  $N_{2,i}$  os horizontes da saída  $i$  e  $N_{u,l}$  o horizonte de controle de  $u_l$ .

A forma de calcular cada  $\mathbf{G}_{i,l}$  segue o mesmo procedimento do caso SISO, calculando as derivadas parciais numericamente para cada controle  $l$ . Finalmente, a matriz completa terá  $n \times m$  blocos:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{1,m} \\ \mathbf{G}_{2,1} & \cdots & \mathbf{G}_{2,m} \\ \vdots & \vdots & \vdots \\ \mathbf{G}_{n,1} & \cdots & \mathbf{G}_{n,m} \end{bmatrix}$$

e o modelo de predição será dado por:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f}, \quad (5.16)$$

com:

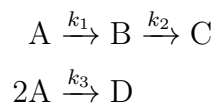
$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}, \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_m \end{bmatrix}.$$

A execução do algoritmo PNMPC para o caso MIMO segue os mesmos passos do caso SISO.

Na Seção 5.5, vamos ilustrar o uso do PNMPC em um processo multivariável e comparar os resultados com os obtidos com os controladores anteriores apresentados neste capítulo.

## 5.5 Estudo de Caso Multivariável – CSTR

Neste estudo de caso, vamos realizar uma análise comparativa dos três algoritmos apresentados: (a) NMPC com otimização não linear geral, (b) NEPSAC e (c) PNMPC. O problema em questão envolve o controle de um Tanque Reator Continuamente Agitado (CSTR, do inglês *Continuous Stirred Tank Reactor*). Reatores são muito importantes em plantas químicas e eles podem se apresentar de diferentes maneiras, mas uma forma muito comum é o CSTR [Beq98]. Esses sistemas podem apresentar dinâmicas altamente não lineares, especialmente quando reações químicas em sequência e em paralelo estão presentes. O processo em consideração realiza a produção de ciclopentanol (B) a partir de ciclopentadieno (A) por meio de uma reação química específica. Esta reação ainda tem como subprodutos o dicitopentadieno (D) e o ciclopentanediol (C) [EK93]. A descrição da reação química é atribuída a van de Vusse [vdV64] e pode ser escrita como:



O diagrama esquemático do processo é apresentado na Figura 5.8.

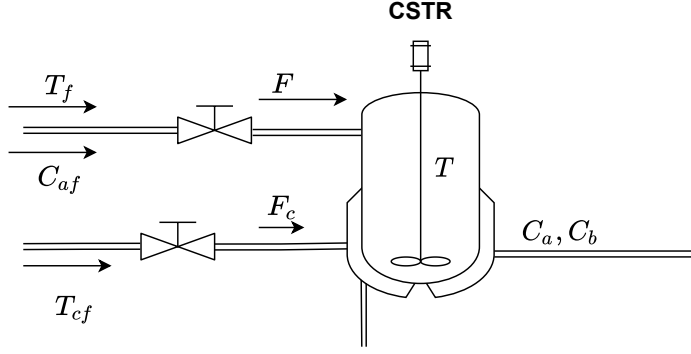


Figura 5.8: Diagrama esquemático do CSTR.

O CSTR MIMO a ser utilizado foi inicialmente apresentado em [DJ14] e tem como variáveis controladas a temperatura dentro do reator  $T$ , e a concentração do produto A na saída  $C_a$ . As variáveis manipuladas são a vazão do processo  $F$ , e a vazão do fluido refrigerante  $F_c$ . As perturbações não medidas são a temperatura de entrada do produto A,  $T_f$ , a temperatura do fluido refrigerante  $T_{cf}$ , e a concentração do produto A na alimentação do reator  $C_{af}$ . As equações diferenciais que descrevem a dinâmica do processo são:

$$\begin{cases} \frac{dC_a(t)}{dt} = \frac{F(t)}{V} [C_{af}(t) - C_a(t)] - k_0 C_a(t) e^{-E/RT(t)} & (5.17a) \\ \frac{dT(t)}{dt} = \frac{F(t)}{V} [T_f(t) - T(t)] + k_1 C_a(t) e^{-E/RT(t)} & (5.17b) \\ \quad + k_2 F_c(t) [1 - e^{-k_3/F_c(t)}] [T_{cf}(t) - T(t)] \end{cases}$$

com

$$\begin{aligned} k_1 &= -\frac{\Delta H k_0}{\rho C_p} = 1,44 \times 10^{13} && \text{L K mol}^{-1} \text{ min}^{-1}, \\ k_2 &= \frac{\rho_c C_{pc}}{\rho C_p V} = 0,01 && \text{L}^{-1}, \\ k_3 &= \frac{h_a}{\rho_c C_{pc}} = 700 && \text{L min}^{-1}, \end{aligned}$$

a partir dos parâmetros dados na Tabela 5.1, juntamente com os valores do ponto de operação como definido em [DJ14].

O modelo a ser utilizado pelos controladores é obtido discretizando as equações do sistema (5.17b) usando a aproximação *forward* (Euler) da deri-

Tabela 5.1: Parâmetros e ponto de operação do MIMO CSTR

Concentração medida do produto A	$C_a^*$	$0,1 \text{ mol L}^{-1}$
Vazão do fluido refrigerante	$F_c^*$	$103,41 \text{ L min}^{-1}$
Concentração de A na alimentação	$C_{af}^*$	$1 \text{ mol L}^{-1}$
Temperatura do fluido refrigerante	$T_{cf}^*$	$350 \text{ K}$
Temperatura do Reator	$T^*$	$438,51 \text{ K}$
Vazão de entrada	$F^*$	$100 \text{ L min}^{-1}$
Temperatura da vazão de entrada	$T_f^*$	$350 \text{ K}$
Termo de transferência de calor	$h_a$	$7 \times 10^5 \text{ cal/min/K}$
Termo de energia de ativação	$E/R$	$1 \times 10^4 \text{ K}$
Densidades dos líquidos	$\rho, \rho_c$	$1 \times 10^3 \text{ g L}^{-1}$
Volume do CSTR	$V$	$100 \text{ L}$
Constante da taxa de reação	$k_0$	$7,2 \times 10^{10} \text{ min}^{-1}$
Calor da reação	$\Delta H$	$-2 \times 10^5 \text{ cal/mol}$
Calor específico	$C_p, C_{pc}$	$1 \text{ cal/g/K}$

vada, com um tempo de amostragem de  $T_s = 0,1 \text{ min}$ :<sup>2</sup>

$$\left\{ \begin{array}{l} C_a(k+1) = C_a(k) + T_s \left( \frac{F(k)}{V} [C_{af}(k) - C_a(k)] \right) \\ \quad - T_s \left( k_0 C_a(k) e^{-E/RT(k)} \right) \\ T(k+1) = T(k) + T_s \left( \frac{F(t)}{V} [T_f(k) - T(k)] \right) \\ \quad + T_s \left( k_1 C_a(k) e^{-E/RT(k)} \right) \\ \quad + T_s \left( k_2 F_c(k) [1 - e^{-k_3/F_c(k)}] [T_{cf}(k) - T(k)] \right) \end{array} \right.$$

Como as perturbações são consideradas não medidas, os modelos nominais usados pelos controladores considerarão os valores do ponto de operação definidos na Tabela 5.1.

O cenário de simulação contempla mudanças de referência e perturbações. O sistema inicia no ponto de operação definido na Tabela 5.1. A referência de  $C_a$  inicia em  $0,1 \text{ mol L}^{-1}$  e muda para  $0,08 \text{ mol L}^{-1}$  em  $t = 8 \text{ min}$ , e para  $0,1 \text{ mol L}^{-1}$  em  $t = 30 \text{ min}$ . A referência de  $T$  inicia em  $438,51 \text{ K}$ , muda para  $443,50 \text{ K}$  em  $t = 15 \text{ min}$ , e para  $438,51 \text{ K}$  em  $t = 45 \text{ min}$ . As perturbações

<sup>2</sup>Para este sistema, a aproximação *forward* da derivada resultou em um bom modelo com este período de amostragem. No entanto, este método de aproximação, apesar de resultar em um modelo mais simples, requer um período de amostragem relativamente menor para funcionar adequadamente [Beq98].

$C_{af}$ ,  $T_f$  e  $T_{cf}$  iniciam no ponto de operação e mudam para  $1,03 \text{ mol L}^{-1}$  em  $t = 60 \text{ min}$ ,  $353,00 \text{ K}$  em  $t = 90 \text{ min}$  e  $347,00 \text{ K}$  em  $t = 75 \text{ min}$ , respectivamente. Para o NEPSAC, foi utilizado como critério de parada uma tolerância de 0,1 ou 100 iterações. O sistema é simulado utilizando (5.17b) e o algoritmo ODE45 do MATLAB. Todos os problemas de otimização foram resolvidos usando o CasADi e o solucionador IPOPT.

As sintonias para o PN MPC e NEPSAC são iguais:  $N_{1,1} = N_{1,2} = 1$ ,  $N_{2,1} = N_{2,2} = 50$ ,  $N_{u,1} = N_{u,2} = 30$ ,  $\delta_1 = 10000$ ,  $\delta_2 = 1$ ,  $\lambda_1 = \lambda_2 = 0,5$ ,  $C(z) = 1$ ,  $D(z) = 1 - z^{-1}$ , e um filtro de referência exponencial de primeira ordem com  $\alpha = 0,85$ . São consideradas restrições nas manipuladas  $F \in [75, 150]$ ,  $F_c \in [60, 130]$ , e nos seus incrementos, que estão limitados a 2 unidades por amostra.

Os resultados das simulações são apresentados na Figura 5.9. Veja que apesar da sintonia ser idêntica, os algoritmos geram respostas diferentes. Isso se dá pela forma de obtenção do controle ótimo: o NMPC utiliza diretamente as equações não lineares gerando um problema de otimização mais complexo, mas mais preciso; o NEPSAC executa uma sequência de otimizações por período de amostragem, como forma de tentar aproximar a solução do problema não linear original; o PN MPC faz apenas uma iteração por amostragem, simplificando o problema. Veja, porém, que as diferenças entre o NMPC e o NEPSAC foram mínimas neste estudo de caso. Essa característica de simplificação do problema do PN MPC é uma vantagem do ponto de vista computacional, mas pode gerar uma resposta sub-ótima em relação ao sistema não linear. As diferenças são visíveis tanto nos trechos de seguimento de referência quanto nos períodos de rejeição de perturbação. Neste caso particular, tanto o NMPC quanto o NEPSAC conseguem respostas consideravelmente melhores do que as do PN MPC sem aumentar de forma significativa o esforço de controle. Um ponto interessante da simulação a se notar, que evidencia a característica não linear do processo, é a mudança de referência por volta de  $t = 45 \text{ min}$ . Note que as ações de controle tomadas pelos três algoritmos são muito similares em termos de transitório e magnitude, mas a resposta do sistema não, especialmente a de  $C_a$ .

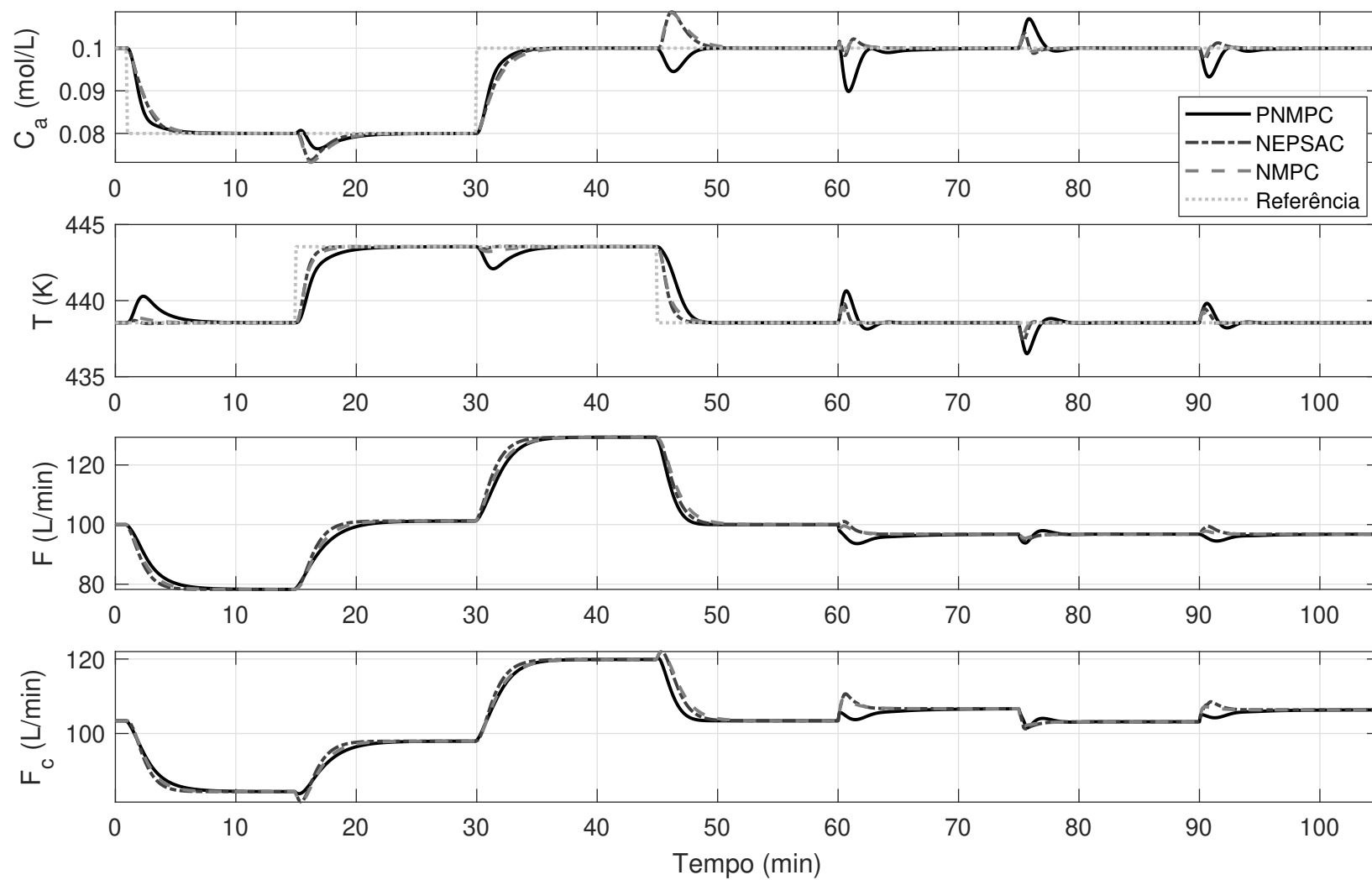


Figura 5.9: Estudo de caso do CSTR – resultados de simulação com PNMPC, NEPSAC e NMPC.



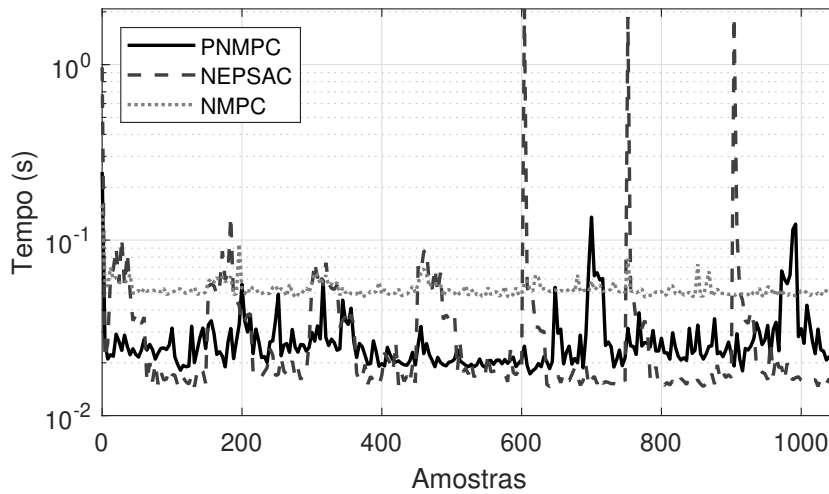


Figura 5.10: Estudo de caso do CSTR – tempo de cômputo dos algoritmos PNMPC, NEPSAC e NMPC.

Apesar de NMPC e NEPSAC obterem respostas melhores, estes têm um custo computacional maior do que o PNMPC, como já enfatizado. A Figura 5.10 ilustra o tempo gasto pelos três algoritmos em cada amostra e é notável a diferença de custo computacional<sup>3</sup>. Obviamente, os critérios de parada do NEPSAC (tolerância e número de iterações) podem ser relaxados para melhorar seu tempo de cômputo, porém isso tende a prejudicar o desempenho do controlador. Além disso, veja que o NMPC, na maior parte do tempo, possui tempos de resolução maiores do que os outros dois algoritmos, o que era de se esperar já que este resolve um problema de otimização mais complexo. No entanto, em certos momentos, o NEPSAC leva mais tempo para resolver o problema do que o NMPC. Isto ocorre pois o NEPSAC não resolve somente um problema de otimização a cada amostra, mas uma sequência de problemas. Assim, em certos momentos, é possível que este leve mais tempo para obter a solução ótima, como mostrado na Figura 5.10. Entretanto, note que, quando o NEPSAC necessita realizar poucas iterações a cada instante de amostragem, ele é capaz de atingir períodos de amostragem mais baixos que os outros dois controladores. Apesar de os tempos de cômputo variarem em função do algoritmo implementado, note que, para este problema específico, os tempos não são críticos e estão bem abaixo do período de amostragem.

<sup>3</sup>Os tempos de execução vão depender muito dos critérios de parada definidos pelo NEPSAC e também das configurações de hardware do computador executando o algoritmo. No entanto, os resultados apresentados aqui ilustram a diferença de custo computacional entre os algoritmos sem perda de generalidade.

## 5.6 Comentários finais

Neste capítulo, apresentamos uma introdução ao problema de controle preditivo não linear (NMPC), entendendo por NMPC as técnicas que utilizam modelos não lineares para o cálculo das predições, mas que mantêm o mesmo tipo de função custo e restrições que os controladores MPC lineares. Inicialmente, mostramos como o problema não linear é conceitualmente muito similar ao do caso linear, e que o uso de ação antecipativa de perturbações medíveis, o uso de filtros de referência e o tratamento de restrições na ação de controle podem ser tratados no NMPC da mesma forma que no caso linear. Por outro lado, mostramos que no NMPC a maior complexidade está associada à solução do problema de otimização que precisa ser resolvido a cada amostra, o que resulta em um problema de otimização não linear. Assim, foram apresentadas algumas formas de resolver o problema, com softwares de otimização não linear e com métodos que aproximam o problema para conseguir achar uma solução fazendo uso apenas de programação quadrática. Do estudo comparativo dos métodos apresentados, primeiro para o caso SISO, e depois para o caso MIMO, concluímos que o resultado é bastante dependente do caso estudado e, em muitas aplicações, as versões mais simples acabam tendo um resultado bem satisfatório. Desta forma, os métodos aproximados apresentados podem ser bastante úteis na prática. Com os estudos de caso apresentados e a discussão relativa à implementação prática de NMPC, este capítulo coloca as bases para que o leitor possa entender e utilizar este tipo de controle na prática.

## 5.7 Exercícios propostos

1. Considere o controle de nível de dois tanques iguais acoplados, de forma tal que a vazão de entrada do tanque 2 é a vazão de saída do tanque 1. Os modelos dos níveis dos tanques  $h_1$  e  $h_2$  são dados por:

$$h_1(k) = h_1(k-1) + K_1 u(k-1) - K_2 q_1(k-1) \sqrt{2gh_1(k-1)},$$

$$h_2(k) = h_2(k-1) + K_2 q_1(k-1) \sqrt{2gh_1(k-1)} - K_2 q_2(k-1) \sqrt{2gh_2(k-1)},$$

com  $g = 9,81$ .  $K_1 = 1,5$  e  $K_2 = 1$  são constantes que definem as características dos tanques e das válvulas. Nestas equações,  $u$  é a abertura da válvula que alimenta o tanque 1, sendo considerada a variável manipulada.  $q_1$  e  $q_2$  são respectivamente as aberturas das válvulas de saída dos tanques 1 e 2. Deseja-se neste problema realizar o controle de nível do tanque 2 buscando que ele possa seguir uma dada referência

de tipo degrau e que rejeite as perturbações causadas pelas variações nas válvulas  $q_1$  e  $q_2$ , que são também do tipo degrau. As aberturas das válvulas estão limitadas ao intervalo  $[0, 1]$  e os níveis limitados ao intervalo  $[0, 10]$ . Ainda considere que as variações de controle estão limitadas ao intervalo  $[-0,05, 0,05]$ .

Implemente um controle NMPC para o sistema dos tanques usando CasADi ou outra ferramenta equivalente para a otimização não linear. Considere dois casos, quando os valores das perturbações podem ser medidos e quando não.

2. Escreva um código genérico para calcular a resposta base e a resposta ótima do NEPSAC, considerando que utiliza uma função para o cálculo da resposta de um sistema dinâmico genérico.
3. Para o problema de controle do NEPSAC, encontre a forma quadrática padrão de  $J$  partindo de:

$$J = (\mathbf{w} - \hat{\mathbf{y}})^T \mathbf{Q}_y (\mathbf{w} - \hat{\mathbf{y}}) + \Delta \mathbf{u}^T \mathbf{Q}_u \Delta \mathbf{u}$$

e usando as relações:

$$\Delta \mathbf{u} = \mathbf{T}^{-1} [\delta \mathbf{u} - \mathbf{1} u(k-1) + \mathbf{u}_b]$$

e

$$\hat{\mathbf{y}} = \mathbf{G}_e \delta \mathbf{u} + \hat{\mathbf{y}}_b.$$

Considere restrições tanto na saída  $\mathbf{y}$  como no controle  $\mathbf{u}$ , com valores máximos e mínimos para todo o horizonte dados por  $[y_m, y_M]$  para a saída e  $[u_m, u_M]$  para o controle.

4. Implemente um controlador NEPSAC para o sistema de controle de nível dos tanques acoplados e compare os resultados com os obtidos com o uso do NMPC direto.
5. Considere o processo de controle de pH do caldo concentrado em uma indústria de cana de açúcar. O controle do pH se realiza adicionando ácido na corrente de caldo e o pH se mede no final da tubulação. Assim, a variável manipulada é a vazão de ácido. O processo é modelado a partir de alguns experimentos usando uma amostragem de  $T_s = 0,2$  s, obtendo a seguinte equação a diferenças não linear:

$$y(k) = 0,9y(k-1) - 0,09(u(k-1))^3 + 0,14q(k-1),$$

com  $y(k)$  a medida do pH,  $u(k)$  a vazão de ácido e  $q(k)$  a vazão de caldo. Neste problema, a vazão de caldo  $q$  é a principal perturbação, dado que pode variar com a produção.

Para este sistema, utilizando a metodologia do PNMPC, encontre a função geral para o cálculo das predições e calcule a forma analítica da matriz  $\mathbf{G}$  considerando um controle com  $N_1 = 1$ ,  $N_2 = 4$  e  $N_u = 2$ .

Escolhendo o ponto de operação dado por  $u = 1$  e  $q = 8$ , calcule a matriz  $\mathbf{G}$  numericamente.

6. Para o caso do item anterior, implemente o PNMPC para controlar o processo em uma faixa de pH entre 2 e 12. Assim, analise primeiro a característica estática do processo, estabeleça as faixas da variável manipulada considerando que a vazão de caldo pode variar dentro do intervalo  $6 \text{ m}^3/\text{min}$  a  $10 \text{ m}^3/\text{min}$ . Utilize estas informações para determinar as restrições de operação para ajustar o PNMPC (defina horizontes e ponderações neste ajuste). Simule o processo considerando variações de  $q$  do tipo degrau e variações de referência de pH nas faixas indicadas. Considere dois casos nas simulações, quando existe uma medida de  $q$  que pode ser usada no cálculo das predições da resposta livre não linear e quando  $q$  não está disponível. Compare os dois casos simulados. Finalmente, compare o controle PNMPC com um NEPSAC, com um GPC linear e com um GPC não linear. Discuta os resultados.
7. Aplique o controle PNMPC ao processo dos tanques acoplados utilizando as mesmas condições usadas para o caso do NEPSAC.
8. O sistema de controle dos tanques acoplados pode ser considerado um sistema multivariável, caso se deseje controlar os dois níveis simultaneamente. Considere neste caso que as duas variáveis manipuladas são as aberturas das válvulas de entrada ao tanque 1 ( $u$ ) e a saída do tanque 2 ( $q_2$ ), e que  $q_1$  é uma perturbação. Implemente e ajuste um controlador NEPSAC e um controlador PNMPC para este sistema. Analise vários cenários de mudanças de referência dentro das faixas de operação dos tanques e aplique perturbações para estudar o comportamento do sistema.

## Capítulo 6

# MPC para Sistemas com Atraso

Neste capítulo, estudaremos em mais detalhes as particularidades da aplicação do MPC para sistemas com atraso. Mostraremos como, no caso sem restrições, o MPC pode ser analisado como um compensador de tempo morto (DTC, do inglês *Dead-Time Compensator*) com controle primário de dois graus de liberdade (controle por realimentação com filtro de referência). Isso permite que utilizemos as propriedades dos DTCs para analisar as propriedades do MPC para esse tipo de processo, principalmente aquelas relativas à robustez e à rejeição de perturbações. **Apresentaremos, inicialmente, esse estudo para o DMC e GPC irrestritos para, posteriormente, discutir o caso não linear, com base no NEPSAC e, finalmente, o caso com restrições e multivariável.**

### 6.1 Entendendo o efeito do atraso no MPC

Como já foi comentado neste livro, o MPC pode tratar de sistemas com atraso de forma natural, dado que o modelo usado para as predições pode incluir atraso. De forma geral, a formulação e ajuste do MPC é realizada como no caso de sistemas sem atraso, considerando que o horizonte de predição inicial de uma certa variável controlada  $y$  é  $N_1 = d + 1$ , sendo  $d$  o atraso do modelo que relaciona a variável manipulada com  $y$ . O MPC faz, implicitamente, uma compensação do atraso, dado que o controlador tem informação do efeito do atraso nas predições usadas na função custo. Vamos ilustrar esta propriedade com um exemplo comparativo SISO entre um DMC, um GPC e uma estrutura clássica de controle DTC baseada no Preditor de Smith (PS). Entretanto, antes revisaremos os conceitos principais da estratégia DTC.

O PS é uma estrutura de compensação de atraso proposta em 1957 por Otto Smith [Smi57]. Na sua versão em tempo discreto, o PS é usado para

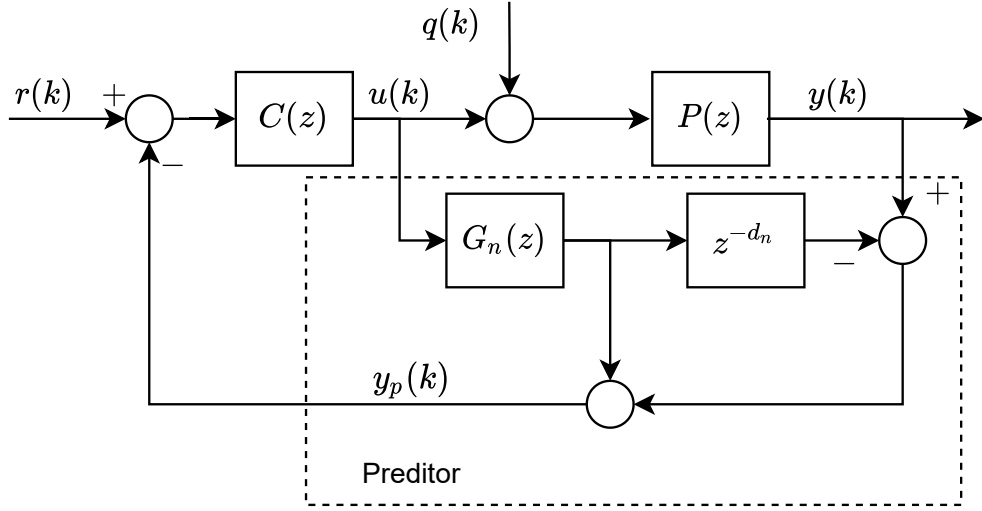


Figura 6.1: Diagrama de blocos do Preditor de Smith (PS).

controlar processos representados por uma função de transferência do tipo  $P(z) = G(z)z^{-d}$ , com  $G(z)$  livre de atraso. Como pode ser observado na Figura 6.1, o PS possui duas partes: (i) um preditor, composto por um modelo nominal do processo sem o atraso  $G_n(z)$  e um modelo nominal do atraso  $z^{-d_n}$ ; e (ii) um controlador primário  $C(z)$ , que atua sobre o erro entre o sinal de referência e a saída predita  $y_p$ . O preditor calcula a previsão da saída do processo sem atraso utilizando  $G_n(z)$  e a previsão da saída do processo com atraso utilizando  $G_n(z)z^{-d_n}$ . Quando temos um modelo perfeito ( $G(z)z^{-d} = G_n(z)z^{-d_n}$ ) e não há perturbações externas,  $y_p$  é uma antecipação da resposta do sistema em  $k + d$ , ou seja,  $y_p(k) = y(k + d)$  e, então, o controle primário pode ser ajustado considerando uma dinâmica do processo sem atraso, já que a ação de controle é dada por

$$U(z) = C(z)[R(z) - Y_p(z)].$$

No entanto, como sempre existirão perturbações e diferenças entre o modelo nominal e a planta, é necessário corrigir as previsões para levar em conta os efeitos dessas diferenças. Assim, o valor predito  $y_p$  enviado ao controle primário é corrigido com o erro entre o valor da saída real do processo e a saída do modelo com atraso:

$$Y_p(z) = G_n(z)U(z) + Y(z) - P_n(z)U(z) = G_n(z)(1 - z^{-d_n})U(z) + Y(z).$$

Desta forma, utilizando o PS, o ajuste do controle é simplificado e se obtém uma função de transferência em malha fechada nominal cujos polos

não dependem do atraso. As relações entre a saída  $Y(z)$  e a referência  $R(z)$  e entre a saída e a perturbação  $Q(z)$  em malha fechada, considerando modelo perfeito no preditor ( $P(z) = P_n(z) = G_n(z)z^{-d_n}$ ), são dadas por:

$$\frac{Y(z)}{R(z)} = \frac{C(z)G(z)z^{-d}}{1 + C(z)G(z)}, \quad (6.1)$$

$$\frac{Y(z)}{Q(z)} = G(z)z^{-d} \left[ 1 - \frac{C(z)G(z)z^{-d}}{1 + C(z)G(z)} \right]. \quad (6.2)$$

Assim, idealmente, o atraso não afeta a dinâmica da resposta depois de  $k + d$  e, teoricamente, o controle primário pode ser ajustado de modo independente do valor de  $d$ . Ainda, teoricamente, se conseguimos polos muito rápidos da função de transferência  $\frac{C(z)G(z)}{1+C(z)G(z)}$  (com controlador de ganho alto), de forma que

$$\frac{C(z)G(z)}{1 + C(z)G(z)} \approx 1,$$

então temos que  $y(k) \approx r(k - d)$ , que é a resposta ideal de um sistema com atraso em malha fechada, já que a saída não pode responder a um estímulo antes do efeito o atraso ter passado.

Para a perturbação, temos propriedades um pouco diferentes. Os polos do modelo em malha aberta do processo aparecem em  $\frac{Y(z)}{Q(z)}$ , o que resulta nas seguintes limitações: (i) o PS somente pode ser usado com plantas estáveis; (ii) a resposta em malha fechada para a perturbação sempre será mais lenta que em malha aberta. Note que a função de transferência é composta pelos termos  $G(z)z^{-d}$  e  $\left[ 1 - \frac{C(z)G(z)z^{-d}}{1+C(z)G(z)} \right]$  e, portanto, o tempo de resposta será sempre maior do que cada um dos termos individualmente. Ainda, para as perturbações na entrada do processo, como é o caso aqui analisado, sempre teremos dois “atrasos” entre o momento da entrada da perturbação e o efeito da ação de controle na saída do processo. Observe que mesmo com um controlador de ganho infinito, com o qual obteríamos  $\frac{C(z)G(z)}{1+C(z)G(z)} \approx 1$ , a saída para uma perturbação é dada por:

$$Y(z) \approx G(z)z^{-d}(1 - z^{-d})Q(z),$$

o que implica que durante  $d$  amostras não há resposta, durante mais  $d$  amostras a saída reage em malha aberta à perturbação  $Q$ , e somente depois desses dois intervalos a perturbação é rejeitada instantaneamente, pois o termo  $1 - z^{-d}$  se anula.

No caso real, com controle não infinito, o transitório da rejeição de perturbação depende do ajuste do controle utilizado, seja este qual for. Assim, observe que o comportamento do controle em malha fechada de um processo

com atraso e perturbação na entrada somente pode ser analisado depois de um tempo de dois atrasos após a aplicação da perturbação. Qualquer controlador sem informação explícita sobre a existência da perturbação, por melhor que ele seja, não será capaz de alterar o comportamento da saída do processo antes de passado um tempo igual a duas vezes o atraso caso a perturbação atue na entrada do sistema. Vamos ilustrar essas propriedades com um exemplo.

**Exemplo 6.1.** *Considere o controle de temperatura de um trocador de calor água-vapor. Nesse sistema, manipula-se a vazão de vapor  $U$  e controla-se a temperatura da água na saída  $T$ . A principal perturbação é a temperatura da entrada da água no trocador  $T_e$ . O modelo do processo foi determinado experimentalmente nas vizinhanças de um ponto de operação como*

$$T(z) = \frac{0,095z^{-d}}{z - 0,905}[U(z) + T_e(z)]$$

*com período de amostragem  $T_s = 0,15$  s, assim as variáveis representam valores incrementais com relação ao ponto de operação. Observe que, neste exemplo, o ganho estático para o controle e para a temperatura de entrada são unitários (nas unidades correspondentes).*

*Neste exemplo, consideramos duas situações diferentes de operação dependendo da localização do sensor de temperatura, o que implica atrasos diferentes. No primeiro caso, temos  $d = 10$ , que corresponde a um atraso igual à constante de tempo do sistema. No segundo caso, temos  $d = 50$ , que representa um atraso mais dominante. O PS é ajustado com um controle primário do tipo PI*

$$C(s) = K_c \frac{1 + T_i s}{T_i s}$$

*com  $T_i = 1,5$  s e  $K_c = 5$ , discretizado com  $T_s = 0,15$  s pelo método de Tustin. Este ajuste busca uma resposta à referência 5 vezes mais rápida que em malha aberta. O atraso do modelo do preditor é considerado, neste exemplo, igual ao do processo, ou seja não há erro de modelagem.*

*No primeiro caso, para  $d = 10$ , compararemos o resultado do PS com o de um PID dado por*

$$C_{PID}(z) = \frac{2,027(z - 0,9048)(z - 0,8182)}{(z - 1)(z - 0,5)}$$

*ajustado usando o método apresentado em [NRC07, Capítulo 4] e discretizando o resultado com o período de amostragem indicado anteriormente. A temperatura e a ação de controle podem ser vistas na Figura 6.2. Na simulação, uma mudança de referência de 0 para 1 é aplicada em  $t = 5$  s e uma*



perturbação de entrada de 0,5 é inserida em  $t = 20$  s. Observe que o PS tem uma resposta bem mais rápida que o PID para a mudança de referência, e isso se deve a uma ação de controle mais rápida e de maior intensidade, antecipando o efeito do atraso. Já para a resposta à perturbação, o PS leva uma vantagem menor, dado que a sua resposta nunca pode ser mais rápida que a de malha aberta, que neste caso é dada pelo polo em  $z = 0,9$ . Note que, depois de aplicar a perturbação, temos um intervalo de  $2d = 20$  amostras até o sistema começar a reagir pelo efeito do controle.

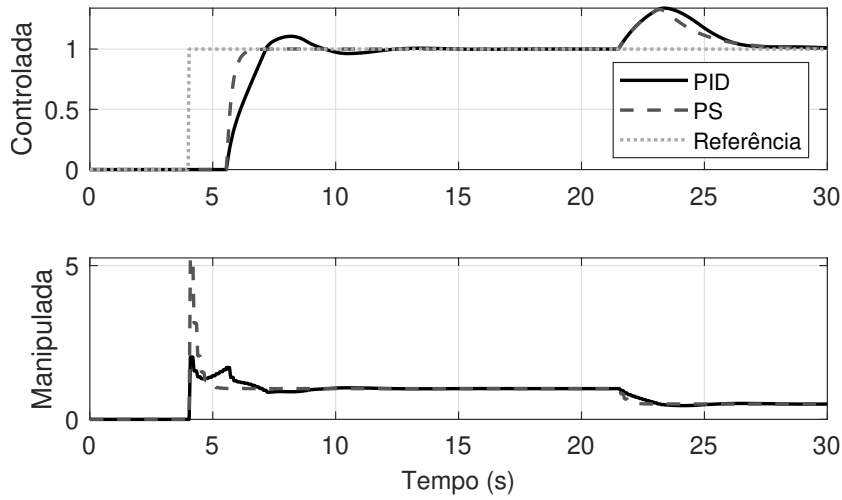


Figura 6.2: Exemplo 6.1 – Controle de temperatura no trocador de calor. Comparação entre PS e PID para atraso  $d = 10$ .

No segundo caso, consideramos o atraso de  $d = 50$ . Nada muda no ajuste do PS além do valor do atraso do modelo. Já no PID, realizamos um novo ajuste pelo mesmo método, obtendo

$$C_{PID}(z) = \frac{0,4699(z - 0,9608)(z - 0,9048)}{(z - 1)(z - 0,875)}.$$

A temperatura e ação de controle podem ser vistas na Figura 6.3. Na simulação, uma mudança de referência de 0 para 1 é aplicada em  $t = 5$  s e uma perturbação de entrada de 0,5 é aplicada em  $t = 50$  s. Observe que as respostas do PS são bem similares às do caso anterior, exceto que estão mais atrasadas, e que, neste caso com maior atraso, as vantagens do PS frente ao PID são maiores. Note, ainda, que a resposta à perturbação continua limitada ao tempo de malha aberta.

Toda a análise anterior considera o caso ideal e, na prática, devemos analisar os efeitos na resposta de malha fechada causados pelos erros de

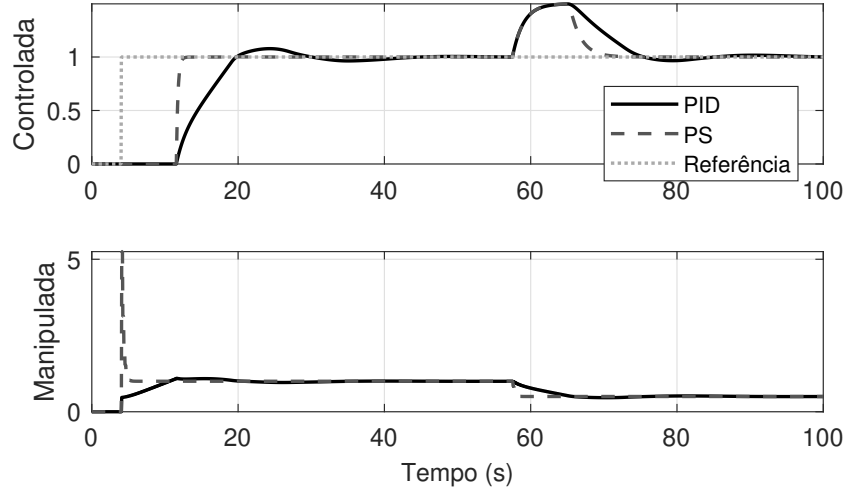


Figura 6.3: Exemplo 6.1 – Controle de temperatura no trocador de calor. Comparação entre PS e PID para atraso  $d = 50$ .

modelagem, isto é, as diferenças entre o processo, representado por  $P(z)$ , e o modelo nominal, representado por  $P_n(z)$ . A estabilidade robusta do sistema pode ser analisada utilizando o teorema do pequeno ganho, obtendo a seguinte condição:

$$\left| \frac{C(z)G_n(z)}{1 + C(z)G_n(z)} \right| < \left| \frac{1}{\delta P(z)} \right|, \quad \forall z = e^{\pm j\omega T_s}, \quad \omega \in [0, \pi/T_s), \quad (6.3)$$

considerando que a representação do processo pode ser relacionada com a do modelo nominal por  $P(z) = P_n(z)(1 + \delta P(z))$ , isto é,  $\delta P(z)$  representa o erro de modelagem multiplicativo e consideramos que  $P$  e  $P_n$  têm o mesmo número de polos com  $|z| \geq 1$ . Assim, se esta condição é satisfeita para toda frequência e para todos os valores esperados do erro de modelagem, o sistema em malha fechada será sempre estável. Esta é uma condição suficiente mas não necessária, mas a experiência prática mostra que, para sistemas com erro de modelagem no atraso, se ela não for satisfeita o sistema em malha fechada tipicamente será instável ou com uma resposta muito oscilatória [NRC07]. Assim, na prática, para obter uma condição de operação com respostas não muito oscilatórias para todos os casos possíveis de erro, devemos considerar um ajuste de  $C(z)$  que cumpra a condição com uma certa folga em todas as frequências. Como, na prática, o valor do erro de modelagem é tipicamente maior em médias e altas frequências do que em baixas frequências, a limitação no ajuste de  $C(z)$  se dá nas frequências médias e altas. Finalmente, caso necessário, o PS pode utilizar um filtro de referência para melhorar a resposta às mudanças de referência, assim como ocorre no controle clássico, atenuando,

por exemplo, picos na resposta ao degrau em  $r(k)$ .

Assim, a partir desta análise, podemos observar que o ajuste do controlador deve buscar um compromisso entre uma resposta nominal rápida depois do atraso, isto é,  $\frac{C(z)G_n(z)}{1+C(z)G_n(z)}$  com valores próximos de 1 em uma larga faixa de frequências, e a condição de robustez calculada com os erros de modelagem esperados, que exige que  $\frac{C(z)G_n(z)}{1+C(z)G_n(z)}$  seja menor que um certo valor menor que 1 nas frequências médias e altas. Como a estabilidade robusta é uma restrição forte, dado que não podemos admitir que o sistema seja instável em malha fechada para nenhuma situação de operação, o ajuste do controle somente pode buscar uma resposta mais rápida contanto que atenda à condição de robustez. Vamos ilustrar esta análise com um exemplo.

**Exemplo 6.2.** Considere o controle de temperatura do trocador de calor água-vapor do Exemplo 6.1 para o caso de menor atraso mas, agora, com erro de modelagem. Assim, o atraso do modelo do preditor,  $d = 10$ , é diferente do atraso do processo, que é assumido como  $d = 12$ , e o polo do modelo do preditor,  $z = 0,9$ , é diferente do polo do processo,  $z = 0,86$ . Novamente, compararemos o resultado do PS e do PID ajustados com os mesmos parâmetros do exemplo anterior. A temperatura e ação de controle podem ser vistas na Figura 6.4 para o mesmo cenário do exemplo anterior. Observe que o PS tem uma resposta muito oscilatória e instável neste caso, dado que o controle foi ajustado sem considerar o erro de modelagem.

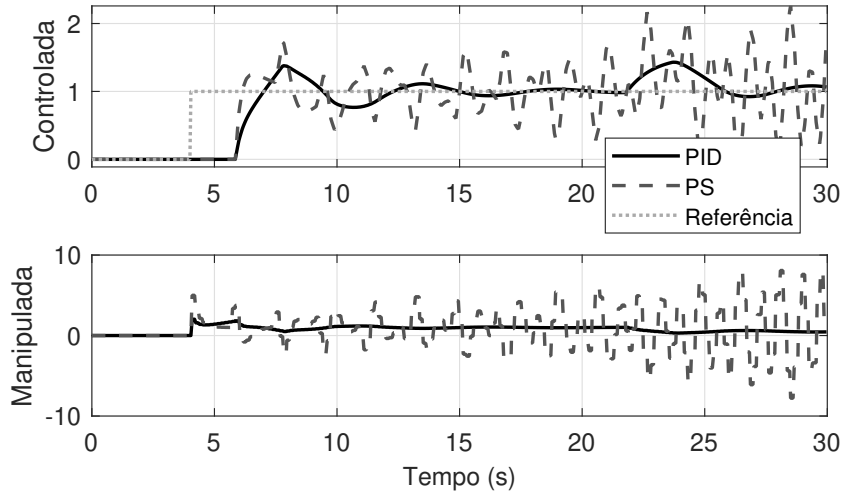


Figura 6.4: Exemplo 6.2 – Controle de temperatura no trocador de calor. Comparação entre PS e PID para atraso  $d = 10$  com erro de modelagem.

Em geral, a condição de robustez depende da frequência de corte  $\omega_c$  de  $\frac{C(z)G_n(z)}{1+C(z)G_n(z)}$  que por sua vez está relacionada ao tempo de resposta do sistema

nominal em malha fechada. Assim, ao ajustar o controle para uma resposta mais rápida, isto é, ao aumentar a frequência de corte  $\omega_c$ , diminuimos, para as frequências  $\omega \geq \omega_c$ , o valor de erro de modelagem que leva o sistema à instabilidade. Por isso, para melhorar a robustez, o ajuste de  $C(z)$  deve ser modificado para uma resposta nominal mais lenta.

Se o ganho do PI é diminuído para  $K_c = 2$ , obtemos as respostas mostradas na Figura 6.5, e observamos que: (i) as oscilações e instabilidade da resposta do PS desaparecem, e (ii) o desempenho do PS é mais lento e acaba sendo similar ao do PID.

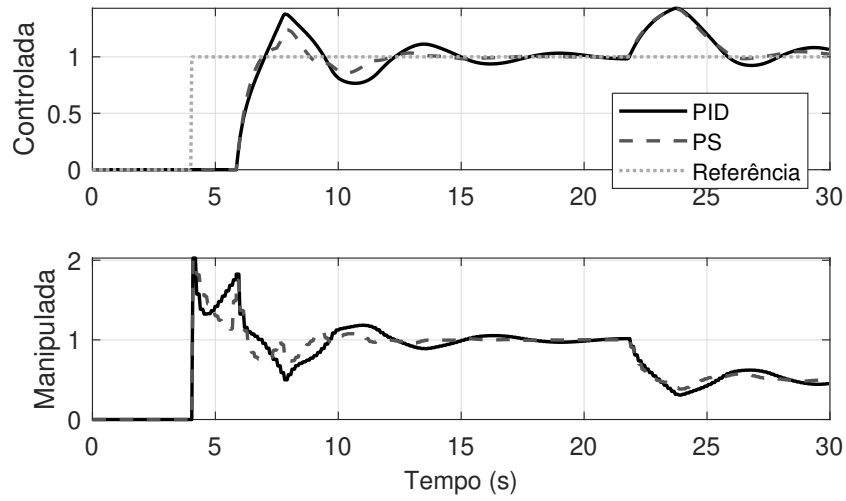


Figura 6.5: Exemplo 6.2 – controle de temperatura no trocador de calor. Comparação entre PS e PID para atraso  $d = 10$  com erro de modelagem e ajuste robusto.

Este resultado é esperado dado que, quando temos erro de modelagem importante como neste exemplo (20% de atraso e 50% na dinâmica), o controle tem que ser mais conservador para conseguir atender à condição de robustez. Nesses casos, utilizar um controle mais sofisticado não traz grandes vantagens dado que ações de controle mais rápidas não podem ser utilizadas.

Assim, mostramos na Figura 6.6 os gráficos dos módulos na frequência da inversa do erro de modelagem,  $|1/\delta P(z)|$  e de  $\left| \frac{C(z)G_n(z)}{1+C(z)G_n(z)} \right|$  para os dois ajustes usados. Observamos que o controlador com ganho  $K_c = 5$  não tem garantia de estabilidade robusta e que o de ganho  $K_c = 2$  tem.

Finalmente, ainda na Figura 6.6, mostramos o gráfico dos módulos na frequência de  $\left| \frac{C_{PID}(z)P_n(z)}{1+C_{PID}(z)P_n(z)} \right|$ , que é a função de transferência em malha fechada para o PID. Observamos que o PID e o controlador PS com  $K_c = 2$  têm garantia de estabilidade robusta e que a distância média entre as curvas

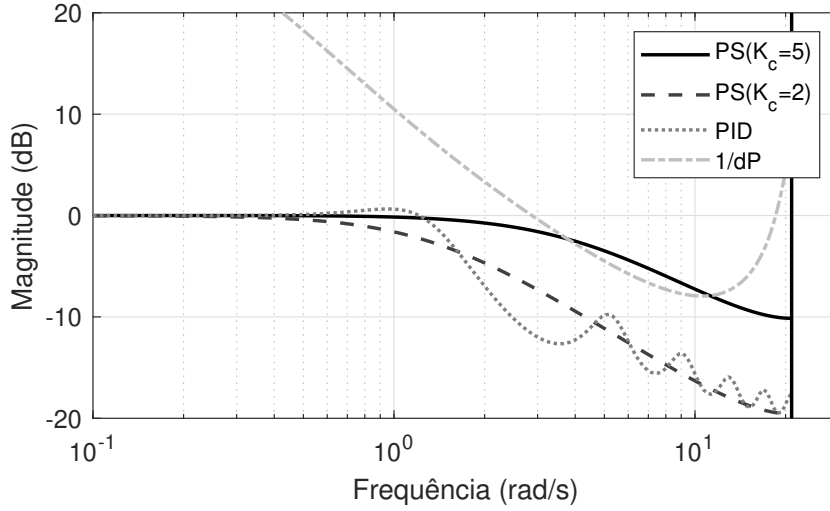


Figura 6.6: Exemplo 6.2 – análise de robustez. Comparação entre os dois ajustes do PS para o caso de atraso nominal  $d = 10$ .

*dos controladores e o erro de modelagem é similar, confirmando o resultado observado na simulação das respostas no tempo.*

Finalmente, vale destacar que a ação de controle implementada pode ser descrita como:

$$U(z) = C_{PS}(z)[R(z) - Y(z)],$$

com  $C_{PS}(z)$  um controlador equivalente que engloba o controle primário e o preditor de tal forma que:

$$C_{PS}(z) = \frac{C(z)}{1 + C(z)G_n(z)(1 - z^{-d_n})}.$$

Podemos chegar aos polinômios usados para a implementação desse controlador definindo  $C(z) = N_c(z)/D_c(z)$  e  $G_n(z) = N_g(z)/D_g(z)$ . Assim:

$$C_{PS}(z) = \frac{\frac{N_c(z)}{D_c(z)}}{1 + \frac{N_c(z)}{D_c(z)} \frac{N_g(z)}{D_g(z)} (1 - z^{-d_n})} = \frac{\frac{N_c(z)}{D_c(z)}}{\frac{D_c(z)D_g(z) + N_c(z)N_g(z)(1 - z^{-d_n})}{D_c(z)D_g(z)}},$$

que pode ser colocado como:

$$C_{PS}(z) = \frac{N_c(z)D_g(z)}{D_c(z)D_g(z) + N_c(z)N_g(z)(1 - z^{-d_n})},$$

ou, usando todos os polinômios em  $z$ :

$$C_{PS}(z) = \frac{N_{PS}(z)}{D_{PS}(z)} = \frac{N_c(z)D_g(z)z^{d_n}}{z^{d_n}D_c(z)D_g(z) + N_c(z)N_g(z)(z^{d_n} - 1)}.$$

Analisando o controlador equivalente do PS observamos que: (i) os polos do modelo aparecem como zeros do controlador, então o PS implementa um controle por cancelamento, o que justifica os fatos de apenas poder ser usado com plantas estáveis e de a resposta à perturbação na entrada ser mais lenta que a de malha aberta; (ii) se o controlador primário  $C(z)$  for integrador, então  $D_c(1) = 0$  e, como o termo  $1 - z^{-d_n}$  é zero em  $z = 1$ , o denominador de  $C_{PS}(z)$  também tem uma raiz em  $z = 1$ , ou seja, o controlador equivalente possui ação integral; (iii) o controlador a ser implementado  $C_{PS}(z) = N_{PS}(z)/D_{PS}(z)$  (transformando a equação em  $z$  em uma equação a diferenças) tem ordem  $n_{PS} = d + n_c + n_g$ , sendo  $n_g$  a ordem do modelo do processo sem atraso e  $n_c$  a ordem do controlador primário.

Uma vez analisado o controlador equivalente do PS, podemos complementar o estudo de robustez e a discussão apresentada no último exemplo. O  $C_{PS}$  é um controlador com ação integral de alta ordem e o PID usado no exemplo é um controle com ação integral de segunda ordem. Dessa forma, o  $C_{PS}$  pode ser considerado como um PID em série com um segundo controle que implementa uma ação de avanço de fase de alta ordem, o que permite melhorar a resposta transitória do sistema no caso nominal. Porém, quando temos erros de modelagem, essa ação de controle de mais alta frequência não pode ser usada, dado que aumenta o módulo de  $\frac{C(z)G_n(z)}{1+C(z)G_n(z)}$  nas médias e altas frequências, fazendo que a condição de robustez não seja satisfeita. Assim, o PS perde suas vantagens frente a um controle mais simples. Em outras palavras, quando ajustamos o PS para uma condição de robustez, em um caso com erro de modelagem importante, podemos ajustar um PID que seja uma aproximação em baixas frequências do  $C_{PS}$  e, assim, obtemos respostas muito similares com os dois controladores. Vamos ilustrar no próximo exemplo essa análise.

**Exemplo 6.3.** *Considere novamente o controle de temperatura do trocador de calor água-vapor do Exemplo 6.2. Vamos comparar a resposta em frequência do  $C_{PS}$  para o ajuste de  $K_c = 2$  com a do PID ajustado com os parâmetros do exemplo anterior. O resultado pode ser visto na Figura 6.7. Observe que, em baixas frequências, o PID aproxima muito bem o  $C_{PS}$ , o que explica o comportamento observado no domínio do tempo.*

Resumindo, o PS se apresenta como uma solução interessante para sistemas estáveis com atraso que não exijam respostas para perturbações na entrada do processo mais rápidas do que em malha aberta. Nos casos de sistemas com pouco erro de modelagem, o PS consegue respostas melhores que um controlador clássico, como o PID. Nos outros casos, o PS ainda consegue um bom compromisso entre robustez e desempenho, porém a vantagem se torna praticamente nula para erros de modelagem grandes.

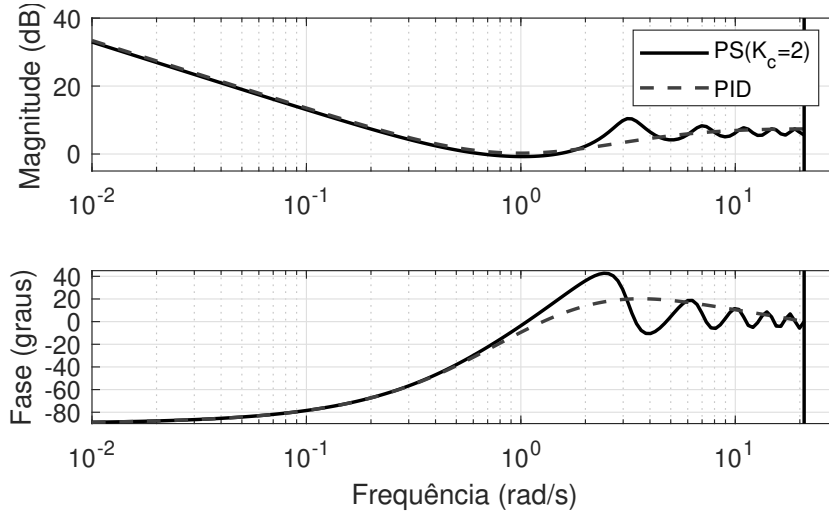


Figura 6.7: Exemplo 6.3 – Comparação das respostas em frequência do controle equivalente do PS e do PID.

Já para processos integradores ou instáveis, temos que usar uma estratégia DTC diferente, que suporte esse tipo de particularidade da planta. Várias alternativas foram propostas na literatura com esse objetivo e muitas delas são discutidas em [NRC07]. Aqui, vamos apresentar uma delas, o Preditor de Smith Filtrado (PSF), que é uma derivação muito simples do PS que pode ser usada em processos com atraso estáveis, integradores e instáveis.

A estrutura do PSF é apresentada Figura 6.8, e é uma modificação do PS que adiciona o filtro  $F_e(z)$  no erro de predição, conferindo mais um grau de liberdade para seu ajuste. Nessa nova configuração, a predição enviada ao controlador primário é dada por:

$$\begin{aligned} Y_p(z) &= G_n(z)U(z) + F_e(z)Y(z) - P_n(z)F_e(z)U(z) \\ &= G_n(z) [1 - z^{-d}F_e(z)] U(z) + F_e(z)Y(z). \end{aligned} \quad (6.4)$$

As funções de transferência de malha fechada com o PSF para a referência e a perturbação, no caso nominal, são dadas, respectivamente, por:

$$\frac{Y(z)}{R(z)} = \frac{C(z)P(z)}{1 + C(z)G(z)}, \quad (6.5)$$

$$\frac{Y(z)}{Q(z)} = P(z) \left[ 1 - \frac{C(z)P(z)}{1 + C(z)G(z)} F_e(z) \right]. \quad (6.6)$$

A estabilidade robusta é garantida se:

$$\left| \frac{C(z)G_n(z)}{1 + C(z)G_n(z)} \right| |F_e(z)| \leq \left| \frac{1}{\delta P(z)} \right|, \quad \forall z = e^{\pm j\omega T_s}, \quad \omega \in [0, \pi/T_s). \quad (6.7)$$

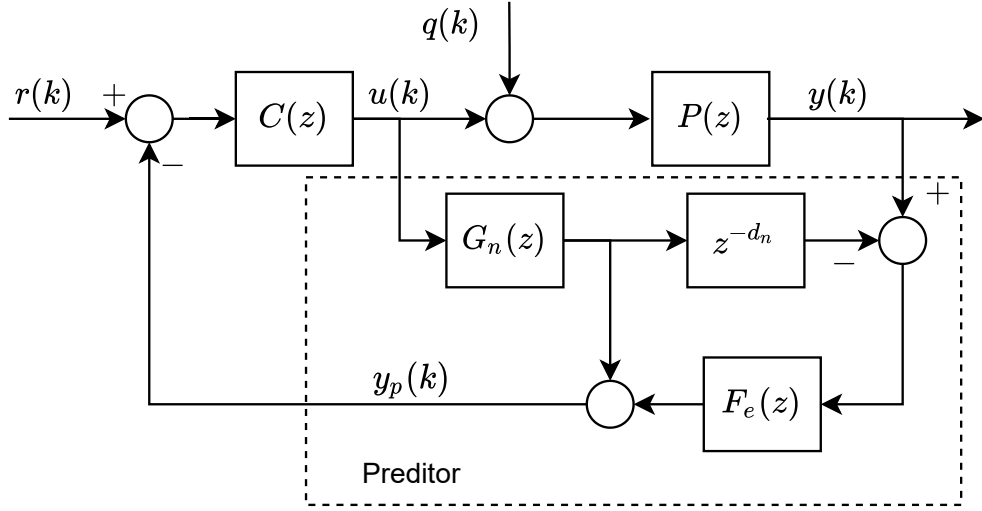


Figura 6.8: Diagrama de blocos do Preditor de Smith Filtrado (PSF).

Observe que o PSF tem a mesma função de transferência  $\frac{Y(z)}{R(z)}$  que o PS e também compensa o atraso do sistema de tal forma que, no caso nominal, o atraso não aparece no denominadores das funções de transferência de malha fechada. Por outro lado, tanto a resposta à perturbação como a condição de robustez são afetadas pelo filtro. Desta forma, escolhendo, no caso nominal, um controle primário  $C(z)$  que atenda uma condição de seguimento de referência, o filtro  $F_e(z)$  pode ser usado para modificar a robustez e a resposta de rejeição de perturbações sem modificar  $\frac{Y(z)}{R(z)}$ .

Adicionalmente, o filtro do erro de predição  $F_e(z)$  pode ser usado para permitir o uso do PSF com plantas integradoras ou instáveis, garantindo a estabilidade interna do sistema. Pode, ainda, permitir melhorar a rejeição de perturbação ou a robustez de malha fechada, dependendo dos requisitos do problema de controle considerado. Observe que podem acontecer situações nas quais essas duas especificações não possam ser atendidas simultaneamente, geralmente quando a faixa de frequências do erro de modelagem e da dinâmica da resposta à perturbação sejam as mesmas. Observe que melhorar a condição de robustez em uma dada faixa de frequências precisa de valores baixos de  $|F_e(e^{\pm\omega T_s j})|$ , o que implica em respostas mais lentas para a perturbação. Esse compromisso deve ser resolvido caso a caso.

Podemos analisar o projeto do filtro utilizando, novamente, o controlador



equivalente do PSF que, neste caso, implementa uma estratégia de controle:

$$U(z) = C_{PSF}(z) [R_f(z) - Y(z)], \quad R_f(z) = \frac{1}{F_e(z)} R(z),$$

com

$$C_{PSF}(z) = \frac{C(z)F_e(z)}{1 + C(z)G_n(z)(1 - z^{-d_n}F_e(z))}. \quad (6.8)$$

Definindo  $F_e(z) = N_e(z)/D_e(z)$ , temos:

$$C_{PSF}(z) = \frac{\frac{N_c(z)N_e(z)}{D_c(z)D_e(z)}}{1 + \frac{N_e(z)}{D_c(z)} \frac{N_g(z)}{D_g(z)} (1 - z^{-d_n} \frac{N_e(z)}{D_e(z)})},$$

$$C_{PSF}(z) = \frac{\frac{N_c(z)N_e(z)}{D_c(z)D_e(z)}}{\frac{D_c(z)D_g(z)D_e(z) + N_c(z)N_g(z)(D_e(z) - z^{-d_n}N_e(z))}{D_c(z)D_g(z)D_e(z)}},$$

que pode ser colocado como:

$$C_{PSF}(z) = \frac{N_c(z)D_g(z)N_e(z)}{D_c(z)D_g(z)D_e(z) + N_c(z)N_g(z)(D_e(z) - z^{-d_n}N_e(z))},$$

ou, usando todos os polinômios em  $z$ :

$$C_{PSF}(z) = \frac{N_c(z)D_g(z)N_e(z)z^{d_n}}{z^{d_n}D_c(z)D_g(z)D_e(z) + N_c(z)N_g(z)(D_e(z)z^{d_n} - N_e(z))}.$$

Analisando este controlador equivalente do PSF, observamos que se  $F_e(z)$  é ajustado para que  $(D_e(z)z^{d_n} - N_e(z))$  seja divisível por  $D_g$ , isto é,

$$D_e(z)z^{d_n} - N_e(z) = D_g(z)D_x(z),$$

sendo  $D_x(z)$  o quociente da divisão, o controlador pode ser escrito como:

$$C_{PSF}(z) = \frac{N_c(z)D_g(z)N_e(z)z^{d_n}}{z^{d_n}D_c(z)D_g(z)D_e(z) + N_c(z)N_g(z)D_g(z)D_x(z)},$$

ou, equivalentemente, dividindo numerador e denominador por  $D_g$ :

$$C_{PSF}(z) = \frac{N_c(z)N_e(z)z^{d_n}}{z^{d_n}D_c(z)D_e(z) + N_c(z)N_g(z)D_x(z)}.$$

Assim, o controlador não cancela mais os polos do modelo e o PSF não é mais um controle por cancelamento, o que permite sua utilização com plantas estáveis e instáveis. Além disso, essa estratégia permite um ajuste da

resposta à perturbação na entrada mais rápida que a de malha aberta, o que não é possível na formulação original do PS. Ainda, se o controlador primário for integrador, basta escolher o filtro com ganho estático unitário,  $F_e(1) = 1$ , para que o termo  $z^{d_n}D_e(z) - N_e(z)$  seja zero em  $z = 1$ . Como consequência dessa escolha, o denominador de  $C_{PSF}(z)$  também tem uma raiz em  $z = 1$ , ou seja, o controlador equivalente do PSF é um controlador com ação integral. Veja que, neste caso, o controlador a ser implementado  $C_{PSF}(z) = N_{PSF}(z)/D_{PSF}(z)$  (transformando a equação em  $z$  em uma equação a diferenças) tem ordem  $n_{PSF} = d + n_c + n_e$ , sendo  $n_e$  a ordem do filtro. Note que mesmo que  $C(z)$  não tenha ação integral, é possível ajustar  $F_e(z)$  para que  $C_{PSF}$  seja integrador fazendo que  $z = 1$  seja raiz do denominador de  $C_{PSF}(z)$ .

Na prática, não é necessário evitar que o controlador cancele todos os polos do modelo, mas apenas aqueles que são indesejados, isto é, que sejam lentos ou possuam módulo maior ou igual a 1 (polos marginalmente estáveis ou instáveis). Assim, a condição que deve ser imposta ao filtro é:

$$(D_e(z)z^{d_n} - N_e(z))\big|_{z=z_i} = 0, \quad \forall z_i, \quad (6.9)$$

para cada polo de malha aberta  $z_i$  indesejado do modelo da planta. Esta condição faz com que estas raízes sejam eliminadas do numerador do controlador. No caso de raízes indesejadas múltiplas, a condição anterior se modifica para:

$$\left. \frac{d^l(D_e(z)z^{d_n} - N_e(z))}{dz^l} \right|_{z=z_i} = 0, \quad \forall z_i, \forall l = 0, 1, \dots, M_i - 1, \quad (6.10)$$

com  $M_i$  a multiplicidade da raiz  $z_i$ .

A condição anterior também pode ser analisada diretamente em (6.6), que pode ser escrita como:

$$\frac{Y(z)}{Q(z)} = \frac{z^{-d_n}N_g(z)}{D_g(z)} \left[ \frac{D_c(z)D_g(z)D_e(z)z^{d_n} + N_c(z)N_g(z)(D_e(z)z^{d_n} - N_e(z))}{(D_c(z)D_g(z) + N_c(z)N_g(z))D_e(z)} \right],$$

onde pode ser observado que, pela condição imposta, o numerador do segundo termo da equação tem raízes em  $z_i$  e, portanto, os polos indesejados do modelo do processo não aparecem mais na resposta à perturbação. O Exercício 1 da lista ao final do capítulo analisa essa dedução.

Finalmente, podemos analisar apenas a estrutura do preditor, isto é, a forma de calcular  $y_p$  como função da entrada  $u$  e saída  $y$  do processo:

$$\begin{aligned} Y_p(z) &= G_n(z)U(z) - P_n(z)F_e(z)U(z) + F_e(z)Y(z) \\ &= G_n(z) [1 - z^{-d_n}F_e(z)] U(z) + F_e(z)Y(z). \end{aligned}$$

Esta equação pode ser escrita de forma compacta como:

$$Y_p(z) = S(z)U(z) + F_e(z)Y(z),$$

com  $S(z) = G_n(z)[1 - z^{-d_n}F_e(z)]$ . Veja que a condição imposta para o filtro  $F_e(z)$  assegura que  $S(z)$  seja sempre estável e não possua os polos indesejados de  $G_n(z)$ , independentemente das características de  $G_n(z)$ , dado que as raízes indesejadas do denominador de  $G_n(z)$  são também raízes do numerador de  $[1 - z^{-d_n}F_e(z)]$ . Porém, é necessário que a implementação do PSF deixe explícito  $S(z)$ , como mostrado na Figura 6.9, e que  $S(z)$  seja implementado com o cancelamento das raízes indesejadas que aparecem tanto no numerador quanto no denominador. Se  $F_e(z)$  tem ganho estático unitário, então  $S(z)$

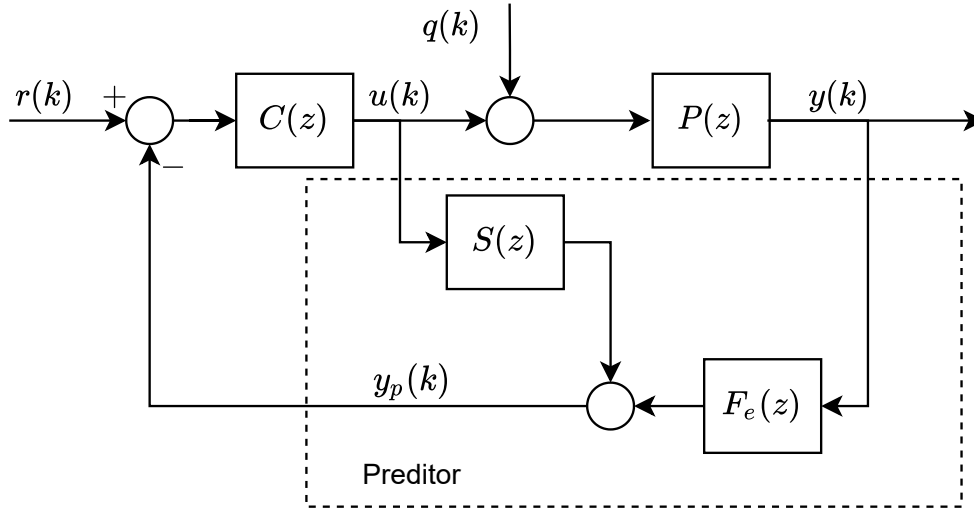


Figura 6.9: Diagrama de blocos do Preditor de Smith Filtrado (PSF) para garantir estabilidade interna.

tem ganho estático zero e sempre  $Y_p = Y$  em regime permanente. Desta forma, e diferentemente do que ocorre no PS, o PSF sempre tem uma predição estável, mesmo que a planta nominal não seja estável. Note que caso o filtro não seja usado, se  $G_n$  é instável (assumindo  $F_e(z) = 1$ , como no PS), então a predição não é convergente.

Note que o ajuste de  $F_e(z)$  regula a velocidade da resposta do preditor. Desta forma, o ajuste de  $F_e(z)$  pode ser realizado, caso desejado, para acelerar a resposta de rejeição de perturbações do sistema sempre que a estabilidade e robustez do sistema sejam atendidas. Ou seja, do ponto de vista prático, o ajuste de  $F_e(z)$  deve atender o compromisso desempenho-robustez,

como em qualquer outra estratégia de controle realimentado, e valem aqui as mesmas considerações realizadas na comparação entre o PS e o PID quando se controlam processos considerando erros de modelagem muito grandes.

**Nota 6.1.** *Note que esta análise do PSF é similar à usada no GDMC para conseguir a estabilidade da resposta livre no caso de processos não estáveis. A forma do filtro e seu ajuste utilizam os mesmos conceitos já apresentados para o GDMC.*

Na sequência, vamos a apresentar um exemplo do uso do PSF para ilustrar as etapas de cálculo para um processo com característica integradora em malha aberta.

**Exemplo 6.4.** *Considere o controle de temperatura de um boiler que tem dinâmica integradora e modelo dado por:*

$$T(z) = \frac{0,2z^{-4}}{z-1}[2U(z) + Q(z)],$$

sendo  $T$  a temperatura a ser controlada,  $U$  a potência aplicada de aquecimento e  $Q$  uma perturbação associada ao sistema de atuação. A amostragem é feita com período de 1 minuto. Com base no modelo do processo, definimos  $G_n(z) = \frac{0,4}{z-1}$  e  $d_n = 4$ .

Deseja-se controlar este sistema para rejeitar as perturbações do tipo degrau de  $Q$  e obter respostas não oscilatórias em malha fechada, com erro zero de seguimento a degraus de referência  $T_r$  e com tempos de resposta depois do atraso de 15 minutos, para ambas as dinâmicas.

Um preditor de Smith, em sua proposta tradicional, não pode ser usado nesse sistema pois não rejeitaria perturbações do tipo degrau. Um PSF pode ser usado para atender todas as especificações. A resposta ao degrau de referência pode ser simplesmente ajustada com um  $C(z) = K_p$ , dado que o processo é integrador. Assim:

$$\frac{T}{T_r} = \frac{K_p G_n(z)}{1 + K_p G_n(z)} z^{-d_n},$$

$$\frac{T}{T_r} = \frac{0,4K_p}{z-1+0,4K_p} z^{-4}.$$

Para uma reposta de 15 minutos depois do atraso, temos uma constante de tempo de 5 minutos e, assim, o polo de tempo discreto desejado é  $z_0 = e^{-1/5} \approx 0,82$ . Para atender essa especificação, o ganho do controlador pode ser definido a partir do denominador da função de transferência de malha fechada deduzida acima e ele tem que satisfazer  $-1+0,4K_p = -0,82$ , logo

$K_p = 0,4532$ . Com este ajuste, a função de transferência de malha fechada resultante para mudanças de referência é:

$$\frac{T}{T_r} = \frac{0,18}{z - 0,82} z^{-4}.$$

Já para a perturbação, temos:

$$\frac{T}{Q} = \frac{0,2}{z - 1} z^{-4} \left[ 1 - \frac{0,18}{z - 0,82} z^{-4} F_e(z) \right] = \frac{0,2}{z - 1} z^{-8} \left[ z^4 - \frac{0,18}{z - 0,82} F_e(z) \right].$$

Para ter uma  $\frac{T}{Q}$  estável e com ganho estático nulo, precisamos que a expressão  $z^4 - \frac{0,18}{z - 0,82} F_e(z)$  tenha dois zeros em  $z = 1$ , um para cancelar o termo integrador do denominador do modelo da planta e outro para adicionar a condição de ganho estático nulo. Isso pode ser obtido com a condição:

$$\left. \frac{d^l \left[ z^4 - \frac{0,18}{z - 0,82} F_e(z) \right]}{dz^l} \right|_{z=1} = 0$$

para  $l = 0$  e  $l = 1$ . Ainda, para conseguir a resposta desejada em termos de constante de tempo, temos que considerar a posição dos polos de  $F_e(z)$ , que vão afetar a resposta para a perturbação. Um ajuste simples do filtro  $F_e(z)$  neste caso pode ser:

$$F_e(z) = \frac{z - a}{z(1 - a)},$$

que tem ganho estático unitário (e assim cumpre a condição para  $l = 0$ ), tem um polo em  $z = 0$  (o que apenas atrasa a resposta uma amostra) e tem um parâmetro de ajuste para conseguir a condição para  $l = 1$ . Observe que com essa solução os polos de malha fechada de  $\frac{T}{Q}$  serão  $z = 0,82$  e  $z = 0$ , garantindo, aproximadamente, a especificação. Usando esse filtro temos:

$$z^4 - \frac{0,18}{z - 0,82} \frac{z - a}{z(1 - a)} = \frac{(1 - a)z^5(z - 0,82) - 0,18z + 0,18a}{(1 - a)z(z - 0,82)}.$$

Como o denominador é diferente de zero em  $z = 1$  podemos considerar apenas o numerador e aplicar a condição para  $l = 1$ , obtendo:

$$\begin{aligned} 5(1 - a)z^4(z - 0,82) + (1 - a)z^5 - 0,18 \Big|_{z=1} &= 0 \\ 5(1 - a)(1 - 0,82) + (1 - a) - 0,18 &= 0 \\ 0,9(1 - a) + (1 - a) - 0,18 &= 0 \\ 1,9(1 - a) &= 0,180 \\ a &= \frac{1,9 - 0,18}{1,9} \\ a &\cong 0,9049, \end{aligned}$$

assim,

$$F_e(z) = \frac{10,517(z - 0,9049)}{z},$$

$$\frac{T}{Q} = \frac{0,2}{z-1} z^{-8} \left[ z^4 - \frac{0,18}{z-0,82} \frac{10,517(z-0,9049)}{z} \right],$$

$$\frac{T}{Q} = \frac{0,2}{z-1} z^{-8} \left[ \frac{z^5(z-0,82) - 1,9063(z-0,9049)}{(z-0,82)z} \right],$$

e é facilmente verificável que

$$z^5(z-0,82) - 1,9063(z-0,9049) = z^6 - 0,82z^5 - 1,906z + 1,725$$

tem duas raízes em  $z = 1$ :

$$z^6 - 0,82z^5 - 0,72z + 0,54 = (z-1)^2(z^2 + 1,861z + 1,316)(z^2 - 0,6795z + 1,31),$$

cumprindo as condições necessárias para a rejeição de perturbações, ou seja,  $\frac{T}{Q}(1) = 0$  e a função de transferência ser estável.

Os resultados de uma simulação considerando o filtro e o controlador primário projetados são apresentados na Figura 6.10, considerando uma mudança de referência em  $t = 5$  min de valor 1, e uma perturbação de  $Q$  de valor 0,2 em  $t = 40$  min. Como pode ser observado, as respostas obtidas para seguimento de referência e rejeição de perturbações atendem às especificações.

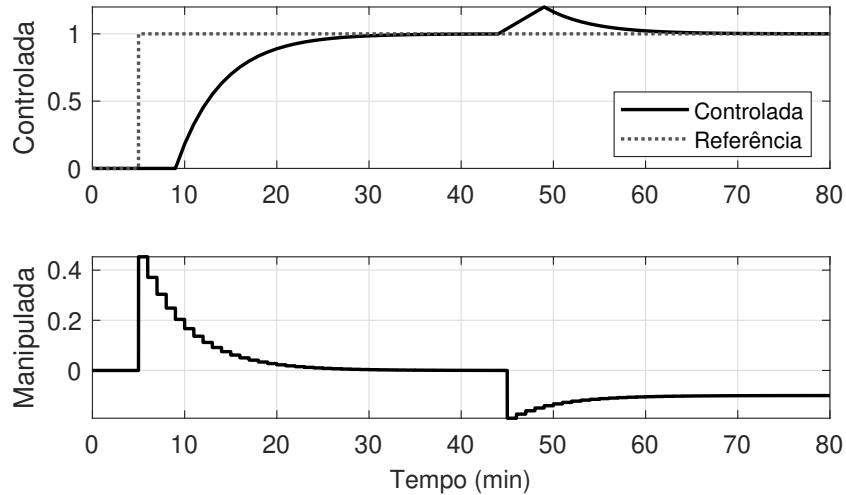


Figura 6.10: Exemplo 6.4 – respostas do PSF para a temperatura do boiler, caso nominal.

Para analisar a robustez do controle, simulamos o segundo cenário, com um erro de 6,5% na estimação do atraso do processo. Os resultados desse segundo cenário podem ser vistos na Figura 6.11. Veja que já há uma deterioração da resposta do sistema, que é mais evidente no sinal de controle, que apresenta um comportamento oscilatório.

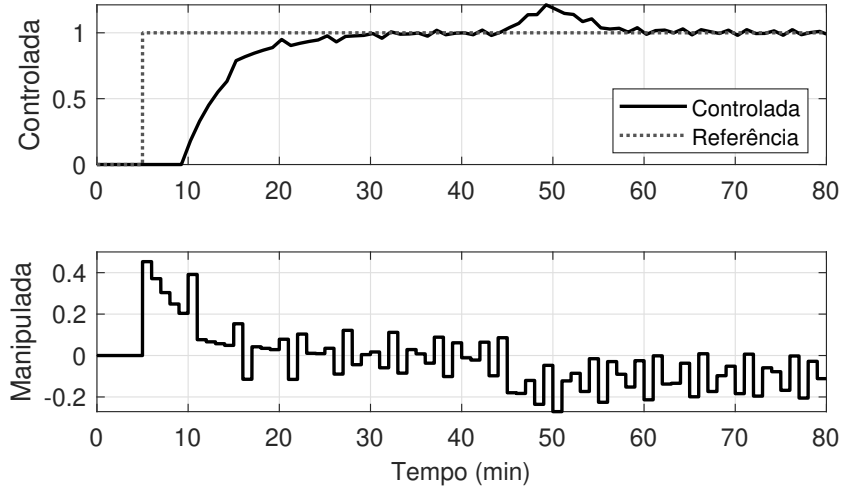


Figura 6.11: Exemplo 6.4 – respostas do PSF para a temperatura do boiler, caso com erro de modelagem.

Uma forma de melhorar este comportamento é tornar o sistema mais lento, o que pode ser conseguido com um novo ajuste do controlador primário e um novo ajuste do filtro, buscando polos de malha fechada no intervalo  $(0.82, 1)$ .

Os resultados obtidos com sinais de perturbação tipo degrau podem ser generalizados para outros tipos de perturbação, bastando, para isso, que as condições no filtro  $F_e(z)$  sejam adaptadas para o tipo de perturbação considerado. Por exemplo, para uma perturbação tipo rampa, precisamos que a função de transferência entre a perturbação e a saída tenha dois zeros em  $z = 1$ , o que implica que a condição:

$$\left. \frac{d^l(D_e(z)z^{d_n} - N_e(z))}{dz^l} \right|_{z=1} = 0, \forall l = 0, 1, \dots, M_1 - 1, \quad (6.11)$$

seja verificada com  $M_1 = 2$ , para o caso de plantas estáveis, e  $M_1 = 3$  para plantas integradoras. Note que, neste último caso, precisamos três zeros em  $z = 1$ , um para eliminar a dinâmica integradora e os dois restantes para conseguir a condição de rejeição em regime permanente para a rampa.

Outros tipos de perturbação bastante comuns na prática são os sinais senoidais, que podem ser boas aproximações de perturbações oscilatórias, como as causadas por malhas de controle liga-desliga dentro do processo. Neste caso, precisamos que a condição no filtro (6.9) seja verificada para  $z_1 = e^{j\omega_0 T_s}$  e  $z_2 = e^{-j\omega_0 T_s}$ , sendo  $\omega_0$  a frequência da perturbação senoidal considerada. O Exercício 3 da lista trata desse problema.

Uma vez analisada a forma como uma estratégia DTC controla um processo com atraso, passamos a um exemplo comparativo entre o PS e o GPC.

**Exemplo 6.5.** *Considere o mesmo controle de temperatura do trocador de calor do Exemplo 6.2, mas consideramos agora também o efeito de uma perturbação atuando diretamente na saída do processo, dada por  $q_e$ . Assim, o modelo considerado é:*

$$T(z) = \frac{0,1}{z - 0,9} z^{-10} [U(z) + T_e(z)] + Q_e(z).$$

Um GPC, ajustado com  $N_1 = 11$ ,  $N_2 = 25$ ,  $N_u = 5$  e  $\lambda = 1$ , foi projetado para controlar a temperatura, e foram simuladas variações da temperatura de referência do tipo degrau, assim como perturbações do tipo degrau na temperatura externa e no sinal  $q_e$ .

Os resultados obtidos com este GPC foram comparados com um PS de tempo discreto que utiliza como controle primário,  $C(z)$ , um PI, inicialmente em um caso sem erro de modelagem. O PS foi ajustado com modelo de predição  $G_n(z) = \frac{0,1}{z-0,9}$  e atraso  $d_n = 10$ . Além do PI, a estrutura de controle considera um filtro de referência que cancela o efeito do zero do controlador primário na resposta à referência:

$$C(z) = \frac{3,70(z - 0,77)}{z - 1}, \quad F(z) = \frac{0,23}{z - 0,77}.$$

Os resultados podem ser vistos na Figura 6.12, considerando uma mudança de referência em  $t = 4$  s de valor 0,5, uma perturbação de  $T_e$  de valor 0,1 em  $t = 15$  s, e uma perturbação de  $q_e$  de valor -0,1 em  $t = 25$  s. Como pode ser observado, as respostas obtidas com os dois controladores, no caso nominal para seguimento de referência, são muito similares e ambas mostram como a ação de controle se antecipa, atuando de forma a prever o resultado depois do atraso. Note que, no seguimento de referência, o sinal de controle começa a ter sua amplitude diminuída antes mesmo de a variável de processo começar a se alterar. Por outro lado, os resultados da rejeição de perturbações não são idênticos. O GPC rejeita melhor a perturbação de  $T_e$  e o PS a de  $Q_e$ . Isto pode ser explicado porque os dois controladores calculam de forma diferente as predições. O PS, como explicado, gera um controle equivalente



por cancelamento dos polos do modelo do processo, o que causa a resposta lenta da rejeição de  $T_e$ , o que não acontece no GPC, que consegue gerar uma ação de controle mais rápida. Por outro lado, esta ação de controle mais rápida causa um pico na resposta para  $Q_e$ , que não é observada no PS. Esta propriedade ficará mais clara depois da análise da estrutura de predição do GPC que apresentaremos neste capítulo.

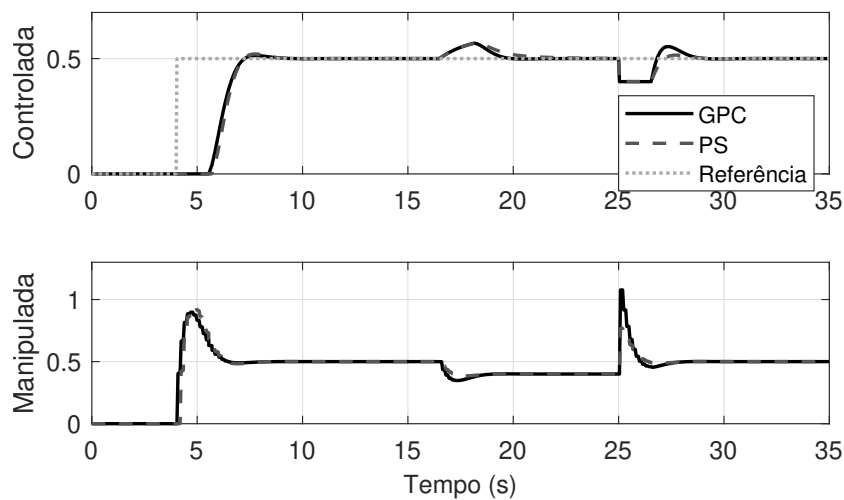


Figura 6.12: Exemplo 6.5 – comparação das respostas do PS e GPC para a temperatura do trocador de calor, caso nominal.

Para analisar a robustez do controle, simulamos, no segundo cenário, um caso no qual existe um erro de 20% na estimação do atraso do processo e 50% de erro na estimação da constante de tempo (mesmo cenário da comparação PS e PID no Exemplo 6.2). Os resultados desse segundo cenário podem ser vistos na Figura 6.13. Observe que o GPC apresenta um comportamento instável e o PS mantém a estabilidade, apesar de ter uma resposta oscilatória. Isto se deve ao fato do GPC ter uma resposta mais rápida para a rejeição de perturbação de entrada  $T_e$ , o que implica em menor índice de robustez, como já discutido na comparação entre o PID e PS.

O exemplo anterior mostra que o GPC atua da mesma forma que o PS, que tem uma compensação de atraso explicitamente colocada na estratégia de controle, mas que apresenta respostas às perturbações e robustez diferentes. Essas diferenças se dão por conta do preditor que o GPC usa internamente, que é diferente do preditor projetado para o PSF. Note que poderíamos repetir o exemplo anterior utilizando um DMC no lugar do GPC, e este caso é explorado no Exercício 6 da lista. Assim, é importante analisar de

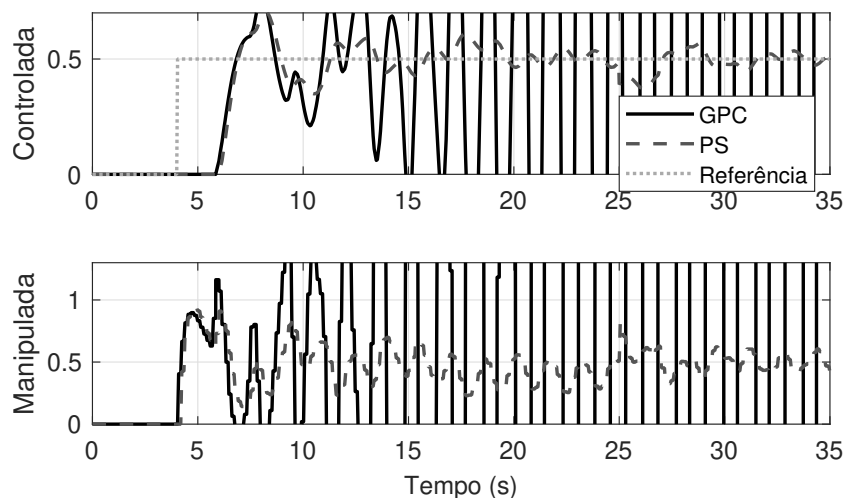


Figura 6.13: Exemplo 6.5 – comparação das respostas do PS e GPC para a temperatura do trocador de calor, caso com erro de modelagem.

forma mais detalhada como funciona a compensação do atraso das principais abordagens de MPC.

Para estudar o efeito do atraso em controladores MPC, é interessante calcular as previsões ótimas em duas etapas: de  $k + 1$  a  $k + d$ , que representa a resposta durante o efeito do atraso, e de  $k + d + 1$  a  $k + N_2$  (ou  $k + d + N$  se usamos um horizonte de predição de tamanho  $N$ ), após o efeito do atraso. Desta forma, o valor da predição em  $k + d$  pode ser colocado em evidência dentro do algoritmo e, assim como no caso do DTC, vamos encontrar uma expressão do controle em função da predição da saída da planta em  $k + d$ . Isso permite separar a parte do algoritmo MPC que calcula a compensação do atraso do restante da lei de controle que usa essa predição para calcular a ação de controle. Iniciaremos o estudo com o GPC e depois realizaremos a análise para o DMC.

## 6.2 Interpretação do GPC como um DTC e o algoritmo DTC-GPC

No Volume 1 desta obra, quando analisamos o controle GPC SISO, foi mostrado como obter o controlador equivalente para o algoritmo GPC no caso sem restrições. Quando o sistema possui atraso de transporte, é possível mostrar que o GPC define implicitamente uma estrutura preditora do tipo PSF, o que explica o comportamento obtido no Exemplo 6.5. Pela forma

como o GPC estima as perturbações, o filtro do erro de predição,  $F_e(z)$ , da estrutura PSF definida implicitamente pelo GPC não é igual ao do PSF definido no exemplo e, por isso, observamos as diferenças no desempenho e na robustez dos dois controladores. Vamos, inicialmente, encontrar o DTC equivalente do GPC para o caso irrestrito para, posteriormente, fazer uma análise e propor uma forma alternativa de lei de controle.

### 6.2.1 Análise do GPC para sistemas com atraso

Nesta seção, vamos obter a estrutura PSF definida implicitamente pelo GPC. Para isso, usaremos a expressão para cálculo das predições futuras da saída dada por:

$$\Delta A(z)\hat{y}(k+j|k) = B(z)\Delta u(k+j-d-1), \quad (6.12)$$

e, como o sistema tem atraso, estamos interessados em calcular as predições para  $j = d+1$  até  $j = d+N$ , que são as utilizadas na função  $J$ , assumindo que  $N_1 = d+1$ , que é a escolha natural para um processo com atraso  $d$ . Para simplificar a notação, assumimos um horizonte de predição com  $N$  termos, definindo  $N_2 = d+N$ .

Vamos, primeiro, calcular as predições de  $j = 1$  a  $j = d$ , que usaremos como condições iniciais para as predições de  $j = d+1$  até  $j = d+N$ . Isso se dá porque uma ação na variável manipulada no instante  $k$  só surtirá efeito em  $k+d+1$ , logo as predições de  $k+1$  a  $k+d$  não dependem do valor atual e nem dos valores futuros da variável manipulada. Observe que, no processo recursivo de cálculo das predições visto no Capítulo do GPC do Volume 1, cada predição é calculada em função de valores passados da saída e do controle, e o número de valores passados a ser usado depende da ordem da equação a diferenças do modelo.

#### Predições de 1 a $d$

Usando a equação diofantina para o caso SISO, as predições futuras podem ser escritas como:

$$\hat{y}(k+j|k) = F_j(z^{-1})y(k) + E_j(z^{-1})B(z^{-1})\Delta u(k+j-d-1), \quad (6.13)$$

e, sabendo que

$$E_j(z^{-1}) = \frac{1 - F_j(z^{-1})z^{-j}}{\Delta A_j(z^{-1})},$$

obtemos:

$$\hat{y}(k+j|k) = F_j(z^{-1})y(k) + \frac{1 - F_j(z^{-1})z^{-j}}{\Delta A(z^{-1})} B(z^{-1})\Delta u(k+j-d-1). \quad (6.14)$$

Observando que a função de transferência do modelo do processo é:

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

podemos simplificar a expressão em (6.14) para:

$$\hat{y}(k+j|k) = F_j(z^{-1})y(k) + (1 - F_j(z^{-1})z^{-j})G(z^{-1})u(k+j-d), \quad (6.15)$$

com  $j = 1, \dots, d$ . Observe a semelhança no cálculo das predições com o PSF. Veja, por exemplo, que para  $j = d$  temos:

$$\hat{y}(k+d|k) = F_d(z^{-1})y(k) + (1 - F_d(z^{-1})z^{-d})G(z^{-1})u(k), \quad (6.16)$$

que tem a mesma estrutura de cálculo das predições do PSF dada por (6.4).

Com (6.15), é possível calcular os valores das predições anteriores a  $d+1$  como funções do sinal medido na planta  $y(k)$  e do sinal de controle aplicado. Esses valores serão utilizados pelo GPC para o cálculo de  $\hat{y}(k+j|k)$  dentro dos horizontes de predição definidos após o atraso ( $j > d$ ). Com isso, podemos fazer um modelo de cálculo das predições usando funções de transferência, o que permite analisar o preditor do GPC usando uma abordagem clássica de controle de tempo discreto.

Vejam, agora, como calcular as predições depois do atraso.

### Predições de $d+1$ a $d+N$

Utilizando novamente os resultados do Capítulo do GPC do Volume 1, considerando o modelo de uma planta com atraso:

$$A(z^{-1})\hat{y}(k+d+j|k) = B(z^{-1})\Delta u(k+j-1), \quad (6.17)$$

e usando a equação diofantina para  $j = 1$  até  $N$ , dada por

$$E_j(z^{-1})A(z^{-1}) = 1 - F_j(z^{-1})z^{-j},$$

com  $F_j(z^{-1})$  de grau  $n_a$ , podemos calcular as predições futuras após o atraso  $d$ . Para isso, multiplicamos (6.17) por  $E_j(z^{-1})$  e usamos a equação diofantina. Assim,

$$[1 - F_j(z^{-1})z^{-j}] \hat{y}(k+d+j|k) = E_j(z^{-1})B(z^{-1})\Delta u(k+j-1), \quad (6.18)$$

ou, de forma equivalente,

$$\hat{y}(k+d+j|k) - F_j(z^{-1})z^{-j}\hat{y}(k+d+j|k) = E_j(z^{-1})B(z^{-1})\Delta u(k+j-1). \quad (6.19)$$

Como  $z^{-j}\hat{y}(k+d+j|k) = \hat{y}(k+d|k)$ , obtemos:

$$\hat{y}(k+d+j|k) = F_j(z^{-1})\hat{y}(k+d|k) + E_j(z^{-1})B(z^{-1})\Delta u(k+j-1). \quad (6.20)$$

Observe que, nesta nova equação, onde  $j$  varia de 1 até  $N$ , as predições futuras após o atraso são escritas em função das predições antes do atraso (de  $k+d$  até  $k+d-n_a$ ), o que nos permite analisar a estrutura PSF equivalente do GPC. Note que, se  $n_a \geq d$ , as predições são as próprias medidas da saída do processo em tempos anteriores a  $k$ , ou seja,  $\hat{y}(k+d-j|k) = y(k+d-j)$ ,  $\forall j \geq d$ . Outro detalhe importante a ressaltar é que, diferentemente do desenvolvimento feito no Volume 1 desta obra (Capítulo relativo ao GPC SISO), aqui usamos os polinômios  $F_j(z^{-1})$  e  $E_j(z^{-1})$  para o cálculo da predição em  $k+d+j$ , e não os polinômios  $F_{d+j}(z^{-1})$  e  $E_{d+j}(z^{-1})$ .

Considere, por exemplo, um sistema de segunda ordem com  $d > 2$ . Assim, o preditor precisa, para o cálculo recursivo das predições de  $d+1$  até  $d+N$ , de três passos anteriores de  $\hat{y}(k+j|k)$  (para um sistema de segunda ordem  $F_j(z^{-1})$  tem grau dois e, desta forma, três coeficientes). Assim, tendo como condição inicial  $\hat{y}(k+d|k)$ ,  $\hat{y}(k+d-1|k)$  e  $\hat{y}(k+d-2|k)$ , podemos calcular todos os valores das predições no horizonte. Isso implica que, usando os polinômios  $F_1(z^{-1})$ ,  $F_2(z^{-1})$  e  $F_3(z^{-1})$ , podemos encontrar a relação entre  $y(k)$ ,  $u(k)$  e as predições usadas, já que, como vimos na seção anterior, as predições até  $k+d$  dependem da saída atual e passada e do controle já aplicado.

Caso o mesmo sistema de segunda ordem tenha  $d \leq 2$ , algumas das predições empregadas serão, na realidade, medições passadas da saída do processo. Se  $d = 1$ , por exemplo, a condição inicial será dada por  $\hat{y}(k+1|k)$ ,  $\hat{y}(k|k) = y(k)$  e  $\hat{y}(k-1|k) = y(k-1)$ .

Vamos considerar, na sequência, que o atraso é maior que a ordem do modelo, ou seja,  $d > n_a$ . Usando os conceitos de resposta livre e forçada, as predições futuras podem, então, ser escritas como:

$$\hat{\mathbf{y}} = \mathbf{G}\Delta \mathbf{u} + \mathbf{f}, \quad (6.21)$$

ou seja, a relação das predições com o controle futuro é a mesma do caso analisado no Capítulo do GPC do Volume 1. Assim,  $\mathbf{G}$  é a mesma já calculada (note que em um sistema com atraso  $d$ , o primeiro coeficiente não nulo

da resposta ao degrau é  $g_{d+1}$ ). Assim, assumindo  $N_2 = d + N$ , temos:

$$\mathbf{G} = \begin{bmatrix} g_{d+1} & 0 & \cdots & 0 \\ g_{d+2} & g_{d+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{d+N} & g_{d+N-1} & \cdots & g_{d+N-N_u+1} \end{bmatrix}$$

e

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k+d+1|k) \\ \hat{y}(k+d+2|k) \\ \vdots \\ \hat{y}(k+d+N|k) \end{bmatrix}.$$

Por outro lado, a resposta livre é escrita de forma diferente:  $\mathbf{f} = \mathbf{F} \underline{\mathbf{y}}_d + \mathbf{H} \underline{\Delta \mathbf{u}}$ , já que agora temos uma dependência com as predições até  $d$ . Nesta equação,  $\underline{\mathbf{y}}_d$  é dado por:

$$\underline{\mathbf{y}}_d = [\hat{y}(k+d|k) \ \hat{y}(k+d-1|k) \ \dots \ \hat{y}(k+d-n_a|k)]^T,$$

e  $\hat{y}(k+d-j|k) = y(k+d-j)$ , se  $j \geq d$ . Já  $\underline{\Delta \mathbf{u}}$  é o vetor de controles passados definido como:

$$\underline{\Delta \mathbf{u}} = \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-n_b) \end{bmatrix},$$

que é similar ao usado no GPC na sua forma tradicional, exceto que o número de controles passados é  $n_b$  e não mais  $n_b + d$ .

O sinal de controle a ser aplicado na planta é obtido minimizando a função custo do problema de controle. Para o caso irrestrito, como mostrado no Volume 1, o incremento do sinal de controle pode ser obtido explicitamente e colocado na forma:

$$\Delta u(k) = \mathbf{K}_1 \left( \mathbf{R} - \mathbf{F} \underline{\mathbf{y}}_d + \mathbf{H} \underline{\Delta \mathbf{u}} \right), \quad (6.22)$$

onde as matrizes  $\mathbf{H} \in \mathbb{R}^{N \times n_b}$  e  $\mathbf{F} \in \mathbb{R}^{N \times n_a+1}$  contêm os coeficientes dos polinômios  $H_j(z^{-1})$  e  $F_j(z^{-1})$ , respectivamente, e são dadas por:

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n_b} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n_b} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N,1} & h_{N,2} & \cdots & h_{N,n_b} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} f_{1,0} & f_{1,1} & \cdots & f_{1,n_a} \\ f_{2,0} & f_{2,1} & \cdots & f_{2,n_a} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N,0} & f_{N,1} & \cdots & f_{N,n_a} \end{bmatrix}.$$

É importante recordar que  $H_j(z^{-1})$  é obtido de  $E_j(z^{-1})B(z^{-1})$  extraindo apenas os coeficientes associados aos controles passados, da mesma forma que foi realizado no GPC original. Ainda na equação, o vetor  $\mathbf{R} \in \mathbb{R}^N$  representa as referências futuras e  $\mathbf{K}_1 \in \mathbb{R}^{1 \times N}$  representa um vetor linha com os ganhos de controlador associados a  $\Delta u(k)$ .

Considerando referências futuras desconhecidas, ou seja, assumindo que as referências futuras são todas iguais à referência atual,  $\mathbf{R} = \mathbf{1}_N r(k)$ , e que a matriz de ganhos pode ser escrita como:

$$\mathbf{K}_1 = [k_1, k_2, k_3, \dots, k_N],$$

então,

$$\Delta u(k) = f_0 r(k) - \sum_{j=0}^{n_a} a_j z^{-j} \hat{y}(k+d|k) - \sum_{j=1}^{n_b} b_j z^{-j} \Delta u(k),$$

onde

$$f_0 = \sum_{i=1}^N k_i, \quad a_j = \sum_{i=1}^N k_i \cdot f_{i,j}, \quad b_j = \sum_{i=1}^N k_i \cdot h_{i,j}. \quad (6.23)$$

Rearranjando

$$\left( \Delta \left[ 1 + \sum_{j=1}^{n_b} b_j z^{-j} \right] \right) u(k) = f_0 r(k) - \sum_{j=0}^{n_a} a_j \hat{y}(k+d-j|k), \quad (6.24)$$

ou, de forma mais compacta, definindo polinômios em  $z$ :

$$D_c(z)u(k) = N_f(z)r(k) - \sum_{j=0}^{n_a} a_j \hat{y}(k+d-j|k), \quad (6.25)$$

com

$$D_c(z) = \Delta \left( 1 + \sum_{j=1}^{n_b} b_j z^{-j} \right), \quad N_f(z) = f_0.$$

Usando o resultado das previsões até  $k+d$  de (6.15), podemos escrever  $\hat{y}(k+d-j|k)$  como:

$$\hat{y}(k+d-j|k) = F_{d-j}(z^{-1})y(k) + (1 - F_{d-j}(z^{-1})z^{-(d-j)})G(z^{-1})u(k-j). \quad (6.26)$$

Então, o último termo de (6.25) se torna

$$\sum_{j=0}^{n_a} \mathfrak{a}_j \hat{y}(k + d - j | k) = \sum_{j=0}^{n_a} \mathfrak{a}_j \left[ F_{d-j}(z^{-1})y(k) + (1 - F_{d-j}(z^{-1})z^{-(d-j)})G(z^{-1})u(k - j) \right].$$

Fatorando os termos que não dependem de  $j$  e dado que  $u(k - j) = z^{-j}u(k)$ , esta última expressão pode ser reescrita como:

$$\begin{aligned} \sum_{j=0}^{n_a} \mathfrak{a}_j \hat{y}(k + d - j | k) &= \left( \sum_{j=0}^{n_a} \mathfrak{a}_j F_{d-j}(z^{-1}) \right) y(k) \\ &\quad + G(z^{-1})u(k) \left( \sum_{j=0}^{n_a} \mathfrak{a}_j z^{-j} \right) \\ &\quad + G(z^{-1})z^{-d}u(k) \left( \sum_{j=0}^{n_a} \mathfrak{a}_j F_{d-j}(z^{-1}) \right). \end{aligned} \quad (6.27)$$

Definindo

$$N_{fe}(z) = \sum_{j=0}^{n_a} \mathfrak{a}_j F_{d-j}(z^{-1}), \quad N_c(z) = \sum_{j=0}^{n_a} \mathfrak{a}_j z^{-j},$$

e dado que o modelo do sistema é

$$P(z) = G(z)z^{-d},$$

é possível escrever a (6.27) como:

$$\begin{aligned} \sum_{j=0}^{n_a} \mathfrak{a}_j z^{-j} \hat{y}(k + d | k) &= N_{fe}(z)y(k) \\ &\quad + G(z)N_c(z)u(k) - G(z)z^{-d}N_{fe}(z)u(k). \end{aligned} \quad (6.28)$$

Fazendo a substituição dessa última equação em (6.25), resulta em:

$$\begin{aligned} D_c(z)u(k) &= N_f(z)r(k) \\ &\quad - \left( N_{fe}(z)y(k) + G(z)N_c(z)u(k) - G(z)z^{-d}N_{fe}(z)u(k) \right), \\ (D_c(z) + G(z)N_c(z) - G(z)z^{-d}N_{fe}(z))u(k) &= N_f(z)r(k) - N_{fe}(z)y(k). \end{aligned}$$



Aplicando a transformada Z, o sinal de controle, no domínio da frequência, é dado por:

$$U(z) = \frac{N_f(z)R(z) - N_{fe}(z)Y(z)}{(D_c(z) + G(z)N_c(z) - G(z)z^{-d}N_{fe}(z))}.$$

Extraindo  $N_{fe}$  do numerador e  $D_c$  do denominador obtemos:

$$U(z) = \frac{\frac{N_{fe}(z)}{D_c(z)}}{1 + G(z)\frac{N_c(z)}{D_c(z)} - G(z)z^{-d}\frac{N_{fe}(z)}{D_c(z)}} \left( \frac{N_f(z)}{N_{fe}} R(z) - Y(z) \right),$$

que pode ser colocado como:

$$U(z) = C_{GPC}(z) (F(z)R(z) - Y(z)),$$

com  $F(z)$  um filtro de referência:

$$F(z) = \frac{N_f(z)}{N_{fe}} = \frac{f_0}{\sum_{j=0}^{n_a} a_j F_{d-j}(z)}, \quad (6.29)$$

e  $C_{GPC}(z)$  um controlador equivalente:

$$C_{GPC}(z) = \frac{\frac{N_{fe}(z)}{D_c(z)}}{1 + G(z)\frac{N_c(z)}{D_c(z)} - G(z)z^{-d}\frac{N_{fe}(z)}{D_c(z)}}.$$

Finalmente, fatorando o segundo termo do denominador de  $C_{GPC}$  com  $G(z)$  e  $N_c/D_c$ , e multiplicando e dividindo o numerador por  $N_c$ , obtemos:

$$C_{GPC}(z) = \frac{\frac{N_c(z)}{D_c(z)} \frac{N_{fe}(z)}{N_c(z)}}{1 + \frac{N_c(z)}{D_c(z)} G(z) \left( 1 - z^{-d} \frac{N_{fe}(z)}{N_c(z)} \right)}.$$

Definindo

$$C(z) = \frac{N_c(z)}{D_c(z)} = \frac{\sum_{j=0}^{n_a} a_j z^{-j}}{\Delta \left( 1 + \sum_{j=1}^{n_b} b_j z^{-j} \right)}, \quad (6.30)$$

$$F_e(z) = \frac{N_{fe}(z)}{N_c(z)} = \frac{\sum_{j=0}^{n_a} a_j F_{d-j}(z)}{\sum_{j=0}^{n_a} a_j z^{-j}}, \quad (6.31)$$

o controlador equivalente do GPC é dado por:

$$C_{GPC} = \frac{C(z)F_e(z)}{1 + C(z)G(z)(1 - z^{-d}F_e(z))},$$

que tem exatamente a mesma expressão da estrutura PSF dada por (6.8). Assim, é possível mostrar que o GPC implementa um preditor de Smith filtrado com filtro de referência com um ajuste definido pelo procedimento de otimização usado no GPC. O cálculo dos coeficientes dos polinômios numerador e denominador das funções de transferência  $F(z)$ ,  $C(z)$  e  $F_e(z)$  pode ser automatizado, como mostrado nos algoritmos disponibilizados na página web do livro<sup>1</sup>.

A partir das expressões deduzidas nesta seção, podemos notar que os ajustes do controlador primário e dos filtros equivalentes do GPC dependem da sintonia utilizada e do modelo do processo. Assim, embora a estrutura seja a mesma que a do PSF, o resultado final em termos de respostas em malha fechada e robustez pode ser diferente. No caso do exemplo anterior, no qual comparamos o PS com o GPC, o ajuste do PI no PS permite obter um  $C(z)$  muito similar ao do GPC. Por isso, os resultados obtidos na resposta para mudanças de referência foram muito similares. Já o filtro do erro de predição tem uma função de transferência muito diferente, o que também explica o resultado tão diferente obtido na rejeição de perturbação e características de robustez. Vamos analisar esse ponto no próximo exemplo.

**Exemplo 6.6.** *Considere o controle do sistema trocador de calor do Exemplo 6.5, com modelo  $T(z) = \frac{0,1}{z-0,9}z^{-10}[U(z) + T_e(z)] + Q_e(z)$ . Calcularemos o controle e filtros equivalentes do GPC para comparar com a solução dada pelo PS, que usa um  $F_e = 1$  e tem um PI como controlador primário.*

*No GPC, usando (6.29), (6.30) e (6.31), podemos calcular:*

$$\begin{aligned} F_e(z) &= \frac{6,7715z - 5,9604}{3,7620z - 2,9509} \approx \frac{1,8(z - 0,8802)}{z - 0,7844}, \\ F(z) &= \frac{0,8111}{6,7715z - 5,9604} \approx \frac{0,1198}{z - 0,8802}, \\ C(z) &= \frac{3,7620z - 2,9509}{z - 1} \approx \frac{3,762(z - 0,7844)}{z - 1}, \end{aligned}$$

*o que mostra que o GPC implementa, neste exemplo, um controle primário do tipo PI e um filtro de predição também de primeira ordem. Como a resposta para a referência não depende do ajuste do  $F_e(z)$  e o PI ajustado pelo GPC tem praticamente o mesmo ganho e zero do PI usado no PS, os resultados obtidos neste caso são, como esperado, muito similares.*

*Já com relação a  $F_e(z)$ , observamos que o filtro implementado pelo GPC tem uma característica passa alta (veja o diagrama de Bode na Figura 6.14), o que explica, para a rejeição de perturbações, os valores maiores das ações*

---

<sup>1</sup><https://danielml.paginas.ufsc.br/livro-mpc/>

de controle no transitório e as respostas mais rápidas. Por outro lado, esta característica passa alta do filtro torna o GPC menos robusto, dado que o filtro afeta a dinâmica de forma a diminuir a tolerância ao erro de modelagem em alta frequência. Como o termo  $\frac{C(z)G(z)}{1+C(z)G_n(z)}$  é praticamente igual nos dois controles, o módulo do filtro estabelece a relação entre a robustez dos dois controles: o PS, neste caso, tem um comportamento mais robusto. A Figura 6.14 mostra a resposta em frequência dos filtros  $F_e(z)$  do GPC e do PS (que vale 1), o que confirma a análise realizada. Para estudar o desem-

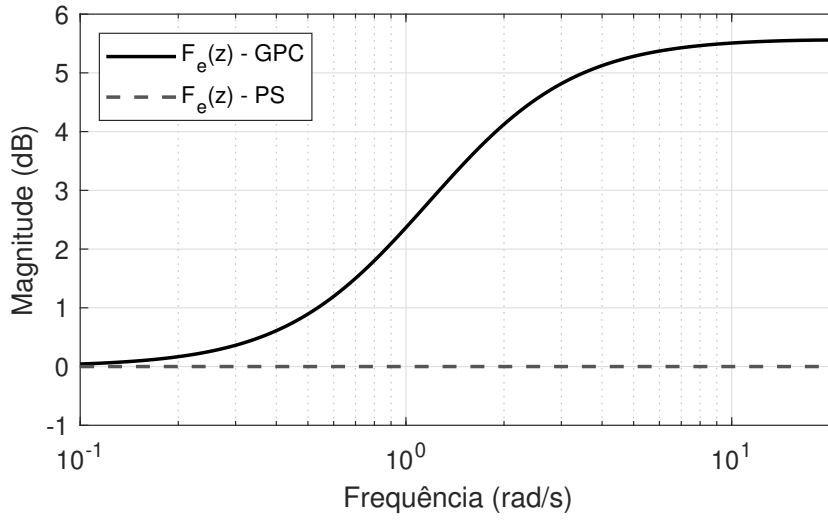


Figura 6.14: Exemplo 6.6 – resposta em frequência dos filtros  $F_e(z)$  do GPC e do PS.

penho, podemos analisar os resultados usando os controladores equivalentes e as funções de transferência  $T/T_e$  e  $T/Q_e$ :

$$\frac{T(z)}{T_e(z)} = P_n(z) \left[ 1 - \frac{C(z)G(z)}{1 + C(z)G_n(z)} z^{-d} F_e(z) \right],$$

$$\frac{T(z)}{Q_e(z)} = \left[ 1 - \frac{C(z)G(z)}{1 + C(z)G_n(z)} z^{-d} F_e(z) \right],$$

considerando que a única diferença entre os dois casos é o valor de  $F_e(z)$ . O zero do  $F_e(z)$  no GPC permite compensar o efeito do polo lento de  $P_n(z)$  em  $T/T_e$ , conseguindo assim a melhor resposta para a perturbação de  $T_e$ , de fato,  $F_e(z)$  força um zero em  $z = 0,9$  no termo  $\left[ 1 - \frac{C(z)G(z)}{1 + C(z)G_n(z)} z^{-d} F_e(z) \right]$ . Porém, como  $T/Q_e$  não tem esse polo lento, o mesmo zero torna-se dominante na resposta da perturbação  $Q_e$ , causando o pico observado no transitório. Por isso, apesar de mais lenta, a resposta do PS para  $Q_e$  acaba sendo melhor

*que no GPC se consideramos, por exemplo a integral do erro de seguimento para a referência. Podemos dizer que o PS tem um melhor compromisso robustez-desempenho que o GPC, pois é mais robusto, tem uma rejeição de  $T_e$  pior que no GPC (mais lenta) mas uma resposta melhor para  $Q_e$  (mais lenta mas com muito menor pico e menor integral do erro). De forma geral, quando o sistema é mais robusto, a rejeição de perturbações na entrada do processo é sempre mais lenta, mas para as perturbações na saída do processo pode ser que essa lentidão ajude a obter uma resposta com menor pico e consequentemente menor integral do erro.*

Da análise anterior, complementada com os resultados do Exemplo 6.6, podemos concluir que para ajustar as características de rejeição de perturbações ou robustez de um GPC, quando se controla um processo com atraso, podemos analisar o seu controlador equivalente e considerar o ajuste do filtro do erro de predição mais adequado para a situação que se deseja resolver. Se utilizamos o ajuste de  $C(z)$  para atender uma especificação desejada para a resposta da referência do sistema em malha fechada, o filtro  $F_e(z)$  será o único parâmetro livre para ajustar o compromisso entre robustez e rejeição de perturbações. Na prática, o problema é que a relação entre o filtro do erro de predição e os ajustes do GPC (horizontes e ponderações) não é direta e nem simples, assim, o ajuste do controle para levar em conta aspectos de robustez pode ser tornar complexo. Por outro lado, o ajuste do controlador primário da estrutura usando a minimização da função objetivo pode ser mais adequada e mais simples do que o ajuste de um controle clássico, principalmente para sistemas complexos. Porém, o ajuste das características do filtro  $F_e(z)$  no PSF é mais direto e intuitivo.

Assim, parece interessante unir as qualidades das duas estratégias de controle em um novo algoritmo, que calcule as predições até o atraso  $d$  usando a estratégia de um PSF e um controlador GPC para encontrar o controlador primário. Isto é, calculam-se as ações de controle ótimas tendo como base as predições até  $d$  calculadas pelo preditor filtrado derivado do PSF. Então, o problema em questão é dividir a estratégia em duas partes: (i) o projeto do GPC é realizado para um processo sem atraso, que tem como saída  $y_p(k)$  e como entrada de controle  $u(k)$ ; (ii) um preditor de Smith filtrado é usado no processo para calcular  $y_p(k)$  a partir do conhecimento da saída do processo  $y(k)$  e do seu controle  $u(k)$ . Com essa estratégia, o GPC enxerga um novo processo a ser controlado, que é a junção do processo real e o preditor, que não possui atraso.

A implementação dessa estratégia é bem simples, bastando, no procedimento aqui apresentado, alterar o cálculo das predições de  $k + 1$  até  $k + d$  e manter exatamente os mesmos cálculos para encontrar o sinal de controle

do GPC, isto é, para o cálculo da ação de controle, seja no caso sem ou com restrições.

Como vimos, as previsões até  $k + d$  apenas afetam a resposta livre e não a resposta forçada do GPC. Desta forma, a modificação proposta apenas altera a resposta livre do GPC e não modifica o restante do algoritmo.

Se voltamos à forma tradicional de implementação do GPC, precisamos, a cada amostra, calcular o vetor de resposta livre que contém as previsões de  $k + N_1$  até  $k + N_2$  com controle futuro nulo. Num cálculo recursivo, calcularemos todas as previsões de 1 a  $N_2$  ficando apenas com as que usaremos na função objetivo. No novo algoritmo, iniciaremos o cálculo da resposta livre usando um PSF até calcular a previsão em  $k + d$ . Com estes valores já calculados, acharemos todas as previsões de  $k + d + N_1$  até  $k + d + N_2$  usando o procedimento padrão do GPC. Este novo algoritmo, que foi denominado DTC-GPC, pois tem um compensador de atraso embutido explicitamente no MPC, foi proposto originalmente em [NR99] e também é descrito em [NRC07].

### 6.2.2 O algoritmo DTC-GPC

Nesta seção, analisaremos alguns aspectos do DTC-GPC que, como explicado, é derivado do PSF e do GPC. As vantagens deste controlador para sistemas com atraso são as seguintes:

- para especificações de seguimento de referência, ele tem exatamente o mesmo desempenho que um GPC;
- para especificações de rejeição de perturbações e robustez, ele permite, de maneira mais simples, atender a um compromisso entre as duas especificações;
- como o algoritmo apenas altera a forma de calcular a resposta livre do GPC, ele pode ser estendido para o caso com restrições de forma direta, dado que a resposta livre é uma das entradas do algoritmo de otimização;
- a extensão para o caso não linear também é direta, sempre que for possível calcular no modelo não linear uma saída do processo sem atraso;
- a extensão para o caso multivariável também é relativamente simples e utiliza as características do PSF multivariável, que pode ser usado para controlar todo tipo de planta com atraso.

A estrutura geral do DTC-GPC para o caso linear monovariável, com e sem restrições, pode ser vista na Figura 6.15. Observamos que existem três blocos nesta estratégia: preditor, resposta livre e otimização. O bloco do preditor é encarregado de calcular a predição da saída do processo até o atraso com base no PSF. Um segundo bloco calcula o vetor de resposta livre, usando como condição inicial a informação fornecida pelo preditor. Finalmente, um bloco de otimização calcula o sinal de controle a ser aplicado no processo e, para isso, utiliza como entradas a resposta livre, a função custo e as restrições do problema. Assim, o algoritmo DTC-GPC em si é dado pelos seguintes passos:

1. projeta-se o GPC para o processo sem atraso;
2. projeta-se o Preditor de Smith filtrado;
3. a cada período de amostragem:
  - (a) calculam-se as saídas  $y_p$  utilizando o PSF;
  - (b) utilizam-se  $y_p$  e os sinais de controle passados para encontrar a nova ação de controle a partir de (6.22) (ou do problema de otimização com restrições);
  - (c) aplica-se o novo sinal de controle.

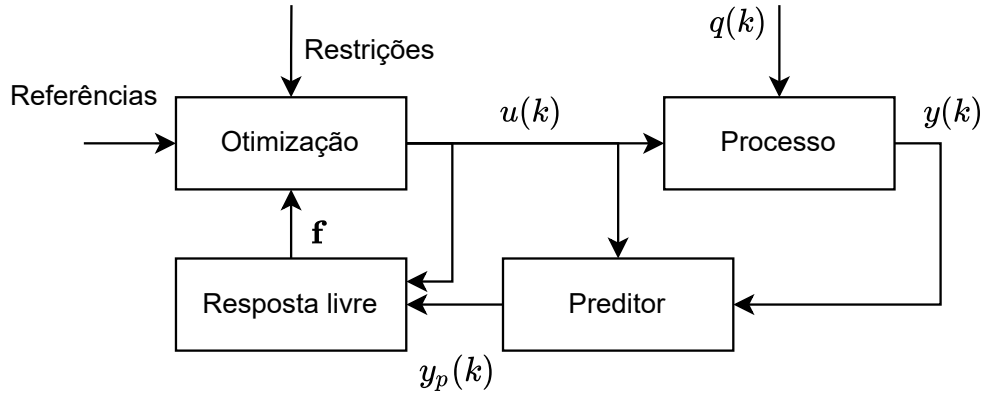


Figura 6.15: Estrutura do DTC-GPC

Para analisar com mais detalhes a implementação das predições no DTC-GPC, basta lembrar que no PSF calculávamos a predição usando a equação:

$$Y_p(z) = S(z)U(z) + F_e(z)Y(z), \quad (6.32)$$

onde  $S(z) = G(z)(1 - F_e(z)z^{-d})$  é estável e todos os polos de  $F_e(z)$  são polos de  $S(z)$ .

Há algumas maneiras de se implementar a estrutura do PSF. Um alternativa é utilizar diretamente a estrutura mostrada na Figura 6.9, implementando  $S(z)$  e  $F_e(z)$  separadamente através das respectivas equações a diferenças, e somando os respectivos sinais de saída para obter  $y_p$ . Lembre-se que, para o cálculo da resposta livre, o GPC precisa do vetor

$$\underline{y}_d = [\hat{y}(k+d|k) \ \hat{y}(k+d-1|k) \ \dots \ \hat{y}(k+d-n_a|k)]^T,$$

e que

$$\hat{y}(k+d-j|k) = y_p(k-j),$$

com  $j = 1$  até  $n_a$ . Dessa forma, é necessário armazenar os valores passados de  $y_p$  até  $y_p(k-n_a)$  para implementar efetivamente a estrutura DTC-GPC.

Uma segunda forma de implementar o cálculo de  $y_p$  envolve rearranjar (6.32) de tal forma que uma única equação a diferenças seja obtida, simplificando a implementação. Para isso, considere que:

$$S(z) = \frac{N_s(z)}{D_s(z)}, \text{ e } F_e(z) = \frac{N_e(z)}{D_e(z)},$$

com  $D_e(z)$  e  $D_s(z)$  mônicos, e também que  $D_s(z) = D_e(z)D_x(z)$ , onde  $D_x(z)$  é o quociente da divisão de  $D_s(z)$  por  $D_e(z)$ . Assim, podemos escrever:

$$Y_p(z) = \frac{N_s(z)}{D_s(z)}U(z) + \frac{N_e(z)}{D_e(z)}Y(z) = \frac{N_s(z)U(z) + N_e(z)D_x(z)Y(z)}{D_s(z)}, \quad (6.33)$$

ou, de forma equivalente,

$$D_s(z)Y_p(z) = N_s(z)U(z) + N_e(z)D_x(z)Y(z),$$

com

$$D_s(z) = 1 + \sum_{i=1}^{m_s} d_{s,i}z^{-i}, \quad N_s(z) = \sum_{i=1}^{m_u} n_{s,i}z^{-i}, \quad N_e(z)D_x(z) = \sum_{i=0}^{m_x} n_{x,i}z^{-i}.$$

sendo que  $m_s$  é o grau de  $D_s(z)$ ,  $m_u$  é o grau de  $N_s(z)$  e  $m_x$  é o grau de  $N_e(z)D_x(z)$ .

A partir de (6.33), é possível obter uma equação a diferenças recursiva para o cálculo de  $y_p(k)$ :

$$y_p(k) = - \sum_{i=1}^{m_s} d_{s,i}y_p(k-i) + \sum_{i=1}^{m_u} n_{s,i}u(k-i) + \sum_{i=1}^{m_x} n_{x,i}y(k-i),$$

e lembrando que ainda é necessário armazenar os valores passados de  $y_p$  até  $y_p(k-n_a)$ . Vamos aplicar esses dois procedimentos em um exemplo.

**Exemplo 6.7.** Considere um processo com modelo de primeira ordem instável dado por:

$$P_n(z) = \frac{0,2}{z - 1,1} z^{-5}.$$

Neste caso, para obter uma predição estável, consideramos um filtro que verifique  $F_e(1) = 1$  e, para  $z_0 = 1,1$ ,

$$z_0^5 - F_e(z_0) = 0 \rightarrow F_e(z_0) = z_0^5.$$

Assim, propomos um filtro de primeira ordem na forma:

$$F_e(z) = \frac{a_{f,0}z + a_{f,1}}{z - z_f}$$

e precisamos ajustar  $a_{f,1}$  e  $a_{f,0}$  para as duas condições especificadas. O polo do filtro é usado para ajustar a resposta para perturbações e robustez. Neste exemplo, definimos  $z_f = 0,9$ . Assim, as duas condições a serem satisfeitas são:

$$F_e(1) = \frac{a_{f,0} + a_{f,1}}{1 - 0,9} = 1,$$

$$F_e(1,1) = \frac{1,1a_{f,0} + a_{f,1}}{1,1 - 0,9} = (1,1)^5,$$

que levam a:

$$a_{f,0} + a_{f,1} = 0,1 \quad 1,1a_{f,0} + a_{f,1} = 0,2(1,1)^5.$$

Veja que essas duas equações definem um sistema de equações lineares que é facilmente solucionado. Este sistema de equações pode ser representado num formato matricial:

$$\begin{bmatrix} 1 & 1 \\ 1,1 & 1 \end{bmatrix} \begin{bmatrix} a_{f,0} \\ a_{f,1} \end{bmatrix} = \begin{bmatrix} 0,1 \\ 0,2(1,1)^5 \end{bmatrix}$$

Resolvendo o sistema, temos:

$$a_{f,0} = 2,2210, \quad a_{f,1} = -2,1210,$$

que define o filtro

$$F_e(z) = \frac{2,2210z - 2,1210}{z - 0,9} = 2,221 \frac{(z - 0,955)}{z - 0,9}.$$

Com esse filtro, obtemos  $S(z)$  que vale:

$$\begin{aligned} S(z) &= \frac{0,2}{z - 1,1} z^{-5} \left[ z^5 - 2,221 \frac{(z - 0,955)}{z - 0,9} \right] \\ &= \frac{0,2z^{-5}}{z - 1,1} \frac{z^5(z - 0,9) - 2,221(z - 0,955)}{z - 0,9}. \end{aligned}$$



Como o polinômio  $z^5(z - 0,9) - 2,221(z - 0,955)$  é divisível por  $z - 1,1$ , podemos reduzir a expressão anterior a:

$$S(z) = \frac{0,2z^{-5}}{z - 0,9} [z^5 + 0,2z^4 + 0,22z^3 + 0,242z^2 + 0,2662z - 1,9282] .$$

Com  $F_e(z)$  e  $S(z)$  calculados, podemos implementar a estrutura PSF de duas maneiras, como mencionado anteriormente. Na primeira delas, com  $F_e(z)$  e  $S(z)$  implementados separadamente, suponha que  $y_f(k)$  e  $y_s(k)$  são as saídas dos respectivos blocos. Assim,

$$\begin{aligned} y_s(k) &= 0,9y_s(k-1) + 0,2u(k-1) + 0,04u(k-2) + 0,044u(k-3) \\ &\quad + 0,0484u(k-4) + 0,0532u(k-5) - 0,3856u(k-6), \\ y_f(k) &= 0,9y_f(k-1) + 2,221y(k) - 2,121y(k-1), \end{aligned}$$

e a saída predita  $y_p$  é calculada somando estes sinais:

$$y_p(k) = y_s(k) + y_f(k).$$

Utilizando a segunda alternativa, obtemos  $y_p$  diretamente a partir de uma única equação a diferenças. Para este sistema, a partir de (6.33), temos

$$Y_p(z) = \frac{N_s(z)}{D_s(z)}U(z) + \frac{N_e(z)}{D_e(z)}Y(z) = \frac{N_s(z)U(z) + N_e(z)D_x(z)Y(z)}{D_s(z)}$$

com

$$D_s(z) = z - 0,9, \quad N_e(z) = 2,221(z - 0,955), \quad D_x(z) = 1,$$

$$N_s(z) = 0,2 + 0,04z^{-1} + 0,044z^{-2} + 0,0484z^{-3} + 0,0532z^{-4} - 0,3856z^{-5},$$

o que resulta na seguinte equação a diferenças para o cálculo de  $y_p$ :

$$\begin{aligned} y_p(k) &= 0,9y_p(k-1) + 0,2u(k-1) + 0,04u(k-2) + 0,044u(k-3) \\ &\quad + 0,0484u(k-4) + 0,0532u(k-5) - 0,3856u(k-6) \\ &\quad + 2,221y(k) - 2,121y(k-1). \end{aligned}$$

Ambos os métodos para se obter  $y_p$  geram os mesmos resultados. Assim, fica a critério do usuário qual utilizar.

Os resultados obtidos com sinais de perturbação do tipo degrau podem ser generalizados para outros tipos de perturbação, bastando, para isso, que as condições no filtro  $F_e(z)$  sejam adaptadas para o tipo de perturbação considerado, da mesma forma que foi analisado no caso do PSF.

Um ponto importante, que deve ser ressaltado aqui, é que no cálculo dos coeficientes dos polinômios que levam à obtenção de  $F_e(z)$  e  $S(z)$ , sempre teremos arredondamentos. Esses arredondamentos podem causar erros numéricos de tal forma que o cancelamento das raízes indesejadas do denominador e numerador de  $S(z)$  não aconteça. Assim, sugerimos sempre ajustar um dos coeficientes do numerador de  $S(z)$  para que a divisão seja exata e que, finalmente, possamos eliminar do denominador de  $S(z)$  todas as raízes indesejadas. Ainda, temos que garantir que o ganho estático de  $S(z)$  seja nulo e que o de  $F_e(z)$  seja unitário no final do procedimento. Essas duas condições garantem a estabilidade do preditor e a convergência da predição ao valor da saída do processo. Caso softwares sejam utilizados para a obtenção de  $F_e(z)$  e  $S(z)$ , tais como Matlab ou Python, por exemplo, o risco de arredondamentos causarem problema é minimizado. No entanto, é possível encontrar situações em que o software não identificará raízes iguais, mesmo se o procedimento para o cálculo de  $F_e(z)$  estiver correto. Nesses casos, sugerimos aumentar a tolerância da função que detecta raízes iguais na função de transferência. Esse procedimento, usualmente, resolve o problema.

Com base na discussão apresentada nesta seção, podemos observar que a mudança no código do GPC para obter o DTC-GPC é simples. Na sequência, vamos apresentar alguns exemplos simulados para ilustrar as diferenças em termos de desempenho e robustez das duas estratégias e que ilustram o fato de que o DTC-GPC é um algoritmo mais apropriado para o controle de processos com atraso.

### 6.2.3 Exemplos

**Exemplo 6.8.** *Vamos considerar um controle de temperatura em um sistema que tem atraso de transporte dominante, dado que o ponto de medida da temperatura está afastado do local onde se encontra a fonte de potência. O modelo do processo considerado é dado por:*

$$T(s) = \frac{3e^{-5s}}{(2s+1)(s+1)}[U(s) + Q_e(s)],$$

com  $T$  a temperatura,  $U$  o sinal de controle e  $Q_e$  uma perturbação causada pelas variações de temperatura do fluido de entrada no aquecedor.

Para este processo, usando uma amostragem de  $T_s = 0,2s$ , foram implementados um GPC e um DTC-GPC utilizando os seguintes parâmetros:  $N = 20$ ,  $N_u = 2$ ,  $\lambda = 2$ . Para mostrar a simplicidade e utilidade prática do filtro do DTC-GPC, escolhemos um filtro simples de primeira ordem de

ganho unitário na forma:

$$F_e(z) = \frac{z - a}{z - b} \frac{1 - b}{1 - a},$$

no qual  $a$  e  $b$  são utilizados para mudar a característica do filtro, mais passa baixa ou passa alta, dependendo da característica desejada.

No primeiro caso, supomos que o modelo é bem conhecido e a resposta às perturbações pode ser acelerada, mantendo inalterada a resposta às mudanças de referência. Assim, o filtro é ajustado com ganho maior que um em alta frequência, com  $a = 0,9$  e  $b = 0,6$ . As respostas podem ser vistas na Figura 6.16 para uma mudança de referência de 0,5 em  $t = 2$  s e uma perturbação de amplitude 0,05 aplicada em  $t = 20$  s. Podemos observar como este ajuste permite que o DTC-GPC seja ligeiramente mais rápido na rejeição da perturbação que o GPC, e ambos os controladores têm, como esperado, a mesma resposta para a mudança de referência.

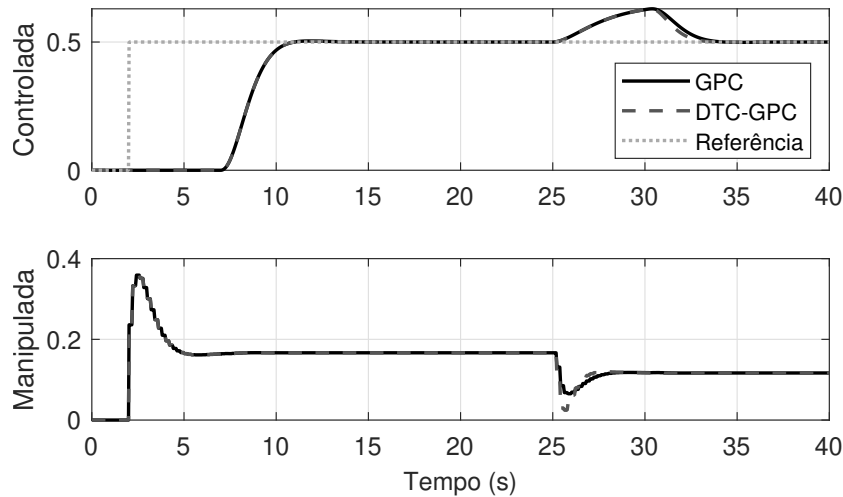


Figura 6.16: Exemplo 6.8 – resposta do sistema de controle de temperatura com  $F_e$  do DTC-GPC ajustado para ganho de velocidade da resposta.

No segundo caso, consideramos que o comportamento do processo não é igual ao do modelo. Assim, o ajuste tem que priorizar a robustez, penalizando a resposta às perturbações. Note que, mesmo neste caso, a resposta às mudanças de referência é mantida inalterada no caso nominal. Assim, o filtro é ajustado com ganho menor que um em alta frequência, com  $a = 0,7$  e  $b = 0,9$ . As respostas, para o mesmo cenário do caso anterior, podem ser vistas na Figura 6.17. Podemos observar como este ajuste deixa o DTC-GPC mais lento na rejeição da perturbação, porém, ambos os controladores têm, como esperado, a mesma resposta para a mudança de referência.

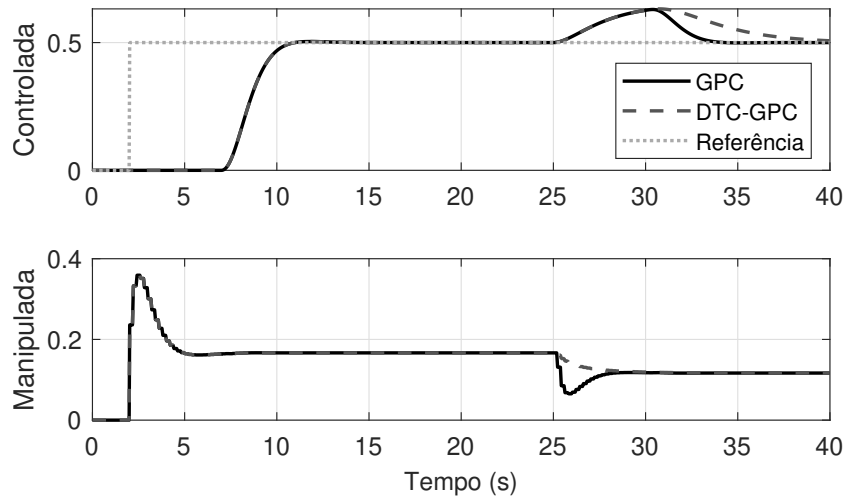


Figura 6.17: Exemplo 6.8 – resposta do sistema de controle de temperatura com  $F_e$  do DTC-GPC ajustado para robustez. Resposta nominal.

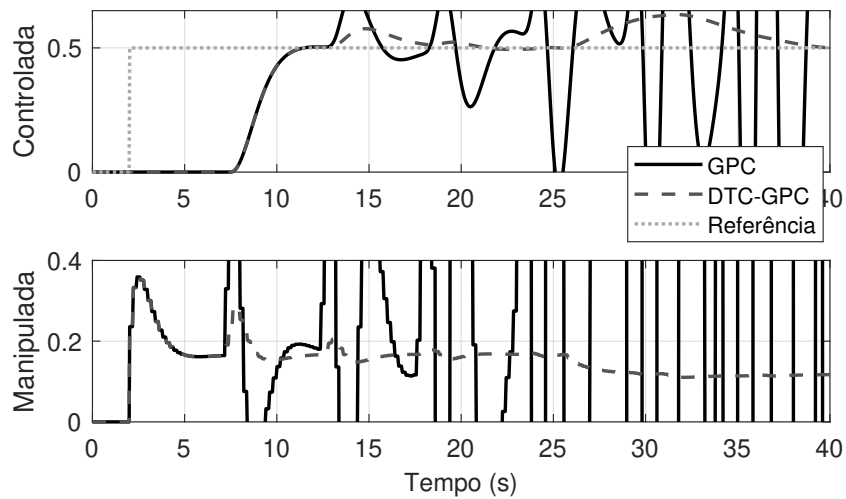


Figura 6.18: Exemplo 6.8 – resposta do sistema de controle de temperatura com  $F_e$  do DTC-GPC ajustado para robustez. Resposta com erro de atraso de 10%.

*Para verificar a maior robustez, na Figura 6.18 mostramos o desempenho dos controladores quando o atraso do processo é aumentado para 5,5s, ainda no mesmo cenário de simulação. Observe como somente o DTC-GPC mantém a estabilidade e, ainda, com uma resposta pouco oscilatória. Neste exemplo mostramos como, para plantas estáveis, o ajuste do filtro pode ser simples e intuitivo para utilização prática, o que facilita o seu uso em dife-*

rentes aplicações.

**Exemplo 6.9.** Vamos analisar agora o caso do sistema instável em malha aberta e com atraso de transporte estudado no Exemplo ??, que tinha um modelo dado por  $A(z^{-1}) = 1 - 1,1z^{-1}$ ,  $B(z^{-1}) = 0,1 + 0,2z^{-1}$  e atraso de uma amostra,  $d = 1$ . Neste exemplo, analisaremos o mesmo caso e também uma situação com atraso maior, com  $d = 5$ . Nas simulações, consideramos a perturbação  $q$  futura desconhecida. O ajuste do GPC é  $\lambda = \delta = 1$ ,  $N = 20$  e  $N_u = 5$ . Estudaremos o ajuste do DTC-GPC para mostrar o compromisso desempenho/robustez e mostraremos como a instabilidade em malha aberta impõe uma limitação no ajuste do filtro em termos de robustez.

Em ambos os casos, utilizaremos um filtro do erro de predição  $F_e(z)$  com o formato:

$$F_e(z) = \frac{z - a}{z - b} \frac{1 - b}{1 - a},$$

de ganho unitário por construção, que deve satisfazer a seguinte condição, já que o sistema é instável em malha aberta,

$$z^d - F_e(z) \Big|_{z=z_0} = 0,$$

sendo  $z_0$  o polo indesejado. Esta última equação pode ser reescrita como:

$$F_e(z_0) = z_0^d.$$

Como a raiz indesejada é instável, então  $|z_0| > 1$  e o termo  $z_0^d$  terá valor maior que 1. Como o filtro  $F_e(z)$  tem ganho unitário, isto quer dizer que o mesmo deverá ter um zero dominante para satisfazer  $F_e(z_0) = z_0^d$  e, além disso, quanto maior o valor de  $d$ , mais dominante vai ser o valor de  $a$  em relação ao valor de  $b$ . Os filtros  $F_e(z)$  para os dois casos, considerando  $b = 0,9$ , como no Exemplo 6.7, são dados abaixo:

$$F_{e,1}(z) = 1,2 \frac{z - 0,9167}{z - 0,9}, \quad F_{e,5}(z) = 2,221 \frac{z - 0,955}{z - 0,9}.$$

Note como no caso  $d = 5$  o zero é mais dominante do que no caso  $d = 1$  para um mesmo valor de  $b$ . Os diagramas de Bode desses dois filtros são mostrados na Figura 6.19. Veja que o filtro para o caso  $d = 5$  tem uma característica consideravelmente mais passa altas do que no caso  $d = 1$ . Na mesma figura são apresentadas as respostas em frequência dos filtros com  $b = 0,95$  (mais lento), que resulta nos seguintes filtros:

$$F_{e,1}(z) = 1,15 \frac{z - 0,9565}{z - 0,95}, \quad F_{e,5}(z) = 1,9158 \frac{z - 0,9739}{z - 0,95}.$$

Veja que deixando o polo do filtro mais lento diminuimos os valores máximos de magnitude de  $F_e(z)$  no domínio da frequência independentemente do atraso. No entanto, sempre é necessário manter uma certa característica passa alta quando lidamos com sistemas com atraso e instáveis, o que os torna mais difíceis de se lidar, especialmente quando consideramos as características de robustez. As respostas temporais para os casos com diferentes atrasos e filtros são apresentadas na Figura 6.20. Nesse cenário de simulação, uma referência de 0,6 é aplicada em  $t = 5$  amostras, e uma perturbação de entrada de amplitude  $-0,2$  é aplicada em  $t = 30$  amostras. Veja que nos casos com  $b = 0,95$ , a resposta de rejeição de perturbação é ligeiramente mais lenta para esse processo, enquanto que o rastreamento de referência não sofre influência do filtro do erro de predição.

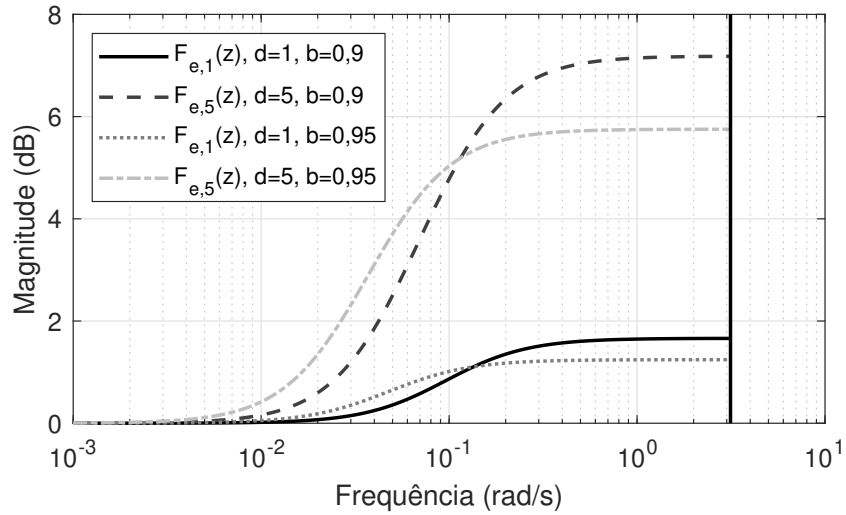


Figura 6.19: Exemplo 6.9 – diagrama de Bode dos filtros  $F_{e,1}(z)$  e  $F_{e,5}(z)$  para  $d = 1$  e  $d = 5$ , e  $b = 0,9$  e  $b = 0,95$ .

Como, nestes casos, teremos uma característica passa altas do filtro  $F_e(z)$  de forma independente de  $d$  ou mesmo de  $b$ , o sistema de malha fechada tende a ser pouco robusto. Isso pode ser parcialmente contornado relaxando as especificações temporais do sistema de malha fechada (tempo de resposta), mas há uma solução alternativa. Podemos melhorar a robustez de malha fechada mudando a estrutura do filtro  $F_e(z)$ . Ao invés de utilizar um filtro de primeira ordem, podemos usar

$$F_e(z) = z^{p-1} \frac{(z - a)}{(z - b)^p} \frac{(1 - b)^p}{(1 - a)}$$

no qual  $p$  determina a ordem desse filtro e o termo  $z^{p-1}$  no numerador apenas

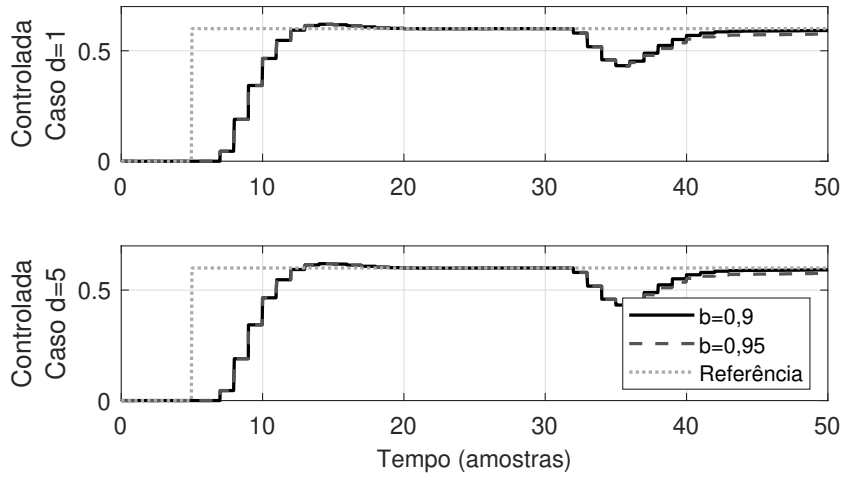


Figura 6.20: Exemplo 6.9 – resposta temporal para diferentes atrasos e filtros de erro de previsão  $F_e(z)$  no caso nominal.

faz com que o filtro não tenha atrasos desnecessários na resposta. Note que se  $p = 1$ , temos o filtro utilizado anteriormente. Para  $p = 2$  e  $b = 0,8449$ , os novos filtros  $F_e(z)$  são:

$$F_{e,1}(z) = 0,4102 \frac{z(z - 0,9414)}{(z - 0,8449)^2}, \quad F_{e,5}(z) = 0,71222 \frac{z(z - 0,9662)}{(z - 0,8449)^2}.$$

O novo valor de  $b$  foi calculado para o filtro ter o mesmo tempo de assentamento do caso  $p = 1$  e  $b = 0,9$ , de forma a tornar mais justa a comparação. Os diagramas de Bode para estes casos são mostrados na Figura 6.21. Veja que as características dos filtros com  $p = 2$  mudam consideravelmente: a magnitude em altas frequências agora é muito mais baixa do que no caso  $p = 1$ . Esse comportamento tende a melhorar a característica de robustez de malha fechada.

Por exemplo, considere agora que o ganho do processo foi mal estimado, havendo um erro de 32%. Os resultados de simulação são mostrados na Figura 6.22, para o mesmo cenário de simulação, exceto que não há mais um sinal de perturbação. No caso  $d = 1$  com o filtro  $b = 0,9$  e  $p = 1$  (o primeiro que foi calculado), veja que a resposta tem uma perda de desempenho em relação ao caso nominal, mas o sistema se mantém estável, o que não ocorre para o caso  $d = 5$ , que é menos robusto por conta da característica mais passa altas de seu filtro  $F_e(z)$ . No entanto, aumentando a ordem do filtro  $F_e(z)$  para  $p = 2$ , veja que a resposta para  $d = 1$  é melhorada, e a resposta para  $d = 5$  passa a ser estável, ou seja, a robustez dos sistemas em malha fechada resultantes foi melhorada com o uso de um filtro mais complexo.

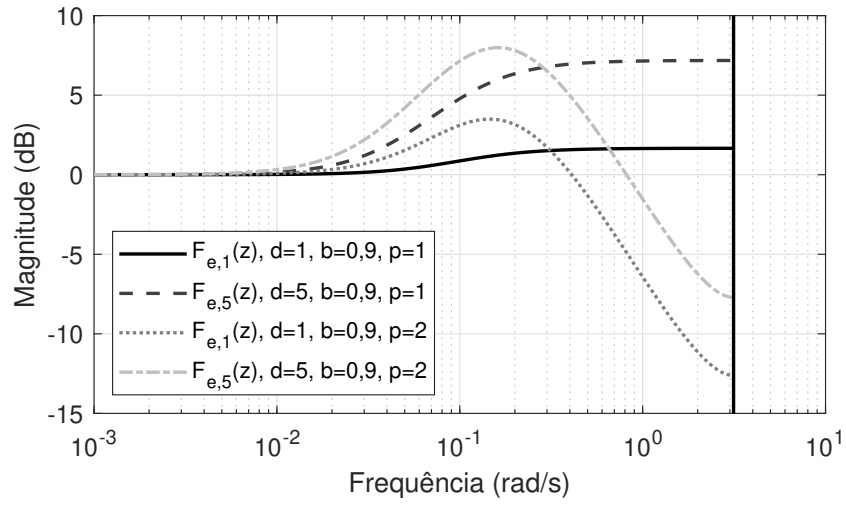


Figura 6.21: Exemplo 6.9 – diagrama de Bode dos filtros  $F_{e,1}(z)$  e  $F_{e,5}(z)$  para  $d = 1$  e  $d = 5$ , e  $p = 1$  e  $p = 2$ .

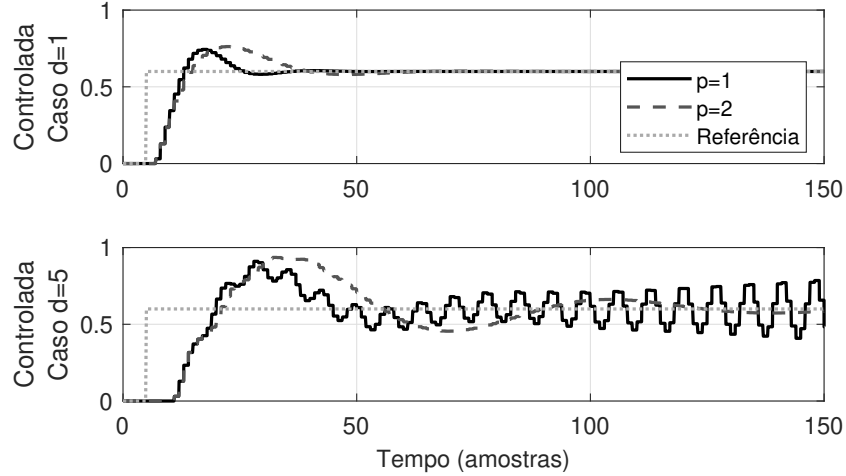


Figura 6.22: Exemplo 6.9 – resposta temporal para diferentes atrasos e filtros de erro de predição  $F_e(z)$  com erro de modelagem.

#### 6.2.4 O caso multivariável

A extensão do DTC-GPC para o caso MIMO segue os mesmos passos do caso SISO, ou seja, devemos apenas modificar as predições calculadas até o tempo de atraso de maneira diferente. Um ponto importante, no entanto, é como definir o atraso de um sistema MIMO. Neste livro, vamos adotar a definição de atraso mínimo por saída, ou atraso efetivo por saída.

Note que, no modelo apresentado no Capítulo 3, cada função de transferência do modelo MIMO tem um atraso  $d_{l,i}$  que corresponde ao atraso da



entrada  $i$  para a saída  $l$ , formando uma matriz de transferência com estrutura dada por:

$$\mathbf{P}(z) = \begin{bmatrix} G_{11}(z)z^{-d_{11}} & \cdots & G_{1m}(z)z^{-d_{1m}} \\ \vdots & \ddots & \vdots \\ G_{n1}(z)z^{-d_{n1}} & \cdots & G_{nm}(z)z^{-d_{nm}} \end{bmatrix}. \quad (6.34)$$

Se consideramos uma dada saída  $l$ , podemos definir o atraso mínimo associado a ela,  $d_l$ , como sendo o mínimo de todos os  $d_{li}$ :

$$d_l = \min_{i=1 \dots m} (d_{li}),$$

visto que por mais que todas as entradas se alterem no instante  $k$ , a saída  $l$  só será alterada em  $k + d_l + 1$ . Desta forma, o modelo MIMO pode ser escrito como o produto de uma matriz diagonal de atrasos mínimos,

$$\mathbf{L}(z) = \begin{bmatrix} z^{-d_1} & & 0 \\ & \ddots & \\ 0 & & z^{-d_n} \end{bmatrix},$$

por  $\mathbf{G}(z)$ , denominado modelo rápido, que não tem esses atrasos:

$$\mathbf{G}_f(z) = \begin{bmatrix} G_{11}(z)z^{-d_{11}+d_1} & \cdots & G_{1m}(z)z^{-d_{1m}+d_1} \\ \vdots & \ddots & \vdots \\ G_{n1}(z)z^{-d_{n1}+d_n} & \cdots & G_{nm}(z)z^{-d_{nm}+d_n} \end{bmatrix}.$$

Logo:

$$\mathbf{P}(z) = \mathbf{L}(z)\mathbf{G}(z),$$

e denominaremos  $G_{l,i}(z)$  ao elemento  $l,i$  da matriz  $\mathbf{G}(z)$ . Como visto no Capítulo 3, a predição de cada saída é calculada de forma independente, considerando apenas todas as entradas que a afetam. Assim, para uma dada saída  $l$  podemos considerar as predições calculadas em duas etapas, de  $k + 1$  até  $k + d_l$ , usando o procedimento do PSF, e de  $k + d_l + 1$  em diante usando o procedimento normal do GPC. Desta forma, considerando um filtro  $F_{e,l}(z)$  por saída, o cálculo das predições usando o PSF, no plano transformado, pode ser feito empregando:

$$\begin{aligned} Y_{p,l}(z) &= F_{e,l}(z)Y_l(z) \\ &\quad + [G_{l,1}U_1(z) + G_{l,2}U_2(z) + \cdots + G_{l,m}U_m(z)](1 - z^{-d_l}F_{e,l}(z)), \\ Y_{p,l}(z) &= F_{e,l}(z)Y_l(z) \\ &\quad + [P_{l,1}U_1(z) + P_{l,2}U_2(z) + \cdots + P_{l,m}U_m(z)](z^{d_l} - F_{e,l}(z)), \end{aligned}$$

que pode ser escrito como:

$$Y_{p,l}(z) = F_{e,l}(z)Y_l(z) + S_{l,1}U_1(z) + S_{l,2}U_2(z) + \dots + S_{l,m}U_m(z),$$

com  $S_{l,i}(z) = (z^{d_l} - F_{e,l}(z))P_{l,i}(z)$  e  $i = 1 \dots m$ .

O que muda em relação ao caso SISO é que o filtro usado no preditor deve ser capaz de estabilizar a predição  $y_{p,l}(k)$ , que neste caso depende das ações de controle  $u_i$  de  $i = 1$  até  $i = m$ , isto é, todas as  $S_{l,i}(z)$  precisam ser estáveis.

O ajuste do filtro é feito da mesma forma que no caso SISO, mas, agora, precisamos eliminar todos polos indesejados de todos modelos associados a uma saída. Assim, o filtro do erro de predição da saída  $l$  deverá ter ganho estático unitário e graus de liberdade suficientes para que:

$$z_x^{d_l} - F_{e,l}(z_x) = 0$$

para todo  $z_x$  que seja polo indesejado dos modelos  $P_{l,i}$ , com  $i = 1, \dots, m$ . O procedimento matemático para a obtenção de  $F_{e,l}(z)$  é o mesmo do caso SISO: é necessário montar um sistema de equações que permita achar os coeficientes do numerador do filtro, considerando o denominador como um parâmetro que permite ajustar a resposta às perturbações e a robustez. Devemos, ainda, lembrar que, como no caso SISO, os cálculos e arredondamentos devem garantir a estabilidade das funções de transferência  $S_{l,i}(z)$  e que todas tenham ganho estático nulo.

Finalmente, uma vez calculados os filtros por saída, o filtro MIMO a ser usado no DTC-GPC-MIMO será diagonal (de dimensão  $n$ ), com elementos  $F_{e,l}(z)$ , de  $l = 1$  a  $l = n$ :

$$\mathbf{F}_e(z) = \begin{bmatrix} F_{e,1}(z) & & 0 \\ & \ddots & \\ 0 & & F_{e,n}(z) \end{bmatrix}.$$

Vamos apresentar, na sequência, um exemplo do cálculo do filtro e da estrutura para obter as predições até o atraso em um sistema MIMO  $2 \times 2$ .

**Exemplo 6.10.** *Considere o sistema de tempo discreto representado por:*

$$\mathbf{P}(z) = \begin{bmatrix} \frac{0,7}{z-1}z^{-3} & \frac{0,1}{z-0,95}z^{-5} \\ \frac{0,3}{z-0,9}z^{-4} & \frac{0,4}{z-0,95}z^{-2} \end{bmatrix}, \quad (6.35)$$

*que tem atrasos mínimos por saída dados por  $d_1 = 3$  e  $d_2 = 2$ . Ainda, para termos uma resposta mais rápida na rejeição de perturbações e um preditor*

estável, é necessário eliminar os modos integradores e lentos. O polo integrador é dado por  $z = 1$  e foi assumido neste exemplo que se deseja ter uma resposta de rejeição de perturbação com dinâmica mais rápida que a definida pelo polo em  $z = 0,95$ . Dessa forma, os polos em  $z = 1$  e  $z = 0,95$  do modelo da planta devem ser eliminados das funções de transferência  $S_{l,i}(z)$  por meio do projeto de  $F_e(z)$ . Devemos, então, calcular dois filtros de erro de predição,  $F_{e,1}(z)$  e  $F_{e,2}(z)$ , um para cada saída. Para a saída 1, veja que o sistema tem um polo integrador e um polo em 0,95, então o filtro  $F_{e,1}(z)$  precisa ser projetado para eliminar esses dois polos e ter ganho estático unitário, o que é obtido com as condições:

$$\begin{aligned} z^3 - F_{e,1}(z) \Big|_{z=0,95} &= 0, \\ \frac{d[z^3 - F_{e,1}(z)]}{dz} \Big|_{z=1} &= 0, \\ F_{e,1}(1) &= 1. \end{aligned}$$

Já para a saída 2, é necessário apenas cancelar o polo em 0,95, fazendo com que as condições a serem satisfeitas sejam:

$$\begin{aligned} z^2 - F_{e,2}(z) \Big|_{z=0,95} &= 0, \\ F_{e,2}(1) &= 0. \end{aligned}$$

Podemos observar que precisamos de três parâmetros de ajuste no filtro  $F_{e,1}(z)$  e dois parâmetros no filtro  $F_{e,2}(z)$ . Assim, escolhemos  $F_{e,1}(z)$  de segunda ordem e  $F_{e,2}(z)$  de primeira ordem:

$$\begin{aligned} F_{e,1} &= \frac{a_1 z^2 + b_1 z + c_1}{(z - z_1)^2}, \\ F_{e,2} &= \frac{a_2 z + b_2}{(z - z_2)}, \end{aligned}$$

com  $z_1$  e  $z_2$  sendo parâmetros de ajuste da velocidade de resposta para perturbações. Os parâmetros  $a_1$ ,  $b_1$  e  $c_1$  são calculados resolvendo as equações:

$$\begin{aligned} \frac{a_1(0,95)^2 + b_1(0,95) + c_1}{(0,95 - z_1)^2} &= (0,95)^3, \\ (2a_1 + b_1)(1 - z_1) - 2(a_1 + b_1 + c_1) &= 3(1 - z_1)^3, \\ \frac{a_1 + b_1 + c_1}{(1 - z_1)^2} &= 1, \end{aligned}$$

e  $a_2$  e  $b_2$  com:

$$\frac{a_2(0,95) + b_2}{(0,95 - z_2)} = (0,95)^2,$$

$$\frac{a_2 + b_2}{(1 - z_2)} = 1.$$

Veja que os parâmetros dos filtros podem ser encontrados por meio da solução dos seguintes sistemas de equações lineares:

$$\begin{bmatrix} 0,95^2 & 0,95 & 1 \\ -2z_1 & -1 - z_1 & -2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} (0,95 - z_1)(0,95)^2 \\ 3(1 - z_1)^3 \\ (1 - z_2)^2 \end{bmatrix},$$

$$\begin{bmatrix} 0,95 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} (0,95 - z_2)(0,95)^2 \\ (1 - z_2) \end{bmatrix}.$$

Neste caso, para o preditor calculamos  $S_{l,i}(z)$  como:

$$S_{1,1} = \frac{0,7z^{-3}}{z - 1} \frac{z^3(z - z_1)^2 - (a_1z^2 + b_1z + c_1)}{(z - z_1)^2} = \frac{(z - 1)N_{s,1,1}(z)}{(z - z_1)^2},$$

$$S_{1,2} = \frac{0,1z^{-5}}{z - 0,95} \frac{z^3(z - z_1)^2 - (a_1z^2 + b_1z + c_1)}{(z - z_1)^2} = \frac{(z - 1)N_{s,1,2}(z)}{(z - z_1)^2},$$

$$S_{2,1} = \frac{0,3z^{-4}}{z - 0,9} \frac{z^2(z - z_1) - (a_2z + b_2)}{(z - z_2)} = \frac{(z - 1)N_{s,2,1}(z)}{(z - 0,9)(z - z_2)},$$

$$S_{2,2} = \frac{0,4z^{-2}}{z - 0,95} \frac{z^2(z - z_1) - (a_2z + b_2)}{(z - z_2)} = \frac{(z - 1)N_{s,2,2}(z)}{(z - z_2)},$$

sendo  $N_{s,l,i}(z)$  os polinômios resultantes da redução dos  $S_{l,i}(z)$  ao cancelar as raízes em  $z = 1$  e em  $z = 0,95$ . Observe que colocamos em evidência um fator  $z - 1$  no numerador dos  $S_{l,i}(z)$  para mostrar que todos têm ganho estático nulo. Com os  $S_{l,i}(z)$  e  $F_{e,l}(z)$  calculados, as previsões são calculadas usando as equações recursivas correspondentes. Teremos uma resposta diferente do sistema para cada escolha de  $z_1$  e  $z_2$ , sendo que para valores destes polos mais próximos de 1 a resposta será mais lenta e mais robusta. O exercício da lista explora este caso.

## 6.3 DMC para sistemas com atraso

Assim como no caso do GPC, para o DMC também foi mostrado como obter o controlador equivalente no caso sem restrições (Volume 1 desta obra). Vamos

mostrar aqui que, quando o sistema possui atraso de transporte, o DMC também define implicitamente uma estrutura preditora do tipo PSF. Além disso, vamos mostrar que o filtro do erro de predição do DMC não é igual ao do GPC, o que justifica as diferenças no desempenho e na robustez entre os algoritmos.

Para apresentar o estudo de uma forma didaticamente mais clara, vamos, inicialmente, encontrar o DTC equivalente do DMC sem restrições para o caso estável. Posteriormente, estenderemos o estudo para o DMC filtrado e, finalmente para, o GDMC, de forma tal a poder aplicar as ideias também a processos integradores e instáveis.

### 6.3.1 DMC como um DTC: plantas estáveis

Lembremos que as predições do DMC SISO são calculadas com:

$$\hat{y}(k+j | k) = \sum_{i=1}^j g_i \Delta u(k+j-i) + \sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i) + \hat{n}(k | k). \quad (6.36)$$

Vamos, inicialmente, obter as predições após  $k+d$ , ou seja, após o atraso do processo, em função da predição em  $k+d$ , assim como foi feito para o GPC. Para isso, a partir de (6.36), temos:

$$\begin{aligned} \hat{y}(k+d+j | k) &= \sum_{i=1}^j g_i \Delta u(k+d+j-i) \\ &\quad + \sum_{i=j+1}^{\infty} g_i \Delta u(k+d+j-i) + \hat{n}(k | k), \end{aligned} \quad (6.37)$$

mas como  $g_i = 0$  para  $i \leq d$ , por causa do efeito do atraso, é possível reescrever a (6.37) da seguinte maneira:

$$\hat{y}(k+d+j | k) = \sum_{i=1}^j g_{d+i} \Delta u(k+j-i) + \sum_{i=1}^{\infty} g_{d+j+i} \Delta u(k-i) + \hat{n}(k | k). \quad (6.38)$$

Agora, utilizando a (6.38) com  $j = 0$ , obtemos a predição da saída em  $k+d$ :

$$\hat{y}(k+d | k) = \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + \hat{n}(k | k). \quad (6.39)$$

Note que  $\hat{y}(k + d | k)$  só depende de termos passados da entrada devido ao atraso. Rearranjando (6.39), obtemos:

$$\hat{n}(k | k) = \hat{y}(k + d | k) - \sum_{i=1}^{\infty} g_{d+i} \Delta u(k - i). \quad (6.40)$$

Substituindo (6.40) em (6.38),

$$\begin{aligned} \hat{y}(k + d + j | k) &= \sum_{i=1}^j g_{d+i} \Delta u(k + j - i) + \sum_{i=1}^{\infty} g_{d+j+i} \Delta u(k - i) \\ &\quad + \hat{y}(k + d | k) - \sum_{i=1}^{\infty} g_{d+i} \Delta u(k - i), \\ \hat{y}(k + d + j | k) &= \sum_{i=1}^j g_{d+i} \Delta u(k + j - i) \\ &\quad + \sum_{i=1}^{\infty} (g_{d+j+i} - g_{d+i}) \Delta u(k - i) + \hat{y}(k + d | k), \end{aligned} \quad (6.41)$$

sendo que o termo com o somatório infinito pode ser truncado, pois  $g_{d+j+i} - g_{d+i} \approx 0$ ,  $\forall i \geq N_{ss}$ . Por fim, obtemos as predições  $k + d + j$  em função da predição em  $k + d$ :

$$\begin{aligned} \hat{y}(k + d + j | k) &= \sum_{i=1}^j g_{d+i} \Delta u(k + j - i) \\ &\quad + \sum_{i=1}^{N_{ss}} (g_{d+j+i} - g_{d+i}) \Delta u(k - i) + \hat{y}(k + d | k). \end{aligned} \quad (6.42)$$

A partir de (6.42), e considerando  $N_1 = d + 1$  e  $N_2 = d + N$ , o vetor de predições futuras é dado por

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f},$$

com  $\mathbf{f} = \mathbf{H} \underline{\Delta \mathbf{u}} + \mathbf{1} \hat{y}(k + d | k)$  e

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k + d + 1 | k) \\ \hat{y}(k + d + 2 | k) \\ \vdots \\ \hat{y}(k + d + N | k) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} g_{d+1} & 0 & \cdots & 0 \\ g_{d+2} & g_{d+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{d+N} & g_{d+N-1} & \cdots & g_{d+N-N_u+1} \end{bmatrix},$$

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(k+d+1 | k) \\ f(k+d+2 | k) \\ \vdots \\ f(k+d+N | k) \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N_{ss}-1} & h_{1,N_{ss}} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,N_{ss}-1} & h_{2,N_{ss}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{N_2,1} & h_{N_2,2} & \cdots & h_{N_2,N_{ss}-1} & h_{N_2,N_{ss}} \end{bmatrix}, \quad \underline{\Delta \mathbf{u}} = \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-N_{ss}) \end{bmatrix},$$

com

$$h_{i,j} = g_{d+j+i} - g_{d+j}, \quad i+j \leq N_{ss},$$

$$h_{i,j} = g_{N_{ss}} - g_{d+j}, \quad i+j > N_{ss}$$

e  $\mathbf{1} \in \mathbb{R}^N$  sendo um vetor formado por elementos iguais a 1.

Minimizando a função custo, de forma similar ao feito com o GPC, obtemos a lei de controle

$$\Delta u(k) = \mathbf{K}_1 \left( \mathbf{1}(r(k) - \hat{y}(k+d | k)) - \mathbf{H} \underline{\Delta \mathbf{u}} \right),$$

com

$$\mathbf{K}_1 = [k_1, k_2, k_3, \dots, k_N],$$

que resulta no seguinte controlador primário  $C(z)$  após a aplicação da transformada Z:

$$C(z) = \frac{U(z)}{E_s(z)} = \frac{f_0}{\Delta \left[ 1 + \sum_{j=1}^{N_{ss}} b_j z^{-j} \right]}, \quad (6.43)$$

onde

$$b_j = \sum_{i=1}^N k_i (g_{d+i+j} - g_{d+j}), \quad f_0 = \sum_{i=1}^N k_i, \quad (6.44)$$

e reforçando que o erro de seguimento é dado por  $e_s(k) = r(k) - \hat{y}(k+d | k)$ . Veja que chegamos novamente a uma forma de controle equivalente similar àquela encontrada para o GPC na Seção 6.2.

Vamos mostrar, agora, que a estrutura preditora utilizada internamente pelo DMC é equivalente à estrutura do PSF. A partir de (6.39) e da definição de  $\hat{n}(k | k)$  em (6.36),

$$\begin{aligned} \hat{y}(k+d | k) &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + \hat{n}(k | k) \\ &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + \left( y(k) - \sum_{i=1}^{\infty} g_i \Delta u(k-i) \right), \end{aligned} \quad (6.45)$$

e note que

$$\begin{aligned}\hat{y}_o(k+d | k) &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) = \sum_{i=1}^{\infty} (g_{d+i} - g_{d+i-1}) u(k-i) \\ &\approx \sum_{i=1}^{N_{ss}} (g_{d+i} - g_{d+i-1}) u(k-i)\end{aligned}$$

é a predição de malha aberta em  $k+d$ , e

$$\begin{aligned}\hat{y}_o(k | k) &= \sum_{i=1}^{\infty} g_i \Delta u(k-i) = \sum_{i=1}^{\infty} (g_i - g_{i-1}) u(k-i) \\ &\approx \sum_{i=1}^{N_{ss}} (g_i - g_{i-1}) u(k-i)\end{aligned}$$

é a predição de malha aberta em  $k$ . Aplicando a transformada Z na expressão de  $\hat{y}_o(k+d | k)$  obtemos:

$$G(z) = \frac{\hat{Y}_{od}(z)}{U(z)} = \sum_{i=1}^{N_{ss}} (g_{d+i} - g_{d+i-1}) z^{-i},$$

e para  $\hat{y}_o(k | k)$ :

$$P(z) = \frac{\hat{Y}_o(z)}{U(z)} = \sum_{i=1}^{N_{ss}} (g_i - g_{i-1}) z^{-i},$$

que permite escrever  $P(z) = G(z)z^{-d}$ , que são os modelos utilizados pelo DMC para encontrar a predição da saída e a da saída antes do atraso.

Utilizando a transformada Z em (6.45), o preditor definido implicitamente pelo DMC filtrado é dado por

$$\hat{Y}_d(z) = G(z)U(z) + (Y(z) - z^{-d}G(z)U(z)) \quad (6.46)$$

$$= G(z)(1 - z^{-d})U(z) + Y(z), \quad (6.47)$$

que é igual à equação de predição do PS. Assim, todas as propriedades do PS podem ser estendidas ao DMC, como a de ter uma resposta às perturbações sempre mais lenta que a de malha aberta, além da conhecida restrição para uso com plantas instáveis.

Esta análise permite concluir que se utilizarmos um filtro no erro de predição do DMC, dado por  $F_e(z)$ , temos os mesmos efeitos do filtro do erro de predição da estrutura PSF, ou seja, as considerações de rejeição de perturbação e robustez apresentadas na Seção 6.1 passam a ser também válidas para o DMC.



### 6.3.2 DMC Filtrado: Plantas estáveis

A proposta do algoritmo DMC filtrado apresentado no Volume 1 desta obra, consiste em usar um erro de predição filtrado:

$$\hat{n}_f(k | k) = F_e(z) (y(k) - \hat{y}_o(k | k)),$$

no lugar do erro de predição tradicional do DMC [LNRP<sup>+</sup>14]. Neste controlador,  $F_e(z)$  é o filtro do erro de predição, e as predições futuras passam a ser dadas por:

$$\hat{y}(k+j | k) = \sum_{i=1}^j g_i \Delta u(k+j-i) + \sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i) + \hat{n}_f(k | k). \quad (6.48)$$

Vamos mostrar agora que esta modificação transforma a estrutura preditora implícita do DMC em um PSF. Para isso, basta considerar a equação da predição em  $k+d$ :

$$\begin{aligned} \hat{y}(k+d | k) &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + \hat{n}_f(k | k) \\ &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + F_e(z) \left( y(k) - \sum_{i=1}^{\infty} g_i \Delta u(k-i) \right), \end{aligned} \quad (6.49)$$

que, para plantas estáveis pode ser escrita como:

$$\begin{aligned} \hat{y}(k+d | k) &= \sum_{i=1}^{N_{ss}} (g_{d+i} - g_{d+i-1}) u(k-i) \\ &\quad + F_e(z) y(k) - F_e(z) \sum_{i=1}^{N_{ss}} (g_i - g_{i-1}) u(k-i). \end{aligned} \quad (6.50)$$

Aplicando a transformada Z e usando as definições de  $G(z) = \sum_{i=1}^{N_{ss}} (g_{d+i} - g_{d+i-1}) z^{-i}$  e  $P(z) = \sum_{i=1}^{N_{ss}} (g_i - g_{i-1}) z^{-i}$ , obtemos:

$$\begin{aligned} \hat{Y}_d(z) &= G(z)U(z) + F_e(z)(Y(z) - P(z)U(z)) \\ &= P(z)(z^d - F_e(z))U(z) + F_e(z)Y(z) = S_e(z)U(z) + F_e(z)Y(z), \end{aligned}$$

que é exatamente a estrutura do PSF, e que também pode ser colocada como  $\hat{Y}_d(z) = G(z)(1 - z^{-d}F_e(z))U(z) + F_e(z)Y(z)$  dado que  $P(z) = G(z)z^{-d}$ .

Assim, podemos usar o filtro para conferir melhores propriedades ao DMC, assim como o PSF faz com o PS, e mais, sem nenhuma alteração no algoritmo DMC já tratado. Desta forma, para as plantas estáveis aqui consideradas, podemos usar o filtro para acelerar a resposta às perturbações ou melhorar a robustez. A ideia do ajuste do filtro segue o mesmo procedimento do GPC: é necessário fazer com que termo  $(z^d - F_e(z))$  seja ajustado para cancelar as dinâmicas lentas de  $P(z)$  quando se deseja acelerar a resposta às perturbações. Se os polos dominantes de  $P(z)$  são calculados, o método usado no DTC-GPC pode ser usado aqui diretamente, escolhendo graus de liberdade suficientes no filtro  $F_e(z)$  e mantendo seu ganho estático unitário. Os polos do filtro são usados para definir o tempo de resposta às perturbações, conjuntamente com os polos de  $P(z)$  não cancelados.

Por outro lado, assim como no DTC-GPC, para aumentar a robustez, é necessário usar filtros com característica mais passa baixas, que tendem a deixar as respostas mais lentas.

Apresentamos, na sequência, um exemplo de ajuste do filtro para ilustrar essas ideias.

**Exemplo 6.11.** *Aqui utilizaremos novamente o sistema trocador de calor do Exemplo 6.5 com modelo  $P_n(z) = \frac{0,1}{z-0,9}z^{-10}$ . Um DMC, ajustado com  $N_{ss} = 80$ ,  $N_1 = 11$ ,  $N_2 = 25$ ,  $N_u = 5$  e  $\lambda = 1$ , foi projetado para controlar a temperatura, e foram simuladas variações da temperatura de referência do tipo degrau, assim como perturbações do tipo degrau na temperatura externa.*

*Além disso, foi projetado um segundo DMC com a mesma sintonia, exceto que agora utilizamos um filtro do erro de predição para eliminar o efeito do polo de malha aberta na resposta às perturbações. O filtro, neste caso, deve verificar duas condições:*

$$\begin{aligned} F_e(1) &= 1, \\ z^{10} - F_e(z)|_{z=0,9} &= 0. \end{aligned}$$

*Para isso, é suficiente utilizar um filtro com dois parâmetros livres, e neste caso propormos um filtro de primeira ordem na forma:*

$$F_e(z) = \frac{a_{f,0}z - a_{f,1}}{z},$$

*e a partir das duas condições estabelecidas chegamos ao seguinte filtro do erro de predição:*

$$F_e(z) = \frac{7,0092(z - 0,8573)}{z},$$

*Dessa forma, como eliminamos o efeito do polo de malha aberta do sistema na resposta da perturbação e usamos um polo em  $z = 0$  no filtro, obtemos*

uma rejeição de perturbações muito mais rápida que a de malha aberta (em detrimento da robustez).

Os resultados da simulação deste DMC no trocador de calor podem ser vistos na Figura 6.23, considerando uma mudança de referência em  $t = 4$  s de valor 0,5 e uma perturbação de temperatura  $T_e$  de valor 0,1 em  $t = 15$  s. Veja que a resposta para a referência é idêntica para ambas as sintonias, mas a rejeição da perturbação com o DMC com o filtro do erro de predição é mais rápida.

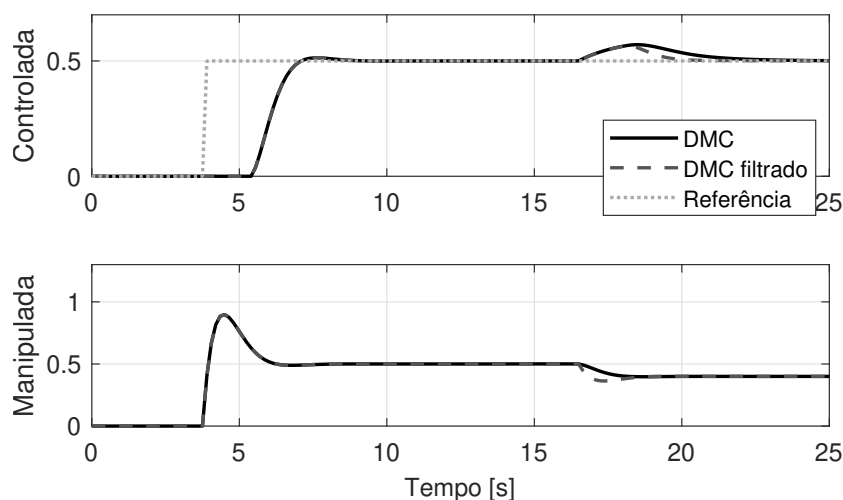


Figura 6.23: Exemplo 6.11 – comparação das respostas do DMC com e sem filtro do erro de predição.

Note que as condições para o projeto do filtro do erro de predição  $F_e(z)$  são as mesmas apresentadas anteriormente para o GPC e para o PSF. No entanto, veja que o ajuste depende da existência da função de transferência de malha aberta do sistema para que os polos indesejados a serem eliminados da dinâmica de malha fechada da perturbação possam ser determinados. No caso do DMC, muitas vezes não temos a função de transferência do modelo (temos unicamente a resposta ao degrau do mesmo). Essa ausência inviabiliza a aplicação direta das condições apresentadas acima, já que esse tipo de modelo é do tipo resposta ao impulso finita (FIR, do inglês *Finite Impulse Response*), que não tem polos definindo a dinâmica do processo. Neste caso, é necessário obter uma aproximação por função de transferência da resposta ao degrau e, daí sim, aplicar as regras para o ajuste do filtro do erro de predição.

Caso se deseje apenas usar os dados do modelo de resposta ao degrau (sem necessidade de cálculo da função de transferência e dos polos do modelo), é

possível usar um método numérico, como por exemplo mínimos quadrados, para ajustar os coeficientes do filtro  $F_e(z)$  para que os coeficientes da resposta impulsiva de  $S_e(z)$  converjam a zero no tempo de resposta desejado, como comentado no estudo do GDMC apresentado no Volume 1. Note que a como a dinâmica de  $S_e(z)$ , que depende de  $F_e(z)$  é sempre mais lenta que a de  $F_e(z)$ ,  $S_e(z)$  é que determina a velocidade com que  $\hat{Y}_d(z)$  converge para  $Y(z)$ . Se os coeficientes  $s_{e,i}$  da resposta impulsiva de  $S_e(z)$  são quase nulos para  $i > N_i$ , isso implica que o tempo de convergência de  $\hat{Y}_d(z)$  para  $Y(z)$  será aproximadamente de  $N_i$  amostras. Como os coeficientes  $s_{e,i}$  podem ser colocados em função dos parâmetros do filtro, um conjunto de equações pode ser resolvido para o ajuste do mesmo.

### 6.3.3 GDMC: plantas integradoras e instáveis

No caso de plantas não estáveis não podemos usar o procedimento já visto para o cálculo das predições em  $k + d$ , através de (6.41), que repetimos aqui:

$$\begin{aligned} \hat{y}(k + d + j \mid k) = & \sum_{i=1}^j g_{d+i} \Delta u(k + j - i) \\ & + \sum_{i=1}^{\infty} (g_{d+j+i} - g_{d+i}) \Delta u(k - i) + \hat{y}(k + d \mid k), \end{aligned} \quad (6.51)$$

porque o termo com o somatório infinito não pode ser truncado. Mas se aplicarmos o filtro  $F_{e,j}(z)$  como proposto no GDMC no erro de predição, a equação anterior se transforma em:

$$\begin{aligned} \hat{y}(k + d + j \mid k) = & \sum_{i=1}^j g_{d+i} \Delta u(k + j - i) \\ & + \sum_{i=1}^{\infty} (g_{d+j+i} - \tilde{g}_{j,d+i}) \Delta u(k - i) + \hat{y}_{f,j}(k + d \mid k), \end{aligned} \quad (6.52)$$

sendo  $\tilde{g}_{j,d+i}$  os coeficientes da resposta ao degrau filtrados e  $\hat{y}_{f,j}(k + d \mid k)$  a predição em  $k + d$  filtrada, ambos com o filtro  $F_{e,j}(z)$ . Se o filtro  $F_{e,j}(z)$  é corretamente ajustado, como visto no Volume 1 e na Seção 3.2.5, podemos truncar a soma em  $N_f$ , pois  $(g_{d+j+i} - \tilde{g}_{j,d+i}) \rightarrow 0$  para  $i \geq N_f$ . Assim, o

cálculo das predições resulta:

$$\begin{aligned}\hat{y}(k+d+j \mid k) &= \sum_{i=1}^j g_{d+i} \Delta u(k+j-i) \\ &+ \sum_{i=1}^{N_f} (g_{d+j+i} - \tilde{g}_{j,d+i}) \Delta u(k-i) + \hat{y}_{f,j}(k+d \mid k),\end{aligned}\quad (6.53)$$

e podemos realizar o mesmo procedimento usado para o caso de plantas estáveis, obtendo as mesmas expressões, mas usando uma nova matriz  $\mathbf{H}$  que usa os coeficientes  $\tilde{g}_{j,d+i}$  no lugar dos  $g_{j,d+i}$  e utilizando  $\hat{y}_{f,j}(k+d \mid k)$  no lugar de  $\hat{y}(k+d \mid k)$  no cálculo da resposta livre.

Considerando agora o horizonte de predição de  $N_1 = d+1$  a  $N_2 = d+N$ , o vetor de predições futuras é dado por

$$\hat{\mathbf{y}} = \mathbf{G} \Delta \mathbf{u} + \mathbf{f},$$

lembrando que  $\mathbf{f} = \mathbf{H} \underline{\Delta \mathbf{u}} + \hat{\mathbf{y}}_f(k+d \mid k)$  e

$$\begin{aligned}\hat{\mathbf{y}} &= \begin{bmatrix} \hat{y}(k+d+1 \mid k) \\ \hat{y}(k+d+2 \mid k) \\ \vdots \\ \hat{y}(k+d+N \mid k) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} g_{d+1} & 0 & \cdots & 0 \\ g_{d+2} & g_{d+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{d+N} & g_{d+N-1} & \cdots & g_{d+N-N_u+1} \end{bmatrix}, \\ \Delta \mathbf{u} &= \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(k+d+1 \mid k) \\ f(k+d+2 \mid k) \\ \vdots \\ f(k+d+N \mid k) \end{bmatrix},\end{aligned}$$

e a nova  $\mathbf{H}$  e o novo  $\underline{\Delta \mathbf{u}}$  dados por:

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N_f-1} & h_{1,N_f} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,N_f-1} & h_{2,N_f} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{N_2,1} & h_{N_2,2} & \cdots & h_{N_2,N_f-1} & h_{N_2,N_f} \end{bmatrix}, \quad \underline{\Delta \mathbf{u}} = \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-N_f) \end{bmatrix},$$

com

$$\begin{aligned}h_{i,j} &= g_{d+j+i} - \tilde{g}_{j,d+j}, \quad i+j \leq N_f, \\ h_{i,j} &= g_{N_f} - \tilde{g}_{j,d+j}, \quad i+j > N_f\end{aligned}$$

e finalmente  $\hat{\mathbf{y}}_f(k+d \mid k) \in \mathbb{R}^N$  sendo um vetor dado por:

$$\hat{\mathbf{y}}_f(k+d | k) = \begin{bmatrix} F_{e,1}(z) \\ F_{e,2}(z) \\ \vdots \\ F_{e,N}(z) \end{bmatrix} \hat{y}(k+d | k) = \mathbf{F}_e(z) \hat{y}(k+d | k)$$

Minimizando a função custo, de forma similar ao feito com o DMC, obtemos a lei de controle com referência futura não conhecida como:

$$\Delta u(k) = \mathbf{K}_1 \left( \mathbf{1}r(k) - \hat{\mathbf{y}}_f(k+d | k) - \mathbf{H} \Delta \mathbf{u} \right),$$

com

$$\mathbf{K}_1 = [k_1, k_2, k_3, \dots, k_N],$$

que resulta em:

$$\Delta u(k) = \mathbf{K}_1 \mathbf{1}r(k) - \mathbf{K}_1 \mathbf{F}_e(z) \hat{y}(k+d | k) - \mathbf{K}_1 \mathbf{H} \Delta \mathbf{u},$$

E com a aplicação da transformada Z:

$$U(z) = \frac{f_0 R(z) - \sum_{j=1}^N k_j F_{e,j}(z) Y_d(z)}{\Delta \left[ 1 + \sum_{j=1}^{N_f} b_j z^{-j} \right]}, \quad (6.54)$$

onde

$$b_j = \sum_{i=1}^N k_i (g_{d+i+j} - \tilde{g}_{j,d+j}), \quad f_0 = \sum_{i=1}^N k_i, \quad (6.55)$$

e  $Y_d(z)$  é a transformada Z de  $\hat{y}(k+d | k)$ . Ainda podemos simplificar a expressão do controle  $U(z)$  definindo um filtro  $F_d(z)$  como:

$$F_d(z) = \sum_{j=1}^N k_j F_{e,j}(z),$$

obtendo:

$$U(z) = \frac{f_0 R(z) - F_d(z) Y_d(z)}{\Delta \left[ 1 + \sum_{j=1}^{N_f} b_j z^{-j} \right]}, \quad (6.56)$$

Veja que chegamos novamente a uma forma de controle equivalente similar àquela encontrada para o DMC, mas agora temos o efeito do filtro sobre a saída predita. Ainda podemos escrever a lei de controle na forma de um filtro de referência e um controle de realimentação da saída predita:

$$U(z) = C_r(z) [F_r(z) R(z) - Y_d(z)], \quad (6.57)$$

com:

$$C_r(z) = \frac{F_d(z)}{\Delta \left[ 1 + \sum_{j=1}^{N_f} b_j z^{-j} \right]}, \quad (6.58)$$

$$F_r(z) = \frac{f_0}{F_d(z)}. \quad (6.59)$$

Estas expressões são similares as encontradas no Volume 1 quando realizamos o cálculo do controle equivalente do GDMC. Assim, todas as conclusões obtidas para aquela análise valem também aqui, com a diferença de considerar o sinal  $Y_d(z)$  e não o  $Y(z)$ .

Observe por exemplo que: (i) temos uma lei de controle com ação integral sobre o erro  $F_r(z)R(z) - Y_d(z)$ ; (ii) o filtro  $F_d(z)$ , e como consequência  $F_r(z)$ , têm ganho estático unitário ( $F_d(1) = 1$ ,  $F_r(1) = 1$ ), pois todos os  $F_{e,j}(z)$  tem ganho estático unitário e  $f_0 = \sum_{i=1}^N k_i$ , assim o controle garante em regime permanente para entradas degrau que  $\hat{y}(k+d | k)$  converge para  $r(k)$ .

Para finalizar a análise, vamos mostrar, agora, assim como fizemos para o DMC, que a estrutura preditora utilizada internamente pelo GDMC é equivalente à estrutura do PSF.

As predições em malha aberta do GDMC são calculadas da mesma forma que no DMC, pois a diferença entre eles está no cálculo do erro de predição, assim:

$$\hat{y}_o(k+d | k) = \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i)$$

é a predição de malha aberta em  $k+d$ , e

$$\hat{y}_o(k | k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i)$$

é a predição de malha aberta em  $k$ .

Aplicando a transformada Z na expressão de  $\hat{y}_o(k+d | k)$  obtemos:

$$G(z) = \frac{\hat{Y}_{od}(z)}{U(z)} = \sum_{i=1}^{\infty} g_{d+i} z^{-i} (1 - z^{-1}),$$

e para  $\hat{y}_o(k | k)$ :

$$P(z) = \frac{\hat{Y}_o(z)}{U(z)} = \sum_{i=1}^{\infty} g_i z^{-i} (1 - z^{-1}) = z^{-d} G(z).$$

Assim, a predição com correção:

$$\begin{aligned}\hat{y}(k+d | k) &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + \hat{n}_f(k | k) \\ &= \sum_{i=1}^{\infty} g_{d+i} \Delta u(k-i) + \left( F_{e,d}(z)y(k) - F_{e,d}(z) \sum_{i=1}^{\infty} g_i \Delta u(k-i) \right),\end{aligned}\tag{6.60}$$

pode ser escrita, utilizando a transformada Z, como:

$$\hat{Y}_d(z) = G(z)U(z) + F_{e,d}(z)Y(z) - F_{e,d}(z)z^{-d}G(z)U(z) \tag{6.61}$$

$$= G(z)(1 - z^{-d}F_{e,d}(z))U(z) + F_{e,d}(z)Y(z) \tag{6.62}$$

$$= S_{e,d}(z)U(z) + F_{e,d}(z)Y(z), \tag{6.63}$$

que é igual à equação de predição do PSF.

Por exemplo, como  $F_{e,d}(z)$  é ajustado para que  $S_{e,d}(z)$  seja estável e  $F_{e,d}(1) = 1$ , temos  $S_{e,d}(1) = 0$ , o preditor garante em regime permanente que  $\hat{y}(k+d | k)$  converge para o valor de equilíbrio de  $y(k)$ , e como o controle  $C_r(z)$  tem ação integral,  $y(k)$  converge para o valor da referência constante.

De forma resumida, todas as propriedades do PSF podem ser estendidas ao GDMC como, por exemplo, a de usar o filtro  $F_{e,d}(z)$  para ajustar um compromisso entre resposta às perturbações e robustez. Em termos de método numérico para o ajuste dos filtros, valem todas as considerações já feitas para o DMC filtrado, ou seja podemos usar modelos de função de transferência equivalentes, ou realizar o ajuste apenas usando os dados das respostas ao degrau. Nos exemplos deste livro usaremos sempre a primeira forma.

Para finalizar esta seção, vamos lembrar da análise do GDMC realizada no Volume 1, onde mostramos que o ajuste dos filtros  $F_{e,j}(z)$  que garantem a aplicação do GDMC para plantas não estáveis também pode ser usado para ajustar um compromisso entre resposta às perturbações e robustez, através do  $F_d(z)$ , dado que eles não afetam (como no PSF) a resposta para a referência. Mas o inconveniente mostrado lá é que o efeito desses filtros não era simples de analisar, dado que há um efeito combinado na malha fechada com os coeficientes do ganho equivalente  $k_i$ . Note que aqui, pela formulação proposta, podemos usar apenas os filtros  $F_{e,j}(z)$  para a condição de estabilidade e apenas o filtro  $F_{e,d}(z)$  para o compromisso entre resposta



às perturbações e robustez, dado que o  $F_{e,d}(z)$  não tem efeito algum sobre os coeficientes do ganho equivalente  $k_i$ .

Vamos ilustrar este estudo com um exemplo.

**Exemplo 6.12.** *Mostrar o uso de Fed num caso instável com atraso.*

### 6.3.4 DMC filtrado MIMO: plantas estáveis

A extensão do DMC para o caso MIMO é realizada da mesma forma que no GPC, considerando as predições de cada uma das saídas do processo. Consideramos aqui o caso do DMC filtrado pois ele é mais geral que o DMC original.

Para sistemas MIMO com atraso, basta usar a mesma forma de atraso efetivo por saída que foi usada no DTC-GPC-MIMO, e o cálculo de cada  $F_{e,l}(z)$  vai depender dos polos estáveis indesejados que se deseja cancelar das funções de transferência associadas às entradas que afetam a saída  $y_l$ . Assim, a ordem do filtro pode aumentar, mas basta projetar um filtro por saída. Vamos ilustrar o procedimento de ajuste com um exemplo.

**Exemplo 6.13.** *Considere o problema do estudo de caso de uma coluna de destilação água-metanol apresentado na Seção 3.4.1, lembrando que, próximo do ponto de operação, o comportamento dinâmico incremental é dado por:*

$$\mathbf{P}(s) = \begin{bmatrix} \frac{12,8e^{-s}}{1 + 16,7s} & \frac{-18,9e^{-3s}}{1 + 21s} \\ \frac{6,6e^{-7s}}{1 + 10,9s} & \frac{-19,4e^{-3s}}{1 + 14,4s} \end{bmatrix}, \quad \mathbf{P}_q(s) = \begin{bmatrix} \frac{3,8e^{-8s}}{1 + 14,9s} \\ \frac{4,9e^{-4s}}{1 + 13,2s} \end{bmatrix},$$

com  $\mathbf{Y}(s) = \mathbf{P}(s)\mathbf{U} + \mathbf{P}_q(s)Q(s)$ ,  $\mathbf{y} = [y_1 \ y_2]^T$  e  $\mathbf{u} = [u_1 \ u_2]^T$ . Neste processo, a vazão de refluxo ( $u_1$ ) e a vazão de vapor no refeedor ( $u_2$ ) são manipuladas para controlar as concentrações de metanol no topo ( $y_1$ ) e na base ( $y_2$ ) da coluna.

Para este processo, vamos utilizar um controlador DMC MIMO com e sem ajuste do filtro do erro de predição  $F_e(z)$ , supondo também  $q(k)$  não mensurável. Para efeitos de simulação, vamos considerar o modelo de matriz de função de transferência como se fosse o processo real. O cenário de simulação considera um tempo total de simulação de 80 segundos sendo que em  $t = 10$  s se aplica uma perturbação tipo degrau de amplitude  $-0,2$  em  $q$ . A resposta para mudanças de referência será omitida, pois o filtro do erro de predição não tem efeito sobre ela no caso nominal, que pode ser visto na Seção 3.4.1. A sintonia é dada por:  $T_s = 1,0$  s,  $N_{ss,l,p} = N_{ss,q,l,p} = 100$ ,  $\forall l, p$ ,  $N_{1,1} = 2$ ,

$N_{1,2} = 3$ ,  $N_{2,1} = 40$ ,  $N_{2,2} = 30$  e  $N_{u,1} = N_{u,2} = 5$ . Já as ponderações são normalizadas pelos horizontes e são dadas por  $\delta_l = \frac{1}{N_l}$ ,  $\lambda_p = \frac{1}{N_{u,p}}$ ,  $\forall l, p$ .

A título de exemplo, usaremos a estrutura do filtro do erro de predição abaixo:

$$F_{e,l} = \frac{az + b}{z}.$$

Veja que o polo do filtro está em 0, tornando o filtro bastante agressivo, o que geraria problemas com robustez. Porém, aqui, estamos preocupados apenas na rejeição de perturbações no caso nominal. Os polos indesejados são os presentes em  $\mathbf{P}_q(s)$ . Após discretização, temos:

$$z_1 = e^{-1/14,9} \approx 0,9351, \quad z_2 = e^{-1/13,2} \approx 0,9270.$$

Assim, sabendo que os atrasos mínimos discretos são 1 e 3 para as saídas 1 e 2, respectivamente, as condições que o filtro do erro de predição precisa satisfazer são:

$$\begin{aligned} F_{e,1}(1) = 1, \quad z^1 - F_{e,1}(z) \Big|_{z=z_1} &= 0 \\ F_{e,2}(1) = 1, \quad z^3 - F_{e,2}(z) \Big|_{z=z_2} &= 0, \end{aligned}$$

o que resulta no seguinte filtro:

$$F_e(z) = \begin{bmatrix} \frac{1,935z-0,9351}{z} & 0 \\ 0 & \frac{3,583z-2,583}{z} \end{bmatrix}.$$

Os resultados para o cenário de rejeição de perturbações proposto podem ser vistos na Figura 6.24. Veja que no caso em que  $F_e(z) = 1$ , ou seja, sem filtro do erro de predição, a resposta é lenta para a rejeição de perturbações, pois os polos de  $\mathbf{P}_q(s)$  aparecem na resposta de malha fechada. Já com  $F_e(z)$  ajustado adequadamente, a rejeição é mais rápida, como já discutido nos vários exemplos mostrados até aqui.

### 6.3.5 O caso não linear

Como vimos no Capítulo 5, se utilizarmos um modelo não linear da planta, é possível utilizar algoritmos como NGPC, NEPSAC ou PNMPC, que permitem resolver o problema de otimização associado unicamente usando um QP. Caso o modelo não linear do processo possa ser escrito de tal forma a separar o atraso do restante da dinâmica não linear, o uso da compensação de atraso acaba sendo direta, e é possível formular estratégias DTC-NMPC usando as mesmas ideias apresentadas no caso do DTC-GPC.

Vamos considerar aqui o modelo genérico de tempo discreto do processo com atraso como no caso do PNMPC, no caso MIMO (sistema com  $n$  saídas,

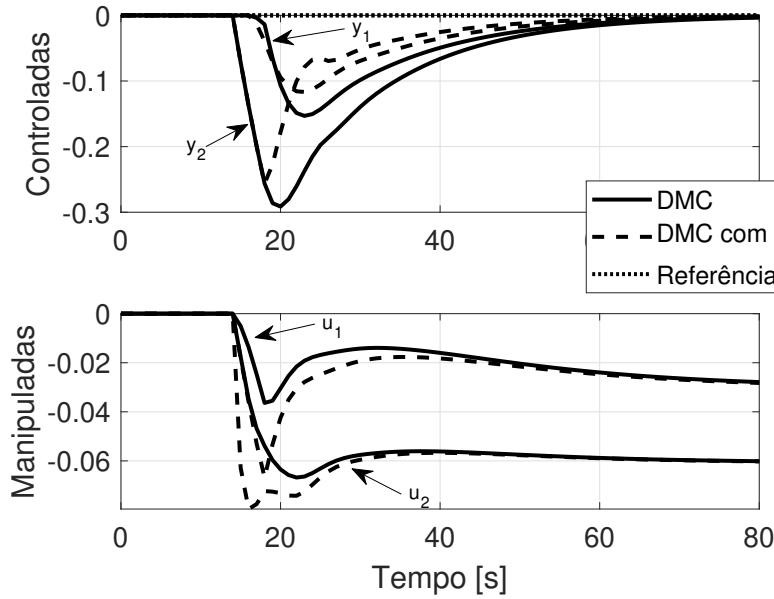


Figura 6.24: Exemplo 6.13 – resposta e sinal de controle em malha fechada para rejeição de perturbações para o DMC com e sem filtro do erro de predição  $F_e(z)$ .

$m$  entradas e  $m_q$  perturbações), onde cada saída  $y_i(k)$  tem associado um atraso efetivo  $d_i$ , que é o tempo que transcorre entre a aplicação de uma variação em todas as ações de controle e o aparecimento do efeito dessa variação na saída. Considere, então, que é possível obter um modelo da saída no tempo de atraso do processo, na forma:

$$\hat{y}_i(k + d_i) = f_i(\hat{\mathbf{y}}_{p,i}(k + d_i - 1), \mathbf{u}_p(k - 1), \mathbf{q}_p(k + d_i - d_{q,i})), \quad i = 1 \dots n,$$

onde  $\hat{\mathbf{y}}_{p,i}(k - 1)$  contém as predições e valores passados da saída,  $\mathbf{u}_p$  e  $\mathbf{q}_p$  são, respectivamente, os vetores contendo as informações passadas dos  $m$  controles e das  $m_q$  perturbações medidas que afetam a saída  $y_i$ , com  $i = 1 \dots n$ . Neste modelo,  $d_{q,i}$  representa o atraso efetivo associado às perturbações medidas. Este modelo permite ter uma predição até o tempo de atraso de cada saída.

Já a predição da saída no instante atual,  $\hat{y}_i(k)$ , é calculada usando:

$$\hat{y}_i(k) = f_i(\mathbf{y}_{p,i}(k - 1), \mathbf{u}_p(k - d_i - 1), \mathbf{q}_p(k - d_{q,i})), \quad i = 1 \dots n.$$

Observe que usando esses modelos podemos construir uma predição não linear até o tempo de atraso usando a forma proposta no PSF, para cada  $i$ , como:

$$\hat{y}_{p,i}(k) = \hat{y}_i(k + d_i) + F_{e,i}(z)[y_i(k) - \hat{y}_i(k)],$$

sendo  $y_i(k)$  a saída medida.

Adicionando essas equações ao modelo do processo, podemos considerar agora um controle PNMPCC que tem  $\mathbf{u}$  como variável manipulada e  $\mathbf{y}_p$  como variável controlada, e que enxerga um modelo do processo sem os atrasos mínimos. Projetando o controle para esse novo modelo da forma tradicional, o conjunto pode ser visto como um DTC-PNPMC, que tem os filtros  $F_{e,i}(z)$  como parâmetros adicionais de ajuste para melhorar a robustez ou rejeição de perturbações o sistema.

A mesma ideia pode ser usada com o NEPSAC ou NGPC, ou qualquer outra abordagem que permita a construção dos modelos de predição não lineares para as saídas e para as saídas adiantadas do atraso mínimo. Conceitualmente, o filtro tem o mesmo efeito que no caso linear, mas matematicamente a demonstração das propriedades é complexa e fica fora do escopo deste livro. Por exemplo, o ajuste do filtro para o caso de modelos não lineares com comportamento instável é um tema de pesquisa. Já no caso de modelos não lineares estáveis, existem resultados que mostram as vantagens da sua utilização.

Vamos apresentar um exemplo do uso desta abordagem em um processo não linear multivariável.

**Exemplo 6.14.** *Colocar o exemplo da tese do Daniel? CSTR com atraso? O mesmo exemplo do cap de NMPC mas colocando atraso nas duas saídas, mostrando erro de modelagem*

## 6.4 Exercícios propostos

1. Para o PSF obtivemos uma função de transferência entre a perturbação e a saída dada por  $H_q = \frac{Y(z)}{Q(z)}$ :

$$H_q = \frac{z^{-d_n} N_g(z)}{D_g(z)} \left[ \frac{D_c(z) D_g(z) D_e(z) z^{d_n} + N_c(z) N_g(z) (D_e(z) z^{d_n} - N_e(z))}{(D_c(z) D_g(z) + N_c(z) N_g(z)) D_e(z)} \right].$$

Considere que o modelo não tem polos múltiplos. Mostre que a condição estabelecida para o filtro  $F_e$

$$(D_e(z) z^{d_n} - N_e(z)) \Big|_{z=z_i} = 0, \quad \forall z_i,$$

para cada polo de malha aberta  $z_i$  indesejado do modelo da planta, implica que  $\frac{Y(z)}{Q(z)}$  não possui os polos indesejados do modelo do processo.

Ainda, se o controle primário for integrador e  $F_e(1) = 1$ ,  $\frac{Y(z)}{Q(z)}$  tem ganho estático nulo.

2. Considere um processo duplo integrador com atraso com modelo função de transferência dado por  $P(z) = \frac{0,1z^{-5}}{(z-1)^2}$ . Ajuste um PSF para que o sistema em malha fechada rejeite, em regime permanente, perturbações do tipo degrau. Analise o ajuste dos polos do filtro de predição  $F_e$  e seu efeito na performance e robustez do sistema em malha fechada. Utilize as respostas no tempo e diagramas frequenciais para a análise.
3. Considere um processo integrador com atraso, como modelo dado por:

$$y(k) = y(k-1) + u(k-5) - q(k),$$

que pode representar a dinâmica de um sistema de armazenamento, onde  $u(k)$  são as entradas de produto (sinal de controle),  $q(k)$  as saídas de produto (perturbação) e  $y(k)$  a quantidade de produto armazenado. O atraso neste processo modela a demora entre a ordem enviada ao fornecedor do produto e a efetiva chegada do mesmo no armazém. Considere que as perturbações podem ser modeladas por  $q(k) = 3\text{sen}(k\pi/60)$ . Ajuste o PSF para que o sistema rejeite esta perturbação e siga referências do tipo degrau com erro nulo, e tenha respostas transitórias sem oscilações.

4. Os modelos de primeira ordem com atraso são os mais utilizados na prática para representar o comportamento de diversos processos. Assim, realize um estudo comparativo de performance e robustez entre o GPC e DTC-GPC para os sistemas de primeira ordem com atraso dados por:

$$P_1(z) = \frac{0,5z^{-10}}{z-0,9},$$

$$P_2(z) = \frac{0,1z^{-10}}{z-1},$$

$$P_3(z) = \frac{0,2z^{-10}}{z-1,2}.$$

Em todos os casos analise, analiticamente, o filtro de predição  $F_e$  equivalente de cada controle para o ajuste que você determinou.

5. Projete um GPC MIMO para o processo do Exemplo 6.10 definindo os horizontes e ponderações. Simule o sistema no caso nominal e com erro de modelagem dos atrasos. Compare os resultados com um DTC-GPC-MIMO com os mesmos horizontes e ponderações e diferentes valores de ajuste dos polos dos filtros  $z_1$  e  $z_2$ . Foque a análise na rejeição de perturbações e na robustez.

6. Realize uma comparação entre o DMC e o PS para o mesmo processo do Exemplo ???. Compare com o GPC e observe as diferenças entre o comportamento dos três controles.
7. Para os sistemas de primeira ordem do Exercício 4 ajuste um controle GDMC e analise o efeito do ajuste do filtro nas respostas e na robustez.
8. Ajuste um GDMC para o mesmo processo do Exercício 5 e compare os resultados com o do DTC-GPC.

## Capítulo 7

# Métodos de Otimização e Algoritmos para o MPC

Como vimos nos capítulos anteriores e mais especificamente no Capítulo 3.1, se o MPC for implementado para o caso com restrições, a obtenção do sinal de controle se dá por meio do cálculo de um problema de otimização. Dessa forma, é necessária a implementação de um algoritmo de otimização junto ao controlador. Muitas vezes os engenheiros de controle utilizam o otimizador como uma caixa preta, ou seja, apenas utilizam sem entender o que se passa internamente. Entretanto, o recurso computacional para resolver o problema de otimização em um período de amostragem pequeno, e se obter a ação de controle a cada iteração, é um ponto que requer atenção para a implementação do MPC. Essa questão se torna ainda mais sensível quando o objetivo é implementar o MPC em sistemas embarcados como microcontroladores, DSPs, FPGAs, etc. Portanto, a escolha do otimizador do MPC é crítico para se obter um algoritmo eficiente. A demanda por algoritmos eficientes de controle é aumentada atualmente pela necessidade de controladores avançados nos níveis mais baixos da pirâmide de produção que, consequentemente, devem operar com períodos de amostragem mais rápidos, como, por exemplo, da ordem de milissegundos. Para que um algoritmo de controle seja considerado eficiente para uma implementação embarcada, podemos destacar algumas características importantes como:

1. ser capaz de computar as ações de controle em um período de amostragem previamente definido, com um pior caso de tempo de execução (WCET, do inglês *Worst Case Execution Time*) previsível, de modo a atender requisitos de tempo real;
2. necessitar de pouca memória para guardar os dados que definem o problema de otimização e o código que implementa a solução;

3. resultar em um código de controle que seja simples o suficiente para ser verificado/validado/certificado, fácil de entender e que sinalize claramente quando em caso de falhas, especialmente em aplicações críticas.

Para auxiliar o leitor que não é familiarizado com a área de otimização convexa, algumas definições importantes são disponibilizadas no Apêndice B e serão citadas no decorrer do texto.

De forma geral, para as formulações lineares como o DMC e o GPC, o problema de otimização se apresenta como um problema de minimização, no qual deseja-se obter o valor do vetor  $\Delta \mathbf{u}$  que minimiza a função custo e está sujeito à restrições:

$$\begin{aligned} \min \quad & \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u} + c \\ \text{s.a.} \quad & \bar{\mathbf{R}} \Delta \mathbf{u} \leq \bar{\mathbf{r}}. \end{aligned} \tag{7.1}$$

Esse tipo de problema é conhecido na área de otimização como um problema de programação quadrática (QP, Definição 5), por ter uma função custo quadrática e restrições lineares. Os detalhes e discussões para a obtenção das matrizes e vetores que compõem o problema de otimização para cada tipo de formulação MPC foram discutidas nas Seções 3.2, 3.3 para o DMC e o GPC, respectivamente. A forma de representação das restrições é bastante genérica e também foi discutida previamente.

Perceba que os algoritmos para a implementação de um MPC podem ser divididos em duas partes, a primeira responsável pela obtenção das matrizes que compõem o controlador e a segunda pelo método de otimização. Dessas duas partes, podemos dividir também o que pode ser computado previamente e que não mudam durante o ciclo de controle que chamaremos de cômputo offline e o que varia durante o ciclo de controle que chamaremos de cômputo online. O foco principal será em reduzir o custo computacional dos algoritmos que participam do cômputo online e que afetam diretamente o tempo de cômputo da ação de controle. Portanto, neste capítulo serão apresentados algoritmos para a implementação do cálculo da ação de controle e alguns métodos de resolução do QP, adequados ao contexto de MPC. Primeiramente serão apresentados opções de algoritmos para a implementação do GPC e DMC, deixando a chamada para o otimizador de forma genérica, e posteriormente os algoritmos dos métodos de otimização que poderão ser utilizados. Será abordado um método clássico como o método de barreira logarítmica e também alguns métodos menos custosos computacionalmente, como o método de projeção de gradiente acelerado dual e o método dos multiplicadores em direções alternadas. Esses métodos podem ser utilizados em plantas com



dinâmicas mais rápidas e até mesmo ser implementados em hardwares mais modestos como microcontroladores, DSPs e FPGAs.

## 7.1 Algoritmo DMC recursivo

A forma recursiva para a implementação do DMC apresentada para o caso SISO no Volume 1 e generalizado para o caso MIMO na Seção 3.2.3 é mais adequada para a formulação do algoritmo DMC. Essa formulação requer menos espaço de memória para armazenar as variáveis que modelam o problema de otimização e um número menor de operações matemáticas, dessa forma, o cômputo da resposta livre fica mais eficiente.

Podemos propor um algoritmo para o DMC recursivo para o caso SISO, como apresentado no Algoritmo 1. Perceba que as variáveis de entrada são o vetor de coeficientes  $\mathbf{g}$  que representa o modelo da planta utilizado para a predição, e também as restrições desejadas, já na sua representação matricial  $\bar{\mathbf{R}}$  e  $\bar{\mathbf{r}}$ . Os dados são os parâmetros utilizados para a sintonização do controlador, como os horizontes e ponderações. Avaliando o algoritmo, percebemos que entre as linhas 1 e 8, as variáveis são inicializadas e a matriz  $\mathbf{H}$  do DMC é definida. Essa parte do algoritmo representa o cômputo offline e não é crítico para o tempo de cômputo da ação de controle. Já a parte entre as linhas 9 e 22 representam a parcela de cômputo online. Perceba que é feita uma divisão para quando o controlador está em modo automático, trabalhando de forma realimentada, e o modo manual. O modo manual foi omitido do algoritmo, mas basicamente permite com que o operador atue diretamente no atuador da planta. O laço condicional da linha 10 deve acontecer periodicamente, em um tempo pré-definido, que corresponde ao período de amostragem do sistema de controle. Frequentemente são utilizadas rotinas de interrupção de temporizadores para cumprir esse requisito em sistemas embarcados. Na linha 11, é feita a obtenção do valor da variável controlada e da referência. Essas tarefas são deixadas de forma genérica, pois podem ser obtidas de diversas formas, como a leitura de conversores analógicos para digitais, recebimento por redes de comunicação, etc. Das linhas 12 até a 15 são definidos os passos para o cálculo da resposta livre do DMC de forma recursiva. Na linha 16 é computado o vetor  $\mathbf{b}$  que é dependente da atualização de  $\mathbf{f}$ . Na linha 17 é computado o vetor de incrementos de controle a partir da resolução do problema de otimização. Nesse algoritmo, fazemos apenas uma chamada de função, no estilo caixa preta, passando como parâmetros as matrizes e vetores do QP e suas restrições. Nas Seções seguintes, apresentaremos alguns possíveis algoritmos de otimização para serem implementados com o controlador. Ao final do algoritmo, linhas 18 e 19, é selecionado apenas o

primeiro incremento de controle calculado e posteriormente a ação de controle a ser aplicada ao atuador. A aplicação da ação de controle também foi omitida do algoritmo pela mesma razão da obtenção da variável controlada e da referência, ou seja, pode ser feito de diversas maneiras, de acordo com aplicação, como, por exemplo, por PWM, redes de comunicação, etc.

---

**Algoritmo 1: DMC SISO**

---

**Entrada:**  $\mathbf{g}, \bar{\mathbf{R}}, \bar{\mathbf{r}}$   
**Saída:**  $u(k)$   
**Dados:**  $\mathbf{Q}_\delta, \mathbf{Q}_\lambda, N_u, N_1, N_2, N_{ss}, u(-1)$

```

1 início
2    $N \leftarrow N_2 - N_1 + 1;$ 
3   para  $i = 1 : N_u$  faça
4      $\mathbf{G}_{i:N,i} \leftarrow \mathbf{g}_{N_1:N_2+1-i};$ 
5   fim
6    $\mathbf{H} \leftarrow 2(\mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_\lambda);$ 
7    $\hat{\mathbf{f}}(-1) \leftarrow \mathbf{1}_{N_{ss}} y(k);$ 
8    $k \leftarrow 0;$ 
9   enquanto modo automático faça
10    se tempo de amostragem então
11      Adquire saída  $y(k)$  e referência  $w(k);$ 
12       $\hat{\mathbf{f}}(k) \leftarrow \hat{\mathbf{f}}(k-1) + \mathbf{g} \Delta u(k-1);$ 
13       $e \leftarrow (y(k) - \hat{\mathbf{f}}_0);$ 
14       $\hat{\mathbf{f}}(k) \leftarrow [\hat{\mathbf{f}}_1 \ \hat{\mathbf{f}}_2 \ \dots \ \hat{\mathbf{f}}_{N_{ss}-1} \ \hat{\mathbf{f}}_{N_{ss}-1}]^T;$ 
15       $\mathbf{f} \leftarrow \hat{\mathbf{f}}_{N_1:N_2} + \mathbf{1}_{N \times 1} e;$ 
16       $\mathbf{b}^T \leftarrow 2(\mathbf{f} - \mathbf{w})^T \mathbf{Q}_\delta \mathbf{G};$ 
17       $\Delta \mathbf{u}(k) \leftarrow \text{Otimizador}(\mathbf{H}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}});$ 
18       $\Delta u(k) \leftarrow [1 \ 0 \ \dots \ 0]_{1 \times N_u} \Delta \mathbf{u};$ 
19       $u(k) \leftarrow u(k-1) + \Delta u(k);$ 
20       $k \leftarrow k + 1;$ 
21    fim
22  fim
23 fim
```

---

O algoritmo para o caso MIMO do DMC recursivo pode ser visto no Algoritmo 2. Os passos são os mesmos do caso SISO, exceto pela generalização das variáveis para o caso MIMO.

---

**Algoritmo 2: MIMO DMC**


---

**Entrada:**  $\mathbf{g}^{(1,1)}, \dots, \mathbf{g}^{(n_o, n_i)}, \bar{\mathbf{R}}, \bar{\mathbf{r}}, u^{(1)}(k-1), \dots, u^{(n_i)}(k-1)$   
**Saída:**  $u^{(1)}(k), \dots, u^{(n_o)}(k)$   
**Dados:**  $\mathbf{Q}_\lambda, \mathbf{Q}_\delta, \mathbf{N}_u, \mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_{ss}, \psi_0 = \psi_{-1} = \mathbf{0}, \epsilon_f, \epsilon_o, j_{max}$

```

1 início
2   para  $p = 1 : n_o$  faça
3      $N^{(p)} \leftarrow N_2^{(p)} - N_1^{(p)} + 1$ ;
4     para  $l = 1 : n_i$  faça
5       para  $i = 1 : N_u^{(l)}$  faça
6          $\mathbf{G}_{i:N^{(p)}, i}^{(p,l)} \leftarrow \mathbf{g}_{N_1^{(p)}:N_2^{(p)}+1-i}^{(p,l)}$ ;
7       fim
8     fim
9      $\mathbf{G}_{mpl} \leftarrow \mathbf{G}^{(pl)}$ ;
10     $\hat{\mathbf{f}}^{(p)} \leftarrow \mathbf{1}_{N_{ss}^{(p)}} y^{(p)}(0)$ ;
11  fim
12   $\mathbf{H} \leftarrow (\mathbf{G}_m^T \mathbf{Q}_\delta \mathbf{G}_m + \mathbf{Q}_\lambda)$ ;
13   $k \leftarrow 0$ ;
14  enquanto modo automático faça
15    se período de amostragem então
16      Obtem as saídas  $y^{(1)}(k) \dots y^{(n_o)}(k)$ ;
17      Define as referências  $\mathbf{w}_m \leftarrow [\mathbf{w}^{(1)}(k) \dots \mathbf{w}^{(n_o)}(k)]^T$ ;
18      para  $p = 1 : n_o$  faça
19         $\hat{\mathbf{f}}^{(p)} \leftarrow \hat{\mathbf{f}}^{(p)} + \mathbf{g}^{(p)} \Delta u^{(p)}(k-1)$ ;
20         $e^{(p)} \leftarrow (y^{(p)}(k) - \hat{f}_0^{(p)})$ ;
21         $\hat{\mathbf{f}}^{(p)} \leftarrow [\hat{f}_1^{(p)} \dots \hat{f}_{N_{ss}^{(p)}-1}^{(p)} \hat{f}_{N_{ss}^{(p)}-1}^{(p)}]^T$ ;
22         $\mathbf{f}^{(p)} \leftarrow \hat{\mathbf{f}}_{N_1^{(p)}:N_2^{(p)}}^{(p)} + \mathbf{1}_{N^{(p)} \times 1} e^{(p)}$ ;
23      fim
24       $\mathbf{f}_m \leftarrow [\mathbf{f}^{(1)} \dots \mathbf{f}^{(n_o)}]^T$ ;
25       $\mathbf{b}^T \leftarrow (\mathbf{f}_m - \mathbf{w}_m)^T \mathbf{Q}_\delta \mathbf{G}_m$ ;
26       $\Delta \mathbf{u}(k) \leftarrow \text{Otimizador}(\mathbf{H}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}})$ ;
27       $i = 1$ ;
28      para  $l = 1 : n_i$  faça
29         $\Delta u^{(l)}(k) \leftarrow \Delta u_i$ ;
30         $i \leftarrow i + N_u^{(l)}$ ;
31         $u^{(l)}(k) \leftarrow u^{(l)}(k-1) + \Delta u^{(l)}(k)$ ;
32      fim
33       $k \leftarrow k + 1$ ;
34    fim
35  fim
36 fim

```

---

## 7.2 Algoritmo GPC recursivo

A forma de obtenção da lei de controle utilizando a equação Diofantina, apresentada na Seção ??, nem sempre é a mais adequada para a utilização em algoritmos embarcados, devido à operação de divisão requerida. Em casos de baixa precisão na representação numérica, a operação de divisão aumenta o acúmulo de erros. Dessa forma, pode-se optar por uma formulação direta, na qual a matriz  $\mathbf{G}$  e o vetor  $\mathbf{f}$  podem ser calculadas recursivamente. Os elementos  $g_i$  que compõem a matriz  $G$  e  $f_i$  que compõem os elementos do valor de resposta livre podem ser computados por:

$$\begin{aligned} g_i &= z(1 - A(z^{-1}))g_{i-1} + B(z^{-1})\bar{u}(i), \\ f_i &= z(1 - \tilde{A}(z^{-1}))f_{i-1} + B(z^{-1})\Delta u(k - d + i - 1), \end{aligned} \quad (7.2)$$

com  $i = 1, \dots, N_2$ ,  $\bar{u}(i)$  sendo um degrau unitário,  $g_i = 0$  para  $i \leq 0$ ,  $f_i = y(k + i)$  para  $i \leq 0$  e  $\Delta u(k + j) = 0$  para  $j \geq 0$ . Após o cômputo dos elementos  $g_i$  e  $f_i$ , a matriz  $\mathbf{G}$  é definida da mesma forma de (??) e o vetor  $\mathbf{f}$  pode ser escrito como:

$$\mathbf{f} = \begin{bmatrix} f_{N_1} \\ f_{N_1+1} \\ \vdots \\ f_{N_2} \end{bmatrix}.$$

Vale ressaltar que em casos práticos, com sistemas estáveis, a matriz  $\mathbf{G}$  frequentemente é obtida de forma experimental a partir dos coeficientes da resposta ao degrau em malha aberta que compõem a primeira coluna  $\mathbf{G}$ .

O algoritmo do GPC recursivo pode ser visto no Algoritmo 3.

No caso MIMO, a matriz  $\mathbf{G}_m$  é formada por outras matrizes  $\mathbf{G}^{(p,l)}$ , onde  $\mathbf{G}^{(p,l)}$  é uma matriz de coeficientes da resposta ao degrau que relaciona a saída  $p$  com a entrada  $l$ . Portanto, a matriz  $\mathbf{G}_m$  é dada por:

$$\mathbf{G}_m = \begin{bmatrix} \mathbf{G}^{(1,1)} & \dots & \mathbf{G}^{(1,n_i)} \\ \mathbf{G}^{(2,1)} & \dots & \mathbf{G}^{(2,n_i)} \\ \vdots & \ddots & \vdots \\ \mathbf{G}^{(n_o,1)} & \dots & \mathbf{G}^{(n_o,n_i)} \end{bmatrix}.$$

Para se obter o vetor de resposta livre generalizado  $\mathbf{f}_m$ , as respostas livres para cada saída do sistema são agrupadas como:

$$\mathbf{f}_m = [\mathbf{f}^{(1)^T} \dots \mathbf{f}^{(n_o)^T}]^T,$$

---

**Algoritmo 3:** Controle preditivo generalizado
 

---

**Entrada:**  $A, B, d, \bar{\mathbf{R}}, \bar{\mathbf{r}}$   
**Saída:**  $u(k)$   
**Dados:**  $\mathbf{Q}_\lambda, \mathbf{Q}_\delta, N_u, N_1, N_2, u(-1)$

```

1 início
2    $\tilde{A}(z^{-1}) \leftarrow (1 - z^{-1})A(z^{-1});$ 
3   para  $i = 0 : N_2$  faça
4      $| \quad g_i \leftarrow z(1 - A(z^{-1}))g_{i-1} + B(z^{-1})\bar{u}_i;$ 
5   fim
6   para  $i = 1 : N_u$  faça
7      $| \quad \mathbf{G}_{i:N,i} \leftarrow \mathbf{g}_{N_1:N_2+1-i};$ 
8   fim
9    $\mathbf{H} \leftarrow 2(\mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_\lambda);$ 
10   $k \leftarrow 0;$ 
11  enquanto modo automatico faça
12    se tempo de amostragem então
13      Adquire saída  $y(k)$  e referência  $w(k)$ ;
14       $f_0 \leftarrow y(k);$ 
15      para  $i = 1 : N_2$  faça
16         $| \quad f_i \leftarrow z(1 - \tilde{A}(z^{-1}))f_{i-1} + B(z^{-1})\Delta u(k - d + i - 1);$ 
17      fim
18       $\mathbf{b}^T \leftarrow 2(\mathbf{f} - \mathbf{w})^T \mathbf{Q}_\delta \mathbf{G};$ 
19       $\Delta \mathbf{u}(k) \leftarrow \text{Otimizador}(\mathbf{H}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}});$ 
20       $\Delta u(k) \leftarrow [1 \quad 0 \quad \cdots \quad 0]_{1 \times N_u} \Delta \mathbf{u};$ 
21       $u(k) \leftarrow u(k - 1) + \Delta u(k);$ 
22       $k \leftarrow k + 1 ;$ 
23    fim
24  fim
25 fim

```

---

onde  $\mathbf{f}_m^{(i)}$  tem a mesma forma de (7.2). Portanto, o vetor de resposta livre para cada saída  $p$  pode ser obtido recursivamente como:

$$f_i^{(p)} = z(1 - \tilde{\mathbf{A}}_{pp}(z^{-1}))f_{i-1}^{(p)} + \sum_{l=1}^{n_i} \mathbf{B}_{pl}(z^{-1})\Delta u^{(l)}(k - \mathbf{D}_{pp} + i - 1), \quad (7.3)$$

com  $i = 1, \dots, N_2^{(p)}$ ,  $p = 1, \dots, n_o$ ,  $l = 1, \dots, n_i$ ,  $f_i^{(p)} = y^{(p)}(k + i)$  para  $i \leq 0$  e  $\Delta u^{(l)}(k + j) = 0$  para  $j \geq 0$ .

O algoritmo do GPC recursivo para o caso MIMO é apresentado no Algoritmo 4.

---

**Algoritmo 4: MIMO GPC**


---

**Entrada:**  $A^{(1)}, \dots, A^{(n_o)}, B^{(1,1)}, \dots, B^{(n_c, n_o)}, \mathbf{D}, \mathbf{R}, \mathbf{r}$

**Saída:**  $u^{(1)}(k), \dots, u^{(n_c)}(k)$

**Dados:**  $n_o, n_c, \mathbf{Q}_\lambda, \mathbf{Q}_\delta, N_u^{(1)}, \dots, N_u^{(n_c)}, N_1^{(1)}, \dots, N_1^{(n_o)}, N_2^{(1)}, \dots, N_2^{(n_o)}, \epsilon_f, \epsilon_o$

```

1 início
2   para  $p = 1 : n_o$  faça
3      $\tilde{A}^{(p)}(z^{-1}) \leftarrow (1 - z^{-1})A^{(p)}(z^{-1});$ 
4     para  $l = 1 : n_c$  faça
5       para  $i = 1 : N_2^{(p)}$  faça
6          $g_i^{(p,l)} \leftarrow z(1 - A^{(p)}(z^{-1}))g_{i-1}^{(p,l)} + B^{(p,l)}(z^{-1})\bar{u}_i;$ 
7       fim
8       para  $i = 1 : N_u^{(l)}$  faça
9          $\mathbf{G}_{i:N_2^{(p)}-N_1^{(p)}+1,i}^{(p,l)} \leftarrow \mathbf{g}_{N_1^{(p)}:N_2^{(p)}+1-i}^{(p,l)};$ 
10      fim
11    fim
12  fim
13   $\mathbf{H} \leftarrow 2(\mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_\lambda);$ 
14   $k \leftarrow 0;$ 
15  enquanto modo automático faça
16    se período de amostragem então
17      para  $p = 1 : n_o$  faça
18        Obtem as saídas  $y^{(p)}(k)$  e referências  $w^{(p)}(k);$ 
19         $f_0^{(p)} \leftarrow y^{(p)}(k);$ 
20        para  $i = 1 : N_2^{(p)}$  faça
21           $f_i^{(p)} \leftarrow z(1 - \tilde{A}^{(p)}(z^{-1}))f_{i-1}^{(p)} + \sum_{l=1}^{n_c} B^{(p,l)}(z^{-1})\Delta u^{(l)}(k -$ 
22             $\mathbf{D}^{(p)} + i - 1);$ 
23        fim
24      fim
25       $\mathbf{f} \leftarrow [\mathbf{f}_{N_1^{(1)}:N_2^{(1)}}^{(1)} \dots \mathbf{f}_{N_1^{(n_o)}:N_2^{(n_o)}}^{(n_o)}]^T;$ 
26       $\mathbf{b}^T \leftarrow 2(\mathbf{f} - \mathbf{w})^T \mathbf{Q}_\delta \mathbf{G};$ 
27       $\Delta \mathbf{u}(k) \leftarrow \text{Otimizador}(\mathbf{H}, \mathbf{b}, \mathbf{R}, \bar{\mathbf{r}});$ 
28       $i = 1;$ 
29      para  $l = 1 : n_c$  faça
30         $\Delta u^{(l)}(k) \leftarrow \Delta \mathbf{u}_i;$ 
31         $i \leftarrow i + N_u^{(l)};$ 
32         $u^{(l)}(k) \leftarrow u^{(l)}(k-1) + \Delta u^{(l)}(k);$ 
33      fim
34       $k \leftarrow k + 1;$ 
35    fim
36  fim

```

---

## 7.3 Método de barreira logarítmica

O método de barreira logarítmica é bastante popular para a resolução do QP. Trata-se de uma adaptação dos métodos do tipo ponto interior para tratar restrições de desigualdade. A classe de métodos de ponto interior partem de um ponto factível dentro dos limites das variáveis para encontrar a solução a partir do problema irrestrito e as condições de otimalidade. Para resolver o problema irrestrito, apesar de qualquer método de minimização linear com restrições de igualdade poder ser utilizado, em geral, utiliza-se o método de Newton por conta da rápida convergência. Para entendermos o método da barreira logarítmica é importante discutirmos o funcionamento básico do método de Newton primeiro.

### 7.3.1 Método de Newton

O método de Newton foi desenvolvido para se obter as raízes de uma determinada função. Entretanto, ele pode ser utilizado para a minimização de funções se for considerado o critério de otimalidade no qual o mínimo de uma função convexa (Definição 2) ocorre quando a derivada da função é igual a zero.

Para obtermos uma forma de calcular os valores que minimizam uma função, podemos aproximar a derivada da função  $J$  por uma série de Taylor truncada no termo linear:

$$J'(\Delta u + \delta) \approx J'(\Delta u) + J''(\Delta u)(\Delta u + \delta - \Delta u). \quad (7.4)$$

Perceba que uma aproximação linear da derivada é um bom modelo para uma função quadrática. Agora, se igualarmos esse modelo à zero,

$$J'(\Delta u) + J''(\Delta u)(\Delta u + \delta - \Delta u) = 0,$$

podemos obter uma forma de calcular o argumento que minimiza a função  $J$ . Isolando o termo a ser calculado,  $\Delta u + \delta$ , temos:

$$\Delta u + \delta = \Delta u - \frac{J'(\Delta u)}{J''(\Delta u)}. \quad (7.5)$$

Perceba que se trata de uma aproximação linear da derivada da função, e que nem sempre esse é um bom modelo para todas as funções convexas, como veremos a partir da utilização da barreira logarítmica. Portanto, é comum representar o algoritmo de Newton de forma iterativa e com um tamanho de passo  $l$  menor do que 1 e até ajustado dinamicamente. De forma simplificada,

podemos pensar que um método iterativo de minimização tem por objetivo encontrar valores menores da função objetivo a cada iteração, tal que:

$$J(\Delta u_{i+1}) < J(\Delta u_i). \quad (7.6)$$

Dessa forma, o método iterativo de Newton é dado por:

$$\Delta u_{i+1} = \Delta u_i - l \frac{J'(\Delta u_i)}{J''(\Delta u_i)}. \quad (7.7)$$

Para generalizar o método para variáveis de dimensões maiores, podemos redefinir a noção de derivada para o gradiente e a Hessiana e dessa forma o passo principal do método iterativo de Newton passa a ser:

$$\Delta \mathbf{u}_{i+1} = \Delta \mathbf{u}_i - l \frac{\nabla J(\Delta \mathbf{u}_i)}{\nabla^2 J(\Delta \mathbf{u}_i)} \quad (7.8)$$

Para ilustrar vamos verificar um exemplo.

**Exemplo 7.1.** *Vamos assumir o caso em que o horizonte de controle é de duas amostras  $N_u = 2$ , ou seja,  $\Delta \mathbf{u} \in \mathbb{R}^2$  é um vetor com dois elementos. Neste caso, a função objetivo tem o formato de um paraboloide em  $\Delta \mathbf{u}$ , e é dada por:*

$$J(\Delta u) = \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u}. \quad (7.9)$$

Definindo  $\mathbf{H} = \mathbf{I}$  e  $\mathbf{b} = [-1 \quad -1]$ , obtemos a representação gráfica apresentada na Figura 7.1.

Se definirmos o ponto inicial como  $\Delta \mathbf{u}_0 = [5 \quad 5]^T$  podemos calcular  $\Delta \mathbf{u}_1$ :

$$\begin{aligned} \Delta \mathbf{u}_1 &= \Delta \mathbf{u}_0 - \frac{\nabla J(\Delta \mathbf{u}_0)}{\nabla^2 J(\Delta \mathbf{u}_0)} \\ \Delta \mathbf{u}_1 &= \Delta \mathbf{u}_0 - \frac{\mathbf{H} \Delta \mathbf{u}_0 + \mathbf{b}}{\mathbf{H}} \\ \Delta \mathbf{u}_1 &= [5 \quad 5]^T - \frac{I[5 \quad 5]^T + [-1 \quad -1]^T}{I} \\ \Delta \mathbf{u}_1 &= [1 \quad 1]^T \end{aligned} \quad (7.10)$$

### 7.3.2 Barreira logarítmica

O método de barreira tem como objetivo aproximar um problema de otimização com restrições de desigualdade de um problema irrestrito ou com apenas restrições de igualdade nos quais o método de Newton pode ser aplicado.



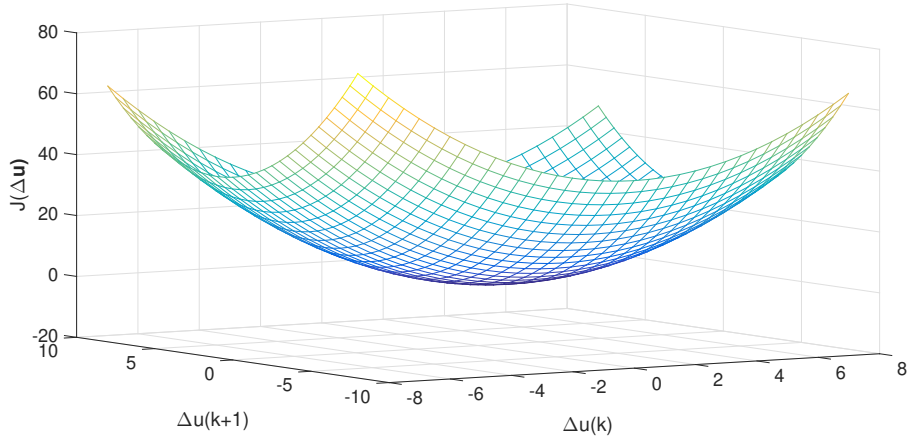


Figura 7.1: Paraboloide

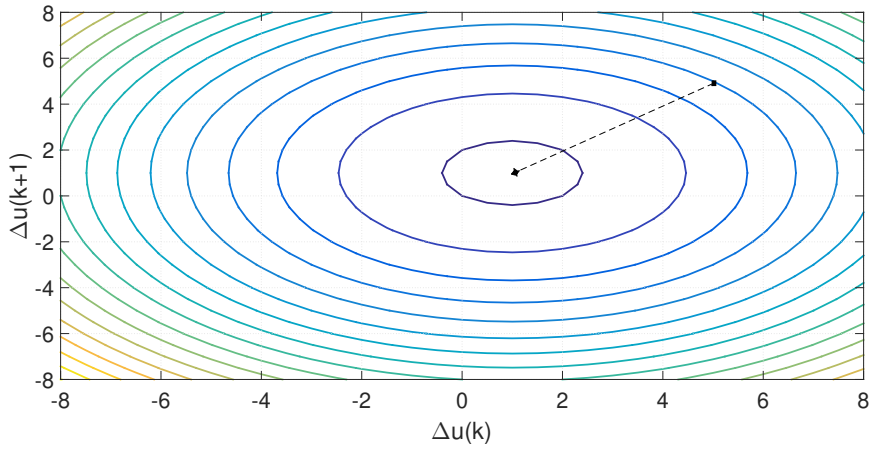


Figura 7.2: Paraboloide

De forma genérica, pode-se reescrever o problema (7.1) com as restrições de desigualdade definidas implicitamente na função objetivo:

$$\min_{\mathbf{u}} J(\mathbf{u}) + \sum_{i=0}^{n_{rin}} I_{-}([\bar{\mathbf{R}}\Delta \mathbf{u} - \bar{\mathbf{r}}]_i) \quad (7.11)$$

onde  $I_{-} : \mathbb{R} \rightarrow \mathbb{R}$  é uma função indicadora para reais não-positivos definida como:

$$I_-([\bar{\mathbf{R}}\Delta\mathbf{u} - \bar{\mathbf{r}}]_i) = \begin{cases} 0, & \text{se } [\bar{\mathbf{R}}\Delta\mathbf{u} - \bar{\mathbf{r}}]_i \leq 0 \\ \infty, & \text{se } [\bar{\mathbf{R}}\Delta\mathbf{u} - \bar{\mathbf{r}}]_i > 0 \end{cases} \quad (7.12)$$

Dessa forma, caso o valor de  $\Delta\mathbf{u}$  levasse à uma violação das restrições, a função objetivo teria um valor infinito e obviamente não seria uma solução de minimização. Esse efeito faz com que a solução obtida seja o mínimo valor da função objetivo que não viola as restrições. Entretanto, apesar de a formulação (7.11) não apresentar restrições de desigualdade, a sua função objetivo não é diferenciável e o método de Newton não pode ser aplicado. De modo a fazer uma aproximação da função indicadora, pode-se utilizar a função logarítmica [BV04]:

$$\hat{I}_- = -(1/k_b) \log(-[\bar{\mathbf{R}}\Delta\mathbf{u} - \bar{\mathbf{r}}]_i), \quad \text{dom } \hat{I}_- = -\mathbb{R}_+,$$

onde  $k_b > 0$  é um parâmetro que ajusta a exatidão da aproximação. Na Figura 7.3 é apresentado o perfil da função indicadora e as aproximações com a função logarítmica. Podemos perceber que quanto maior o parâmetro  $k_b$  melhor a aproximação da função logarítmica com a função indicadora.

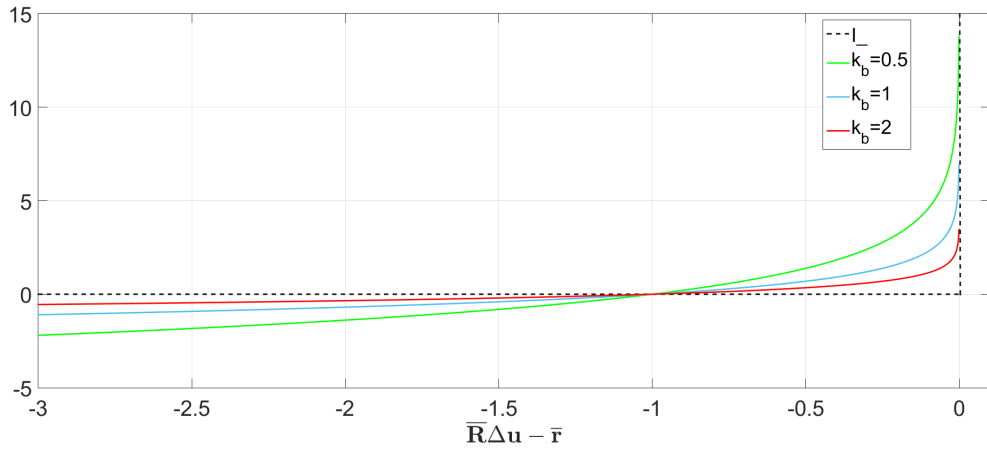


Figura 7.3: Função indicadora e as aproximações com a função logarítmica e diferentes valores de  $k_b$ .

Substituindo  $\hat{I}_-$  por  $I_-$  em (7.11) resulta o problema aproximado:

$$\min_{\Delta\mathbf{u}} J(\Delta\mathbf{u}) + \sum_{i=0}^{n_{rin}} -(1/k_b) \log(-[\bar{\mathbf{R}}\Delta\mathbf{u} - \bar{\mathbf{r}}]_i). \quad (7.13)$$

A função objetivo de (7.13) é convexa e diferenciável, logo o método de Newton se aplica. A função

$$\eta = \sum_{i=0}^{n_{rin}} -\log(-[\bar{\mathbf{R}}\Delta\mathbf{u} - \bar{\mathbf{r}}]_i),$$

com **dom**  $\eta = \Delta\mathbf{u} \in \mathbb{R}^n \mid \Delta\mathbf{u} < 0, i = 1, \dots, n_{rin}$ , é chamada de barreira logarítmica para o problema (7.13).

Como para valores altos de  $k_b$  a convergência do método de Newton é mais difícil, a ideia do método de barreira é começar com um valor mais baixo de  $k_b$  e, conseqüentemente, uma aproximação pior da função indicadora, ir aumentando o parâmetro  $k_b$  e a partir da solução anterior construir um algoritmo iterativo para obter a solução ótima. Esse conjunto de pontos de soluções intermediárias formam o chamado caminho central. A partir da definição do caminho central podemos deduzir propriedades importantes para o desenvolvimento do método.

### 7.3.3 Caminho central

Um conceito importante para o método de barreira formulado em (7.13) é o do caminho central. Simplificando a notação de (7.13) e multiplicando por  $k_b$ , obtemos:

$$\min_{\Delta\mathbf{u}} k_b J(\Delta\mathbf{u}) + \eta(\Delta\mathbf{u}) \quad (7.14)$$

o qual tem os mesmos mínimos. Assumindo que o problema (7.14) pode ser resolvido pelo método de Newton e, em particular, que tem solução única para cada  $k_b > 0$ , define-se  $\Delta\mathbf{u}^*(k_b)$  como a sua solução. O caminho central associado ao problema (??) é definido como o conjunto de pontos  $\Delta\mathbf{u}^*(k_b)$ , com  $k_b > 0$ , os quais são chamados de pontos centrais [BV04]. Os pontos centrais são caracterizados por atenderem as seguintes condições necessárias e suficientes:  $\Delta\mathbf{u}^*(k_b)$  é estritamente factível se e somente se satisfaça

$$\bar{\mathbf{R}}\Delta\mathbf{u}^* - \bar{\mathbf{r}} \leq 0, \quad (7.15)$$

e

$$k_b \nabla J(\Delta\mathbf{u}^*(k_b)) + \nabla \eta(\Delta\mathbf{u}^*(k_b)) = 0.$$

ou

$$k_b \nabla J(\Delta\mathbf{u}^*(t)) + \sum_{i=0}^{n_{rin}} \frac{1}{-[\bar{\mathbf{R}}\Delta\mathbf{u}^*(k_b) - \bar{\mathbf{r}}]_i} = 0. \quad (7.16)$$

A partir da condição de otimalidade (7.16), pode-se derivar uma importante propriedade do caminho central: todo ponto central implica a existência

de um ponto dual factível e consequentemente um limite inferior para valor ótimo primal  $J^*$  [BV04]. Para se verificar essa propriedade pode-se definir:

$$\psi^*(t) = -\frac{1}{k_b[\bar{\mathbf{R}}\Delta\mathbf{u}^*(k_b) - \bar{\mathbf{r}}]_i}, \quad i = 1, \dots, n_{rin}, \quad (7.17)$$

e expressar a condição de otimalidade (7.16) como:

$$k_b \nabla J(\Delta\mathbf{u}^*(k_b)) + \sum_{i=0}^{n_{rin}} \psi_i^*(k_b) [\bar{\mathbf{R}}\Delta\mathbf{u}^*(k_b) - \bar{\mathbf{r}}]_i = 0. \quad (7.18)$$

Pode-se verificar que  $\Delta\mathbf{u}^*(k_b)$  minimiza a função lagrangiana com os multiplicadores associados às restrições de desigualdade,  $\psi$ ,

$$\mathbb{L}(\Delta\mathbf{u}, \psi) = J(\Delta\mathbf{u}) + \sum_{i=0}^{n_{rin}} \psi_i [\bar{\mathbf{R}}\Delta\mathbf{u}^*(k_b) - \bar{\mathbf{r}}]_i, \quad (7.19)$$

para  $\psi = \psi^*$ , o que significa que  $\psi^*(k_b)$  é um ponto dual factível [BV04].

Portanto, a função dual  $J_d(\psi^*(k_b))$  é finita e

$$\begin{aligned} J_d(\psi^*(k_b)) &= J(\Delta\mathbf{u}^*(k_b)) + \sum_{i=0}^{n_{rin}} \psi_i^*(t) [\bar{\mathbf{R}}\Delta\mathbf{u}^*(k_b) - \bar{\mathbf{r}}]_i \\ &= J(\Delta\mathbf{u}^*(k_b)) - n_{rin}/k_b. \end{aligned}$$

Nesse caso particular, o resíduo dual associado com  $\Delta\mathbf{u}^*(k_b)$  e o ponto factível  $\psi^*(k_b)$  é  $n_{rin}/k_b$ . Como uma importante consequência, tem-se que:

$$J(\Delta\mathbf{u}^*(k_b)) - J^* \leq n_{rin}/k_b,$$

ou seja,  $\Delta\mathbf{u}^*(k_b)$  é no máximo  $n_{rin}/k_b$ -subótimo. Este resultado confirma a ideia intuitiva de que  $\Delta\mathbf{u}^*(k_b)$  converge para um ponto ótimo quando  $t \rightarrow \infty$ .

A partir dos conceitos apresentados pode-se formular o algoritmo do método barreira. Uma versão simplificada pode ser vista no Algoritmo 5 e se baseia na resolução de uma sequência de problemas de minimização irrestritos, utilizando o último ponto encontrado como o ponto de início da próxima iteração. A cada iteração o valor de  $k_b$  é incrementado, até que  $k_b \geq n_{rin}/\epsilon$ , o que garante que seja obtida uma solução  $\epsilon$ -subótima do problema original.

### 7.3.4 Sintonização

Em [BV04] são apresentadas várias características práticas para a sintonia do método de barreira logarítmica. A primeira delas é quanto à exatidão

---

**Algoritmo 5:** Método de barreira

---

**Entrada:**  $\Delta \mathbf{u}_0$  estritamente factível,  $k_b = k_{b0} > 0, \mu > 1$ , tolerância  $\epsilon > 0$

**Saída:**  $\Delta \mathbf{u}$

```
1 enquanto  $n_{rin}/k_b > \epsilon$  faça
2   | Centralização: computa  $\Delta \mathbf{u}^*(k_b)$  minimizando  $k_b J + \eta$ 
   | começando em  $\Delta \mathbf{u}_0$ ;
3   | Atualização:  $\Delta \mathbf{u} \leftarrow \Delta \mathbf{u}^*(k_b)$ ;
4   | barreira:  $k_b \leftarrow \mu k_b$  ;
5 fim
```

---

do passo de centralização. Os autores defendem que computar  $\Delta \mathbf{u}^*(k_b)$  no passo de centralização de forma exata é desnecessário, uma vez que o caminho central tem pouca relevância no direcionamento da solução para o problema original quando  $k_b \rightarrow \infty$ . A centralização inexata ainda continuará resultando em uma sequência de pontos  $\Delta \mathbf{u}_{(k)}$  que convergem para um ponto ótimo. Dessa forma, devido à rápida convergência do método de Newton, o passo de centralização pode ser executado com apenas algumas poucas iterações.

Para a escolha da taxa de atualização de  $k_b$ , denotada por  $\mu$ , é necessário ponderar entre o número de iterações na centralização, chamadas de iterações internas, ou de iterações no método propriamente dito, chamadas de iterações externas. Se  $\mu$  for escolhido pequeno (próximo de 1), então a cada iteração externa  $k_b$  é incrementado por um fator pequeno. Como resultado, o ponto inicial para o processo de Newton é um bom ponto inicial e o número de passos de Newton é pequeno. Por outro lado, o número de iterações externas é grande, uma vez que o resíduo é também reduzido por valores pequenos. Na prática, para uma grande faixa de valores entre 3 e 100, aproximadamente, os dois efeitos praticamente se cancelam e o número de passos de Newton fica aproximadamente constante. Valores usuais de  $\mu$  estão na faixa entre 10 e 20 [BV04].

O último parâmetro é o valor inicial de  $k_b$ . A escolha de  $k_{b0}$  grande implica uma primeira iteração externa com um grande número de iterações internas. Já para a escolha de um  $k_{b0}$  pequeno, o algoritmo irá requerer uma quantidade extra de iterações externas. Portanto, já que  $n_{rin}/k_{b0}$  é o resíduo dual que resultará do primeiro passo de centralização, uma escolha razoável de  $k_{b0}$  é tal que  $n_{rin}/k_{b0}$  seja aproximadamente da mesma ordem de grandeza da tolerância de resíduo dual desejada ou  $\mu$  vezes essa quantidade [BV04].

### 7.3.5 Algoritmo de barreira logarítmica para MPC

O gradiente  $\mathcal{G}$  e a Hessiana  $\mathcal{H}$  da função custo de (7.13) são dadas por:

$$\mathcal{G} = \mathbf{H}\Delta\mathbf{u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/k_b)}{\bar{\mathbf{R}}_{li}\Delta\mathbf{u} - \bar{\mathbf{r}}_i} \bar{\mathbf{R}}_{li}^T$$

$$\mathcal{H} = \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/k_b)}{(\bar{\mathbf{R}}_{li}\Delta\mathbf{u} - \bar{\mathbf{r}}_i)^2} \bar{\mathbf{R}}_{li}^T \bar{\mathbf{R}}_{li}.$$

O algoritmo de Newton aplicado ao problema (7.13) pode ser visto no Algoritmo 6.

---

**Algoritmo 6:** Método de Newton

---

**Entrada:**  $\Delta\mathbf{u}^0$  estritamente factível,  $t$ ,  $\mathbf{H}$ ,  $\mathbf{b}$ ,  $\bar{\mathbf{R}}$ ,  $\bar{\mathbf{r}}$

**Saída:**  $\Delta\mathbf{u}$

**Dados:**  $L_{nt}$

1 **início**

2     **para**  $i = 1 : L_{nt}$  **faça**

3         Atualiza  $\mathcal{G} \leftarrow \mathbf{H}\Delta\mathbf{u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{\bar{\mathbf{R}}_{li}\Delta\mathbf{u} - \bar{\mathbf{r}}_i} \bar{\mathbf{R}}_{li}^T$  ;

4         Atualiza  $\mathcal{H} \leftarrow \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/t)}{(\bar{\mathbf{R}}_{li}\Delta\mathbf{u} - \bar{\mathbf{r}}_i)^2} \bar{\mathbf{R}}_{li}^T \bar{\mathbf{R}}_{li}$  ;

5         Computa o passo de Newton:  $\Delta\mathbf{x}_{nt} \leftarrow -\mathcal{H}^{-1}\mathcal{G}$  ;

6         Escolhe o tamanho do passo  $l$  por busca retroativa linear ;

7         Atualiza  $\Delta\mathbf{u} \leftarrow \Delta\mathbf{u} + l\Delta\mathbf{x}_{nt}$  ;

8     **fim**

9 **fim**

---

A partir dos métodos apresentados, pôde-se chegar ao algoritmo completo do método de barreira proposto para o problema de programação quadrática do MPC, que pode ser visto no Algoritmo 7.

As orientações para se escolher os parâmetros do método de barreira forma discutidas no Seção 7.3.4. O método de Newton, usado para computar o passo de centralização, foi limitado em  $L_{nt}$  iterações. O tamanho do passo de Newton foi computado utilizando-se o algoritmo de busca retroativa linear, em duas partes. Na primeira, pode-se perceber uma busca por um ponto factível a cada iteração. Na segunda parte, a busca pela minimização da função custo, com o parâmetro  $\varpi$  sendo a fração aceitável de decaimento da função custo predita por extrapolação linear. Em [BV04] afirma-se que os valores típicos de  $\varpi$  são escolhidos entre 0,01 e 0,3, significando que é

---

**Algoritmo 7:** Algoritmo do método de barreira para MPC
 

---

**Entrada:**  $\check{\mathbf{H}} = \mathbf{H}^{-1}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}}$   
**Saída:**  $\Delta \mathbf{u}(k)$   
**Dados:**  $k_b = k_{b0} > 0, \mu > 1, \epsilon > 0, \beta \in (0, 1), \varpi \in (0, 0,5), L_{gc}, L_{nt}$

```

1  início
2  enquanto  $n_{rin}/k_b > \epsilon$  faça
3       $l = 1$ ;
4      para cada  $i = 1 : L_{nt}$  faça
5           $\mathcal{G} \leftarrow \mathbf{H}\Delta \mathbf{u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/k_b)}{\bar{\mathbf{R}}_{li}\Delta \mathbf{u} - \bar{\mathbf{r}}_i} \bar{\mathbf{R}}_{li}^T$ ;
6           $\mathcal{H} \leftarrow \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/k_b)}{(\bar{\mathbf{R}}_{li}\Delta \mathbf{u} - \bar{\mathbf{r}}_i)^2} \bar{\mathbf{R}}_{li}^T \bar{\mathbf{R}}_{li}$ ;
7           $\Delta \mathbf{x}_{nt} \leftarrow -\mathcal{H}^{-1}\mathcal{G}$ ;
8          enquanto  $\bar{\mathbf{R}}(\Delta \mathbf{u} + l\Delta \mathbf{x}_{nt}) - \bar{\mathbf{r}} > 0$  faça
9               $l \leftarrow \beta l$ ;
10         fim
11         enquanto  $J(\Delta \mathbf{u} + l\Delta \mathbf{x}_{nt}) > J(\Delta \mathbf{u}) + l\varpi \mathcal{G}^T \Delta \mathbf{x}_{nt}$  faça
12              $l \leftarrow \beta l$ ;
13         fim
14          $\Delta \mathbf{u} \leftarrow \Delta \mathbf{u} + l\Delta \mathbf{x}_{nt}$ ;
15     fim
16      $k_b \leftarrow \mu k_b$ ;
17 fim
18 fim
  
```

---

aceitável um decaimento da função custo de 1% a 30% da predição baseada na extrapolação linear.

Ao se avaliar o Algoritmo 6, pode-se constatar que o cômputo mais custoso se refere ao passo de Newton, pois deve ser calculado a inversão da matriz Hessiana. Nos casos em que o algoritmo será implementado em hardwares com poucos recursos de processamento ou processos de dinâmicas mais rápidas, esse cálculo pode ser inconveniente. Dessa forma, pode-se reformular esse passo como o cômputo de um sistema linear do tipo  $\mathbf{Ax} = \mathbf{b}$ , que tipicamente requer  $O(N^3 n_{rin}^3)$  operações em um processador serial [WMN11]. Esse tipo de sistema poderia ser resolvido de forma eficiente por meio da decomposição das matrizes e exploração da estrutura da matriz  $\mathbf{A}$ , como a decomposição de Cholesky. Entretanto, a decomposição só é eficiente se a estrutura de  $\mathbf{A}$  for esparsa. Para o caso do GPC, a matriz tipicamente é densa, então não é trivial explorar a estrutura. Portanto, uma outra abordagem é a utilização de um método iterativo. Um método possível é o do gradiente conjugado [GVL96]. Essa abordagem foi utilizada no contexto de SSMPC e FPGA em [RLSCK09] e [WMN11]. Para entender como o método do gradiente conjugado pode ser utilizado para resolver o problema  $\mathbf{Ax} = \mathbf{b}$ , pode-se

considerar o problema de minimização:

$$\min_{\mathbf{x}} h_{gc}(x) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}, \quad (7.20)$$

com  $\mathbf{A} \in \mathbb{R}^{n \times n}$  simétrica positiva definida e  $\mathbf{b} \in \mathbb{R}^n$ . O mínimo de  $h_{gc}$  é  $-\mathbf{b}^T \mathbf{A}^{-1} \mathbf{b}/2$ , com  $x^* = \mathbf{A}^{-1} \mathbf{b}$ . Dessa forma, minimizar  $h_{gc}$  e resolver  $\mathbf{A} \mathbf{x} = \mathbf{b}$  são problemas equivalentes. O Algoritmo 8 apresenta o método do gradiente conjugado com busca exata de linha (em inglês, *exact line search*) aplicado ao cômputo do passo de Newton.

---

**Algoritmo 8:** Método do Gradiente Conjugado

---

**Entrada:**  $\Delta \mathbf{x}_{nt}^0$  estritamente factível,  $\mathcal{H}, \mathcal{G}$

**Saída:**  $\Delta \mathbf{x}_{nt}$

```

1 início
2   Calcula o resíduo inicial:  $\mathbf{r} \leftarrow \mathcal{G} - \mathcal{H} \Delta \mathbf{x}_{nt}^0$ ;
3    $i \leftarrow 0$ ;
4   enquanto  $\mathbf{r} \neq 0$  faça
5      $i \leftarrow i + 1$  ;
6     se  $i = 1$  então
7       Calcula a direção do passo:  $\mathbf{p} \leftarrow \mathbf{r}$ ;
8     fim
9     senão
10      Calcula a direção do passo:  $\mathbf{p} \leftarrow \mathbf{r} + \frac{\mathbf{r}^T \mathbf{r}}{\mathbf{r}^T \mathbf{r}} \mathbf{p}$ ;
11    fim
12    Calcula o tamanho do passo  $\alpha \leftarrow \mathbf{r}^T \mathbf{r} / (\mathbf{p}^T \mathcal{H} \mathbf{p})$ ;
13    Atualiza  $\Delta \mathbf{x}_{nt} \leftarrow \Delta \mathbf{x}_{nt} + \alpha \mathbf{p}$ ;
14    Guarda valor anterior:  $\mathbf{r}^- \leftarrow \mathbf{r}$ ;
15    Atualiza o resíduo:  $\mathbf{r} \leftarrow \mathbf{r} - \alpha \mathcal{H} \mathbf{p}$  ;
16  fim
17 fim
```

---

O algoritmo de barreira sem a necessidade do cálculo de inversão de matriz pode ser visto no Algoritmo 9.

## 7.4 Método de projeção de gradiente

O método de projeção de gradiente foi originalmente proposto por [Gol64] e [LP66]. A ideia principal do algoritmo é assumir o problema sem restrições, iterativamente dar passos na direção de descida da função e posteriormente fazer uma projeção da solução no conjunto de restrições.



---

**Algoritmo 9:** método de barreira e sem inversão online de matriz
 

---

**Entrada:**  $\check{\mathbf{H}} = \mathbf{H}^{-1}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}}$   
**Saída:**  $\Delta \mathbf{u}(k)$   
**Dados:**  $k_b = k_{b0} > 0, \mu > 1, \epsilon > 0, \beta \in (0, 1), \varpi \in (0, 0,5), L_{gc}, L_{nt}$

```

1 início
2   enquanto  $n_{rin}/k_b > \epsilon$  faça
3      $l = 1$ ;
4     para cada  $i = 1 : L_{nt}$  faça
5        $\mathcal{G} \leftarrow \mathbf{H}\Delta \mathbf{u} + \mathbf{b} + \sum_{i=0}^{n_{rin}} \frac{(1/k_b)}{\bar{\mathbf{R}}_{li}\Delta \mathbf{u} - \bar{\mathbf{r}}_i} \bar{\mathbf{R}}_{li}^T$ ;
6        $\mathcal{H} \leftarrow \mathbf{H} + \sum_{i=0}^{n_{rin}} \frac{(1/k_b)}{(\bar{\mathbf{R}}_{li}\Delta \mathbf{u} - \bar{\mathbf{r}}_i)^2} \bar{\mathbf{R}}_{li}^T \bar{\mathbf{R}}_{li}$ ;
7        $\mathbf{r} \leftarrow -\mathcal{G} - \mathcal{H}\Delta \mathbf{x}_{nt}$ ;
8       para cada  $j = 1 : L_{gc}$  faça
9         se  $j = 1$  então
10            $\mathbf{p} \leftarrow \mathbf{r}$ ;
11         fim
12         senão
13            $\mathbf{p} \leftarrow \mathbf{r} + \frac{\mathbf{r}^T \mathbf{r}}{\mathbf{r}^T \mathbf{r}} \mathbf{p}$ ;
14         fim
15          $\alpha \leftarrow \mathbf{r}^T \mathbf{r} / \mathbf{p}^T \mathcal{H} \mathbf{p}$ ;
16          $\Delta \mathbf{x}_{nt} \leftarrow \Delta \mathbf{x}_{nt} + \alpha \mathbf{p}$ ;
17          $\mathbf{r}^- \leftarrow \mathbf{r}$ ;
18          $\mathbf{r} \leftarrow \mathbf{r} - \alpha \mathcal{H} \mathbf{p}$ ;
19       fim
20       enquanto  $\bar{\mathbf{R}}(\Delta \mathbf{u} + l\Delta \mathbf{x}_{nt}) - \bar{\mathbf{r}} > 0$  faça
21          $l \leftarrow \beta l$ ;
22       fim
23       enquanto  $J(\Delta \mathbf{u} + l\Delta \mathbf{x}_{nt}) > J(\Delta \mathbf{u}) + l\varpi \mathcal{G}^T \Delta \mathbf{x}_{nt}$  faça
24          $l \leftarrow \beta l$ ;
25       fim
26        $\Delta \mathbf{u} \leftarrow \Delta \mathbf{u} + l\Delta \mathbf{x}_{nt}$ ;
27     fim
28      $k_b \leftarrow \mu k_b$ ;
29   fim
30 fim
  
```

---

A partir do problema de programação quadrática do MPC 7.1 e assumindo primeiramente o caso sem restrições, podemos verificar qual a direção de maior descida da função objetivo. Dessa forma, podemos fazer uma aproximação linear da função objetivo, para um dado  $\Delta \mathbf{u}_i$  e um passo  $\delta_i$  a partir de uma série de Taylor:

$$J(\Delta \mathbf{u}_i + \delta) \approx J(\Delta \mathbf{u}_i) + \nabla J(\Delta \mathbf{u}_i) \cdot (\delta), \quad (7.21)$$

ou seja,

$$J(\Delta \mathbf{u}_i + \delta) - J(\Delta \mathbf{u}_i) \approx \langle \nabla J(\Delta \mathbf{u}_i) | (\delta) \rangle, \quad (7.22)$$

onde  $\langle \cdot | \cdot \rangle$  denota o produto escalar. Como o produto escalar é definido por:

$$\langle \mathbf{a} | \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\phi), \quad (7.23)$$

onde  $\phi$  é o ângulo formado entre os vetores  $\mathbf{a}$  e  $\mathbf{b}$ . Portanto, podemos verificar que o maior aumento na função objetivo ocorre quando o ângulo  $\phi$  é igual a zero, ou seja, o passo  $\delta$  é dado no direção do gradiente  $\nabla J(\Delta \mathbf{u}_i)$ . Como o objetivo é diminuir a função objetivo a cada passo, a direção de maior descida ocorre quando o ângulo  $\phi$  é igual a  $180^\circ$ , ou seja, o passo  $\delta$  é dado no direção do gradiente  $\nabla J(\Delta \mathbf{u}_i)$  e no sentido negativo  $\delta = -l_j \nabla J(\Delta \mathbf{u}_j)$ . Portanto, o passo do método do gradiente para o caso sem restrições é dado por:

$$\Delta \mathbf{u}_{j+1} = \Delta \mathbf{u}_j - l_j \nabla J(\Delta \mathbf{u}_j), \quad (7.24)$$

onde  $l_j > 0$  é o tamanho do passo e deve garantir que o algoritmo iterativo forme uma chamada sequência de relaxação tal que:

$$J(\Delta \mathbf{u}_{j+1}) \leq J(\Delta \mathbf{u}_j), \quad \forall j \geq 0. \quad (7.25)$$

Várias regras para a definição do comprimento do passo  $l_j$  podem ser encontrados na literatura, como o comprimento de passo fixo, a regra de Armijo e a busca linear retroativa [TN20]. A regra mais simples é a de passo fixo  $l_j = l > 0$ ,  $\forall j \geq 1$  e é geralmente utilizada, uma vez que a função  $J$  é convexa e foi provado em [Nes14] ter a mesma taxa de convergência das outras regras para este caso específico. Para se definir um tamanho de passo que garanta a convergência do método do gradiente para o MPC, uma das condições é que o gradiente da função seja Lipschitz contínuo em um conjunto  $\Omega$  (Definição 4), ou seja, que exista uma constante Lipschitz  $L \geq 0$  tal que [RJM09]:

$$\|\nabla J(\Delta \mathbf{u}_{j+1}) - \nabla J(\Delta \mathbf{u}_j)\| \leq L \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\|, \quad \forall \Delta \mathbf{u}_{j+1}, \Delta \mathbf{u}_j \in \Omega. \quad (7.26)$$

A partir da equação (7.26) pode-se definir um limite superior quadrático para  $J$  dado por [RJM09]:

$$J(\Delta \mathbf{u}_{j+1}) \leq J(\Delta \mathbf{u}_j) + \nabla J(\Delta \mathbf{u}_j)^T (\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j) + \frac{L}{2} \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\|^2, \quad \forall \Delta \mathbf{u}, \Delta \mathbf{u}_j \in \Omega. \quad (7.27)$$

Agora, se assumirmos que o próximo valor do incremento de controle,  $\Delta \mathbf{u}_{j+1}$  será dado pelo passo principal do método de gradiente apresentado em (7.24), e substituirmos em (7.27) obtemos:

$$\begin{aligned}
J(\Delta \mathbf{u}_{j+1}) &\leq J(\Delta \mathbf{u}_j) + \nabla J(\Delta \mathbf{u}_j)^T (\Delta \mathbf{u}_j - l_j \nabla J(\Delta \mathbf{u}_j) - \Delta \mathbf{u}_j) \\
&\quad + \frac{L}{2} \|\Delta \mathbf{u}_j - l_j \nabla J(\Delta \mathbf{u}_j) - \Delta \mathbf{u}_j\|^2, \\
J(\Delta \mathbf{u}_{j+1}) &\leq J(\Delta \mathbf{u}_j) + \nabla J(\Delta \mathbf{u}_j)^T (-l_j \nabla J(\Delta \mathbf{u}_j)) + \frac{L}{2} \|-l_j \nabla J(\Delta \mathbf{u}_j)\|^2 \\
J(\Delta \mathbf{u}_{j+1}) &\leq J(\Delta \mathbf{u}_j) - l \nabla J(\Delta \mathbf{u}_j)^T \nabla J(\Delta \mathbf{u}_j) + \frac{Ll^2}{2} \|\nabla J(\Delta \mathbf{u}_j)\|^2 \\
J(\Delta \mathbf{u}_{j+1}) &\leq J(\Delta \mathbf{u}_j) - l(1 - \frac{Ll}{2}) \|\nabla J(\Delta \mathbf{u}_j)\|^2 \tag{7.28}
\end{aligned}$$

Pode-se observar que se o tamanho da passo for definido com  $0 < l < \frac{2}{L}$ , que são as raízes, de  $l - \frac{Ll^2}{2}$  podemos garantir a sequência de relaxação  $J(\Delta \mathbf{u}_{j+1}) \leq J(\Delta \mathbf{u}_j)$ . De fato, podemos derivar  $l - \frac{Ll^2}{2}$  em relação a  $l$  e igualar a zero para obter o maior valor na diminuição da função objetivo. Dessa forma, obtemos que o maior valor ocorre com  $l = \frac{1}{L}$  e portanto a iteração principal do método de gradiente com passo fixo é dado por:

$$\Delta \mathbf{u}_{j+1} = \Delta \mathbf{u}_j - \frac{1}{L} \nabla J(\Delta \mathbf{u}_j). \tag{7.29}$$

Para encontrar o valor da constante Lipschitz  $L$  para o QP (7.1), podemos retomar a definição de Lipschitz continuidade da derivada (7.26), e substituir para a função custo:

$$\begin{aligned}
\|\mathbf{H}\Delta \mathbf{u}_{j+1} + \mathbf{b} - (\mathbf{H}\Delta \mathbf{u}_j + \mathbf{b})\| &\leq L \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\| \\
\|\mathbf{H}\Delta \mathbf{u}_{j+1} - \mathbf{H}\Delta \mathbf{u}_j\| &\leq L \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\| \\
\|\mathbf{H}(\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j)\| &\leq L \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\| \\
\|\mathbf{H}\| \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\| &\leq L \|\Delta \mathbf{u}_{j+1} - \Delta \mathbf{u}_j\| \\
\|\mathbf{H}\| &\leq L \tag{7.30}
\end{aligned}$$

Portanto, podemos definir  $L$  com o mesmo valor da norma espectral da Hessiana que nesse caso seria  $L = \|\mathbf{H}\|$ . A norma espectral pode ser calculada como:

$$L = \|\mathbf{H}\| = \sqrt{\lambda_{max}(\mathbf{H}^T \mathbf{H})}. \tag{7.31}$$

Alguns autores também utilizam a norma Frobenius para a definição de  $L$ , uma vez que,

$$\|\mathbf{H}\|_F \geq \|\mathbf{H}\|. \quad (7.32)$$

A norma Frobenius da Hessiana pode ser obtida por:

$$L = \|\mathbf{H}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |h_{ij}|^2}, \quad (7.33)$$

onde  $n$  é a dimensão da matriz Hessiana e  $h_{ij}$  é o elemento da matriz correspondente à linha  $i$  e coluna  $j$ .

### 7.4.1 Projeção ortogonal

Agora vamos retomar o caso com restrições. Após calcular o passo na direção de descida da função objetivo, o método prevê uma projeção ortogonal dos pontos obtidos no conjunto formado pelas restrições. Dessa forma, o passo principal do método de projeção de gradiente é dado por:

$$\Delta \mathbf{u}_{j+1} = P(\Delta \mathbf{u}_j - \frac{1}{L} \nabla J(\Delta \mathbf{u}_j)),$$

onde  $P$  é a função projeção ortogonal (Definição 14). A ideia básica da projeção é encontrar o vetor de soluções factível (dentro do conjunto das restrições) que tenha a menor distância Euclidiana ao vetor calculado. Entretanto, o cômputo da projeção do vetor no conjunto  $\Omega$  pode ser complicado para alguns tipos de restrições do problema de otimização. Uma solução é a abordagem através da propriedade da dualidade. A partir dessa propriedade, o ótimo primal (problema original) pode ser encontrado através da solução do problema dual, que é dado por:

$$\begin{aligned} \min_{\psi} \quad & -J_d(\psi) \\ \text{s.a.} \quad & \psi \geq \mathbf{0}, \end{aligned} \quad (7.34)$$

com  $J_d : \mathbb{R}^n \rightarrow \mathbb{R}$  sendo a função dual associada a  $J$  (Definição 10) e  $\psi \in \mathbb{R}^n$  o vetor de multiplicadores de Lagrange associados às restrições de desigualdade. Dessa forma, uma vez obtidos os multiplicadores de Lagrange ótimos  $\psi^*$ , assumindo que a dualidade forte se aplica (Definição 12), o ótimo primal pode ser obtido por:

$$\Delta \mathbf{u}^* = \operatorname{argmin}(\mathcal{L}(\Delta \mathbf{u}, \psi^*)). \quad (7.35)$$

Dessa forma, pode-se perceber que as restrições associadas a (7.34) são apenas  $\psi > \mathbf{0}$  e portanto a projecção é a menor distância de  $\psi$  para valores positivos, o que pode ser representado matematicamente como:

$$P(\psi) = \max(\psi, \mathbf{0}). \quad (7.36)$$

Trazendo as definições para o QP (7.1), o primeiro passo para se obter a forma dual é definir o Lagrangiano associado:

$$\mathcal{L}(\Delta \mathbf{u}, \psi) = \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u} + \psi^T (\mathbf{R} \Delta \mathbf{u} - \mathbf{r}), \quad (7.37)$$

onde  $\psi \in \mathbb{R}^n$  é o vetor de multiplicadores de Lagrange associados às restrições de desigualdade.

O problema de otimização dual, apresentado como uma minimização, pode ser dado como:

$$\begin{aligned} \min_{\psi} \quad & -J_d(\psi) = \frac{1}{2} \psi^T \mathbf{R} \mathbf{H}^{-1} \mathbf{R}^T \psi + (\mathbf{R} \mathbf{H}^{-1} \mathbf{b} + \mathbf{r})^T \psi + \frac{1}{2} \mathbf{b}^T \mathbf{H}^{-1} \mathbf{b} \\ \text{s.a.} \quad & \psi \geq \mathbf{0}, \end{aligned} \quad (7.38)$$

o qual é um QP com dualidade forte, a partir da condição de Slater (Definição 13), uma vez que há um ponto estritamente factível no problema primal [BV04, Sec. 5.2.3].

Portanto, o problema em (??) pode ser reescrito como:

$$\begin{aligned} \psi_{j+1} &= P \left( \psi_j - \frac{1}{L} \nabla J_d(\psi_j) \right) \\ \psi_{j+1} &= \max \left( \psi_j - \frac{1}{L} (\mathbf{R} \mathbf{H}^{-1} \mathbf{R}^T \psi_j + \mathbf{R} \mathbf{H}^{-1} \mathbf{b} + \mathbf{r}), \mathbf{0} \right), \end{aligned} \quad (7.39)$$

o qual pode ser dividido em dois passos, para que o cálculo de  $\Delta \mathbf{u}$  fique explícito a cada iteração:

$$\begin{aligned} \Delta \mathbf{u}_j &= -\mathbf{H}^{-1} (\mathbf{b} + \mathbf{R}^T \psi_j), \\ \psi_{j+1} &= \max \left( \psi_j + \frac{1}{L} (\mathbf{R} \Delta \mathbf{u}_j + \mathbf{r}), \mathbf{0} \right). \end{aligned} \quad (7.40)$$

A partir das definições apresentadas, um algoritmo do método de projecção pode ser visto no Algoritmo 10.

---

**Algoritmo 10:** Método de gradiente projetado dual

---

**Entrada:**  $J_d$ ,  $\psi_0 = \psi_{-1} = \mathbf{0}$ ,  $L$ , tolerância  $\epsilon > 0$

**Saída:**  $\Delta \mathbf{u}_k$

```
1  $j \leftarrow 0$ ;  
2 enquanto  $|\psi_j - \psi_{j-1}| > \epsilon$  faça  
3    $\Delta \mathbf{u}_j = -\mathbf{H}^{-1} (\mathbf{b} + \mathbf{R}^T \psi_{j-1})$ ;  
4    $\psi_j \leftarrow \max(\psi_{j-1} - \frac{1}{L} \nabla J_d(\psi_{j-1}), \mathbf{0})$ ;  
5    $j \leftarrow j + 1$  ;  
6 fim
```

---

### 7.4.2 Aceleração do método de projeção de gradiente

Em geral, o método de projeção de gradiente pode precisar de muitas iterações para convergir em casos de tolerâncias muito pequenas. Dessa forma, uma versão acelerada do método de projeção de gradiente foi proposta por [Nes83] e vem sendo amplamente utilizada no contexto de MPC de cômputo rápido. O ponto principal para a aceleração do método é a não observância da propriedade da sequência de relaxação definida em (7.25) e a sua substituição pela chamada sequência estimada. Dessa forma, é inserida uma operação *a priori* no cálculo de  $\Delta \mathbf{u}$ :

$$\mathbf{w}_j = \Delta \mathbf{u}_j + \theta_j(\theta_{j-1}^{-1} - 1)(\Delta \mathbf{u}_j - \Delta \mathbf{u}_{j-1}), \quad (7.41)$$

$$\Delta \mathbf{u}_{j+1} = P(\mathbf{w}_j - \frac{1}{L} \nabla J_d(\mathbf{w}_j)), \quad (7.42)$$

com  $\theta_{j+1} = (\sqrt{(\theta_j)^4 + 4(\theta_j)^2} - (\theta_j)^2)/2$ .

O cômputo de  $\theta$  apresenta operações algébricas mais custosas computacionalmente, como as potências e a raiz quadrada, e essas são operações indesejadas para sistemas embarcados. Dessa forma, como foi demonstrada a convergência do método para quaisquer escolhas de  $\theta$  tal que:

$$\frac{1 - \theta_{j+1}}{(\theta_{j+1})^2} \leq \frac{1}{(\theta_j)^2}, \quad (7.43)$$

uma escolha possível que atende (7.43) é  $\theta_j = 2/(j+2)$  [Tse08]. Substituindo  $\theta_j = 2/(j+2)$  em (7.41) obtém-se uma equação mais simples:

$$\theta_j(\theta_{j-1}^{-1} - 1) = \frac{j-1}{j+2},$$

e se definirmos uma variável  $\beta = \frac{j-1}{j+2}$ , a equação (7.41) torna-se:

$$\mathbf{w}_j = \Delta \mathbf{u}_j + \beta(\Delta \mathbf{u}_j - \Delta \mathbf{u}_{j-1}). \quad (7.44)$$

A versão do método de gradiente acelerado dual é chamada de GPAD.

### 7.4.3 Algoritmo GPAD para MPC

Como critério de parada do algoritmo, a factibilidade e a otimalidade primal podem ser utilizadas. Para a factibilidade primal, é possível testar se as restrições são atendidas dentro da tolerância  $\epsilon_f$ , ou seja:

$$\mathbf{R}\Delta\mathbf{u}_j - \mathbf{r} \leq \epsilon_f. \quad (7.45)$$

Para o critério de otimalidade primal, uma propriedade derivada da dualidade forte pode ser utilizada, o que implica:

$$J(\Delta\mathbf{u}) - J(\Delta\mathbf{u}^*) \leq J(\Delta\mathbf{u}) - J_d(\psi). \quad (7.46)$$

Portanto, uma solução é considerada  $\epsilon_o$ -ótima se:

$$J(\Delta\mathbf{u}) - J_d(\psi) \leq \epsilon_o, \quad (7.47)$$

e, então, substituindo a função  $J(\Delta\mathbf{u})$ , definida em (7.1), e a função  $J_d(\psi)$ , definida em (7.38), a solução será  $\epsilon_o$ -ótima se:

$$-\psi^T(\mathbf{R}\Delta\mathbf{u}_j - \mathbf{r}) \leq \epsilon_o. \quad (7.48)$$

A partir dos desenvolvimentos apresentados e do algoritmo recursivo para a obtenção das matrizes do GPC detalhadas em [PLFNR20], o GPCGPAD pode ser resumido no Algoritmo 11, onde  $j_{max}$  é o número máximo de iterações e  $\bar{u}_i$  é um sinal de degrau unitário com  $\bar{u}_i = 1$  para  $i \geq 0$  e  $\bar{u}_i = 0$  para  $i < 0$ .

## 7.5 Métodos de operador de partição

Os métodos de operador de partição baseiam-se na ideia de que, a partir de uma função objetivo separável, o problema de otimização pode ser separado em problemas mais simples. Dessa forma, esses problemas simples podem ser resolvidos paralelamente. Para resolver cada um dos problemas reduzidos, existem técnicas que podem utilizar a formulação primal, dual ou até primal-dual. Os métodos que se destacam são o Método dos Multiplicadores com Direções Alternadas (ADMM, do inglês *Alternating Direction Method of Multipliers*), o Algoritmo de Minimização Alternada (AMA, do inglês *Alternating Minimization Algorithm*) e o Algoritmo Primal-Dual (PDA, do inglês *Primal-Dual Algorithm*) [SSS<sup>+</sup>16].

---

**Algoritmo 11:** Algoritmo GPAD para MPC

---

**Entrada:**  $\mathbf{H}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}}$ **Saída:**  $\Delta \mathbf{u}(k)$ **Dados:**  $\psi_0 = \psi_{-1} = \mathbf{0}$ ,  $\epsilon_f$ ,  $\epsilon_o$ ,  $j_{max}$ 

```
1 início
2   enquanto  $j < j_{max}$  faça
3       se  $j = 0$  então
4            $\beta \leftarrow 0$ ;
5       fim
6       senão
7            $\beta \leftarrow \frac{j-1}{j+2}$ ;
8       fim
9        $\mathbf{a} \leftarrow \psi_j + \beta(\psi_j - \psi_{j-1})$ ;
10       $\Delta \mathbf{u}_m \leftarrow -\bar{\mathbf{H}}(\bar{\mathbf{R}}^T \mathbf{a} + \mathbf{b})$ ;
11       $\mathbf{s} \leftarrow (1/L)(\bar{\mathbf{R}} \Delta \mathbf{u}_m - \bar{\mathbf{r}})$ ;
12      se  $\mathbf{s}_i \leq \epsilon_f/L$ ,  $\forall i = 1, \dots, n_{rin}$  então
13          se  $-\mathbf{a}^T \mathbf{s} < \epsilon_o/L$  então
14              retorna;
15          fim
16      fim
17       $\psi_{j+1} \leftarrow \max(\mathbf{a} + \mathbf{s}, \mathbf{0})$ ;
18       $j \leftarrow j + 1$ ;
19 fim
20 fim
```

---

O ADMM é um dos métodos mais populares e trata-se de um algoritmo simples, mas que se adapta muito bem a problemas de otimização convexa distribuída. Ele se apresenta na forma de um algoritmo de decomposição/-coordenação, no qual as soluções de pequenos subproblemas locais são coordenadas para encontrar uma solução para um problema global de grande escala [OSB13]. O ADMM pode ser interpretado como uma tentativa de colher ambos os benefícios da decomposição dual e dos métodos de Lagrangiano aumentado para otimização com restrições [BPC<sup>+</sup>11].

Devido à possibilidade de paralelização dos métodos de partição, o ADMM se apresenta como uma boa abordagem para as soluções modernas de hardware, como os processadores de múltiplos núcleos e os FPGAs. Em [OSB13] é apresentada uma forma de implementação de MPC com ADMM denominada *operator splitting for control* (OSC). Essa formulação, em muitos casos, pode não necessitar de operações de divisão, o que facilita a implementação



em arquiteturas de aritmética de ponto fixo, como FPGAs. Os pontos fracos do ADMM são a dependência do tipo do problema no desempenho do método e a baixa precisão nos resultados. Em [RC14] é proposto um mecanismo de detecção de infactibilidade e o seu desempenho é exemplificado em uma aplicação de MPC.

A seguir são apresentadas a formulação e alguns conceitos importantes relacionados ao ADMM e suas variações úteis para o MPC.

### 7.5.1 Precusores

Pode-se afirmar que os precusores do ADMM são os métodos conhecidos como o método ascendente dual e o método dos multiplicadores [BPC<sup>+</sup>11]. O método ascendente dual assume um problema de otimização convexa com restrições de igualdade da seguinte forma:

$$\begin{aligned} \min_{\Delta \mathbf{u}} \quad & J(\Delta \mathbf{u}) \\ \text{s.a.} \quad & \bar{\mathbf{M}}\Delta \mathbf{u} = \bar{\mathbf{m}} \end{aligned} \tag{7.49}$$

com  $J : \mathbb{R}^n \rightarrow \mathbb{R}$  convexa,  $\Delta \mathbf{u} \in \mathbb{R}^n$ ,  $\bar{\mathbf{M}} \in \mathbb{R}^{n_{req} \times n}$ ,  $\bar{\mathbf{m}} \in \mathbb{R}^{n_{req}}$ .

A partir de (7.49), o problema dual pode ser obtido através do Lagrangiano e da função dual  $J_d$  (Definição 10) associados. O Lagrangiano é dado por:

$$\mathcal{L}(\Delta \mathbf{u}, \nu) = J(\Delta \mathbf{u}) + \nu^T (\bar{\mathbf{M}}\Delta \mathbf{u} - \bar{\mathbf{m}}),$$

com  $\nu \in \mathbb{R}^{n_{req}}$  sendo o vetor de multiplicadores de Lagrange associados às restrições de igualdade.

O problema dual recai em:

$$\begin{aligned} \max_{\nu} \quad & J_d(\nu) \\ \text{s.a.} \quad & \nu \geq \mathbf{0}. \end{aligned} \tag{7.50}$$

Dessa forma, uma vez obtidos os multiplicadores de Lagrange ótimos  $\nu^*$ , assumindo que a dualidade forte (Definição 12) se aplica, o ótimo primal pode ser obtido a partir do ótimo dual:

$$\Delta \mathbf{u}^* = \operatorname{argmin}(\mathcal{L}(\Delta \mathbf{u}, \nu^*)). \tag{7.51}$$

O método ascendente dual consiste em iterativamente computar a variável primal e atualizar a variável dual na direção do gradiente em relação a  $\nu$  de  $J_d$  com um passo de comprimento  $l$ , ou seja:

$$\begin{aligned} \Delta \mathbf{u}_{j+1} &= \operatorname{argmin}(\mathcal{L}(\Delta \mathbf{u}, \nu_j)) \\ \nu_{j+1} &= \nu_j + l \nabla_{\nu} J_d = \nu_j + l(\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} - \bar{\mathbf{m}}). \end{aligned} \tag{7.52}$$

Como o método ascendente dual necessita de algumas propriedades para a convergência, como a finitude e a convexidade estrita (Definição 3) da função  $J$ , o método de multiplicadores foi desenvolvido para contornar esses requisitos.

O método dos multiplicadores utiliza o Lagrangiano aumentado que para (7.49) é dado por:

$$\mathcal{L}_\rho(\Delta \mathbf{u}, \nu) = J(\Delta \mathbf{u}) + \nu^T(\bar{\mathbf{M}}\Delta \mathbf{u} - \bar{\mathbf{m}}) + \frac{\rho}{2}\|\bar{\mathbf{M}}\Delta \mathbf{u} - \bar{\mathbf{m}}\|^2,$$

onde  $\rho > 0$  é chamado de parâmetro de penalização. A vantagem de incluir esse termo com penalização no Lagrangiano é que a função dual obtida é diferenciável sob condições menos restritas do que na formulação original. O gradiente do Lagrangiano aumentado pode ser obtido da mesma forma que o Lagrangiano normal e, dessa forma, o método dos multiplicadores se apresenta da seguinte forma:

$$\begin{aligned}\Delta \mathbf{u}_{j+1} &= \operatorname{argmin}(\mathcal{L}_\rho(\Delta \mathbf{u}, \nu_j)) \\ \nu_{j+1} &= \nu_j + \rho \nabla_\nu J_d = \nu_j + \rho(\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} - \bar{\mathbf{m}}).\end{aligned}\tag{7.53}$$

Pode-se perceber que o método dos multiplicadores segue o mesmo padrão do ascendente dual, exceto que o passo de minimização de  $\Delta \mathbf{u}$  usa o Lagrangiano aumentado e que o parâmetro de penalização  $\rho$  é usado como o comprimento do passo  $l$ . O método dos multiplicadores converge sob condições mais gerais que o ascendente dual, incluindo casos nos quais  $h$  tende a infinito ou não é estritamente convexa. A escolha de  $\rho$  como comprimento do passo na direção do gradiente pode ser justificada como uma boa escolha para manter pontos factíveis, tanto primal quanto dual. Essa característica pode ser vista se avaliadas as condições de otimalidade Karush-Kuhn-Tucker (KKT) (Teorema B.0.2) de (7.49). Supondo  $h$  convexa, as condições de otimalidade primal e dual são [SSS<sup>+</sup>16]:

$$\bar{\mathbf{M}}\Delta \mathbf{u}^* - \bar{\mathbf{m}} = \mathbf{0}, \quad \nabla J(\Delta \mathbf{u}^*) + \bar{\mathbf{M}}^T \nu^* = \mathbf{0},$$

respectivamente. Como, por definição,  $\Delta \mathbf{u}_{j+1}$  minimiza o Lagrangiano aumentado  $\mathcal{L}_\rho(\Delta \mathbf{u}, \nu_j)$ , então

$$\begin{aligned}\mathbf{0} &= \nabla_u \mathcal{L}_\rho(\Delta \mathbf{u}_{j+1}, \nu_j) \\ &= \nabla_u J(\Delta \mathbf{u}_{j+1}) + \bar{\mathbf{M}}^T(\nu_j + \rho(\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} - \bar{\mathbf{m}})) \\ &= \nabla_u J(\Delta \mathbf{u}_{j+1}) + \bar{\mathbf{M}}^T \nu_{j+1}.\end{aligned}\tag{7.54}$$

Pode-se perceber a partir de (7.54) que a escolha de  $\rho$  como o comprimento de passo, na atualização da variável dual do método, fornece um par  $\{\Delta \mathbf{u}_{j+1}, \nu_{j+1}\}$  factível dual.

A melhora nas propriedades de convergência do método dos multiplicadores em relação ao ascendente dual implica um ônus na possibilidade de decomposição do mesmo. Quando  $J$  é separável, o Lagrangiano aumentado não é separável e portanto o passo de minimização de  $\Delta \mathbf{u}$  não pode ser computado de forma paralela, como ocorre no método ascendente dual [SSS<sup>+</sup>16].

### 7.5.2 Formulação ADMM

O ADMM foi desenvolvido de modo a reunir as propriedades de decomposição do método ascendente dual com as boas propriedades de convergência do método dos multiplicadores. Para isso, a variável  $\Delta \mathbf{u}$  é particionada em duas novas variáveis, o que torna a função objetivo separável. Para entender o ADMM, pode-se assumir que método resolve problemas da seguinte forma:

$$\begin{aligned} \min_{\Delta \mathbf{u}, \mathbf{z}} \quad & J(\Delta \mathbf{u}) + \zeta(\mathbf{z}) \\ \text{s.a.} \quad & \bar{\mathbf{M}}\Delta \mathbf{u} + \bar{\mathbf{D}}\mathbf{z} = \bar{\mathbf{m}} \end{aligned} \quad (7.55)$$

com  $\Delta \mathbf{u} \in \mathbb{R}^n$ ,  $\mathbf{z} \in \mathbb{R}^m$ ,  $\bar{\mathbf{M}} \in \mathbb{R}^{n_{req} \times n}$ ,  $\bar{\mathbf{D}} \in \mathbb{R}^{n_{req} \times m}$ ,  $\bar{\mathbf{m}} \in \mathbb{R}^{n_{req}}$  e  $J$  e  $\zeta$  sendo funções convexas.

Para incluir a restrição na função objetivo, a partir de (7.55), pode ser formado o Lagrangiano aumentado como no método dos multiplicadores:

$$\mathcal{L}_\rho(\Delta \mathbf{u}, \mathbf{z}, \nu) = J(\Delta \mathbf{u}) + \zeta(\mathbf{z}) + \nu^T(\bar{\mathbf{M}}\Delta \mathbf{u} + \bar{\mathbf{D}}\mathbf{z} - \bar{\mathbf{m}}) + \frac{\rho}{2}\|\bar{\mathbf{M}}\Delta \mathbf{u} + \bar{\mathbf{D}}\mathbf{z} - \bar{\mathbf{m}}\|^2.$$

Assumindo que a dualidade forte se aplica, os valores ótimos para o problema primal e dual são os mesmos. Dessa forma, o ótimo do problema primal  $\Delta \mathbf{u}^*$  pode ser obtido através do ponto ótimo dual  $\nu^*$ :

$$\Delta \mathbf{u}^* = \operatorname{argmin} \mathcal{L}_\rho(\Delta \mathbf{u}, \nu^*)$$

Portanto o ADMM pode ser computado de forma recursiva, alternando entre as minimizações de  $\Delta \mathbf{u}$  e  $\mathbf{z}$  e atualizando a variável  $\nu$  da seguinte forma:

$$\Delta \mathbf{u}_{j+1} = \operatorname{argmin} \mathcal{L}_\rho(\Delta \mathbf{u}_j, \mathbf{z}_j, \nu_j) \quad (7.56)$$

$$\mathbf{z}_{j+1} = \operatorname{argmin} \mathcal{L}_\rho(\Delta \mathbf{u}_{j+1}, \mathbf{z}_j, \nu_j) \quad (7.57)$$

$$\nu_{j+1} = \nu_j + \rho(\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_{j+1} - \bar{\mathbf{m}}). \quad (7.58)$$

### 7.5.3 Sintonização

O comprimento do passo no ADMM coincide com o parâmetro de penalização  $\rho$  do Lagrangiano aumentado e pode ser utilizado como um grau de

liberdade para uma convergência mais rápida do algoritmo. Na forma padrão do ADMM, o valor de  $\rho$  é mantido constante, entretanto, existem variações do ADMM que propõem diferentes valores de  $\rho_j$  a cada iteração [BPC<sup>+</sup>11]. As equações de atualização do ADMM sugerem que grandes valores de  $\rho$  causam uma grande penalização nas violações da factibilidade primal e tendem a produzir pequenos resíduos primais. Por outro lado, a definição do resíduo dual sugere que pequenos valores de  $\rho$  tendem a diminuir o resíduo dual, ao custo de reduzirem a penalização da factibilidade primal, o que pode resultar em um maior resíduo primal. Dessa forma, o valor de  $\rho$  deve ser escolhido de forma a reduzir o número de iterações necessárias para o ponto  $\epsilon$ -ótimo.

#### 7.5.4 Condições de otimalidade

Assumindo  $J$  e  $\zeta$  diferenciáveis, as condições de otimalidade necessárias e suficientes para o ADMM são a factibilidade primal,

$$\bar{\mathbf{M}}\Delta\mathbf{u}^* + \bar{\mathbf{D}}\mathbf{z}^* - \bar{\mathbf{m}} = \mathbf{0}, \quad (7.59)$$

e a factibilidade dual,

$$\begin{aligned} \mathbf{0} &= \nabla J(\Delta\mathbf{u}^*) + \bar{\mathbf{M}}^T \nu^*, \\ \mathbf{0} &= \nabla \zeta(\mathbf{z}^*) + \bar{\mathbf{D}}^T \nu^*. \end{aligned} \quad (7.60)$$

Uma vez que, por definição,  $\Delta\mathbf{u}_{j+1}$  minimiza o Lagrangiano  $\mathcal{L}_\rho(\Delta\mathbf{u}_j, \mathbf{z}_j, \psi_j)$  tem-se, a partir de (7.60), que:

$$\begin{aligned} \mathbf{0} &= \nabla J(\Delta\mathbf{u}_{j+1}) + \bar{\mathbf{M}}^T(\nu_j + \rho(\bar{\mathbf{M}}\Delta\mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_j - \bar{\mathbf{m}})) \\ &= \nabla J(\Delta\mathbf{u}_{j+1}) + \bar{\mathbf{M}}^T(\psi_j + \rho(\bar{\mathbf{M}}\Delta\mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_{j+1} - \bar{\mathbf{m}} - \bar{\mathbf{D}}\mathbf{z}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_j)) \\ &= \nabla J(\Delta\mathbf{u}_{j+1}) + \bar{\mathbf{M}}^T \nu_{j+1} + \rho \bar{\mathbf{M}}^T \bar{\mathbf{D}}(\mathbf{z}_j - \mathbf{z}_{j+1}) \\ \rho \bar{\mathbf{M}}^T \bar{\mathbf{D}}(\mathbf{z}_{j+1} - \mathbf{z}_j) &= \nabla J(\Delta\mathbf{u}_{j+1}) + \bar{\mathbf{M}}^T \nu_{j+1}, \end{aligned}$$

e, portanto, a parcela

$$\mathbf{s}_{j+1} = \rho \bar{\mathbf{M}}^T \bar{\mathbf{D}}(\mathbf{z}_{j+1} - \mathbf{z}_j)$$

pode ser considerada o resíduo dual na iteração  $j + 1$ .

A partir de (7.59), tem-se que o resíduo primal é dado por:

$$\mathbf{r}_{j+1} = \bar{\mathbf{M}}\Delta\mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_{j+1} - \bar{\mathbf{m}}.$$

Dessa forma, como um critério de parada frequentemente utilizado em ADMM, é utilizada a comparação dos resíduos com as tolerâncias primal  $\epsilon_{pri}$  e dual  $\epsilon_{dual}$ , de acordo com algum critério de otimalidade desejado, ou seja:

$$\|\mathbf{r}_j\| \leq \epsilon_{pri} \quad \text{e} \quad \|\mathbf{s}_j\| \leq \epsilon_{dual}. \quad (7.61)$$

### 7.5.5 Convergência

Uma prova da convergência do Algoritmo ADMM pode ser encontrada em [BPC<sup>+</sup>11]. A partir da prova de convergência do algoritmo, pode-se concluir que as iterações do ADMM, assumindo  $J$  e  $\zeta$  funções próprias, convexas e fechadas e que o Lagrangiano associado tem um ponto de sela, satisfazem o seguinte:

- convergência residual:  $\mathbf{r}_j \rightarrow 0$  se  $j \rightarrow \infty$ ;
- convergência do objetivo:  $J(\Delta \mathbf{u}_j) + \zeta(\mathbf{z}_j) \rightarrow \mathbf{p}^*$  se  $j \rightarrow \infty$ , onde  $\mathbf{p}^*$  é o ponto ótimo primal;
- convergência da variável dual:  $\nu_j \rightarrow \nu^*$  se  $j \rightarrow \infty$ , onde  $\nu^*$  é o ponto ótimo dual.

### 7.5.6 Forma escalonada

O ADMM pode ser representado de uma forma mais compacta e que pode ser mais conveniente para a implementação. Essa forma é chamada de escalonada e pode ser obtida por meio da definição de uma nova variável  $\phi_j = (\frac{1}{\rho})\nu_j$ , chamada de variável escalonada dual. Dessa forma, o ADMM escalonado pode ser computado recursivamente da seguinte forma:

$$\begin{aligned}\Delta \mathbf{u}_{j+1} &= \operatorname{argmin} \left( J(\Delta \mathbf{u}_j) + \frac{\rho}{2} \|\bar{\mathbf{M}}\Delta \mathbf{u}_j + \bar{\mathbf{D}}\mathbf{z}_j - \bar{\mathbf{m}} + \phi_j\|_2^2 \right), \\ \mathbf{z}_{j+1} &= \operatorname{argmin} \left( \zeta(\mathbf{z}_j) + \frac{\rho}{2} \|\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_j - \bar{\mathbf{m}} + \phi_j\|_2^2 \right), \\ \phi_{j+1} &= \phi_j + (\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_{j+1} - \bar{\mathbf{m}}).\end{aligned}\tag{7.62}$$

A partir dos conceitos apresentados, o algoritmo genérico do ADMM para o problema (7.55) na forma escalonada pode ser visto no Algoritmo 12.

### 7.5.7 Algoritmo ADMM para MPC

A partir do QP do MPC, definido em (??), o primeiro passo proposto é representar as restrições por meio de uma função indicadora  $I_-(\mathbf{x})$ . Pode-se definir  $J(\Delta \mathbf{u}) = \frac{1}{2}\Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u}$  e reescrever o problema rearranjando as restrições com uso da função indicadora  $I_-(x)$  na função objetivo, resultando em:

$$\min_{\Delta \mathbf{u}} J(\Delta \mathbf{u}) + I_-(\bar{\mathbf{R}}\Delta \mathbf{u} - \bar{\mathbf{r}}).\tag{7.63}$$

Para se obter a forma padrão do ADMM, apresentada em (7.55), pode-se definir uma nova variável  $\mathbf{z} = \bar{\mathbf{R}}\Delta \mathbf{u} - \bar{\mathbf{r}}$  para a segunda equação e inserir uma

---

**Algoritmo 12:** Método dos multiplicadores em direções alternadas (ADMM) na forma escalonada

---

**Entrada:**  $J, \zeta, \bar{\mathbf{M}}, \bar{\mathbf{D}}, \bar{\mathbf{m}}, \Delta \mathbf{u}_0$  factível,  $\mathbf{z}_0$  factível

**Dados:**  $\rho > 0, \phi^0 = \mathbf{0}, \epsilon_{pri} > 0, \epsilon_{dual} > 0, j_{max}$

**Saída:**  $\Delta \mathbf{u}$

```

1  $j \leftarrow 0$ ;
2 enquanto  $j < j_{max}$  faça
3    $\Delta \mathbf{u}_{j+1} \leftarrow \operatorname{argmin} \left( J(\Delta \mathbf{u}_j) + \frac{\rho}{2} \|\bar{\mathbf{M}}\Delta \mathbf{u}_j + \bar{\mathbf{D}}\mathbf{z}_j - \bar{\mathbf{m}} + \phi_j\|^2 \right)$  ;
4    $\mathbf{z}_{j+1} \leftarrow \operatorname{argmin} \left( \zeta(\mathbf{z}_j) + \frac{\rho}{2} \|\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_j - \bar{\mathbf{m}} + \phi_j\|^2 \right)$ ;
5    $\phi_{j+1} \leftarrow \phi_j + (\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_{j+1} - \bar{\mathbf{m}})$ ;
6   se  $\|\bar{\mathbf{M}}\Delta \mathbf{u}_{j+1} + \bar{\mathbf{D}}\mathbf{z}_{j+1} - \bar{\mathbf{m}}\| \leq \epsilon_{pri}$  e  $\|\rho \bar{\mathbf{M}}^T \bar{\mathbf{D}}(\mathbf{z}_{j+1} - \mathbf{z}_j)\| \leq \epsilon_{dual}$ 
7     então
8       retorna;
9   fim
10   $j \leftarrow j + 1$  ;
11 fim

```

---

restrição de igualdade para manter o acoplamento entre as variáveis:

$$\begin{aligned} \min_{\Delta \mathbf{u}, \mathbf{z}} \quad & J(\Delta \mathbf{u}) + I_-(\mathbf{z}) \\ \text{s.a.} \quad & \bar{\mathbf{R}}\Delta \mathbf{u} - \mathbf{z} = \bar{\mathbf{r}}. \end{aligned} \quad (7.64)$$

A partir da forma padrão de (7.64), o Lagrangiano aumentado  $\mathcal{L}_\rho$  pode ser definido como:

$$\mathcal{L}_\rho(\Delta \mathbf{u}, \mathbf{z}, \nu) = J(\Delta \mathbf{u}) + I_-(\mathbf{z}) + \nu^T (\bar{\mathbf{R}}\Delta \mathbf{u} - \mathbf{z} - \bar{\mathbf{r}}) + \frac{\rho}{2} \|\bar{\mathbf{R}}\Delta \mathbf{u} - \mathbf{z} - \bar{\mathbf{r}}\|^2.$$

Com o Lagrangiano aumentado e os passos básicos do ADMM escalonado descritos em (7.62), o GPCADMM pode ser computado como:

$$\Delta \mathbf{u}_{j+1} = \operatorname{argmin} \left( J(\Delta \mathbf{u}_j) + \frac{\rho}{2} \|\bar{\mathbf{R}}\Delta \mathbf{u} - \mathbf{z}_j - \bar{\mathbf{r}} + \phi_j\|^2 \right), \quad (7.65)$$

$$\mathbf{z}_{j+1} = \operatorname{argmin} \left( I_-(\mathbf{z}) + \frac{\rho}{2} \|\bar{\mathbf{R}}\Delta \mathbf{u}_{j+1} - \mathbf{z}_j - \bar{\mathbf{r}} + \phi_j\|^2 \right), \quad (7.66)$$

$$\phi_{j+1} = \phi_j + (\bar{\mathbf{R}}\Delta \mathbf{u}_{j+1} - \mathbf{z}_{j+1} - \bar{\mathbf{r}}). \quad (7.67)$$

Para computar o passo da equação (7.65), pode-se abrir o termo da norma e agrupar da seguinte forma:

$$\Delta \mathbf{u}_{j+1} = \operatorname{argmin} \left( \frac{1}{2} \Delta \mathbf{u}^T (\mathbf{H} + \bar{\mathbf{R}}^T \bar{\mathbf{R}}) \Delta \mathbf{u} + (\mathbf{b}^T + \bar{\mathbf{R}}^T (\phi - \mathbf{z} - \bar{\mathbf{r}})) \Delta \mathbf{u} \right).$$

Portanto, se forem definidos  $\tilde{\mathbf{H}} = \mathbf{H} + \bar{\mathbf{R}}^T \bar{\mathbf{R}}$  e  $\tilde{\mathbf{b}} = \mathbf{b}^T + \bar{\mathbf{R}}^T(\phi - \mathbf{z} - \bar{\mathbf{r}})$  recai-se num problema padrão de programação quadrática sem restrições:

$$\min_{\Delta \mathbf{u}} \quad \frac{1}{2} \Delta \mathbf{u}^T \tilde{\mathbf{H}} \Delta \mathbf{u} + \tilde{\mathbf{b}}^T \Delta \mathbf{u}, \quad (7.68)$$

que possui solução analítica:

$$\Delta \mathbf{u}_{j+1} = -\tilde{\mathbf{H}}^{-1} \tilde{\mathbf{b}}.$$

Como em um problema típico de GPC a matriz  $\tilde{\mathbf{H}}$  pode ser computada offline, então a sua inversa pode ser armazenada *offline* e a solução desse passo pode ser computada apenas com uma multiplicação matricial.

Para computar o passo (7.66), percebe-se que este tipo de problema pode ser reescrito como um operador proximal [SSS<sup>+</sup>16]. O operador proximal de uma função escalada  $\rho h(x)$  é definido como:

$$\text{prox}_{\rho f}(x) = \underset{z}{\text{argmin}} \left( f(z) + \frac{1}{2\rho} \|z - x\|^2 \right). \quad (7.69)$$

Portanto, aplicando a definição de operador proximal na função indicadora  $I_-(z)$  e com um passo de tamanho  $\rho$ , o cômputo de (7.66) pode ser representado como:

$$\mathbf{z}_{j+1} = \text{prox}_{\rho I_-}(\mathbf{R} \Delta \mathbf{u}_{j+1} - \mathbf{r} + (1/\rho) \hat{v}_j). \quad (7.70)$$

Dado que  $I_-(\mathbf{z})$  é uma função indicadora no ortante negativo  $\mathbb{R}_-^n$ , o qual é um conjunto convexo fechado e não vazio, o operador proximal se torna uma projeção sobre  $\mathbb{R}_-^n$  [? ]. Portanto, (7.70) pode ser obtido com a seleção da parte negativa de cada elemento ou zero:

$$\mathbf{z}_{j+1} = \min(\bar{\mathbf{R}} \Delta \mathbf{u}_{j+1} - \bar{\mathbf{r}} + \phi_j, \mathbf{0}).$$

Os resíduos das condições de otimalidade podem ser usados como critério de parada. O resíduo primal  $\mathbf{r}$  e o resíduo dual  $\mathbf{s}$  são dados por:

$$\begin{aligned} \mathbf{r}_{j+1} &= \bar{\mathbf{R}} \Delta \mathbf{u}_{j+1} - \mathbf{z}_{j+1} - \bar{\mathbf{r}}, \\ \mathbf{s}_{j+1} &= \rho \bar{\mathbf{R}}^T (\mathbf{z}_{j+1} - \mathbf{z}_j). \end{aligned} \quad (7.71)$$

Portanto, um critério de parada razoável é que os resíduos devem ser menores que um valor de tolerância desejado:

$$\|\mathbf{r}_j\| \leq \epsilon_p \quad \text{e} \quad \|\mathbf{s}_j\| \leq \epsilon_d. \quad (7.72)$$

A partir dos desenvolvimentos apresentados, pode-se resumir o ADMM para MPC no Algoritmo 13.

---

**Algoritmo 13:** Algoritmo ADMM para MPC

---

**Entrada:**  $\check{\mathbf{H}} = \mathbf{H}^{-1}, \mathbf{b}, \bar{\mathbf{R}}, \bar{\mathbf{r}}$ **Saída:**  $\Delta \mathbf{u}(k)$ **Dados:**  $\rho > 0, \phi_0 = \mathbf{0}, \epsilon_p, \epsilon_d$ 

```
1 início
2    $j \leftarrow 0$ ;
3   enquanto  $j < j_{max}$  faça
4      $\tilde{\mathbf{b}} \leftarrow \mathbf{b}^T + \rho \bar{\mathbf{R}}^T (\phi_j - \mathbf{z}_j - \bar{\mathbf{r}})$ ;
5      $\Delta \mathbf{u} \leftarrow -\check{\mathbf{H}} \tilde{\mathbf{b}}$ ;
6      $\mathbf{z}_{j+1} \leftarrow \min(\bar{\mathbf{R}} \Delta \mathbf{u} - \bar{\mathbf{r}} + \phi_j, \mathbf{0})$ ;
7      $\phi_{j+1} \leftarrow \phi_j + (\bar{\mathbf{R}} \Delta \mathbf{u} - \mathbf{z}_{j+1} - \bar{\mathbf{r}})$ ;
8     se  $\|\bar{\mathbf{R}} \Delta \mathbf{u} - \mathbf{z}_{j+1} - \bar{\mathbf{r}}\|_2 \leq \epsilon_p$  e  $\|\rho \bar{\mathbf{R}}^T (\mathbf{z}_{j+1} - \mathbf{z}_j)\|_2 \leq \epsilon_d$  então
9       retorna;
10    fim
11     $j \leftarrow j + 1$ ;
12  fim
13 fim
```

---

## 7.6 Considerações finais

Neste Capítulo foram abordados algoritmos para as formulações do GPC e do DMC por serem as mais utilizadas na indústria. O algoritmo de otimização baseado no método de barreira é uma solução que pode ser útil nos casos onde não são desejadas grandes variações no número de iterações do otimizador, apesar de ter um tempo de cômputo médio mais elevado. Os métodos de otimização de primeira ordem apresentados, GPAD e ADMM são mais simples de serem implementados e necessitam de operações matemáticas mais simples o que os tornam adequados para aplicações com dinâmicas rápida, e controladores com recursos computacionais mais modestos. Ambos possuem a característica de não necessitar de operações de divisão, o que possibilita a utilização de aritmética de ponto fixo para a representação numérica com um acúmulo menor de erro. O ADMM, apresenta o inconveniente de necessitar do ajuste de um parâmetro  $\rho$ , que nem sempre tem uma escolha trivial.

De modo a orientar a escolha do algoritmo de controle, na Tabela 7.1 é apresentada uma comparação qualitativa que destaca os algoritmos mais adequados com as métricas consideradas relevantes para a implementação.

## 7.7 Exercícios propostos

1. Considerando um sistema cujo modelo é dado por  $A = 1 - 1,783z^{-1} + 0,819z^{-2}$  e  $B = 0,0187z^{-1} + 0,01746z^{-2}$ , calcule a predição de resposta



Tabela 7.1: Tabela qualitativa para orientar a escolha do algoritmo de otimização.

Requisito	GPAD	ADMM	Barreira logarítmica
Maior rapidez	✓	✓	-
Menos recursos computacionais	✓	✓	-
Mais simples de codificar	✓	✓	-
Previsibilidade WCET	✓	-	✓
Menor variabilidade temporal	-	-	✓
Maior precisão	-	-	✓

livre de forma recursiva a partir da equação (7.2). Assuma o horizonte  $N_1 = 1$  e  $N_2 = 3$  e as condições iniciais  $y[k] = 0,9$ ,  $y[k-1] = 0,8$ ,  $y[k-2] = 0,7$ ,  $\Delta \mathbf{u}[k-1] = 0,1$  e  $\Delta \mathbf{u}[k-1] = 0,2$ .

2. Implemente um código genérico em uma linguagem de programação que calcule a predição de resposta livre de forma recursiva. Teste o código com o exemplo do exercício 1.
3. Dadas as matrizes que modelam o problema do QP de um determinado sistema de controle:  $\mathbf{H} = \begin{bmatrix} 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix}$ ,  $\mathbf{b}^T = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix}$ ,  $\bar{\mathbf{R}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  e  $\bar{\mathbf{r}} = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}$ , calcule de forma manual  $\Delta \mathbf{u}$  pelo algoritmo de barreira logarítmica apresentado no algoritmo 7. Utilize uma tolerância  $\epsilon = 0,1$ ,  $k_b = 2$ ,  $\mu = 4$ ,  $\varpi = 0,2$  e  $\beta = 0,5$ .
4. Implemente um código genérico em uma linguagem de programação que calcule o incremento de controle  $\Delta \mathbf{u}$  pelo algoritmo de barreira logarítmica apresentado no algoritmo 7. Teste o código com o exemplo do exercício 3.
5. Dadas as matrizes que modelam o problema do QP de um determinado sistema de controle:  $\mathbf{H} = \begin{bmatrix} 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix}$ ,  $\mathbf{b}^T = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix}$ ,  $\bar{\mathbf{R}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  e  $\bar{\mathbf{r}} = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}$ , calcule de forma manual  $\Delta \mathbf{u}$  pelo algoritmo GPAD apresentado no algoritmo 11. Utilize as tolerâncias  $\epsilon_f = 0,1$  e  $\epsilon_o = 0,1$ .
6. Implemente um código genérico em em uma linguagem de programação

que calcule o incremento de controle  $\delta \mathbf{u}$  pelo algoritmo GPAD apresentado no algoritmo 11. Teste o código com o exemplo do exercício 5.

7. Dadas as matrizes que modelam o problema do QP de um determinado sistema de controle:  $\mathbf{H} = \begin{bmatrix} 0,1 & 0,2 \\ 0,2 & 0,1 \end{bmatrix}$ ,  $\mathbf{b}^T = \begin{bmatrix} 0,1 \\ 0,2 \end{bmatrix}$ ,  $\bar{\mathbf{R}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  e  $\bar{\mathbf{r}} = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix}$ , calcule de forma manual  $\Delta \mathbf{u}$  pelo algoritmo ADMM apresentado no algoritmo 13. Utilize as tolerâncias  $\epsilon_f = 0,1$ ,  $\epsilon_o = 0,1$  e  $\rho = 0,5$ .
8. Implemente um código genérico em uma linguagem de programação que calcule o incremento de controle  $\Delta \mathbf{u}$  pelo algoritmo ADMM apresentado no algoritmo 13. Teste o código com o exemplo do exercício 7.

# Apêndice A

## Apêndice: Simuladores

Neste apêndice apresentamos brevemente um simulador interativo para o controlador GPC MIMO. O simulador pode ser descarregado da página web do professor Jose Luis Guzman Sanchez, da Universidade de Almeria na Espanha [https://w3.ual.es/personal/joguzman/material\\_docente\\_itools.shtml](https://w3.ual.es/personal/joguzman/material_docente_itools.shtml).

O arquivo é executável e não precisa de nenhuma instalação. Ainda da mesma página podem ser baixados os manuais de utilização. Por ser interativa, é possível modificar quaisquer parâmetro da simulação e observar os efeitos causados no desempenho do sistema de forma imediata. Por isso, é uma ferramenta bem útil para ilustrar os principais conceitos de MPC, a pesar de ser baseada apenas no GPC.

### A.1 Simulador GPC MIMO

Neste simulador apenas podem ser simulados 4 casos de estudo, mas usando a mesma ferramenta de alocação de polos e zeros no plano complexo, algumas das características dos modelos podem ser alteradas. Neste caso MIMO a ferramenta tem duas janelas, uma para especificar a planta e modelo, e outra para especificar os parâmetros da simulação e controle.

A Figura A.1 mostra a janela da ferramenta para especificar a planta e o modelo, neste caso para um sistema  $2 \times 2$ . Na parte inferior a esquerda podemos ver os gráficos da saída  $y$  e do controle  $u$ . Aqui existe a possibilidade de ver todas as variáveis ou apenas algumas. Na parte direita se observa a janela que mostra a localização de polos e zeros de cada um dos 4 elementos da função de transferência do processo e modelo. Assim como no caso SISO é possível modificar a posição de polos e zeros do processo ou modelo.

Para a etapa de ajuste do controle, deve ser escolhida a opção *Design*

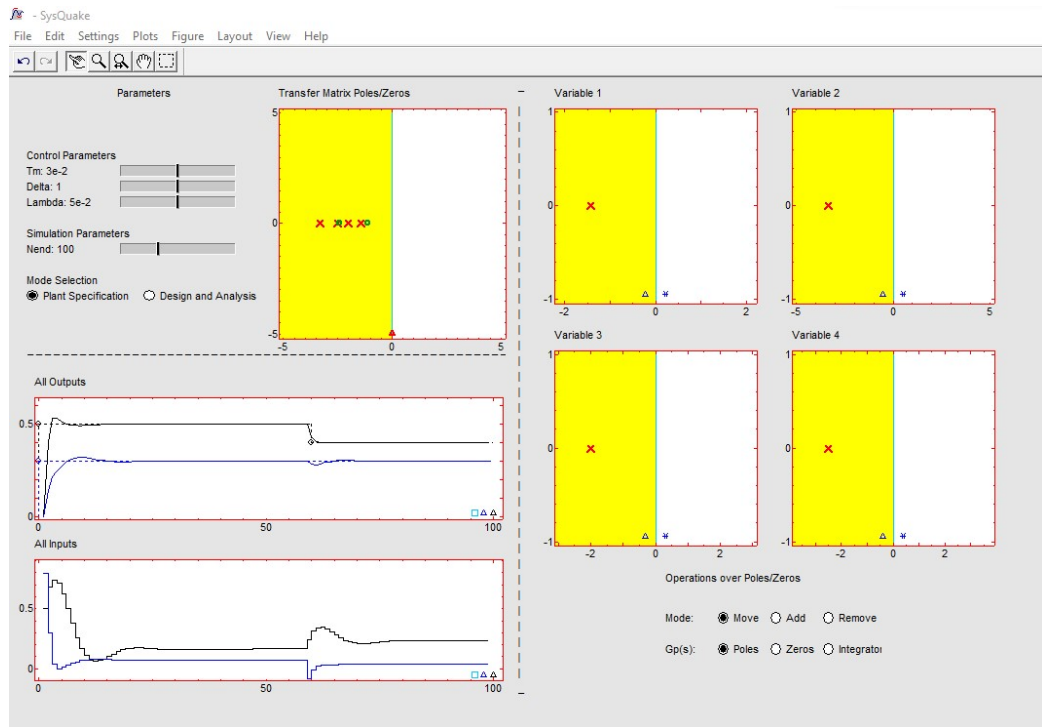


Figura A.1: Janela da ferramenta interativa de simulação do GPC MIMO. Definição do processo e modelo.

*and Analysis* mediante a seleção da opção correspondente na tela vista na Figura A.2. Neste caso a parte direita da tela apresenta gráficos para a saída e os controles. Pode ser escolhida a saída (mediante um *click* nas setinhas mostradas na parte inferior direita de cada gráfico) a ser analisada e então ajustar horizontes, ponderações e restrições correspondentes. Assim como no caso SISO, barras verticais pontilhadas indicam os horizontes de predição no gráfico de  $y$  e de controle no gráfico de  $u$ , agora para cada um dos controles. É possível mudar as referências e perturbações aplicadas assim como o tempo de simulação igual que no caso SISO. Para o ajuste de quaisquer parâmetro basta mover as barras para esquerda ou direita. Para escolha do tipo de restrição a ser usada basta marcar a caixa correspondente. Nos gráficos, é possível modificar os horizontes, tempos de aplicação das mudanças de referência e perturbações, assim como suas amplitudes, apenas mexendo diretamente com o cursor no elemento correspondente.

Para maiores detalhes sugerimos a leitura do manual, que pode ser acessado da mesma página web da ferramenta.

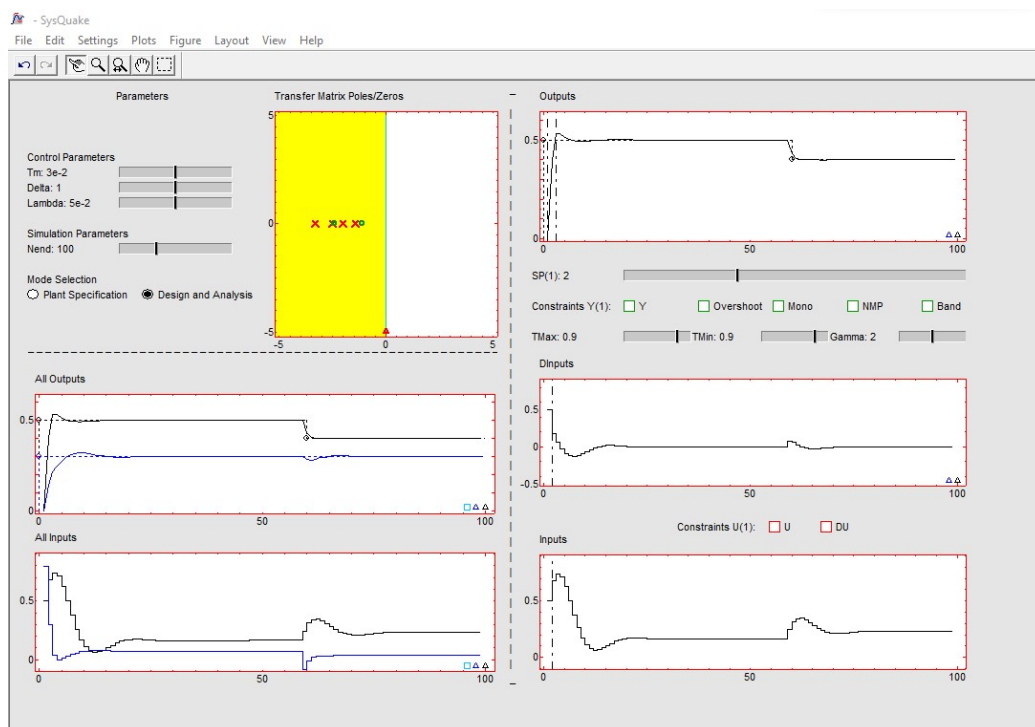


Figura A.2: Janela da ferramenta interativa de simulação do GPC MIMO. Definição do controle e simulação.



## Apêndice B

### Definições de otimização convexa

Neste apêndice são apresentadas algumas definições dos principais termos da área de otimização que são utilizados neste livro. Para um maior aprofundamento nessa temática, recomenda-se a leitura de [BV04], material empregado como base para redação deste apêndice.

**Definição 1** (Conjunto convexo). *Um conjunto  $\Omega \subset \mathbb{R}^n$  é dito convexo, se  $\forall \theta \in [0,1]$ :*

$$\theta x + (1 - \theta)y \in \Omega, \quad \forall x, y \in \Omega.$$

**Definição 2** (Função convexa). *Uma função  $f : \Omega \rightarrow \mathbb{R}$  definida sobre um conjunto convexo  $\Omega \subset \mathbb{R}^n$  é dita convexa em  $\Omega$ , se  $\forall \theta \in [0,1]$ :*

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y), \quad \forall x, y \in \Omega.$$

**Definição 3** (Função estritamente convexa). *Uma função  $f : \Omega \rightarrow \mathbb{R}$  definida sobre um conjunto convexo  $\Omega \subset \mathbb{R}^n$  é dita estritamente convexa em  $\Omega$ , se  $\forall \theta \in [0,1]$ :*

$$f(\theta x + (1 - \theta)y) < \theta f(x) + (1 - \theta)f(y), \quad \forall x, y \in \Omega.$$

**Definição 4** (Função Lipschitz contínua). *Uma função é dita Lipschitz contínua no conjunto  $\Omega$  se existir uma constante Lipschitz  $L \geq 0$  que satisfaça a seguinte condição:*

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \Omega$$

**Definição 5** (Programa quadrático). *O problema de otimização*

$$\begin{aligned} QP : \min_{\mathbf{x}} \quad & \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{b}^T \mathbf{x} \\ \text{s. a.} \quad & \bar{\mathbf{R}} \mathbf{x} \leq \bar{\mathbf{r}} \\ & \bar{\mathbf{C}} \mathbf{x} = \bar{\mathbf{c}}, \end{aligned} \tag{B.1}$$

com  $\mathbf{x} \in \mathbb{R}^n$  sendo o vetor variáveis de decisão,  $\mathbf{H} \in \mathbb{R}^{n \times n}$  sendo uma matriz Hessiana positiva semidefinida simétrica, o vetor gradiente  $\mathbf{b} \in \mathbb{R}^n$ , a matriz de restrições de desigualdade  $\bar{\mathbf{R}} \in \mathbb{R}^{n_{rin} \times n}$ , o vetor de restrições de desigualdade  $\bar{\mathbf{r}} \in \mathbb{R}^{n_{rin}}$ , a matriz de restrições de igualdade  $\bar{\mathbf{C}} \in \mathbb{R}^{n_{req} \times n}$  e o vetor de restrições de igualdade  $\bar{\mathbf{c}} \in \mathbb{R}^{n_{req}}$ , é chamado de programa quadrático e sua solução ótima é dada por  $\mathbf{x}^*$ .

**Definição 6** (Programa quadrático convexo). *Um programa quadrático da forma de (B.1) é dito convexo, se e somente se, sua matriz Hessiana  $\mathbf{H}$  for simétrica positiva definida, ou seja:*

$$\mathbf{H} = \mathbf{H}^T; \quad \nu^T \mathbf{H} \nu \succeq \mathbf{0}, \quad \forall \nu \in \mathbb{R}^n.$$

**Definição 7** (Programa quadrático factível). *Um programa quadrático da forma de (B.1) é dito factível, se e somente se, seu conjunto factível*

$$\mathcal{F} \triangleq \{\check{\mathbf{x}} \in \mathbb{R}^n | \bar{\mathbf{R}} \check{\mathbf{x}} \leq \bar{\mathbf{r}}, \bar{\mathbf{C}} \check{\mathbf{x}} = \bar{\mathbf{c}}\},$$

*for não vazio.*

**Definição 8** (Programa quadrático limitado). *Um programa quadrático da forma de (B.1) é dito limitado (por baixo), se e somente se, existir um número  $\theta$  tal que*

$$\theta \leq \frac{1}{2} \check{\mathbf{x}}^T \mathbf{H} \check{\mathbf{x}} + \mathbf{b}^T \check{\mathbf{x}}, \quad \forall \check{\mathbf{x}} \in \mathcal{F}.$$

**Definição 9** (Lagrangiano). *O Lagrangiano  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{n_{rin}} \times \mathbb{R}^{n_{req}} \rightarrow \mathbb{R}$  associado ao QP (B.1) é definido como:*

$$\mathcal{L}(\mathbf{x}, \psi, \nu) \triangleq \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{b}^T \mathbf{x} + \psi^T (\bar{\mathbf{R}} \mathbf{x} - \bar{\mathbf{r}}) + \nu^T (\bar{\mathbf{C}} \mathbf{x} - \bar{\mathbf{c}})$$



**Definição 10** (Função dual). *Define-se a função dual  $f_d : \mathbb{R}^{n_{rin}} \times \mathbb{R}^{n_{req}} \rightarrow \mathbb{R}$  como o mínimo valor do Lagrangiano sobre  $x$ , ou seja:*

$$f_d(\psi, \nu) \triangleq \inf_x (\mathcal{L}(\mathbf{x}, \psi, \nu)).$$

**Definição 11** (Programa quadrático dual). *Define-se o programa quadrático dual referente ao QP (B.1) como*

$$\begin{aligned} QP_{dual} : \max_{\psi} \quad & f_d(\psi) \\ \text{s. a.} \quad & \psi \geq 0 \end{aligned} \tag{B.2}$$

onde todas as quantidades são definidas como em (B.1) e sua solução ótima é dada por  $\psi^*$ .

**Definição 12** (Dualidade forte). *A dualidade forte se aplica para um QP do tipo (B.1), se a brecha dual (duality gap) for nula, ou seja, o valor ótimo da função objetivo do problema dual  $d^*$  é igual ao valor ótimo da função objetivo primal  $p^*$ :*

$$p^* = d^*.$$

**Definição 13** (Condição de Slater). *É dito que um QP do tipo (B.1) satisfaz a condição de Slater se ele é estritamente factível, ou seja:*

$$\exists \bar{\mathbf{x}} \in \Omega : \bar{\mathbf{R}}\bar{\mathbf{x}} < \bar{\mathbf{r}}, \quad \bar{\mathbf{C}}\bar{\mathbf{x}} = \bar{\mathbf{c}}.$$

**Teorema B.0.1** (Dualidade forte via condição de Slater). *Se o problema primal (B.1) é convexo, e satisfaz a condição de Slater, então a dualidade forte se aplica.*

*Demonstração.* Uma prova da dualidade forte via condição de Slater pode ser vista em [BV04, p. 234]  $\square$

**Teorema B.0.2** (Condições Karush-Kuhn-Tucker (KKT) de otimalidade para QP). *Assumindo um QP do tipo (B.1) estritamente convexo e factível,*

então existe um único  $\mathbf{x}^* \in \mathbb{R}^n$  e um vetor  $\psi^* \in \mathbb{R}^{n_{rin}}$  que satisfazem as seguintes condições:

$$\begin{aligned}\bar{\mathbf{R}}\mathbf{x}^* - \bar{\mathbf{r}} &\leq \mathbf{0}, \\ \bar{\mathbf{C}}\mathbf{x}^* - \bar{\mathbf{c}} &= \mathbf{0}, \\ \psi^* &\geq \mathbf{0}, \\ \psi^{*T}(\bar{\mathbf{R}}\mathbf{x}^* - \bar{\mathbf{r}}) &= \mathbf{0}, \\ \mathbf{H}\mathbf{x}^* + \mathbf{b} + \bar{\mathbf{R}}^T\psi^* &= \mathbf{0},\end{aligned}\tag{B.3}$$

e além disso,  $\mathbf{x}^*$  é o único mínimo global do QP primal (B.1),  $(\mathbf{x}^*, \psi^*)$  é a solução ótima do QP dual (B.2) e os valores ótimos das funções objetivo do QP primal e dual são iguais.

*Demonstração.* Diversos livros de otimização apresentam demonstrações de que as condições KKT são necessárias e suficientes para caracterizar soluções de um QP convexo, recomenda-se [BV04, p. 44].  $\square$

**Definição 14** (Projeção em um conjunto). *Dados um conjunto  $\Omega \subset \mathbb{R}^n$  e um ponto factível  $\mathbf{x}_0 \in \Omega$  define-se a projeção de  $\mathbf{x} \in \mathbb{R}^n$  sobre  $\Omega$ , do seguinte modo:*

$$P(\mathbf{x}) \triangleq \operatorname{argmin}\{\|\mathbf{x}_0 - \mathbf{x}\| : \mathbf{x}_0 \in \Omega\}$$

**Teorema B.0.3** (Teorema da projeção). *Seja  $\Omega \subset \mathbb{R}^n$  um conjunto fechado e não-vazio. Então para todo ponto  $\mathbf{x} \in \mathbb{R}^n$  existe uma projeção de  $\mathbf{x}$  sobre  $\Omega$ . Se, além de ser fechado, o conjunto  $\Omega$  for convexo, então para todo ponto  $\mathbf{x} \in \mathbb{R}^n$  existe uma única projeção de  $\mathbf{x}$  sobre  $\Omega$ .*

*Demonstração.* Uma demonstração pode ser encontrada nas páginas 10 e 94 de [IS14].  $\square$

# Referências Bibliográficas

- [AGH<sup>+</sup>19] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11:1—36, 2019.
- [Beq98] B. W. Bequette. *Process dynamics*. Prentice-Hall international series in the physical and chemical engineering sciences. Prentice Hall PTR, 1998.
- [BPC<sup>+</sup>11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends on Machine Learning*, 3(1):1–122, January 2011.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, Cambridge, MA, March 2004.
- [CB04] E. F. Camacho and C. Bordons. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer, London, 2004.
- [Che13] C.-T. Chen. *Linear System Theory and Design*. Oxford University Press, Cary, 4 edition, 2013.
- [CR80] C. R. Cutler and B. L. Ramaker. ... In ..., ..., 1980.
- [DJ14] J. Du and T. A. Johansen. A gap metric based weighting method for multimodel predictive control of MIMO nonlinear systems. *Journal of Process Control*, 24(9):1346–1357, 2014.
- [EK93] S. Engell and K.-U. Klatt. Nonlinear control of a non-minimum-phase CSTR. In *American Control Conference, 1993*, pages 2941–2945, June 1993.

- [FPEN13] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Sistemas de Controle para Engenharia*. Bookman Editora, 2013.
- [Gol64] A. A. Goldstein. Convex programming in hilbert space. *Bulletin of the American Mathematical Society*, 70(5):709–710, 09 1964.
- [Gup98] Y.P. Gupta. Control of integrating processes using dynamic matrix control. *Chemical Engineering Research and Design*, 76(4):465–470, 1998. Process Operations and Control.
- [GVL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 3 edition, 1996.
- [HJS08] E. Hendricks, O. Jannerup, and P. H. Sørensen. *Linear systems control*. Springer, 2008.
- [IS14] A. Izmailov and M. Solodov. *Otimização-Volume 1-Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*. IMPA, 3 edition, 2014.
- [KL03] Robin De Keyser and Mircea Lazar. A ‘gent’-le approach to nonlinear predictive control. In *Plenary Paper in the ACA-Congress of the Automatic Control Association*, Medellin - Colombia, 2003.
- [Lat06] B. P. Lathi. *Sinais e Sistemas Lineares*. Bookman, 2 edition, 2006.
- [Ław07] M. Ławryńczuk. A family of model predictive control algorithms with artificial neural networks. *International Journal of Applied Mathematics and Computer Science*, 17(2):217–232, 2007.
- [LNRP<sup>+</sup>14] D. M. Lima, J. E. Normey-Rico, A. Plucenio, T. L. Santos, and M. V. C. Gomes. Improving robustness and disturbance rejection performance with industrial MPC. In *Proc. 20th Brazilian Conference on Automation (CBA)*, pages 3229–3236, Belo Horizonte, Brazil, 2014. SBA.
- [LP66] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Computational Mathematics and Mathematical Physics*, pages 1–50, 1966.
- [Nes83] Y. Nesterov. A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

- [Nes14] Y. Nesterov. *Introductory Lectures on Convex Optimization*. Springer Publishing Company, Incorporated, Boston, 2014.
- [NR99] J. E. Normey-Rico. *Prediction for Control*. PhD thesis, Escuela Sup. de Ingenieros, Univ. de Sevilla, 1999.
- [NRC07] J. E. Normey-Rico and E. F. Camacho. *Control of Dead-Time Processes*. Springer, 2007.
- [OSB13] B. O’Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442, Nov 2013.
- [PLFNR20] V. B. Peccin, D. M. Lima, R. C. C. Flesch, and J. E. Normey-Rico. Fast constrained generalized predictive control with ADMM embedded in an FPGA. *IEEE Latin America Transactions*, 18(2):422–429, 2020.
- [Plu10] Agostinho Plucênio. *Desenvolvimento de Técnicas de Controle Não Linear para Elevação de Flúidos Multifásicos*. PhD thesis, Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, 2010.
- [PPBNR07] A. Plucenio, D. J. Pagano, A. H. Bruciapaglia, and J. E. Normey-Rico. A practical approach to predictive control for nonlinear processes. In *Proceedings of NOLCOS 2007 – 7<sup>th</sup> IFAC Symposium on Nonlinear Control Systems*, Pretoria, South Africa, 2007.
- [RC14] A. U. Raghunathan and S. D. Cairano. Infeasibility detection in alternating direction method of multipliers for convex quadratic programs. In *Proc. 53rd IEEE Conference on Decision and Control*, pages 5819–5824, Los Angeles, USA, Dec 2014. IEEE.
- [RJM09] S. Richter, C. N. Jones, and M. Morari. Real-time input-constrained MPC using fast gradient methods. In *Proc. 48th IEEE Conference on Decision and Control (CDC)*, pages 7387–7393, Shanghai, China, Dec 2009. IEEE.
- [RLSCK09] A. Roldao-Lopes, A. Shahzad, G. A. Constantinides, and E. C. Kerrigan. More flops or more precision? accuracy parameterizable linear equation solvers for model predictive control. In *Proc.*

- 17th IEEE Symposium on Field Programmable Custom Computing Machines*, pages 209–216, Napa, USA, April 2009. IEEE.
- [RRJP76] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Algorithm control for industrial processes. In *Proc. 4th IFAC Symp. on Identification and System Parameter Estimation*, Tbilisi, URSS, 1976.
  - [Smi57] O. J. M. Smith. Closed control of loops with dead-time. *Chemical Engineering Progress*, 53:217–219, 1957.
  - [SSS<sup>+</sup>16] G. Stathopoulos, H. Shukla, A. Szucs, Y. Pu, and C. N. Jones. Operator splitting methods in control. *Foundations and Trends in Systems and Control*, 3(3):249–362, 2016.
  - [TN20] T. T. Truong and H. T. Nguyen. Backtracking gradient descent method and some applications in large scale optimisation. *Applied Mathematics & Optimization*, pages 1–30, 2020.
  - [Tse08] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, University of Washington, Seattle, WA, 2008.
  - [vdV64] J.G. van de Vusse. Plug-flow type reactor versus tank reactor. *Chemical Engineering Science*, 19(12):994 – 996, 1964.
  - [WB06] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.
  - [WMN11] A. Wills, A. Mills, and B. Ninness. FPGA implementation of an interior-point solution for linear model predictive control. In *Proc. 18th IFAC World Congr.*, Milano, Italy, 2011. IFAC.
  - [Yan16] B. B. Picon Yang. Contribuições aos algoritmos PNMPC e DMC com passo de estabilidade. Dissertação (mestrado), Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, 2016.