

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TASSIANO NEUHAUS

**PROJETO DE CONTROLADORES
NÃO LINEARES UTILIZANDO
REFERÊNCIA VIRTUAL**

Porto Alegre
2012

TASSIANO NEUHAUS

**PROJETO DE CONTROLADORES
NÃO LINEARES UTILIZANDO
REFERÊNCIA VIRTUAL**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.
Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Alexandre Sanfelice Bazanella

Porto Alegre
2012

TASSIANO NEUHAUS

**PROJETO DE CONTROLADORES
NÃO LINEARES UTILIZANDO
REFERÊNCIA VIRTUAL**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Alexandre Sanfelice Bazanella, Doutor pela Universidade Federal de Santa Catarina, UFSC - Florianópolis, Brasil

Banca Examinadora:

Prof. Dr. Gustavo Henrique da Costa Oliveira, UNICAMP
Doutor em Automação, UNICAMP – Campinas, Brasil

Prof. Dra. Luciola Campestrini, UFRGS
Doutora em Engenharia Elétrica, UFRGS – Porto Alegre, Brasil

Prof. Dr. Luis Fernando Alves Pereira, ITA
Doutor em Engenharia Elétrica, ITA – São Paulo, Brasil

Coordenador do PPGE: _____
Prof. Dr. João Manoel Gomes da Silva Júnior

Porto Alegre, Setembro de 2012.

DEDICATÓRIA

Dedico este trabalho aos meus pais.

RESUMO

Este trabalho tem o intuito de apresentar alguns conceitos relativos à identificação de sistemas, tanto lineares quanto não lineares, além da ideia de referência virtual para, em conjunto com a teoria de projeto de controladores baseados em dados, propor uma forma de projeto de controladores não lineares baseados em identificação de sistemas. A utilização de referência virtual para a obtenção dos sinais necessários para a caracterização do controlador ótimo de um sistema é utilizado no método VRFT (*Virtual Reference Feedback Tuning*). Este método serve como base para o desenvolvimento da proposta deste trabalho que, em conjunto com a teoria de identificação de sistemas não lineares, permite a obtenção do controlador ótimo que leva o sistema a se comportar como especificado em malha fechada. Em especial optou-se pela caracterização do controlador utilizando estrutura de modelos racional, por esta ser uma classe bastante abrangente no que diz respeito à quantidade de sistemas reais que ela é capaz de descrever. Para demonstrar o potencial do método proposto para projeto de controladores, são apresentados exemplos ilustrativos em situações onde o controlador ideal consegue ser representado pela classe de modelos, e quando isso não é possível.

Palavras-chave: Identificação de sistemas, Referência Virtual, Sistemas não lineares, Modelos racionais, Projeto de controladores baseado em dados.

ABSTRACT

This work aims to present some concepts related to linear and nonlinear system identification, as well as the concept of virtual reference that, together with data based controller design's theory, provides design framework for nonlinear controllers. The Virtual Reference Feedback Tuning method (VRFT) is used as a basis for the current proposal, where we propose to unite nonlinear system identification algorithms and virtual reference to obtain the ideal controller: the one which makes the system behave as desired in closed loop. It was chosen to model the controller as a rational model due the wide variety of practical systems that can be represented by this model structure. For rational system identification we used an iterative algorithm which, based on the signal from input and output of the plant, allows to identify the parameters of the pre defined controller structure with the signals obtained by virtual reference. To demonstrate the operation of the proposed identification controller methodology, illustrative examples are presented in situations where the ideal controller can be represented by the class of models, and also when it is not possible.

Keywords: System identification, Virtual reference, nonlinear system, Data driven controller design.

LISTA DE ILUSTRAÇÕES

Figura 1:	Diagrama de blocos para o método do erro de predição	28
Figura 2:	Diagrama de blocos do conjunto ARMAX de modelos. Ambos os diagramas são equivalentes.	33
Figura 3:	Estimativas de um sistema e a elipse representando a região de confiança para um χ de 99%. No centro da elipse destaca-se a média de todas as estimativas.	45
Figura 4:	Sinal PRBS para $n = 7$	47
Figura 5:	Acima: modelo de Hammerstein. Abaixo: Modelo de Wiener.	54
Figura 6:	Representação de uma rede neural multi-camadas.	55
Figura 7:	Representação de redes neurais recorrentes. Onde q^{-1} representa um atraso de tempo na realimentação do sistema.	56
Figura 8:	Função multi-quadráticas inversa para alguns valores de σ	57
Figura 9:	Estimativas obtidas nos 100 experimentos de Monte Carlo realizados para as variáveis θ_1 e θ_2 . Destaca-se a elipse de confiança para $\chi^2 = 95\%$	64
Figura 10:	Comparativo da resposta do sistema real e do modelo estimado para uma entrada do tipo degrau unitário	65
Figura 11:	Circuito do conversor CC-CC Buck	66
Figura 12:	Comparativo entre a saída do sistema real descrito pela equação(82) e a saída do sistema racional descrito pela classe de modelos apresentada em (83) quando os parâmetros estimados para um valor inicial de 31.5, como apresentado em (84).	68
Figura 13:	Representação de um sistema de controle em malha fechada, com ruído aditivo na saída. São apresentados também os principais sinais referenciados.	73
Figura 14:	Diagrama de blocos para o método VRFT. São destacados os sinais reais $u(t)$ e $y(t)$ além dos sinais virtuais $\epsilon(t)$ e $\bar{r}(t)$. Em pontilhado está o controlador que quando aplicado sobre a malha, faz com que o sistema se comporte como desejado: $T_d(q)$	76

Figura 15:	Gráfico da função custo de algum sistema hipotético e a respectiva função custo proposta pelo método VRFT que é mais simples de encontrar o ponto de mínimo pois é quadrática em θ , não recaindo em mínimos locais. O valor θ^* é o ponto de mínimo de ambas as funções custo, logo, minimizando a função custo $J_{VR}(\theta)$ é o equivalente a minimizar $J_y(\theta)$ sob condições ideais.	78
Figura 16:	Exemplo hipotético de funções custo na situação onde o controlador ideal não consegue ser representado pelo modelo. O uso do filtro $L(q)$ reduz o erro de polarização.	80
Figura 17:	Resultado das 100 estimativas de Monte Carlo dos parâmetros θ_1 e θ_2 para o controlador apresentado em (114). Com $\sigma_e^2 = 0.005$	83
Figura 18:	100 estimativas Monte Carlo dos parâmetros θ_1 e θ_2 para o controlador apresentado em (114) com um ruído de variância de $\sigma_e^2 = 0.02$	84
Figura 19:	Resultado das 100 estimativas de Monte Carlo dos parâmetros θ_1 e θ_2 para o controlador apresentado em (114) com variância do ruído $\sigma_e^2 = 0.02$ utilizando variáveis instrumentais para estimar os parâmetros.	85
Figura 20:	100 estimativas Monte Carlo dos parâmetros θ_1 , θ_2 e θ_3 para o controlador apresentado em (117) com variância do ruído $\sigma_e^2 = 0.05$ utilizando variáveis instrumentais	87
Figura 21:	Comparativo da resposta do sistema a um degrau unitário quando o controlador inserido é obtido pelo método VRFT utilizando o filtro L e quando não se utiliza este artifício.	89
Figura 22:	Diagrama de Bode para as funções de transferência dos controladores estimados utilizando VRFT com e sem o filtro $L(q)$ e variáveis instrumentais.	90
Figura 23:	Representação em blocos do sistema em malha fechada para obtenção dos sinais $\epsilon(t)$ e $\bar{r}(t)$ utilizando referência virtual. Traçado está a realimentação e o controlador, que não estão presentes no procedimento para obtenção dos sinais.	95
Figura 24:	Um sistema do tipo Wiener como descrito por (123) e (124) . . .	97
Figura 25:	Um controlador do tipo Hammerstein como descrito por (125) e (126)	97
Figura 26:	Diagrama de blocos para um sistema não linear do tipo Wiener	98
Figura 27:	Comparativo da resposta a um degrau unitário do modelo de referência $T_d(q)$ com o sistema real quando o controlador não linear é inserido na planta.	101

Figura 28:	Sinais $v(t)$ e $z(t)$ do sistema apresentado na Figura 26 quando este é excitado por um degrau unitário.	102
Figura 29:	Relação entre os sinais $v(t)$ e $z(t)$ quando o sistema é alimentado por um degrau unitário	103
Figura 30:	Erro entre a resposta esperada e a obtida para o sistema quando submetido a uma entrada do tipo degrau unitário.	105
Figura 31:	100 experimentos de Monte Carlo das variáveis θ_1 e θ_2	106
Figura 32:	Exemplo onde $\mathcal{C}_d \notin \mathcal{C}$: Resposta do sistema ao degrau unitário para o sistema desejado $T_d(q)$ e o sistema quando o controlador parametrizado por θ como descrito em (156).	108
Figura 33:	Exemplo onde $\mathcal{C}_d \notin \mathcal{C}$: Erro entre o comportamento do sistema em malha fechada e o comportamento em malha fechada desejado.108	

LISTA DE TABELAS

Tabela 1:	Alguns modelos comuns para sistemas SISO. Casos especiais da equação (24).	34
Tabela 2:	Polinômios $A(q)$ que geram o máximo período M para sinais PRBS de ordem n . $a_k = 1$ para os k indicados, 0 para os demais. Diversas outras escolhas existem para os mesmos n	46
Tabela 3:	Algumas funções Radiais de base comumente utilizadas.	57
Tabela 4:	Tabela com os valores das estimativas do conversor de corrente contínua para diferentes valores de condições iniciais. Todas as simulações utilizaram $N = 100$	67
Tabela 5:	Valor dos custos J_{VR}^N e J_y além da variância das estimativas para diferentes valores de σ_e^2 quando o método VRFT não utiliza variáveis instrumentais para a estimativa dos parâmetros θ	85
Tabela 6:	Valor dos custos J_{VR}^N e J_y além da variância das estimativas para diferentes valores de σ_e^2 quando o método VRFT utiliza variáveis instrumentais para a estimativa dos parâmetros θ	86
Tabela 7:	Valor dos custos J_{VR}^N e J_y além da variância das estimativas para diferentes valores de σ_e^2 quando o método VRFT utiliza variáveis instrumentais para a estimativa dos parâmetros θ do controlador (117)	87
Tabela 8:	Valor dos custos J_{VR}^N e J_y para o sistema controlado por $C(q)$ e $C_L(q)$	89

LISTA DE ABREVIATURAS

ARARMAX	<i>Autoregressive, autoregressive moving average model with exogenous input</i>
ARARX	<i>Autoregressive, autoregressive model with exogenous input</i>
ARMA	<i>Autoregressive moving average</i>
ARMAX	<i>Autoregressive moving average with exogenous input</i>
ARX	<i>Autoregressive with exogenous input</i>
BJ	<i>Box-Jenkins</i>
CbT	<i>Correlation-based Tuning</i>
CC	Corrente contínua
<i>den</i>	Denominador
FDT	<i>Frequency domain Tuning</i>
FIR	<i>Finite impulse response</i>
IFT	<i>Iterative feedback tuning</i>
IV	<i>Instrumental variables</i>
LMI	<i>Linear Matrix Inequality</i>
LTI	<i>Linear time invariant</i>
MA	<i>Moving average</i>
MMQ	Método dos mínimos quadrados
NARMAX	<i>Nonlinear autoregressive moving average model with exogenous variables</i>
NARX	<i>Nonlinear autoregressive model with exogenous variables</i>

<i>num</i>	Numerador
OE	<i>Output error</i>
PEM	<i>Prediction error method</i>
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
PRBS	<i>Pseudo random binary sequence</i>
RBF	<i>Radial basis functions</i>
SISO	<i>Single input single output</i>
VRFT	<i>Virtual Reference Feedback Tuning</i>

LISTA DE SÍMBOLOS

A^{-1}	inverso de A
A^T	transposto de A
$E(\cdot)$	valor esperado
$res(x, 2)$	resíduo de x dividido por 2
$G_0(q)$	Função de transferência que representa a planta real do sistema.
$G(q, \theta)$	Função de transferência que representa a planta a ser estimada na identificação.
$H_0(q)$	filtro do ruído branco que atua sobre o sistema.
$H(q, \theta)$	filtro do ruído branco que atua sobre o sistema, a qual se quer identificar.
θ_0	conjunto de parâmetros que faz com que o modelo identificado seja igual ao sistema real.
θ^*	conjunto de parâmetros quando a quantidade de dados $N \rightarrow \infty$
θ	conjunto qualquer de parâmetros estimado do sistema.
$\hat{\theta}_N$	estimativa para um certo valor de N pontos.
\mathcal{S}	sistema real sob análise.
\mathcal{M}	classe de modelos utilizada para identificar o sistema real.
\mathcal{C}	classe de modelos utilizada para identificar o controlador do sistema.
$T(q)$	comportamento do sistema em malha fechada.
$T_d(q)$	comportamento do sistema em malha fechada desejado.
$u(t)$	sinal de saída do controlador ou sinal de entrada da planta.
$y(t)$	sinal de saída da planta.

$e(t)$	ruído branco.
$\nu(t)$	ruído resultante depois de alterado pelo filtro $H_0(q)$
ϕ	variável de regressão utilizada para identificação do sistema.
$Z(t)$	instrumento utilizado no método de variáveis instrumentais.
Φ_a	espectro do sinal a .
\prod	produtório
\sum	somatório
$L(q)$	filtro utilizado no método VRFT.
σ_e^2	variância do ruído.
$\chi^2(n)$	distribuição Qui-quadrado com n graus de liberdade
$J_y(\cdot)$	função custo para critério de seguimento de referência.
\mathbb{R}	conjunto dos números reais
\mathbb{R}^n	espaço euclidiano de ordem n
q	operador de avanço
$\varepsilon(t, \theta)$	erro de predição
$\epsilon(t)$	sinal de entrada do controlador, erro entre a saída $y(t)$ e a referência $r(t)$
Z^N	conjunto de dados de tamanho N

SUMÁRIO

1	INTRODUÇÃO	21
2	IDENTIFICAÇÃO DE SISTEMAS LINEARES INVARIANTES NO TEMPO DE TEMPO DISCRETO	25
2.1	Definições	25
2.1.1	Elementos da identificação	27
2.2	Premissas para o sucesso do experimento de identificação	28
2.2.1	Persistência de Excitação	28
2.2.2	Experimentos Informativos	29
2.2.3	Escolha do modelo e sobre-modelagem	30
2.3	Estruturas clássicas de modelos	32
2.3.1	Forma geral de classes de modelos	32
2.4	Métodos de identificação	35
2.4.1	Método dos mínimos quadrados	35
2.4.2	Método das variáveis instrumentais	36
2.4.3	Modelagem do ruído	37
2.5	Propriedades estatísticas da estimativa	38
2.5.1	Conjunto de dados	38
2.5.2	Consistência e identificabilidade	39
2.5.3	Incertezas nas estimativas dos parâmetros	40
2.5.4	Distribuição assintótica das estimativas dos parâmetros	42
2.6	Projeto de Experimento	44
2.6.1	Sinais de entrada mais comumente utilizados	45
2.6.2	Projeto de experimento	46
2.7	Considerações Finais	49
3	IDENTIFICAÇÃO DE SISTEMAS NÃO LINEARES DE TEMPO DISCRETO	51
3.1	Conceitos básicos	51

3.1.1	Tipos de não linearidades	52
3.2	Estruturas de modelos para sistemas não lineares	53
3.2.1	Modelos de Wiener e Hammerstein	53
3.2.2	Série de Volterra	54
3.2.3	Redes Neurais	54
3.2.4	Funções Radiais de Base	56
3.2.5	Modelos NARMAX	58
3.2.6	Modelo polinomial	59
3.3	Modelo Racional	60
3.3.1	Algoritmo para identificação de modelos racionais	60
3.3.2	Exemplos ilustrativos	62
3.4	Considerações Finais	68
4	CONTROLE BASEADO EM DADOS	71
4.1	Definições	71
4.1.1	Critérios de Desempenho	73
4.2	Virtual reference feedback tuning	75
4.2.1	O método	76
4.2.2	Utilização do Filtro $L(q)$	77
4.2.3	Dados corrompidos por ruído	81
4.2.4	Exemplos ilustrativos	82
4.3	Considerações Finais	89
5	IDENTIFICAÇÃO DE CONTROLADORES NÃO LINEARES UTILIZANDO REFERÊNCIA VIRTUAL	93
5.1	Introdução	93
5.2	Definições	94
5.3	Método	94
5.3.1	Classe do modelo	95
5.4	Wiener / Hammerstein	96
5.4.1	Convexidade	98
5.4.2	Exemplo numérico	99
5.5	Controladores Racionais	102
5.5.1	Controlador ideal representado pela classe de modelos: $\mathcal{C}_d \in \mathcal{C}$	104
5.5.2	Controlador ideal não sendo representado pela classe de modelos: $\mathcal{C}_d \notin \mathcal{C}$	106
5.6	Considerações Finais	109
6	CONCLUSÃO	111
	REFERÊNCIAS	113

APÊNDICE A	ALGORITMO DE IDENTIFICAÇÃO DE SISTEMAS NÃO LINEARES	117
A.1	Rotina principal	117
A.2	Rotinas auxiliares	120
APÊNDICE B	ROTINAS PARA IDENTIFICAÇÃO CLASSES DE MODELOS RACIONAIS	127
B.1	Identificação de um conversor de corrente contínua do tipo Buck	127
B.2	Identificação de uma classe de modelos racionais	129
APÊNDICE C	ROTINAS PARA IDENTIFICAÇÃO DE CONTROLADORES NÃO LINEARES COM O USO DE REFERÊNCIA VIRTUAL	133
C.1	Rotinas para uso do método VRFT em sistemas lineares	133
C.2	Identificação de um controlador polinomial para um planta do tipo Wiener	135
C.3	Identificação de um controlador racional para uma planta também racional	139

1 INTRODUÇÃO

A teoria de identificação de sistemas é foco de estudo há diversas décadas por inúmeros engenheiros e pesquisadores em diversas áreas (SÖDERSTRÖM; STOICA, 1989; LJUNG, 1999; AGUIRRE, 2004) . Muitas aplicações existentes se baseiam direta ou indiretamente no uso de identificação de sistemas e muitas outras aplicações estão em processo de desenvolvimento. Dentro deste contexto, o estudo da teoria de identificação de sistemas é muito vasto e neste trabalho tem-se o interesse em apresentar alguns pontos relativos a este assunto que são base para o desenvolvimento deste trabalho.

Basicamente a identificação de um sistema, ou uma parte de um sistema maior, pode ser dividido em três componentes principais: é necessário que exista um sistema real; uma classe de modelos, que pretende representar este sistema real; um critério que elenca, dentre os infinitos modelos que podem ser obtidos a partir de uma classe, qual é o que melhor representa este sistema. Do sistema real coletam-se os dados que serão utilizados para proceder o processo de identificação, e muitas das propriedades das estimativas obtidas são baseadas na qualidade e quantidade das informações que estes dados coletados conseguem trazer do sistema real.

Nesta questão de informatividade existe uma grande e vasta quantidade de pesquisa desenvolvida e em desenvolvimento (GEVERS et al., 2009). Questões como projetos de experimentos que neste trabalho são abordados superficialmente e que para a identificação de sistemas são uma ferramenta formidável, possuem aplicabilidade em diversas áreas (JANSSON, 2004). Em engenharia, a escolha do sinal que será utilizado para excitar o sistema pode, entre outras características, reduzir erros da estimativa, reduzir tempo e custo do processo de identificação, aumentar a confiabilidade das estimativas obtidas.

Entretanto não é apenas o sinal de entrada do sistema que determina as propriedades estatísticas dos experimentos de identificação executados. A escolha da classe de modelos para identificar o sistema é de crucial importância para que haja sucesso no processo de identificação (LJUNG, 1999). A escolha desta classe deve ser tomada com base em diversas diretrizes, o que torna sua escolha outro foco

extremamente diversificado de estudo para a teoria de identificação de sistemas.

Observa-se com certa frequência que uma classe de modelos é mais performática para representar certos tipos de sistemas. Isso é devido ao fato de que as classes de modelos ou estruturas de modelos se diferenciam umas das outras devido ao uso de bases diferentes para representar cada sistema. Desta forma, sistemas que possuem uma afinidade maior com certas bases são representados com um número menor de parâmetros do que quando uma base qualquer é utilizada. A busca por uma classe de modelos simples, mas que ainda assim consiga representar o sistema real é também foco de diversos estudos. A identificação de um sistema que não possua um número desnecessário de parâmetros é sempre almejado, desta forma reduz-se a possibilidade de existência de dinâmicas no modelo que não estão presentes no processo real, além de reduzir a complexidade dos algoritmos utilizados na identificação.

Em se tratando de aplicações na área da engenharia, o controle de sistemas tem um destaque especial no uso da teoria de identificação de sistemas para a obtenção dos parâmetros do controlador desejado. Muitos dos projetos de controladores baseados em dados se utilizam desta teoria para firmar suas bases e determinar seu funcionamento (CAMPI; LECCHINI; SAVARESI, 2002; BILLINGS; ZHU, 1991; HJALMARSSON et al., 1998; ECKHARD et al., 2009).

Assim como identificação de sistemas, projetos de controladores baseados em dados são foco de estudo em várias e diversas linhas de pesquisa há décadas (BAZANELLA et al., 2008; BAZANELLA; CAMPESTRINI; ECKHARD, 2012; CAMPI; LECCHINI; SAVARESI, 2002). Neste trabalho será dado enfoque ao uso de referência virtual para a obtenção dos sinais necessários para a identificação do controlador. Referência virtual é largamente utilizada pelo método VRFT (*Virtual Reference Feedback Tuning*) que, por ser um método direto, não requer que haja um conhecimento do modelo da planta para que o controlador possa ser projetado. Este método propõe-se a encontrar o controlador ótimo com apenas um conjunto de dados proveniente do sistema (CAMPI; LECCHINI; SAVARESI, 2000, 2002).

Estas características são bastante atrativas para diversas aplicações, sobretudo àquelas onde a parada do processo para execução de testes e experimentos é indesejada, ou por vezes, inviável. Entretanto o uso do método VRFT possui algumas limitações, a maioria deles já apontadas em diversos trabalhos (CAMPESTRINI et al., 2011; BAZANELLA; CAMPESTRINI; ECKHARD, 2012; CAMPESTRINI, 2010). Contudo, o uso deste método para sistemas não lineares é ainda bastante incipiente e os trabalhos existentes nesta área exigem que haja um considerável conhecimento da planta e do sinal aplicado para que as estimativas sejam realizadas com margens de erro conhecidas (CAMPI; SAVARESI, 2006).

O trabalho aqui apresentado tem por objetivo endereçar uma parte deste problema. Para isso usou-se a ideia de referência virtual utilizada pelo método VRFT

para determinar quais são os sinais necessários para identificar o controlador ideal, que leva o sistema a se comportar em malha fechada como especificado. De posse destes sinais obtidos de forma virtual além daqueles medidos diretamente do sistema, utilizam-se estas informações para alimentar um algoritmo de identificação de sistemas não lineares.

Para limitar o escopo do trabalho desenvolvido aqui, algumas decisões sobre estrutura de modelos foram tomadas: optou-se pela identificação de controladores não lineares que são representados por modelos racionais e polinomiais.

Para demonstrar o funcionamento desta proposta de identificação de controladores não lineares, diversos exemplos serão abordados, tanto do algoritmo implementado quanto para o conjunto do uso de referência virtual e classes de modelos racional. Os exemplos abordados têm o intuito de apresentar propriedades das estimativas e avaliar o desempenho da proposta em atingir seus objetivos.

Este trabalho tem a seguinte organização: no Capítulo 2 serão abordados assuntos básicos da teoria de identificação de sistemas. Questões como a escolha da classe de modelos, do método de identificação, das propriedades estatísticas que as estimativas obtidas possuem e uma breve introdução ao que é conhecido como projeto de experimentos para determinar a escolha de sinais de entrada e de objetivos de identificação. No Capítulo 3 são introduzidas algumas características de sistemas não lineares. São apresentadas as classes de modelos mais comumente utilizadas para descrição destes sistemas. Neste capítulo também é apresentado o algoritmo de identificação de modelos tanto racionais como polinomiais e alguns exemplos de uso.

No Capítulo 4 são introduzidos alguns conceitos sobre projeto de controladores baseados em dados. Apresentam-se brevemente alguns dos principais métodos conhecidos e estende-se a discussão para apresentar o método VRFT e com ele a ideia de referência virtual. São também apresentados neste capítulo alguns exemplos de uso do método VRFT e suas propriedades para situações onde o sistema é corrompido por ruído ou quando a classe de modelos não consegue representar o controlador ideal desejado.

No Capítulo 5 são apresentadas as contribuições deste trabalho com a união de referência virtual e o algoritmo de identificação de sistemas não lineares que podem ser descritos por classes de modelos racionais. São apresentados exemplos de uso para sistemas onde a não linearidade está na dinâmica do processo e quando está acoplada de forma estática na entrada ou na saída do processo. São apresentados também exemplos onde a classe de modelos não consegue representar a totalidade das dinâmicas do controlador desejado. Por fim no Capítulo 6 são apresentadas conclusões gerais sobre o que foi desenvolvido neste trabalho.

No Apêndice A são apresentadas as rotinas desenvolvidas em Matlab do algoritmo de identificação de sistemas não lineares utilizando modelos racionais ou

polinomiais. Já no Apêndice B são apresentadas as rotinas desenvolvidas para simular os exemplos apresentados para identificação de sistemas não lineares. No Apêndice C são apresentadas as rotinas utilizadas para o projeto de controladores não lineares utilizando referência virtual.

2 IDENTIFICAÇÃO DE SISTEMAS LINEARES INVARIANTES NO TEMPO DE TEMPO DISCRETO

Identificação de sistemas possui uma vasta aplicabilidade em inúmeros ramos do conhecimento, e não é escopo deste trabalho enumerar todas as possibilidades de utilização desta ferramenta, nem tão pouco esmiuçar sua teoria. Serão aqui abordadas as principais características e princípios que compõem a teoria de identificação de sistemas lineares.

Na Seção 2.1 serão introduzidos os tipos de sistemas lineares que são objetivo de estudo deste capítulo além das definições básicas que serão utilizadas no decorrer do texto. Já na Seção 2.2 serão apresentadas algumas das premissas básicas para que o sucesso do procedimento de identificação possa ser assegurado. Na Seção 2.3 serão descritas algumas das principais famílias de modelos utilizadas para identificação de sistemas lineares de tempo discreto. Em seguida na Seção 2.4 serão apresentados alguns dos métodos mais comuns para identificação e suas características. Na Seção 2.5 serão apresentadas algumas das propriedades estatísticas das estimativas obtidas e quais erros e incertezas são intrínsecos à identificação. Na Seção 2.6 será introduzido o conceito de projeto de experimentos, dando uma ideia geral de escolha do sinal mais apropriado para excitar o sistema de forma eficiente, além de apresentar quais são os sinais mais comumente utilizados. Ao fim na Seção 2.7 será apresentado uma breve conclusão do que foi visto neste capítulo.

2.1 Definições

Neste capítulo tem-se o interesse de estudar sistemas lineares invariantes no tempo (LTI - *linear time invariant*) de tempo discreto SISO (*Single input single output*). Este tipo de sistema pode ser representado como:

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \quad (1)$$

onde q é o operador de avanço de tempo $qx(t) = x(t+1)$, t representa o tempo discreto, $G_0(q)$ é a função de transferência que descreve o comportamento da planta real, $H_0(q)$ é a função de transferência que atua sobre o ruído branco $e(t)$; $u(t)$ é o sinal de entrada aplicado sobre a planta e $y(t)$ é o sinal de saída, ambos medidos e conhecidos.

Para identificar o sistema apresentado em (1) utiliza-se uma família de modelos parametrizada por um vetor $\theta \in \mathbb{R}^d$. Desta forma o sistema a ser identificado pode ser representado como:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (2)$$

onde as funções de transferência $G(q, \theta)$ e $H(q, \theta)$ são funções racionais do operador q .

O objetivo da identificação de sistemas é encontrar um valor para o vetor θ que faça com que $G(q, \theta)$ e $H(q, \theta)$ sejam o mais próximos possíveis de $G_0(q)$ e $H_0(q)$. O vetor θ que faz com que estas duas funções de transferências sejam iguais, quando existir, é denotado por θ_0 :

$$\begin{aligned} G(q, \theta_0) &= G_0(q) \\ H(q, \theta_0) &= H_0(q) \end{aligned} \quad (3)$$

Outra definição que acompanha o que aqui será abordado é a definição de sinal quasi-estacionário:

Definição 2.1. (LJUNG, 1999) *Um processo quasi-estacionário $s(t)$ pode ser definido como:*

- $\bar{E}[s(t)] = m_s(t), \quad |m_s| \leq C, \quad \forall t;$
- $\bar{E}[s(t)s(r)] = |R_s(t, s)| \leq C, \quad \forall t, r;$
- $\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N R_s(t, t - \tau) = R_s(\tau), \quad \forall \tau,$

onde $m_s(t)$ é o valor médio de $s(t)$ e $R_s(t, r)$ é a covariância de s nos instantes t e r .

O operador $\bar{E}[\cdot]$ é definido por (LJUNG, 1999):

$$\bar{E}[f(t)] \triangleq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N E[f(t)] \quad (4)$$

2.1.1 Elementos da identificação

A identificação de sistemas é dependente de três componentes básicas. A primeira delas é o sistema real, definido pelo símbolo \mathcal{S} e que, no caso de sistemas lineares, pode ser definido a partir de (1) como abaixo:

$$\mathcal{S} : y(t) = G_0(q)u(t) + H_0(q)e(t) \quad (5)$$

A segunda componente é a classe de modelos escolhida, denotada por \mathcal{M} , onde:

$$\mathcal{M} : \{G(q, \theta), H(q, \theta) | \theta \in D_{\mathcal{M}}\} \quad (6)$$

A terceira componente é o critério de escolha que diz qual modelo, dentro da classe é melhor que outro. Em outras palavras, qual modelo melhor representa o sistema real \mathcal{S} de acordo com o critério escolhido. Dentre os mais diversos critérios possíveis o mais utilizado é o de erro de predição, descrito a seguir.

Preditores são equações que predizem qual será o próximo valor de saída do sistema baseado na classe do modelo e nos valores de dados coletados até aquele instante. Os sinais $y(t)$ e $u(t)$ são os sinais da saída e entrada medidos do sistema real enquanto que o sinal $\hat{y}(t, \theta)$ é o sinal de saída do preditor. A diferença entre o valor do preditor e o valor do sistema real é conhecida como erro de predição:

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t, \theta) \quad (7)$$

A Figura 1 apresenta um diagrama de blocos de como é organizado o método do erro de predição. Existe um processo que se deseja identificar, para isso escolhe-se uma classe de modelos onde os parâmetros a serem ajustados são representados por θ . O preditor é então ajustado, baseado nas diferentes possibilidades de escolha de θ . Este ajuste é feito a partir do erro entre o sistema real $y(t)$ e a saída do preditor $\hat{y}(t, \theta)$.

O preditor ótimo é dado por (LJUNG, 1999):

$$\hat{y}(t|t-1, \theta) = H^{-1}(q, \theta)G(q, \theta)u(t) + \{1 - H^{-1}(q, \theta)\} y(t) \quad (8)$$

e portanto

$$\varepsilon(t, \theta) = H^{-1}(q, \theta) \{y(t) - G(q, \theta)u(t)\}$$

De posse da definição do erro de predição $\varepsilon(t, \theta)$ é conveniente introduzir o critério mais utilizado para elencar qual é o melhor modelo dentro da classe dentre os infinitos possíveis. Este critério será utilizado no decorrer do texto e é a base para a maioria dos métodos de identificação usualmente utilizados.

$$V(\theta) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} \varepsilon^2(t, \theta) \quad (9)$$

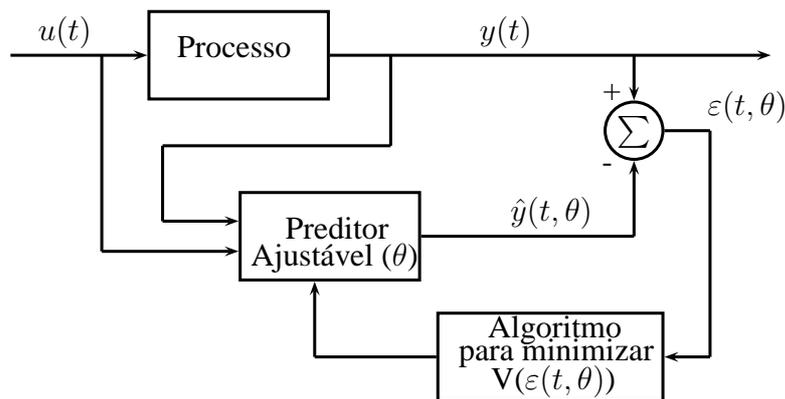


Figura 1: Diagrama de blocos para o método do erro de predição

Este critério é também conhecido como função custo da identificação de sistemas por apresentar o valor do erro quadrático entre os dados obtidos do sistema real \mathcal{S} e os dados obtidos por meio do modelo estimado.

2.2 Premissas para o sucesso do experimento de identificação

Em aplicações de engenharia, deseja-se que a solução de um problema seja única. Desta forma existem algumas premissas que devem ser satisfeitas para que o processo de identificação consiga atingir seus objetivos, obtendo um único vetor θ que consiga representar melhor o sistema real \mathcal{S} de acordo com o critério de desempenho escolhido.

Devem ser levadas em consideração algumas características sobre o sinal de excitação do sistema e sobre a escolha da classe de modelos que irá representar o sistema real \mathcal{S} , para que o processo de identificação possa ser único e representativo, contendo o menor erro possível, além de que esta margem de erro seja conhecida.

2.2.1 Persistência de Excitação

Um sinal quasi-estacionário $u(t)$, com espectro $\Phi_u(\omega)$ é dito *persistentemente excitante de ordem n* se, para todos os filtros de forma:

$$M_n(q) = m_1 q^{-1} + \dots + m_n q^{-n} \quad (10)$$

a relação

$$|M_n(e^{i\omega})|^2 \Phi_u(\omega) \equiv 0, \text{ implica que } M_n(e^{i\omega}) \equiv 0 \quad (11)$$

Outra caracterização pode ser dada em termos da função de covariância, onde $R_u(\tau) = u(t)$ é um sinal quasi-estacionário, e \bar{R}_n uma matriz $n \times n$ definida como:

$$\bar{R}_n = \begin{bmatrix} R_u(0) & R_u(1) & \dots & R_u(n-1) \\ R_u(1) & R_u(2) & \dots & R_u(n-2) \\ \vdots & \vdots & \vdots & \vdots \\ R_u(n-1) & R_u(n-2) & \dots & R_u(0) \end{bmatrix} \quad (12)$$

Então $u(t)$ é persistentemente excitante de ordem n se e somente se \bar{R}_n for não singular (LJUNG, 1999).

A partir da equação (11) pode-se extrair interpretações mais explícitas. Uma delas é que para que o sinal seja persistentemente excitante de ordem n , ele precisa ter n componentes de frequência distintas no intervalo $-\pi < \omega \leq \pi$.

Se um sinal quasi-estacionário é filtrado por uma função de transferência estável, então o sinal resultante também é um sinal quasi-estacionário e desta forma se $|M_n(e^{j\omega})|^2 \Phi_u(\omega)$ é o espectro do sinal $\varrho(t) = M_n(q)u(t)$ então este sinal não perde sua persistência de excitação se filtrado pelo filtro $M_n(q)$.

Considere o somatório de senoides:

$$u(t) = \sum_{k=1}^n \mu_k \cos(\omega_k t), \quad \omega_k \neq \omega_j, \quad \omega_k \neq 0, \quad \omega_k \neq \pi \quad (13)$$

cada uma possui duas linhas espectrais em $\pm \omega_k$, fazendo com que este sinal seja persistentemente excitante de ordem $2n$.

2.2.2 Experimentos Informativos

Na Seção 2.2.1 foi visto como caracterizar sinais que são suficiente informativos. Considere um conjunto de modelos \mathcal{M} para um sistema SISO descrito por (6) tendo a função de transferência $G(q, \theta)$ a função racional:

$$G(q, \theta) = \frac{B(q, \theta)}{F(q, \theta)} = \frac{q^{n_b}(b_1 + b_2 q^{-1} + \dots + b_{n_b} q^{-n_b+1})}{1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f}} \quad (14)$$

Um experimento em malha aberta é informativo se a sua entrada for persistentemente excitante. Observa-se que é necessário que a ordem de excitação seja igual ao número de parâmetros a serem estimados (LJUNG, 1999). No caso da equação apresentada em (14) a ordem de excitação necessária é $n_b + n_f$. Em (GEVERS et al., 2009) foram apresentadas as condições necessárias e suficientes para informatividade em estruturas de modelos arbitrárias.

Garantir a persistência de excitação para um sinal utilizado na identificação de um sistema é garantir que o sinal terá componentes de frequência suficientes para excitar o sistema ao ponto de ser possível observar sua dinâmica de forma

satisfatória. Inclui-se, desta forma, na identificação informações suficientes para que todos os parâmetros da classe de modelos possam ser identificados.

2.2.3 Escolha do modelo e sobre-modelagem

A escolha de um conjunto de modelos \mathcal{M} onde a ordem do modelo é maior do que a ordem do sistema real \mathcal{S} faz com que existam infinitas combinações de θ , resultando em infinitos modelos dentro do conjunto \mathcal{M} , que conseguem descrever exatamente o sistema \mathcal{S} .

Considere como exemplo o seguinte sistema real:

$$\mathcal{S}: \quad G_0(q) = \frac{a}{q-b}, \quad H_0(q) = \frac{1}{q-b} \quad (15)$$

e a estrutura de modelos:

$$\mathcal{M}: \quad G(q, \theta) = \frac{\theta_1(q - \theta_2)}{(q - \theta_3)(q - \theta_4)}, \quad H(q, \theta) = \frac{1}{q - \theta_5} \quad (16)$$

Observa-se que existem infinitos conjuntos de parâmetros θ que fazem com que a classe de modelos em (16) consiga representar o sistema real em (15):

$$\theta = [a \quad X \quad X \quad b \quad b] \quad (17)$$

ou

$$\theta = [a \quad X \quad b \quad X \quad b]$$

com X podendo assumir qualquer valor. Deve ficar claro que esta situação apresentada de sobre modelagem irá apresentar erros com relação ao sistema real, no momento que houverem mínimas diferenças entre os valores estimados de X no numerador e X no denominador.

A escolha de uma classe de modelos \mathcal{M} é o ponto mais crucial para que o processo de identificação de sistemas tenha sucesso. Esta escolha deve ser feita com base no entendimento do procedimento de identificação e das percepções e conhecimentos que se tem sobre o sistema a ser identificado (LJUNG, 1999).

O preço ou custo de uma classe de modelos pode ser mensurado em alguns aspectos e a escolha da classe de modelos se dará por alguma ponderação sobre estes critérios, que podem ser (LJUNG, 1999):

- Flexibilidade: Usando classes de modelos que tenham boas capacidades de descrever tipos diferentes de sistemas. Flexibilidade pode ser obtida usando-se vários parâmetros ou alocando-os em “posições estratégicas”.
- Parcimônia: Não usar desnecessariamente uma quantidade elevada de parâmetros.

- Complexidade do algoritmo: A complexidade de obter o erro de predição $\varepsilon(t, \theta)$ e as demais informações necessárias à identificação são fortemente correlacionadas com a escolha de \mathcal{M} e a sua ordem.
- Propriedades da função critério. Dependendo do formato da curva, o algoritmo para encontrar o mínimo pode ser mais custoso no que diz respeito a iterações matemáticas.
- O uso pretendido do modelo.

O uso de classes de modelos com diversos parâmetros só deve ser utilizado se modelos com menos parâmetros não passarem pelos testes de validação do modelo escolhido. Desta forma mantém-se a ideia de que quanto mais simples melhor, garantindo a parcimônia da classe escolhida e conseguindo cumprir com o uso pretendido do sistema.

Uma das primeiras justificativas para o desenvolvimento de métodos para a escolha de classes de modelos é a dificuldade em trabalhar com classes de modelos com muitos parâmetros, pelas complexidades adicionadas aos algoritmos e as incertezas matemáticas inseridas. Além disso existe o problema destes modelos serem numericamente mal condicionados além da convicção de que alguns parâmetros são redundantes e poderiam ser removidos do modelo (aplicando algum algoritmo de simplificação de modelo). Um modelo com um número excessivo de parâmetros pode exibir dinâmicas que não são observadas no sistema real. Desta forma não existe apenas o problema numérico para a identificação, mas também um problema de dinâmica em utilizar um modelo sobre dimensionado (AGUIRRE; JACOME, 1998).

O procedimento que testa se a classe de modelos é simples e apropriada para descrever o sistema é aplicar alguma técnica de redução à classe de modelos. Se a ordem da classe de modelos pode ser reduzida sem afetar as propriedades de entrada e saída do sistema, então a classe de modelos original era desnecessariamente complexa (LJUNG, 1999).

Por fim, ao escolher uma classe de modelos para a identificação, deve-se levar em conta diversos pontos e informações. Depois de escolhido o modelo, um processo de validação é indicado a ser executado. Este procedimento é simples e consiste em algumas perguntas que o projetista do experimento de identificação deve fazer:

- A classe de modelos escolhida consegue descrever suficientemente bem os dados observados?
- A classe de modelos é boa suficiente para meus objetivos?
- A classe de modelos descreve o sistema real (\mathcal{S}) ?

Se todas as perguntas forem justificadas, então o processo de escolha do modelo é satisfatório para o processo de identificação.

2.3 Estruturas clássicas de modelos

Algo importante a se destacar antes do processo de modelagem do sistema é a escolha do que deseja-se modelar deste sistema. Uma modelagem completa de todas as características é muitas vezes inviável e, na maioria dos sistemas reais, desnecessário. Usualmente, tem-se a necessidade de interagir, seja controlando ou observando, um conjunto restrito de informações do sistema, deve-se então focar o modelo nestas características desejadas.

2.3.1 Forma geral de classes de modelos

Modelo de um sistema é a descrição matemática de *algumas* de suas propriedades. Na Seção 2.1 apresentou-se a definição para um conjunto de modelos:

$$\mathcal{M} : \{G(q, \theta), H(q, \theta) | \theta \in D_{\mathcal{M}}\}$$

Resultando em um sistema que pode ser descrito como abaixo:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \quad (18)$$

onde as funções de transferência $G(q, \theta)$ e $H(q, \theta)$ são parametrizadas por θ e para cada combinação de θ um novo modelo é definido.

Provavelmente o modelo mais simples para descrever a relação entre entrada e saída de um sistema seja obtido descrevendo-o como uma equação linear (LJUNG, 1999):

$$y(t) + a_1y(t-1) + \dots + a_nay(t-n_a) = b_1u(t-1) + \dots + b_nbu(t-n_b) + e(t) \quad (19)$$

O ruído branco $e(t)$ aqui entra como um erro direto na equação. Os parâmetros ajustáveis neste caso são:

$$\theta = [a_1 \ a_2 \ \dots \ a_{na} \ b_1 \ \dots \ b_{nb}]^T \quad (20)$$

Definem-se os polinômios:

$$A(q, \theta) = 1 + a_1q^{-1} + \dots + a_naq^{-n_a}$$

$$B(q, \theta) = b_1q^{-1} + \dots + b_nbq^{-n_b}$$

O modelo definido em (19) também é conhecido como **ARX**, onde AR refere-se à parte de $A(q)y(t)$ auto-regressiva, e X como a entrada extra $B(q)u(t)$.

Uma das limitações deste modelo é a falta de liberdade para descrever as propriedades dos distúrbios sobre o sistema. Pode-se então adicionar certo grau de liberdade descrevendo a equação do erro como uma média móvel do ruído branco, o que remete a:

$$y(t) + a_1y(t-1) + \dots + a_nay(t-n_a) = b_1u(t-1) + \dots + b_nbu(t-n_b) + e(t) + k_1e(t-1) + \dots + k_nke(t-n_k) \quad (21)$$

Define-se o polinômio que filtra o ruído como:

$$K(q, \theta) = 1 + k_1q^{-1} + \dots + k_nkq^{-n_k}$$

e as funções de transferência como abaixo:

$$G(q, \theta) = \frac{B(q)}{A(q)}, \quad H(q, \theta) = \frac{K(q)}{A(q)} \quad (22)$$

tem-se então que os parâmetros à estimar são:

$$\theta = [a_1 \ a_2 \ \dots \ a_{na} \ b_1 \ \dots \ b_{nb} \ k_1 \ \dots \ k_{nk}]^T \quad (23)$$

O modelo definido em (21) também é conhecido como **ARMAX**, onde MA define a média móvel $K(q, \theta)$ do ruído $e(t)$. Na Figura 2 é apresentado o diagrama de blocos para o modelo **ARMAX**.

A partir do equacionamento do sistema apresentado em (22) e (1) pode-se facilmente generalizar para o equacionamento apresentado em:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{K(q)}{D(q)}e(t) \quad (24)$$

onde os polinômios $A(q), \dots, F(q)$ podem ser descritos como:

$$\begin{aligned} A(q, \theta) &= 1 + a_1q^{-1} + \dots + a_naq^{-n_a} \\ B(q, \theta) &= b_1q^{-1} + \dots + b_nbq^{-n_b} \\ K(q, \theta) &= 1 + k_1q^{-1} + \dots + k_nkq^{-n_k} \\ D(q, \theta) &= 1 + d_1q^{-1} + \dots + d_ndq^{-n_d} \\ F(q, \theta) &= 1 + f_1q^{-1} + \dots + f_nfq^{-n_f} \end{aligned}$$

Quando apenas um conjunto dos polinômios apresentados na equação (24) é utilizado, obtém-se diferentes tipos de estruturas de modelos. As mais conhecidas e utilizadas são apresentadas da Tabela 1.

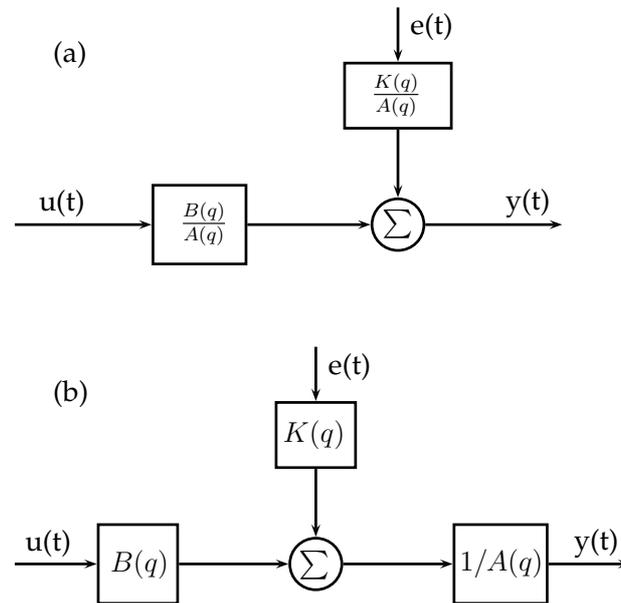


Figura 2: Diagrama de blocos do conjunto ARMAX de modelos. Ambos os diagramas são equivalentes.

Tabela 1: Alguns modelos comuns para sistemas SISO. Casos especiais da equação (24).

Polinômios diferentes de 1	Nome da estrutura do modelo
B	FIR (finite impulse response)
AB	ARX
ABK	ARMAX
AK	ARMA
ABD	ARARX
ABKD	ARARMAX
BF	OE (output error)
BFKD	Box-Jenkins

As equações que descrevem a saída do sistema em função da entrada $u(t)$ e do ruído $e(t)$ como apresentado em (1) e que podem ser caracterizados pela utilização de vários polinômios, nada mais são do que famílias ou conjuntos de modelos. Existe para cada conjunto de modelos uma infinidade de possíveis saídas para uma mesma entrada, bastando para isso que os parâmetros dos coeficientes dos polinômios sejam escolhidos apropriadamente.

2.4 Métodos de identificação

Identificação de sistemas é formada por três premissas básicas: o sistema real \mathcal{S} , a classe de modelos escolhida \mathcal{M} e o critério que determina qual é o melhor modelo dentro da classe \mathcal{M} para representar o sistema real \mathcal{S} .

Métodos de identificação de sistemas envolvem procedimentos, ou algoritmos, para encontrar o mínimo de uma função custo, conhecida também como critério. Na Seção 2.1 foi apresentado o critério de minimização mais utilizado para a identificação:

$$V(\theta) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} \varepsilon^2(t, \theta) \quad (25)$$

Nesta Seção serão apresentados métodos que minimizam esta função custo, suas propriedades e principais características.

2.4.1 Método dos mínimos quadrados

Existem diversos métodos para a estimativa de parâmetros. O mais conhecido remete ao ano de 1809 utilizado por Gauss para determinação da órbita dos planetas (SÖDERSTRÖM; STOICA, 1989), método este que veio a ser conhecido pelo nome de método dos mínimos quadrados (MMQ).

Define-se um preditor como em:

$$\hat{y}(t, \theta) = \varphi^T(t) \theta \quad (26)$$

onde $\hat{y}(t)$ é a predição, $\varphi(t)$ é comumente chamado de *variável de regressão* e θ é o vetor de parâmetros a ser identificado.

A partir do erro de predição apresentado em (7) e a equação (26) podemos redefinir o erro de predição como:

$$\varepsilon(t, \theta) = y(t) - \varphi^T(t) \theta$$

A estimativa dos mínimos quadrados é definido como o vetor $\hat{\theta}$ que minimiza a função custo (25). O valor de $\hat{\theta}$ que minimiza esta função custo é dado por (SÖDERSTRÖM; STOICA, 1989):

$$\hat{\theta} = (\varphi(t)\varphi^T(t))^{-1}\varphi(t)y(t) \quad (27)$$

Desta forma o mínimo da função custo fica como em:

$$\min_{\theta} V(\theta) = V(\hat{\theta}) = \frac{1}{N} [y - \varphi(\varphi\varphi^T)^{-1}\varphi y] \quad (28)$$

O método dos mínimos quadrados é simples de ser aplicado, mas tem o inconveniente de que para que não existam erros de polarização na estimativa, a variável de regressão $\varphi(t)$ não pode estar correlacionada com o distúrbio estocástico $\nu(t)$, ou seja:

$$E[\varphi(t)\nu(t)] = 0 \quad (29)$$

Assume-se que o sistema real é dado por:

$$y(t) = \varphi^T(t)\theta_0 + \nu(t) \quad (30)$$

Para que as estimativas de θ tendam a θ_0 quando o número de amostras tende a infinito é necessário que:

1. $E[\varphi(t)\varphi^T(t)]$ seja não singular.
2. $E[\varphi(t)\nu(t)] = 0$

Para a primeira condição algumas exceções são apresentadas (SÖDERSTRÖM; STOICA, 1989):

- Se a entrada não for persistentemente excitante.
- Os dados não são afetados por ruído ($\nu(t) \equiv 0$) e a ordem da classe de modelos é escolhida com um valor muito alto, com relação a ordem do sistema o que implica que $A(q)$ e $B(q)$ possuem fatores comuns.
- A entrada $u(t)$ é gerada por uma realimentação de saída fazendo com que os regressores sejam diretamente relacionados com o ruído por meio do sinal $y(t)$.

Quando o número de amostras tende ao infinito e a estimativa tende para um valor diferente de θ_0 , este erro é chamado de *bias* ou erro de polarização da estimativa. Isso ocorrerá quando $\nu(t)$ não for ruído branco. Para contornar este erro existem duas opções que serão abordadas em seguida: utilização de variáveis instrumentais e inclusão da modelagem do ruído na classe de modelos do sistema.

2.4.2 Método das variáveis instrumentais

Como foi apresentado anteriormente, o método dos mínimos quadrados é simples de ser aplicado mas carece de exatidão quando existe correlação entre a variável de regressão $\varphi(t)$ e o ruído estocástico $\nu(t)$, causando *bias* (erro de polarização). Nesta seção será apresentada uma breve discussão sobre um dos métodos que se propõe a sanar este inconveniente: método das variáveis instrumentais.

Seja $Z(t)$ uma matriz $n \times n$ que possui sinais não correlacionados com o distúrbio $\nu(t)$. O parâmetro θ deve obedecer a restrição da equação:

$$\frac{1}{N} \sum_{t=1}^N Z(t)\varepsilon(t) = \frac{1}{N} \sum_{t=1}^N Z(t) [y(t) - \varphi^T(t)\theta] = 0 \quad (31)$$

Se a dimensão da matriz $Z(t)$ for a mesma dimensão do vetor θ temos o estimador do método das variáveis instrumentais:

$$\hat{\theta} = \left[\sum_{t=1}^N Z(t)\varphi^T(t) \right]^{-1} \left[\sum_{t=1}^N Z(t)y(t) \right] \quad (32)$$

Os elementos da matriz $Z(t)$ são normalmente chamados de instrumentos. O estimador das variáveis instrumentais é uma generalização do estimador dos mínimos quadrados, quando $Z(t) = \varphi(t)$ (SÖDERSTRÖM; STOICA, 1989).

O estimador de variáveis instrumentais evita a polarização desde que o vetor de erro seja não correlacionado com as variáveis instrumentais. Esta condição é menos restritiva que a condição dos mínimos quadrados para que não haja erro de polarização. O custo a ser pago por isso envolve (AGUIRRE, 2004):

1. Escolha das variáveis instrumentais.
2. O estimador resultante é assintoticamente não polarizado, ao invés de ser apenas não polarizado. Ou seja, envolve a necessidade de um conjunto maior de dados para que a estimativa convirja para o valor final de θ .

Ao escolher as variáveis instrumentais é importante notar que a escolha não deve ser apenas para evitar a correlação entre o vetor de erro e os instrumentos. A razão para isso é que os instrumentos devem ser tão correlacionadas quanto possível com os regressores do modelo, caso contrário $Z(t)\varphi(t)^T$ seria próxima a singular e sua inversa muito mal condicionada. Portanto os instrumentos devem ser, idealmente, pouco correlacionados com o erro e muito correlacionados com os regressores do modelo (AGUIRRE, 2004).

A correlação dos instrumentos com os regressores ($E [Z(t)\varphi(t)^T]$) sempre será inferior à correlação dos mínimos quadrados ($E [\varphi(t)\varphi(t)^T]$) onde temos o regressor correlacionado consigo mesmo. Isso propicia que o uso de variáveis instrumentais trará uma estimativa com uma variância maior que o método dos mínimos quadrados, pois a variância da estimativa está relacionada com o inverso de $E [Z(t)\varphi(t)^T]$.

2.4.3 Modelagem do ruído

O método dos mínimos quadrados possui erro de polarização quando existe correlação entre o regressor $\varphi(t)$ e $\nu(t)$. Apresentou-se em seguida uma alternativa ao método dos mínimos quadrados que resolve este inconveniente ao custo da escolha de um instrumento que não seja correlacionado com o ruído: método das variáveis instrumentais (do inglês *intrumental variables - IV*).

Deve-se entretanto salientar que a escolha do instrumento não é o único inconveniente do método variáveis instrumentais. O aumento do erro de variância, como foi apresentado anteriormente, justifica a busca por outras alternativas para a estimativa dos parâmetros, almejando um erro de polarização nulo e com o menor erro de variância possível.

Desta forma, tem-se a possibilidade de incluir o modelo do ruído ao processo de identificação. A adição de $H(q, \theta)$ (com denominador diferente de $G(q, \theta)$) à identificação traz a vantagem de que quando a classe de modelos \mathcal{M} consegue representar a totalidade das dinâmicas contidas no sistema real \mathcal{S} então o erro de polarização é zerado. Por questões de simplicidade de implementação e utilização, deseja-se que seja possível a utilização do método dos mínimos quadrados para a realização da estimativa, o que só é possível quando o denominador de $G(q, \theta)$ e $H(q, \theta)$ são iguais. No caso da estrutura de modelos completa apresentada em (24) e replicado aqui:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

o sistema não é linearmente parametrizável quando todos os polinômios são diferentes do caso unitário ($A(q), \dots, F(q) \neq 1$) e com isso a função custo não é convexa, adicionando um grau considerável de complexidade aos algoritmos para encontrar o mínimo global da equação (SÖDERSTRÖM; STOICA, 1989).

2.5 Propriedades estatísticas da estimativa

Nesta seção tem-se o objetivo de descrever algumas das propriedades estatísticas relevantes para a identificação de sistemas. Tem-se o intuito de formalizar quais são os erros que podem estar contidos nas estimativas obtidas e quais são as

condições para evitá-las, ou reduzi-las.

Conhecer a qualidade das estimativas que são obtidas de um processo de identificação e suas propriedades estatísticas são de suma importância para aplicações de engenharia. O conhecimento do valor de θ sem a informação de quão confiável esta informação é pouco útil em muitas aplicações.

2.5.1 Conjunto de dados

O conjunto de dados coletados do sistema pode ser descrito como em:

$$Z^N = \{u(1), y(1), \dots, u(N), y(N)\} \quad (33)$$

e é o ponto de partida básico para a identificação de sistemas. As propriedades calculadas de $\hat{\theta}_N$ para um certo montante de dados serão levadas junto para quando $N \rightarrow \infty$, é natural que as condições sobre os dados, sejam relativos a um conjunto infinito Z^∞ (LJUNG, 1999).

Assume-se que o conjunto de dados Z^∞ é tal que para alguns filtros da forma $\{d_t^{(i)}(k)\}$:

$$\begin{aligned} y(t) &= \sum_{k=1}^{\infty} d_t^1(k)r(t-k) + \sum_{k=0}^{\infty} d_t^2(k)e_0(t-k) \\ u(t) &= \sum_{k=0}^{\infty} d_t^3(k)r(t-k) + \sum_{k=0}^{\infty} d_t^4(k)e_0(t-k) \end{aligned} \quad (34)$$

onde:

1. $\{r(t)\}$ é uma entrada externa, determinística e limitada em amplitude.
2. $\{e_0(t)\}$ é uma sequência randômica independente com média zero, limitada em amplitude e momentos com ordem $4 + \delta$ para algum $\delta > 0$.

Além disso:

3. a família de filtros $\{d_t^{(i)}(k)\}_{k=1}^{\infty}$, $i = 1 - 4$; $t = 1, 2, \dots$ é uniformemente estável.
4. os sinais $\{y(t)\}$ e $\{u(t)\}$ são juntamente quasi-estacionários.

2.5.2 Consistência e identificabilidade

Suponha que o sistema real possa ser descrito por \mathcal{S} (equação (5)) e o modelo \mathcal{M} seja como apresentado em (6) e além disso que $\mathcal{S} \in \mathcal{M}$. Os resultados básicos de consistência são quase imediatos e consistem em determinar que quando o sistema real consegue ser representado pela classe de modelos, então não haverá erro de polarização nas estimativas obtidas. O teorema a seguir apresenta estas conclusões (LJUNG, 1999):

Teorema 2.2. *Suponha que o conjunto de dados Z^∞ está sujeito ao que foi apresentado na Seção 2.5.1 e que $\mathcal{S} \in \mathcal{M}$. Assume-se que Z^∞ é suficientemente informativo com respeito a \mathcal{M} . Se o sinal de entrada contém a saída por realimentação então também se assume que exista um atraso de tempo tanto em $G_0(q)$ quanto em $G(q, \theta)$. Então:*

$$D_c = D_T(\mathcal{S}, \mathcal{M}) \quad (35)$$

onde

$$D_c = \arg \min_{\theta \in D_{\mathcal{M}}} V(\theta) = \left\{ \theta \mid \theta \in D_{\mathcal{M}}, V(\theta) = \min_{\theta' \in D_{\mathcal{M}}} V(\theta') \right\} \quad (36)$$

e

$$D_T(\mathcal{S}, \mathcal{M}) = \{ \theta \in D_{\mathcal{M}} \mid G(e^{j\omega}, \theta) = G_0(e^{j\omega}); \\ H(e^{j\omega}, \theta) = H_0(e^{j\omega}); -\pi \leq \omega \leq \pi \}$$

Se além disso, a classe de modelos é globalmente identificável em $\theta_0 \in D_{\mathcal{M}}$, então:

$$D_c = \{ \theta_0 \} \quad (37)$$

Desta forma, e como:

$$\hat{\theta}_N \rightarrow D_c, \text{ quando } N \rightarrow \infty \quad (38)$$

tem-se que as funções de transferência obedecem a:

$$G(e^{j\omega}, \hat{\theta}_N) \rightarrow G_0(e^{j\omega}), \quad (39)$$

$$H(e^{j\omega}, \hat{\theta}_N) \rightarrow H_0(e^{j\omega}), \text{ quando } N \rightarrow \infty$$

2.5.3 Incertezas nas estimativas dos parâmetros

Um dos principais desejos para a identificação de sistemas é que este não possua erros, ou ao menos que este erro, se existir, seja o menor possível e limitado dentro de uma região conhecida. Para tanto é necessário introduzir os conceitos de *erro de variância* e *erro de polarização*.

Seja θ^* o limite de convergência para a minimização do erro de predição:

$$\lim_{N \rightarrow \infty} \hat{\theta}_N = \theta^* \quad (40)$$

e sejam os vetores:

$$Q_0(q) = [G_0(q) \quad H_0(q)]^T$$

$$\hat{Q}_N(q) = [G(q, \hat{\theta}_N) \quad H(q, \hat{\theta}_N)]^T$$

A qualidade da estimativa pode ser calculada em termos da diferença entre o processo real $Q_0(q)$ e o melhor sistema que o modelo pode atingir (quando a quantidade de dados é ilimitada ($N \rightarrow \infty$)) (CAMPESTRINI, 2010; SOLARI, 2005).

$$\Delta Q_N(q) \equiv \hat{Q}_N(q) - Q_0(q) \quad (41)$$

Define-se então:

$$Q^*(q) = [G(q, \theta^*) \quad H(q, \theta^*)]^T \quad (42)$$

Entende-se por $Q^*(q)$ como a melhor aproximação que o método pode proporcionar quando a quantidade de dados coletados $N \rightarrow \infty$. Adicionando e subtraindo a equação (42) na equação (41) chega-se à definição dos erros de polarização e de variância:

$$\Delta Q_N(q) \equiv \underbrace{\hat{Q}_N(q) - Q^*(q)}_{\text{Erro de variância}} + \underbrace{Q^*(q) - Q_0(q)}_{\text{Erro de polarização}} \quad (43)$$

A partir da equação (43) compreende-se o erro de polarização como sendo a distância entre a melhor aproximação possível e o valor real para o parâmetro. Já o erro de variância é a média que cada uma das estimativas está distante do valor ótimo possível para a estimativa, que é tão menor quanto mais dados forem usados para a identificação.

Seguidamente é mais importante ter uma boa estimativa da função de transferência da planta $G(\cdot)$ do que do filtro do ruído $H(\cdot)$ (LJUNG, 1999). Suponha que o conjunto de modelos:

$$\mathcal{G} = \{G(e^{j\omega}, \theta) \mid \theta \in D_{\mathcal{M}}\}$$

é abrangente o suficiente para que

$$G_0(q) \in \mathcal{G} \quad (44)$$

mas a função de transferência do filtro real do ruído $H_0(q)$ não consegue ser totalmente descrita pela classe de modelos. Desta forma $\mathcal{S} \notin \mathcal{M}$ e supondo que o conjunto de dados Z^∞ esteja sujeito ao que foi descrito na Seção 2.5.1 e que este seja suficientemente informativo. Se o experimento é realizado em malha aberta, ou seja, quando $u(t)$ e $e(t)$ são descorrelacionados e se $G(q, \theta)$ e $H(q, \theta)$ são parametrizados de forma independente:

$$\theta = \begin{bmatrix} \rho \\ \eta \end{bmatrix}, \quad G(q, \theta) = G(q, \rho), \quad H(q, \theta) = H(q, \eta) \quad (45)$$

então $G_0(q)$ pode ser estimada de forma consistente, independente de $H_0(q)$ pertencer ou não à classe de modelos (LJUNG, 1999). Neste caso, o erro entre o processo real e o modelo estimado é dado pelos erros de polarização e de variância da estimativa (GEVERS, 2006; CAMPESTRINI, 2010), onde:

$$\begin{aligned} \text{Erro de variância} &= G(q, \hat{\theta}_N) - G(q, \theta^*) \\ \text{Erro de polarização} &= G(q, \theta^*) - G_0(q) \end{aligned}$$

O caso da parametrização independente em (45) cobre o caso do *Output Error* apresentado na Tabela 1 além de modelos com estruturas fixas para o filtro do ruído $H(q, \theta)$ como em:

$$y(t) = G(q, \theta)u(t) + H_*(q)e(t)$$

onde $H_*(q)$ é um filtro do ruído fixo, onde $\mathcal{S} \notin \mathcal{M}$.

Este modelo também cobre o caso da estrutura de modelos Box-Jenkins (Tabela 1). Consequentemente esta estrutura de modelos tem a vantagem de que a função de transferência $G(q, \theta)$ pode ser consistentemente estimada, mesmo quando o modelo do filtro do ruído é muito simples para admitir-se uma completa descrição do sistema (LJUNG, 1999).

2.5.4 Distribuição assintótica das estimativas dos parâmetros

Um meio comum de medir a qualidade das estimativas é estudar suas propriedades assintóticas. Quando o valor da quantidade de dados N cresce muito, a estimativa pertencerá a alguma distribuição. As propriedades desta irão determinar a qualidade das estimativas obtidas (JANSSON, 2004).

Uma vez tendo entendido as propriedades de convergência de $\hat{\theta}_N$, a pergunta que surge é quão rápido esta estimativa chega ao limite. Se θ^* denota o limite, estamos interessados na variável $\hat{\theta}_N - \theta^*$ que é uma variável randômica, e seu “tamanho” pode ser caracterizado pela sua matriz de covariância, ou pela sua distribuição probabilística. $\hat{\theta}_N - \theta^*$ tipicamente decai a uma taxa $1/\sqrt{N}$. Então a distribuição da variável randômica:

$$\sqrt{N}(\hat{\theta}_N - \theta^*)$$

irá convergir para uma distribuição bem definida Gaussiana (LJUNG, 1999).

Assume-se que dentro do conjunto $D_{\mathcal{M}}$ exista apenas um ponto de convergência θ^* . Em (40) foi apresentado que $\hat{\theta}_N \rightarrow \theta^*$ quando $N \rightarrow \infty$ e considerando que :

$$\sqrt{N}D_N \rightarrow 0, \text{ quando } N \rightarrow \infty$$

onde D_N pode ser definido como:

$$D_N = E \left[\frac{1}{N} \sum_{t=1}^N [\psi(t, \theta^*)\varepsilon(t, \theta^*) - E\psi(t, \theta^*)\varepsilon(t, \theta^*)] \right] \quad (46)$$

e

$$\psi(t, \theta^*) = -\frac{d}{d\theta}\varepsilon(t, \theta) |_{\theta=\theta^*} = \frac{d}{d\theta}\hat{y}(t | \theta) |_{\theta=\theta^*}$$

Então tem-se que:

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \rightarrow \mathcal{N}(0, P(\theta)) \text{ quando } N \rightarrow \infty \quad (47)$$

onde $\mathcal{N}(0, P(\theta))$ denota uma distribuição normal e $P(\theta)$ é dado por:

$$P(\theta^*) \triangleq \lim_{N \rightarrow \infty} N E(\hat{\theta}_N - \theta^*)(\hat{\theta}_N - \theta^*)^T \quad (48)$$

Para o caso onde $\mathcal{S} \in \mathcal{M}$ e assumindo ainda que $\varepsilon(t, \theta_0) = e_0(t)$ é uma sequência randômica independente de média zero e variância λ_0 . Desta forma tem-se

$$P(\theta) = \lambda_0 [E \psi(t, \theta_0)\psi^T(t, \theta_0)]^{-1} \quad (49)$$

Assim quando o sistema real pertence à classe de modelos escolhida a estimativa converge para θ_0 e a covariância da estimativa é dada por (LJUNG, 1999):

$$\text{Cov}\hat{\theta}_N \sim \frac{1}{N}P(\theta) \quad (50)$$

Não é apenas o tamanho (N) do experimento que irá influenciar na qualidade da estimativa. Introduz-se o espectro:

$$\Phi_{\chi_0} = \begin{bmatrix} \Phi_u & \Phi_{ue} \\ \Phi_{ue} & \lambda_0 \end{bmatrix} \quad (51)$$

onde Φ_u é o espectro da entrada relativo ao sinal $u(t)$, Φ_{ue} é o espectro cruzado entre os sinais $u(t)$ e $e(t)$. A distribuição da frequência do espectro Φ_{χ_0} também influenciará na qualidade da estimativa, como pode ser visto no Lemma 2.1.

Lemma 2.1. *A inversa da matriz de covariância, $P^{-1}(\theta_0)$, é uma função linear do espectro Φ_{χ_0} dada por:*

$$P^{-1}(\theta_0) = \frac{1}{2\pi\lambda_0} \int_{-\pi}^{\pi} \mathcal{F}(\theta_0)\Phi_{\chi_0}(\theta_0)\mathcal{F}^*(\theta_0)d\omega \quad (52)$$

onde $\mathcal{F}(q, \theta_0) = [\mathcal{F}_u(q, \theta_0) \quad \mathcal{F}_e(q, \theta_0)]$ e:

$$\mathcal{F}_u(\theta_0) = H^{-1}(\theta_0) \frac{dG(\theta_0)}{d\theta}$$

$$\mathcal{F}_e(\theta_0) = H^{-1}(\theta_0) \frac{dH(\theta_0)}{d\theta}$$

Como $P(\theta)$ é a medida do tamanho do erro nos parâmetros, o Lemma 2.1 mostra como este erro é relacionado com o espectro Φ_{χ_0} e especialmente com o espectro de entrada Φ_u e o espectro cruzado Φ_{ue} . É interessante perceber que as únicas informações capazes de moldar a covariância $P(\theta)$ são exatamente os espectro de entrada e cruzado, todos os outros fatores são dependentes do desconhecido sistema real. Outro ponto a observar é que o espectro cruzado Φ_{ue} é zero quando o sistema esta operando em malha aberta (JANSSON, 2004).

2.5.4.1 Margens de confiança das estimativas

A partir da distribuição assintótica normal em (48) segue:

$$(\hat{\theta}_N - \theta_0)^T P_N^{-1}(\hat{\theta}_N - \theta_0) \rightarrow \chi^2(n) \text{ quando } N \rightarrow \infty$$

Com $P_N = P/N$ e onde:

$$U_\theta = \left\{ \theta \mid (\hat{\theta}_N - \theta_0)^T P_N^{-1}(\hat{\theta}_N - \theta_0) \leq \chi_\alpha^2(n) \right\} \quad (53)$$

é a região de confiança, que assintoticamente inclui θ_0 com probabilidade α . Desta forma tem-se o valor real θ_0 com probabilidade α estará contido no elipsóide U_θ (JANSSON, 2004).

Na Figura 3 é apresentado um exemplo de região de confiança para um χ de 99%. O ponto em destaque é a média de todas as estimativas e está localizado no centro da região de confiança.

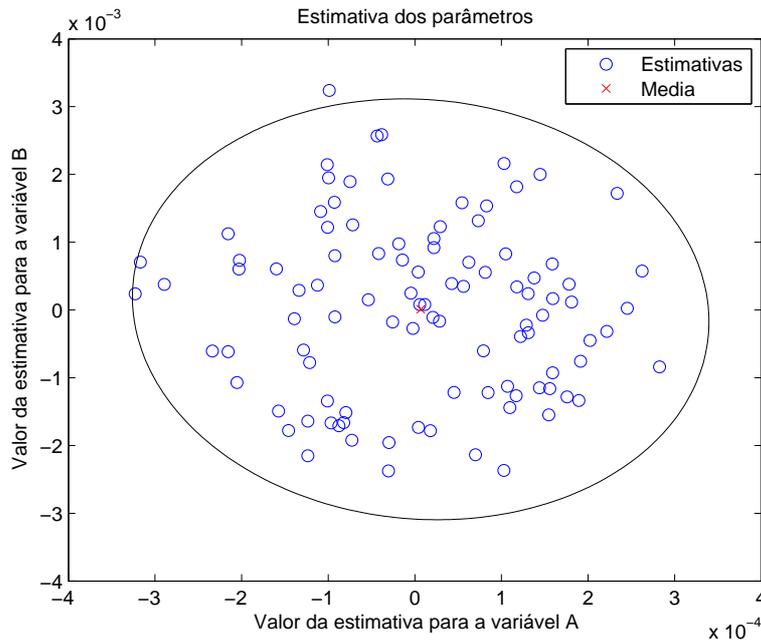


Figura 3: Estimativas de um sistema e a elipse representando a região de confiança para um χ de 99%. No centro da elipse destaca-se a média de todas as estimativas.

2.6 Projeto de Experimento

Projeto de experimentos pode ser entendido como o procedimento para que se escolha o melhor sinal de entrada para a identificação dos parâmetros desejados para o experimento. Desta forma muitas variáveis podem ser levadas em consideração, refletindo em propriedades que podem ou não ser o foco do projeto de experimentos.

Uma forma de organizar o projeto de experimentos é desenvolvê-lo como um problema de otimização convexa, onde entre muitas vantagens está o fato de que é possível a utilização de métodos matemáticos para o cálculo e sua formulação pode ser feita por LMIs (*Linear Matrix Inequality*). Em (JANSSON, 2004) este tópico é explorado em mais profundidade, sendo aqui apenas apresentada a sua ideia básica.

A escolha de um sinal mais apropriado para o experimento traz diversas vantagens, que podem ser bastante significativas para o tempo e esforço despendido sobre o projeto do controlador, ou da identificação do sistema. Uma destas vantagens é o tempo de dados coletados; aplicando-se sinais com componentes de frequência que são mais informativos, tem-se uma eficiência maior nos dados amostrados, bastando um montante menor de dados, para que sejam obtidos os mesmos índices

de qualidade.

2.6.1 Sinais de entrada mais comumente utilizados

Vários sinais que podem ser gerados de maneira fácil e que possuem um grande número de componentes de frequência foram elaborados.

2.6.1.1 Sinal Binário Pseudo-Randômico - PRBS

Um *sinal binário pseudo-randômico* é um sinal periódico com algumas propriedades semelhantes à do ruído branco. Este sinal é gerado pela equação:

$$u(t) = \text{rem}(A(q)u(t), 2) = \text{rem}(a_1u(t-1) + \dots + a_nu(t-n), 2) \quad (54)$$

onde $\text{rem}(x, 2)$ é o resto inteiro da divisão de x por 2. O sinal $u(t)$ deve ser periódico de pelo menos 2^n valores diferentes. Como uma sequência de zeros não é um sinal válido, por ser nulo, temos que o máximo período é de tamanho $M = 2^n - 1$. Na verdade o período vai depender da escolha de $A(q)$. Pode-se entretanto mostrar que para cada n existem escolhas de $A(q)$ que proporcionam o tamanho máximo. Tais escolhas são apresentadas na Tabela 2 (LJUNG, 1999).

Tabela 2: Polinômios $A(q)$ que geram o máximo período M para sinais PRBS de ordem n . $a_k = 1$ para os k indicados, 0 para os demais. Diversas outras escolhas existem para os mesmos n .

Ordem n	$M = 2^n - 1$	a_k não zeros para k
2	3	1, 2
3	7	2, 3
4	15	1, 4
5	31	2, 5
6	63	1, 6
7	127	3, 7
8	255	1, 2, 7, 8
9	511	4, 9
10	1023	7, 10
11	2047	9, 11

O espectro de um sinal PRBS é dado por:

$$\Phi_u(\omega) = \frac{2\pi\bar{u}^2}{M} \sum_{k=1}^{M-1} \delta(\omega - 2\pi k/M), \quad 0 \leq \omega < 2\pi \quad (55)$$

Observa-se que este é um sinal persistentemente excitante de ordem $M - 1$. Na Figura 4 é apresentado um exemplo de sinal PRBS para $n = 7$.

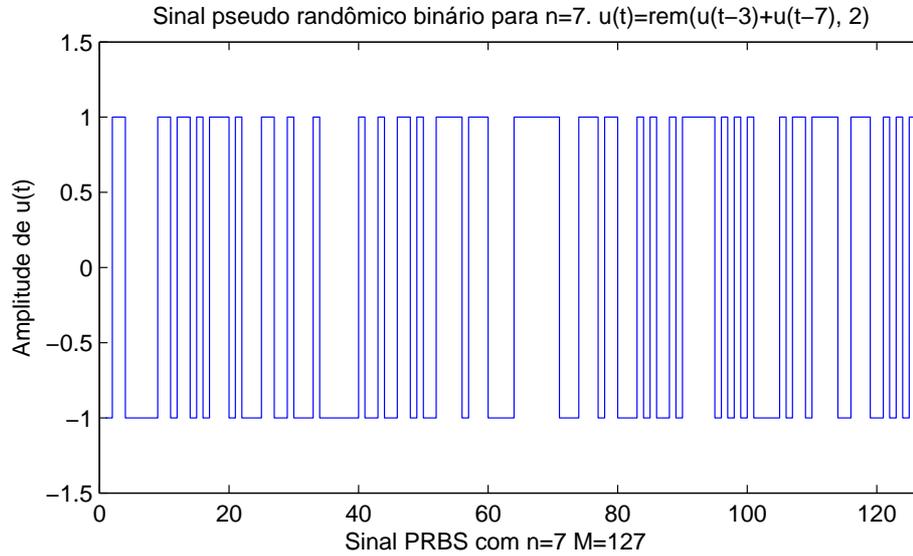


Figura 4: Sinal PRBS para $n = 7$

2.6.1.2 Ruído Branco Filtrado

Um das escolhas mais simples de sinais, é gerar um ruído Gaussiano e então filtrá-lo com algum filtro linear. Desta forma, teoricamente, é possível atingir qualquer espectro de sinal, bastando apenas a correta escolha do filtro. Como este sinal é gerado *off-line*, é possível aplicar filtros não causais e eliminar efeitos transientes do sinal, o que proporciona um comportamento espectral melhor (CAMPESTRINI, 2010; LJUNG, 1999).

2.6.2 Projeto de experimento

O problema de projeto de experimento pode ser considerado como uma forma geral apresentada em:

$$\begin{array}{ll}
 \text{minimize} & \text{Objetivo} \\
 \Phi_{x_0} & \\
 \text{Sujeito a:} & \text{Requisitos de qualidade} \\
 & \text{Requisitos de sinais}
 \end{array} \tag{56}$$

De forma geral os requisitos de qualidades são funções da covariância de P . Por esta razão é natural usar o espectro da entrada Φ_u e eventualmente o espectro cruzado Φ_{ue} como variáveis do projeto. A inclusão de limitações nos sinais e sua

inclusão como variáveis de projeto são úteis para evitar que se chegue em resultados onde a energia de entrada precise ser infinita para se obter os critérios desejados, ou largura de banda que não são facilmente atingíveis em projetos reais.

Típicos projetos de experimentos são intratáveis em sua forma original, pelos seguintes motivos (JANSSON, 2004):

- I Algumas restrições são tipicamente não convexas.
- II Existem restrições que são de dimensão infinita.
- III Existe ainda o problema de encontrar um sinal realizável que tenha as propriedades desejadas para o espectro.
- IV A variância assintótica depende tipicamente dos parâmetros do sistema θ_0 , $P = P(\theta_0)$ que são desconhecidos.

Os primeiros 3 itens listados acima são contornáveis pela inserção de uma parametrização finita do espectro de entrada e algumas vezes até do espectro cruzado, tornando o problema convexo, que pode ser tratado de forma bem mais fácil.

O último item onde a solução do problema depende das características do sistema a ser identificado é intrínseco de quase todos os problemas de otimização, sendo então inevitável. Em problemas reais este fato pode ser contornado utilizando alguns métodos conhecidos (JANSSON, 2004).

A formulação de um projeto de experimento pode ser particionado nas seguintes partes:

- *Parametrização do espectro*

A escolha de usar uma parametrização finita do espectro ou uma parametrização parcial da correlação é regido pelos seguintes aspectos:

- Relativos a otimização
- Computacionais
- Limitação de sinais
- Robustez

A parametrização parcial da correlação é ótima e pode usar um número mínimo de parâmetros, utilizando assim, menos processamento computacional. Entretanto certas limitações de sinais não podem ser garantidas nesta situação, e a parametrização pode depender do sistema real. Parametrização finita do espectro geralmente não atinge um mínimo global mas as funções bases não precisam ser funções do sistema real. Este método pode gerenciar problemas de limitações de sinal de frequência por frequência.

- *Restrições de qualidade*

Assumindo que erros de variância são de grande importância é natural que qualquer medida de qualidade do modelo leve em conta a matriz de covariância P . Esta matriz pode ser manipulada pelos espectros de entrada Φ_u e o espectro cruzado Φ_{ue} como apresentado em (57).

$$P^{-1}(\theta_0) = \frac{1}{2\pi\lambda_0} \int_{-\pi}^{\pi} \mathcal{F}(e^{j\omega}, \theta_0) \begin{bmatrix} \Phi_u(\omega) & \Phi_{ue}(\omega) \\ \Phi_{ue}^0(\omega) & \lambda_0 \end{bmatrix} \mathcal{F}(e^{j\omega}, \theta_0) d\omega \quad (57)$$

O principal desafio neste ponto é tornar as restrições convexas em P^{-1} .

- *Restrições de sinal*

Podem ser incluídos, restrições de energia para os sinais de entrada e saída do sistema bem como restrições no sinal dependentes da frequência deste.

- *Restrições de robustez*

A solução de boa parte dos problemas de otimização recaem na necessidade de se conhecer o sistema real. Isso nem sempre é possível e uma das formas de contornar este problema é substituir o sistema real por uma aproximação deste. Devido aos erros de estimativa deste sistema, tem-se a necessidade de se utilizar métodos que garantam robustez deste sistema, para que quando submetido ao sistema real, o projeto ainda seja válido.

2.7 Considerações Finais

Neste capítulo foram introduzidos os principais conceitos para a identificação de sistemas LTI de tempo discreto. A identificação de sistemas contém três elementos fundamentais: o sistema real \mathcal{S} e o conjunto de dados coletados deste sistema Z^N , a classe de modelos \mathcal{M} escolhida para representar o sistema real e o critério que elenca qual é o modelo dentro da classe que melhor descreve o sistema real.

De posse do conjunto de dados, da classe de modelos, e do critério, apresentou-se o método mais comumente utilizado para identificação: Método dos mínimos quadrados. Foram apresentadas suas características e suas propriedades. Devido às incertezas da estimativa dos parâmetros quando o regressor está correlacionado com o ruído, foram introduzidos métodos alternativos a este para sanar este problema. Foram apresentados para isso os métodos das variáveis instrumentais e também o método PEM onde leva-se em conta a modelagem do ruído na identificação.

Apresentou-se as propriedades estatísticas das estimativas obtidas e quais são os erros esperados para situações onde a classe de modelos \mathcal{M} consegue representar

o sistema real \mathcal{S} e quando $\mathcal{S} \notin \mathcal{M}$. Em adição a estas propriedades discutiu-se quais são as condições que propiciam erro de polarização e quais influenciam no erro de variância das estimativas. Introduziu-se também o conceito de nível de confiança baseado nas características do experimento realizado.

Ao fim foram apresentados os principais aspectos de um projeto de identificação de sistemas. Partindo-se da escolha do sinal de excitação para o experimento onde pode-se determinar o que é de fundamental importância para a identificação e focar os esforços para reduzir ao máximo erros nestes aspectos. Apresentou-se neste quesito de projeto de experimento, a ideia de tornar a escolha do sinal, um problema de otimização, onde as restrições são as margens de qualidade que desejassem para as propriedades assintóticas da estimativa, limitações de sinais e margens de robustez.

É perceptível que a identificação de um sistema é uma tarefa que depende de um conjunto relativamente grande de fatores. De um lado isso é extremamente interessante, pois tem-se liberdade de escolha e a possibilidade de focar o projeto em pontos que são mais interessantes e importante, em detrimento de outros que não são relevantes para o sistema em questão. Por outro lado o conjunto de variáveis que, se não compreendidas corretamente, podem tornar este processo oneroso e por vezes até ineficiente.

3 IDENTIFICAÇÃO DE SISTEMAS NÃO LINEARES DE TEMPO DISCRETO

Anteriormente foram introduzidos conceitos básicos de identificação de sistemas lineares invariantes no tempo. Este capítulo tem como objetivo apresentar os principais conceitos para identificação de sistemas não lineares de tempo discreto.

Na Seção 3.1 serão apresentados conceitos básicos sobre não linearidades, algumas das principais não linearidades encontradas e uma breve descrição de suas características.

Na Seção 3.2 serão apresentados alguns modelos tipicamente utilizados para caracterização de sistemas não lineares. Na Seção 3.2.5 será dada ênfase aos modelos racionais e na Seção 3.3.1 será apresentado um algoritmo para a identificação de sistemas não lineares quando descritos por esta estrutura de modelos. No Apêndice A são apresentadas as rotinas desenvolvidas em Matlab para o desenvolvimento deste algoritmo.

Ao fim, serão apresentadas breves conclusões sobre identificação de sistemas não lineares (Seção 3.4).

3.1 Conceitos básicos

Muitos, se não todos sistemas reais são não lineares. Existe entretanto um grande número destes sistemas que podem, e assim o são, representados por sistemas lineares em determinadas faixas de operação. Os sistemas desta forma conseguem representar satisfatoriamente o sistema observado. Todavia alguns sistemas para serem linearizados exigem uma faixa de atuação muito estreita, fazendo com que o sistema em sua operação normal já saia desta faixa, tornando o modelo linear não representativo para aquele sistema em questão.

Tem-se desta forma uma das justificativas de escolher um modelo não linear para caracterizar o sistema. Ganha-se de um lado no quesito de faixa de atuação mais ampla e perde-se por abdicar da simplicidade de sistemas lineares.

3.1.1 Tipos de não linearidades

Existem duas limitações básicas para sistemas linearizados. Primeiro que a linearização é uma aproximação ao redor do ponto de operação, logo ele pode prever apenas o comportamento nesta localidade do sistema e não o comportamento global. Segundo, as dinâmicas de sistemas não lineares são muito mais ricas que as de sistemas lineares. Existem portanto alguns fenômenos essenciais, não lineares, que não conseguem ser descritos ou preditos por modelos lineares, estes são (KHALIL, 1996):

- Tempo de fuga finito

O estado de um sistema linear instável tende ao infinito na medida que o tempo tende ao infinito. Um sistema não linear instável, entretanto, pode divergir para o infinito em um tempo finito.

- Múltiplos pontos de equilíbrios isolados

Um sistema linear assintoticamente estável tem apenas um ponto de equilíbrio. Desta forma, este sistema tem apenas um ponto do espaço de estados que atrai o estado do sistema, independentemente da condição inicial destes estados. Um sistema não linear pode ter mais de um ponto de equilíbrio isolado. Desta forma o estado do sistema pode convergir para um destes pontos de equilíbrios ou para outro, dependendo do estado inicial.

- Ciclos limites

Para um sistema linear e invariante no tempo oscilar ele deve ter um par de autovalores sobre o eixo imaginário, o que é uma condição não robusta praticamente impossível de manter na presença de perturbações. Mesmo que isso seja possível de manter, a amplitude da oscilação dependerá das condições iniciais do sistema. Na vida real, oscilações estáveis são atingidas apenas com sistemas não lineares. Alguns sistemas possuem oscilações com amplitude e frequência constantes, independente das condições iniciais. Chama-se isso de ciclos limites.

- Sub harmônicas, harmônicas ou oscilações quase periódicas

Um sistema linear estável submetido a uma entrada senoidal periódica produz uma saída senoidal na mesma frequência. Sistemas não lineares alimentados por sinais periódicos podem oscilar com frequências que podem ser sub múltiplos ou múltiplos da frequência de entrada.

- Caos

Um sistema não linear pode ter um espaço de estados mais complexo e seu comportamento não pode ser descrito por equilíbrio, oscilações periódicas

ou quase periódicas. Este comportamento normalmente é conhecido como caos. Alguns destes movimentos caóticos exibem comportamento randômico, independentemente da natureza do sistema.

3.2 Estruturas de modelos para sistemas não lineares

Famílias ou conjuntos de modelos para sistemas não lineares podem ser divididos em dois grupos principais baseados na natureza de suas não linearidades: não linearidades estáticas, onde a dinâmica do sistema pode ser bem caracterizada por um modelo linear enquanto que a parte não linear está concentrada ou na entrada ou na saída do sistema de forma estática. Estes sistemas são chamados de Wiener para não linearidades na saída do processo ou Hammerstein, quando a não linearidade está na entrada do processo.

Para os demais casos, onde a não linearidade está na dinâmica do processo existem várias estruturas de modelos que podem ser utilizados como será visto no decorrer deste capítulo. De forma geral todos estes conjuntos possuem em comum a ideia da escolha de uma base que seja representativa e que possa reduzir a quantidade de termos a ser identificado e com isso ter uma boa aproximação do sistema real com a menor quantidade de parâmetros possível.

Percebe-se então que um mesmo sistema pode ser representado por diversos destes modelos, mas dependendo das características deste sistema, uma das estruturas de modelos será mais adequada, por sua base possuir mais afinidade em representar certos tipos de não linearidades. Entende-se por “maior afinidade” a característica de conseguir representar o sistema com um número de parâmetros menor que outro modelo que use outra base.

Um dos passos mais desafiadores na construção de modelos não lineares é a escolha da estrutura de modelo. Quando este modelo é não linear, existe uma grande quantidade de opções e com isso o perigo de escolher um modelo desnecessariamente complexo é evidente. Este perigo baseia-se no *princípio da parcimônia* que basicamente determina que o modelo deve ser o mais simples possível (AGUIRRE, 1997). Na Seção 2.2.3 foram apresentadas algumas ideias relacionadas a este assunto.

Juntamente com a escolha do modelo para representar o sistema não linear, saber se este é identificável é de grande importância. Em (GLAD; LJUNG, 1990) é apresentado um procedimento para determinar a identificabilidade de sistemas não lineares.

3.2.1 Modelos de Wiener e Hammerstein

Modelos de Wiener e Hammerstein são normalmente utilizados em situações onde a dinâmica do sistema pode ser bem descrita por um sistema linear, mas

existem algumas não linearidades estáticas atreladas à entrada e/ou à saída. Este será o caso de atuadores não lineares com características como saturação, zona morta, entre outros. Um modelo com não linearidades na entrada é chamado de *modelo de Hammerstein* e para não linearidades na saída chama-se *modelo de Wiener* (LJUNG, 1999). Na Figura 5 observa-se o diagrama de blocos para os modelos de Hammerstein e Wiener.

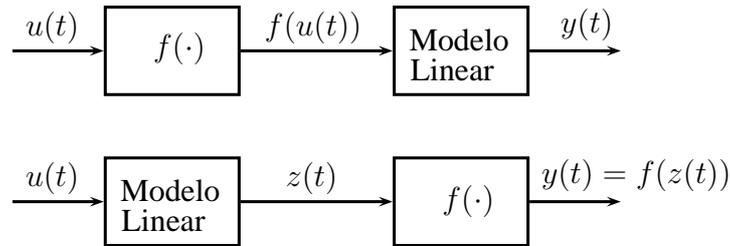


Figura 5: Acima: modelo de Hammerstein. Abaixo: Modelo de Wiener.

3.2.2 Série de Volterra

Um sistema não linear pode ser descrito pela série de Volterra:

$$y(t) = \sum_{j=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_j(\tau_1, \dots, \tau_j) \prod_{i=1}^j u(t - \tau_i) d\tau_i \quad (58)$$

onde h_j são generalizações não lineares da resposta ao impulso $h_1(t)$. Quando $j = 1$ a equação de Volterra se reduz à integral de convolução (AGUIRRE, 2004).

As expansões funcionais em séries de Volterra relacionam os sinais passados da entrada com o valor atual da saída do sistema e isso inevitavelmente significa que um conjunto tipicamente elevado de parâmetros seja necessário para descrever até mesmo simples sistemas não lineares e conseqüentemente apenas poucas aplicações práticas foram apresentadas (CHEN; BILLINGS, 1989a).

3.2.3 Redes Neurais

As redes neurais artificiais são nada mais do que um tipo de base em particular que podem ser usadas para modelagem. As redes neurais são compostas por camadas de neurônios interconectados. A saída de um neurônio com n entradas é apresentada na equação:

$$x = f \left(\sum_{j=1}^n \omega_j x_j + b \right) \quad (59)$$

onde b é conhecido como *bias* e ω_j são constantes e $f(\cdot)$ é chamada função de ativação. ω_j e b são os parâmetros a serem identificados. A função de ativação mais comum é (AGUIRRE, 2004):

$$f(q) = \frac{1}{1 + e^{-q}}$$

3.2.3.1 Redes Neurais multi-camadas

Uma típica rede multi-camadas pode ser descrita como na Figura 6. Na prática redes neurais multi-camadas têm grande apelo no reconhecimento de padrões. Do ponto de vista teórico os sistemas de redes multi-camadas podem ser considerados como mapas não lineares onde os elementos das matrizes de peso são os parâmetros (NARENDRA; PARTHASARATHY, 1990).

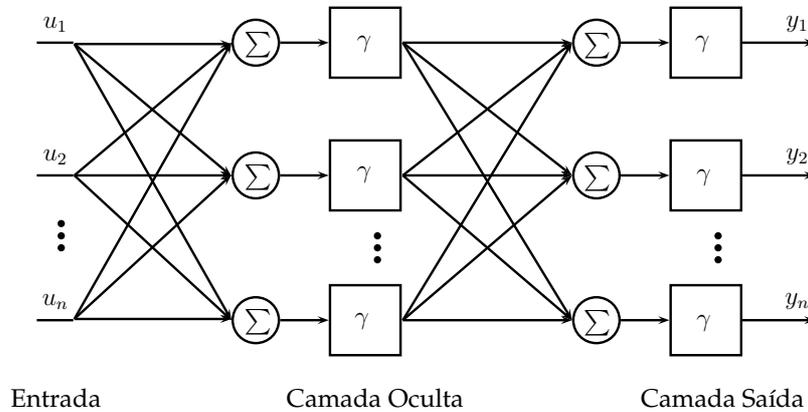


Figura 6: Representação de uma rede neural multi-camadas.

A saída do sistema para uma rede multi-camadas pode ser descrita como:

$$y(t) = f_s \left\{ \sum_{i=1}^m \omega_i f_i \left(\sum_{j=1}^n \omega_{ij} x_j + b_i \right) + b_s \right\} \quad (60)$$

onde f_s é a função de ativação do neurônio da camada de saída. Esta função não precisa ser igual a f_i , $i = 1, \dots, m$ que por sua vez não precisam ser iguais entre si. As constantes b_s são os termos de polarização dos neurônios da camada de saída, ω_i são os pesos da saída de cada neurônio da camada oculta e ω_{ij} são os pesos da entrada j , vista pelo i -ésimo neurônio da camada oculta (AGUIRRE, 2004).

3.2.3.2 Redes Neurais recorrentes

Redes neurais recorrentes, trabalho introduzido por Hopfield em (HOPFIELD, 1982) provê uma alternativa para o reconhecimento de padrões. O método pro-

posto consiste em ter uma rede neural de apenas uma camada adicionada de uma realimentação com um atraso de tempo como apresentado na Figura 7 (NARENDRA; PARTHASARATHY, 1990).

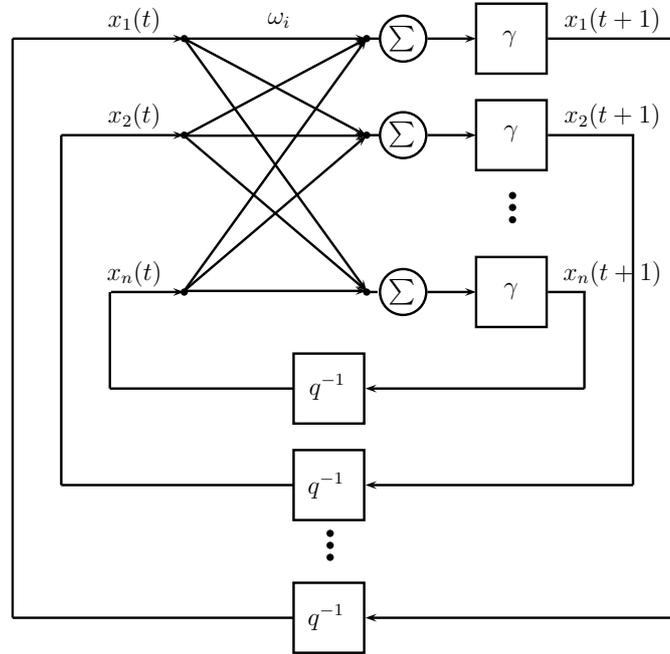


Figura 7: Representação de redes neurais recorrentes. Onde q^{-1} representa um atraso de tempo na realimentação do sistema.

3.2.4 Funções Radiais de Base

Funções radiais de base (*RBF - Radial basis functions*) são uma tradicional técnica para interpolação em espaços multidimensional (CHEN et al., 1990) e podem ser descritas como mapeamentos do tipo:

$$f(y) = w_0 + \sum_i w_i \phi(\|y - c_i\|) \quad (61)$$

onde $y \in \mathbb{R}^{d_e}$ (d_e é conhecido como dimensão de imersão), $\|\cdot\|$ é a norma euclidiana, $w_i \in \mathbb{R}$ são pesos, $c_i \in \mathbb{R}^{d_e}$ são centros e $\phi(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}$ é uma função, normalmente escolhida a priori, como por exemplo (AGUIRRE, 2004):

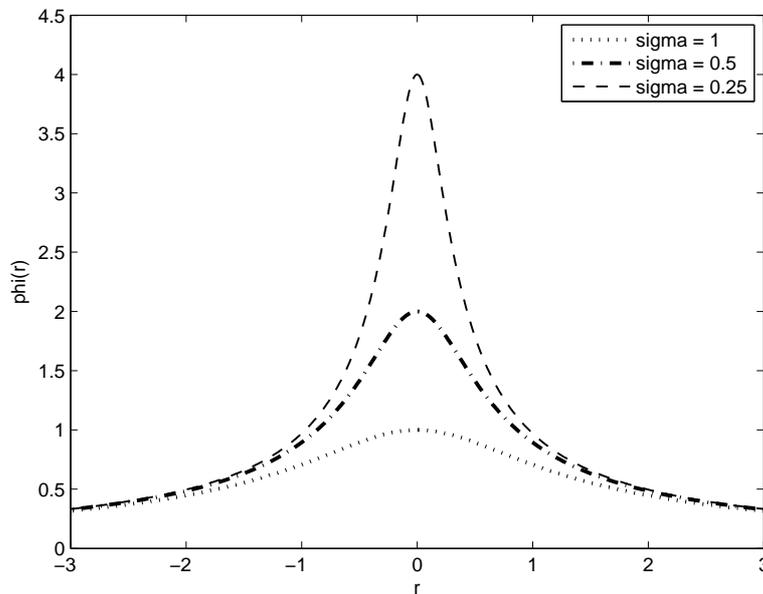
$$\phi(\|y - c_i\|) = \exp\left(-\frac{\|y - c_i\|^2}{\sigma_i^2}\right)$$

Outras funções de base usadas são apresentadas na Tabela 3.

Tabela 3: Algumas funções Radiais de base comumente utilizadas.

Nome	Função
Multi quadrática inversa	$\phi(r) = (r^2 + \sigma^2)^{-1/2}$
Linear	$\phi(r) = r$
Cúbica	$\phi(r) = r^3$
Multi-quadrática	$\phi(r) = \sqrt{r^2 + \sigma^2}$
<i>Thin - plate spline</i>	$\phi(r) = r^2 \log(r)$

onde $r = \|y - c_i\|$ e σ definem a largura do chapéu no caso de funções do tipo Gaussianas e multi-quadráticas, como pode ser visto na Figura 8.

Figura 8: Função multi-quadráticas inversa para alguns valores de σ .

Este tipo de representação tem boas propriedades locais e pode ser interpretada como uma técnica de interpolação global. Funções radiais de base são casos particulares de redes neurais, porém neste caso, lineares nos parâmetros w_i (AGUIRRE, 2004).

No contexto de identificação de sistemas é comum adicionar termos auto-regressivos lineares, bem como termos de entrada à equação (61) resultando em:

$$y(k) = w_0 + \sum_i w_i \phi(\|\mathbf{y}(k-1) - c_i\|) + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + e(k)$$

onde

$$\mathbf{y}(k-1) = [y(k-1) \quad \dots \quad y(k-n_y) \quad u(k-1) \quad \dots \quad u(k-n_u)]$$

3.2.5 Modelos NARMAX

Os modelos *NARX* (do termo inglês *nonlinear autoregressive model with exogenous variables*) são modelos discretos no tempo que caracterizam o valor da saída em função dos valores passados da entrada e da própria saída. Algumas vezes, para evitar a polarização da estimativa dos parâmetros adiciona-se termos do modelo do ruído ao modelo do sistema. Quando isso é feito o modelo passa a ser chamado de modelo *NARMAX* (do termo inglês *nonlinear autoregressive moving average model with exogenous variables*), introduzido por (LEONTARITIS; BILLINGS, 1985).

Este modelo provê uma representação unificada para a descrição de sistemas discretos não lineares:

$$y(t) = \theta^T \Phi_{nl}(y, u, e) \quad (62)$$

onde $\Phi_{nl}(\cdot)$ denota um campo vetorial que depende dos valores passados de $y(t)$ e presente e passados de $u(t)$ e $e(t)$; θ é o vetor de parâmetros a ser identificado.

As classes de modelos NARMAX podem então ser compreendidas como um conjunto ou somatório de funções não lineares, que são parametrizados linearmente. Vale aqui esclarecer que a forma apresentada em (62) não é uma regressão linear. Para uma regressão linear:

$$y(t) = \theta^T \phi(y(t), u(t)) \quad (63)$$

onde $\phi(\cdot)$ é dependente apenas dos sinais de entrada e saída do sistema, usualmente sendo uma matriz formada a partir dos dados coletados do sistema. Diferentemente do equacionamento apresentado em (62), onde $\Phi_{nl}(\cdot)$ é dependente também do ruído. Outra diferença está no fato de que em (62) é possível ter regressores não lineares.

3.2.6 Modelo polinomial

Um modelo polinomial pode ser representado como

$$y(t) = \sum_i c_i \prod_{j=1}^{n_y} y(t-j) \prod_{r=1}^{n_u} u(t-r) \prod_{q=0}^{n_e} e(t-q) \quad (64)$$

para este caso duas considerações serão levadas em conta:

1. O sistema tem um atraso puro de tempo τ_d .
2. Nenhum termo cujo parâmetro tenha que ser estimado pode depender de $e(t)$.

A segunda consideração implica em tornar a equação independente de $e(t)$. O que equivale a dizer que $q = 1$ em (64) resultando em:

$$y(t) = F[y(t-1), \dots, y(t-n_y), u(t-\tau_d), \dots, u(t-\tau_d-n_u+1), e(t-1), \dots, e(t-n_e)] + e(t) \quad (65)$$

onde $e(t)$ indica que todos os efeitos não podem ser bem representados. $F^l[\cdot]$ é uma função polinomial de $y(t)$, $u(t)$ e $e(t)$ com grau de não linearidade $l \in \mathbb{N}$. Portanto a parte não determinística da equação (65) pode ser expandida como um somatório de termos com grau de não linearidade variando de $1 \leq m \leq l$. Assim sendo, cada termo de grau m poderá conter um valor de grau p do tipo $y(t-i)$ e um fator de grau $(m-p)$ do tipo $u(t-i)$ sendo multiplicado por um parâmetro representado por $c_{p,m-p}(n_1, \dots, n_m)$. Obtendo (AGUIRRE, 2004):

$$y(t) = \sum_{m=0}^l \sum_{p=0}^m \sum_{n_1, n_m}^{n_y, n_u} c_{p,m-p}(n_1, \dots, n_m) \prod_{i=1}^p y(t-n_i) \prod_{i=p+1}^m u(t-n_i) \quad (66)$$

sendo que

$$\sum_{n_1, n_m}^{n_y, n_u} \equiv \sum_{n_1=1}^{n_y} \cdots \sum_{n_m=1}^{n_y}$$

e o limite superior é n_y se o somatório se refere a fatores do tipo $y(t-n_i)$ ou n_u se os fatores forem do tipo $u(t-n_i)$.

Os modelos racionais tratados em seguida são diferenciados dos demais apresentados neste trabalho, pois são foco de estudo posterior nos capítulos seguintes e necessita de um algoritmo especial para a identificação de seus parâmetros, por isso será tratado em uma seção própria, separado dos outros modelos não lineares, aqui apresentados.

3.3 Modelo Racional

Um modelo racional tem a seguinte forma geral:

$$y(t) = \frac{a(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))}{b(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))} + c(e(t-1), \dots, e(t-n_e)) + e(t) \quad (67)$$

onde $a(\cdot)$ e $b(\cdot)$ são polinômios e é conveniente definir o numerador e denominador de como em:

$$a(\cdot) = \sum_{j=1}^{N_n} p_{nj} \theta_{nj} = \psi_n^T(t-1) \theta_n \quad (68)$$

$$b(\cdot) = \sum_{j=1}^{N_d} p_{dj} \theta_{dj} = \psi_d^T(t-1) \theta_d \quad (69)$$

onde θ_{nj} e θ_{dj} são os parâmetros dos regressores, possuindo informações até o instante $t-1$. Desta forma o vetor de parâmetros a ser estimados possui tamanho $N_n + N_d$.

A equação (67) possui não linearidade nos parâmetros, tornando a identificação mais complexa por não ser possível a utilização do método dos mínimos quadrados para a estimativa dos parâmetros. Para isso será apresentado a seguir um algoritmo iterativo que fará a identificação dos parâmetros do sistema.

Uma das maiores desvantagens do modelo racional quando comparado com o modelo polinomial é que o modelo polinomial é linear nos parâmetros. Muitos resultados para identificação de sistemas lineares podem ser estendidos para modelos polinomiais não lineares e várias rotinas de determinação de estrutura já foram desenvolvidas para isso (CHEN; BILLINGS, 1989a; BILLINGS; ZHU, 1995).

3.3.1 Algoritmo para identificação de modelos racionais

Esta seção descreve um algoritmo para determinar os parâmetros de modelos racionais (67). Este algoritmo foi proposto por (CORRÊA, 1997) e é uma modificação do algoritmo originalmente proposto por (BILLINGS; ZHU, 1991).

O ruído é modelado por um polinômio que pode ou não ser linear. A consideração básica por trás de (67) é que o erro de regressão pode ser representado por um modelo *MA* (*Move average*), possivelmente não linear. Assim sendo sugere-se o seguinte procedimento (AGUIRRE, 2004):

1. Faça $i = 0$. Monte a matriz Ψ de regressão e estime os parâmetros usando o método dos mínimos quadrados:

$$\begin{bmatrix} \hat{\theta}_n^i \\ \hat{\theta}_{d1}^i \end{bmatrix} = [\Psi^T \Psi]^{-1} \Psi^T y^* \quad (70)$$

onde o índice i indica a iteração. Além disso a matriz de regressores Ψ é formada tomando-se os vetores de regressores $\psi_n(t-1)$ e $\psi_{d1}(t-1)$ ao longo da janela de dados do tamanho N , ou seja:

$$\Psi = \begin{bmatrix} \psi_n^T(t-1) & \psi_{d1}^T(t-1) \\ \vdots & \vdots \\ \psi_n^T(t+N-2) & \psi_{d1}^T(t+N-2) \end{bmatrix}$$

Analogamente o vetor $y^* \in \mathbb{R}^{N \times 1}$ é formado tomando os dados, ou seja:

$$y^{*T} = [y^*(t), y^*(t+1), \dots, y^*(t+N-1)]$$

2. Faça $i = i + 1$. Determine os resíduos e sua variância, respectivamente como:

$$\xi^i(t) = y(t) - \frac{\psi_n^T(t-1)\hat{\theta}_n}{\psi_d^T(t-1)\hat{\theta}_d} \quad (71)$$

$$(\sigma_\xi^2)^i = \frac{1}{N - m_d} \sum_{i=m_d+1}^N (\xi^i(t))^2 \quad (72)$$

onde N é o tamanho dos dados e $m_d = \max(n_y, n_u, n_e)$.

3. Usando-se os resíduos determinados no passo anterior, atualize $\Psi^T \Psi$ e $\Psi^T y^*$ usando:

$$\Psi = \begin{bmatrix} \psi_n^T(t-1) & y(t)\psi_{d1}^T(t-1) & \psi_\xi^T(t-1) \\ \vdots & \vdots & \vdots \\ \psi_n^T(t+N-2) & y(t)\psi_{d1}^T(t+N-2) & \psi_\xi^T(t+N-2) \end{bmatrix} \quad (73)$$

onde ψ_ξ é o vetor de regressores do modelo do ruído. Pelo fato do ruído não ser medido, os resíduos do passo (2) são utilizados.

4. Determine

$$\Phi = \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \sum_{t=1}^N p_{d2}^2 & \dots & \sum_{t=1}^N p_{d2} p_{d_{N_d}} & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \sum_{t=1}^N p_{d_{N_d}} p_{d2} & \dots & \sum_{t=1}^N p_{d_{N_d}}^2 & \dots & 0 & \dots & 0 \end{bmatrix} \quad (74)$$

$$\phi = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\sum_{k=1}^N p_{d2} p_{d1} \\ \vdots \\ -\sum_{k=1}^N p_{dN_d} p_{d1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (75)$$

E estime novamente os parâmetros utilizando:

$$\begin{bmatrix} \hat{\theta}_n^i \\ \hat{\theta}_{d1}^i \end{bmatrix} = [\Psi^T \Psi - (\sigma_\xi^2)^i \Phi]^{-1} [\Psi^T y^* - (\sigma_\xi^2)^i \phi] \quad (76)$$

5. Volte ao passo 2 até atingir convergência (de parâmetro ou de variância de resíduo).

A convergência do algoritmo depende da convergência da estimativa da sequência do ruído (BILLINGS; ZHU, 1991).

3.3.2 Exemplos ilustrativos

Nesta seção serão apresentados alguns casos de uso do algoritmo descrito anteriormente. Inicia-se com um exemplo simples, onde o sistema é racional e a classe de modelos escolhida consegue representar completamente este sistema ($\mathcal{S} \in \mathcal{M}$). Em seguida será apresentado um exemplo onde o sistema ainda é racional, mas a classe de modelos escolhida não consegue representar completamente o sistema em questão, ou seja, $\mathcal{S} \notin \mathcal{M}$. Por fim será apresentado um exemplo de um conversor de corrente contínua para corrente contínua do tipo Buck.

Serão apresentados os resultados obtidos a fim de verificar a qualidade e a confiabilidade do algoritmo de identificação de sistemas polinomial e racional.

3.3.2.1 Sistemas originalmente racionais

Considere o sistema real:

$$y(t) = \frac{0.2y(t-1) + 0.1y(t-1)u(t-1) + u(t-1)}{1 + y^2(t-1) + y^2(t-2)} + e(t) \quad (77)$$

O modelo escolhido para representar o sistema real é:

$$y(t) = \frac{\theta_1 y(t-1) + \theta_2 y(t-1)u(t-1) + \theta_3 u(t-1)}{1 + \theta_4 y^2(t-1) + \theta_5 y^2(t-2)} \quad (78)$$

Utilizando para simulação como entrada $u(t)$ um sinal PRBS de 635 pontos e adicionando-se um ruído $e(t)$ de variância $\sigma_e^2 = 0.005$ os resultados da média de 100 estimativas de Monte Carlo foram:

$$\theta_{\text{média}} = [0.1991 \quad 0.0995 \quad 0.9963 \quad 0.9899 \quad 0.9912]$$

A matriz de covariância encontrada:

$$\theta_{\text{Covariância}} = 1.0 \times 10^{-4} \begin{bmatrix} 0.0089 & -0.0005 & 0.0112 & 0.0213 & 0.0339 \\ -0.0005 & 0.0073 & 0.0057 & 0.0220 & 0.0091 \\ 0.0112 & 0.0057 & 0.1082 & 0.2529 & 0.2589 \\ 0.0213 & 0.0220 & 0.2529 & 0.7179 & 0.4987 \\ 0.0339 & 0.0091 & 0.2589 & 0.4987 & 0.7910 \end{bmatrix}$$

Utilizando a média das estimativas para simular a saída do modelo obtido com o sistema real, obteve-se um custo:

$$V(\theta) = 1.1269 \times 10^{-4}$$

A fim de melhor ilustrar a estimativa obtida, na Figura 9 são apresentadas as estimativas dos parâmetros θ_1 e θ_2 e a elipse de confiança para $\chi^2 = 95\%$.

Devido à presença de ruído e ao erro de polarização apresentado ser menor que a variância do ruído adicionado, sabe-se que com o aumento de N o valor de $\hat{\theta}_N \rightarrow \theta_0$.

3.3.2.2 Sistemas originalmente racionais, quando $\mathcal{S} \notin \mathcal{M}$

Considere o sistema real:

$$y(t) = \frac{0.2y(t-1) + 0.1y(t-1)u(t-1) + u(t-1)}{1 + y^2(t-1) + y^2(t-2)} + e(t) \quad (79)$$

O sistema real \mathcal{S} não consegue ser representado pela classe de modelos escolhida:

$$y(t) = \frac{\theta_1 y(t-1) + \theta_2 u(t-1)}{1 + \theta_3 y^2(t-1) + \theta_4 y^2(t-2)} \quad (80)$$

Utilizando para simulação um sinal PRBS de 635 pontos e um ruído de variância $\sigma_e^2 = 0.005$ os resultados da média de 100 estimativas de Monte Carlo foram:

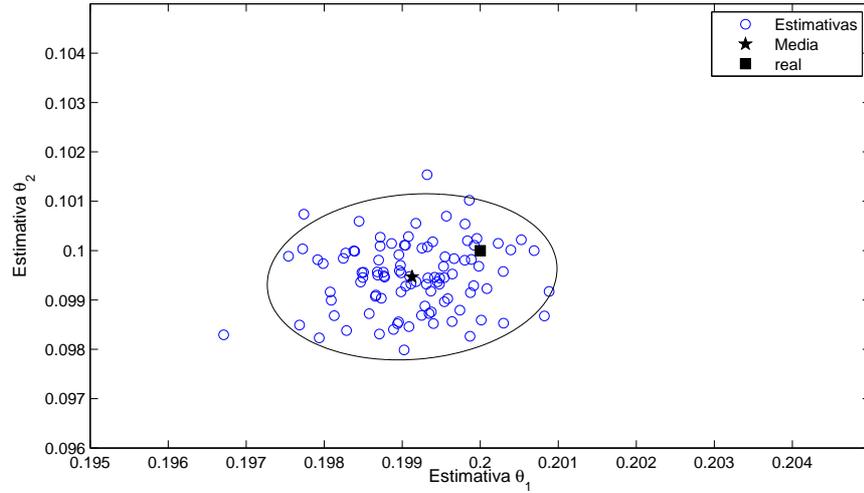


Figura 9: Estimativas obtidas nos 100 experimentos de Monte Carlo realizados para as variáveis θ_1 e θ_2 . Destaca-se a elipse de confiança para $\chi^2 = 95\%$

$$\theta_{\text{média}} = [0.2279 \quad 1.1341 \quad 1.2716 \quad 1.3769]$$

A fim de comparar a qualidade dos resultados obtidos, na Figura 10 são apresentadas as respostas do sistema real (79) e do sistema obtido pela estimativa utilizando a classe de modelos (80) para uma entrada do tipo degrau unitário.

O custo obtido para o modelo estimado foi de:

$$V(\theta) = 1.1239$$

Como não há um valor de θ_0 para comparação, visto que o sistema real não consegue ser representado pelo modelo, optou-se por apresentar o resultado do sistema a uma resposta do tipo degrau unitário. Observa-se com isso que o sistema estimado é apenas semelhante ao real, possuindo um sobrepasso maior e possuindo um ganho em regime ligeiramente diferente do sistema real.

3.3.2.3 Conversor CC-CC Buck

O conversor de corrente contínua do tipo Buck possui um mapa não linear que pode ser obtido a partir da equação do circuito apresentado na Figura 11 e que tem a forma como em (TSE, 1994):

$$y(t) = \alpha y(t-1) + \frac{h(d_n)^2 \beta R [R - y(t-1)]}{y(t-1)} \quad (81)$$

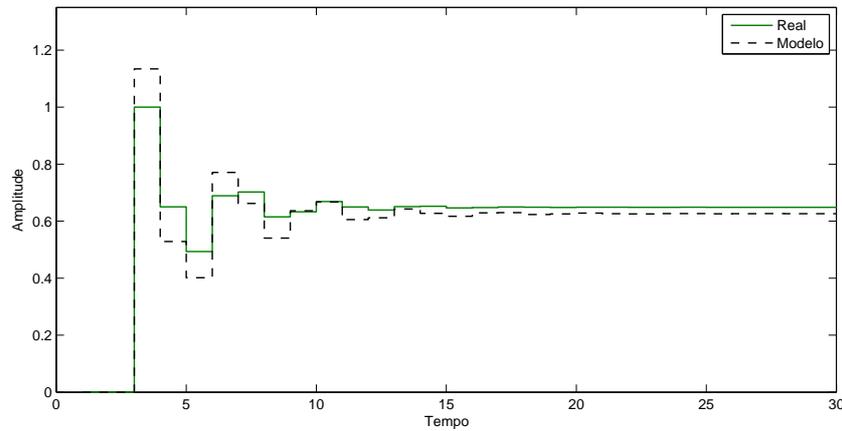


Figura 10: Comparativo da resposta do sistema real e do modelo estimado para uma entrada do tipo degrau unitário

onde $\alpha = 0.8872$, $\beta = 1.2$ e $R = 33$ são constantes que dependem apenas dos componentes do circuito eletrônico. d_n é um sinal de tensão que implementa a ação de controle e a saturação $h(d_n)$ é dada por:

$$h(d_n) = \begin{cases} 0 & \text{se } d_n < 0 \\ 1 & \text{se } d_n > 1 \\ d_n & \text{caso contrario.} \end{cases}$$

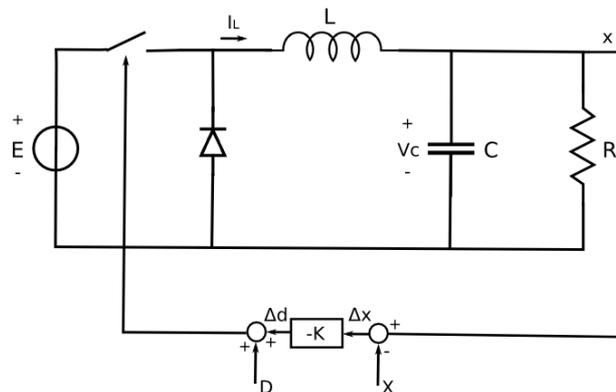


Figura 11: Circuito do conversor CC-CC Buck

Modelos polinomiais não resultam em bons modelos para a dinâmica de (81). Um modelo com estrutura *ad hoc*, que aproxima relativamente bem a característica estática do conversor é (AGUIRRE, 1997):

$$y(t) = Oexp [22 - y(t - 1)] + \frac{a y(t - 1)^2 - b y(t - 1) + c}{y(t - 1)} \quad (82)$$

onde $O = 46.429$, $a = 2.6204$, $b = 99.875$ e $c = 1.4171 \times 10^3$.

Para aplicar o algoritmo, escolhe-se inicialmente uma classe de modelos para o sistema em análise:

$$y(t) = \frac{\theta_1 + \theta_2 y(t - 1)^3 + \theta_3 y(t - 1)^2}{1 + \theta_4 y(t - 1) + \theta_5 y(t - 1)^2} \quad (83)$$

Como o sistema não possui entrada, foram feitas diversas simulações com valores iniciais diferentes a fim de demonstrar a influência deste valor inicial sobre as estimativas e sobre tudo a qualidade destas estimativas obtidas. Na Tabela 4 são apresentados os valores estimados utilizando o algoritmo de identificação de sistemas racionais para o sistema apresentado em (81). Ao utilizar-se 200 pontos e um valor inicial de 31.5 obteve-se o seguinte conjunto de parâmetros θ :

$$\theta = [11.832 \quad 0.0019514 \quad -0.065624 \quad -0.095531 \quad 0.002306] \quad (84)$$

com uma covariância de

$$\theta_{\text{Cov}} = \begin{bmatrix} 0.18146 & 4.61 \times 10^{-5} & -1.33 \times 10^{-3} & -7.87 \times 10^{-4} & 3.58 \times 10^{-5} \\ 4.61 \times 10^{-5} & 1.17 \times 10^{-8} & -3.40 \times 10^{-7} & -2. \times 10^{-7} & 9.12 \times 10^{-9} \\ -1.33 \times 10^{-3} & -3.40 \times 10^{-7} & 9.85 \times 10^{-6} & 5.80 \times 10^{-6} & -2.64 \times 10^{-7} \\ -7.87 \times 10^{-4} & -2 \times 10^{-7} & 5.80 \times 10^{-6} & 3.41 \times 10^{-6} & -1.55 \times 10^{-7} \\ 3.58 \times 10^{-5} & 9.12 \times 10^{-9} & -2.64 \times 10^{-7} & -1.55 \times 10^{-7} & 7.08 \times 10^{-9} \end{bmatrix}$$

e um custo $J_y(\theta) = 9.1789$. Na Figura 12 é apresentado o comparativo entre o sinal obtido do sistema real definido por 82 e o sistema estimado com θ como em (84).

Pelos resultados apresentados na Figura 12, observa-se que a estimativa obtida consegue representar bem o sistema em questão. Na Tabela 4 são apresentados os valores obtidos de custo e de θ para diferentes valores iniciais.

É interessante observar que os parâmetros estimados θ variam consideravelmente com as alterações dos valores iniciais de $y(0)$, mas que o custo entre o sinal real e o estimado não tem uma variação tão pronunciada, fazendo com que todas estimativas feitas com valor inicial superior a 23.25 convergissem para um custo próximo. Para valores inferiores a 23.25 o custo da estimativa diverge, fazendo com que os parâmetros estimados não mais consigam representar o sistema real.

Tabela 4: Tabela com os valores das estimativas do conversor de corrente contínua para diferentes valores de condições iniciais. Todas as simulações utilizaram $N = 100$.

Valor inicial	$J_y(\theta)$	θ_1	$\theta_2 \times 10^{-3}$	$\theta_3 \times 10^{-2}$	$\theta_4 \times 10^{-2}$	$\theta_5 \times 10^{-3}$
23	50.089	10.0202	0.763182	-3.46527	-6.81581	1.13627
23.25	5.5352	-0.81237	-1.28468	2.79502e	-4.01844	-0.21438
23.5	7.6047	11.3058	1.78578	-6.10171	-9.23685	2.16415
23.75	5.9539	12.8959	2.20649	-7.31006	-9.97201	2.49790
24	7.8558	13.1524	2.26985	-7.49464	-10.0780	2.54633
24.5	6.6557	13.9802	2.49699	-8.14400	-10.4804	2.72801
24.75	5.3027	12.8416	2.20702	-7.30300	-9.98869	2.50449
25	6.693	12.4921	2.09369	-6.98853	-9.77019	2.40692
25.5	5.2074	12.7096	2.15963	-7.17432	-9.89270	2.46168
25.75	5.9741	12.5451	2.10720	-7.02813	-9.79257	2.41698
26	7.4629	12.8809	2.19478	-7.27905	-9.94607	2.48714
26.5	6.9013	13.6151	2.39946	-7.86409	-10.3097	2.65066
26.75	5.4738	13.4205	2.35664	-7.73609	-10.2448	2.62060
27	8.4976	11.8107	1.93084	-6.51041	-9.50420	2.28510
27.5	6.983	12.6303	2.14560	-7.13057	-9.87420	2.45267
27.75	8.501	12.2226	2.04532	-6.83696	-9.70788	2.37700
28	7.575	12.234	2.03861	-6.82282	-9.68797	2.36873
28.5	5.5844	12.6729	2.15105	-7.14821	-9.88018	2.45613
28.75	6.5158	12.8559	2.21262	-7.31848	-9.99959	2.50919
29	7.8105	12.8799	2.21777	-7.33352	-10.0080	2.51323
29.5	6.4722	12.8431	2.19824	-7.28332	-9.96343	2.49356
29.75	7.5409	11.823	1.91388	-6.47248	-9.45460	2.26415
30	7.3062	12.5771	2.12420	-7.07062	-9.83379	2.43552
30.5	6.6314	12.7644	2.16924	-7.20325	-9.90581	2.46835
30.75	7.3702	12.4053	2.07591	-6.93389	-9.74557	2.39560
31	6.9049	12.0157	1.96101	-6.60985	-9.53295	2.30004
31.5	4.7764	13.2659	2.30172	-7.58446	-10.1352	2.57236
31.75	6.0537	12.8071	2.16287	-7.19490	-9.87593	2.45604
32	8.9643	11.7958	1.89441	-6.42284	-9.40820	2.24413
32.5	6.9209	9.995	1.44211	-5.11152	-8.64115	1.89389
32.75	9.9668	8.6553	1.07533	-4.06248	-7.99042	1.60003
33	5.5441	6.0089	0.423059	-2.16225	-6.89908	1.10128
33.5	5.9131	2.6155	-0.454214	0.370902	-5.39045	0.41545
33.75	5.5792	1.5481	-0.751479	1.22007	-4.85755	0.17591
34	6.1125	0.67708	-0.951813	1.81111	-4.53803	0.02826
34.5	6.1698	-0.61482	-1.26130	2.71727	-4.03054	-0.20474
34.75	6.8971	-0.83524	-1.31433	2.87267	-3.94360	-0.24457

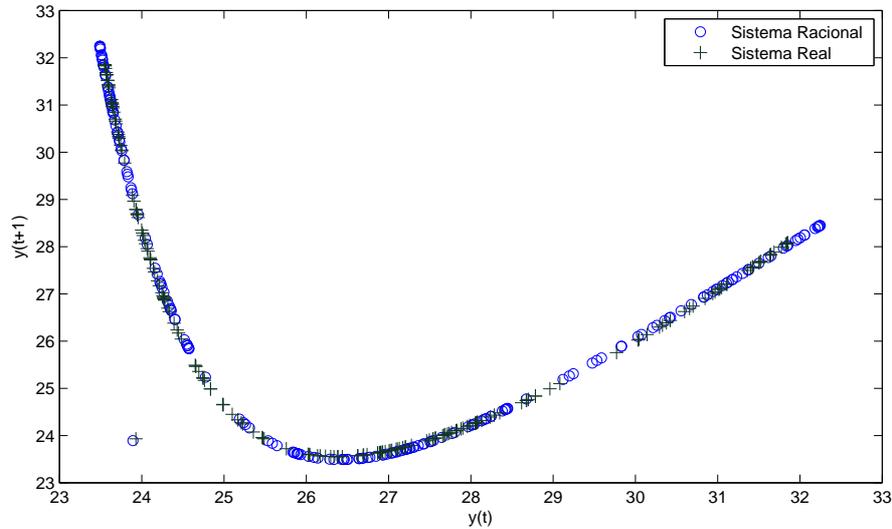


Figura 12: Comparativo entre a saída do sistema real descrito pela equação(82) e a saída do sistema racional descrito pela classe de modelos apresentada em (83) quando os parâmetros estimados para um valor inicial de 31.5, como apresentado em (84).

3.4 Considerações Finais

Neste capítulo apresentou-se conceitos básicos de identificação de sistemas não lineares. Inicialmente foram apresentados os tipos de linearidades existentes. Muitas vezes os sistemas comportam-se de forma linear em torno da região de operação do sistema, desta forma identificar um sistema com um modelo linear é o que normalmente se utiliza. Para os casos onde há necessidade de modelar a dinâmica não linear, ou que se deseja modelar características específicas do sistema, uma classe de modelos não linear é mais apropriada.

Apresentou-se alguns dos conjuntos de modelos mais utilizados para identificação de sistemas não lineares de tempo discreto e algumas de suas características. Destacou-se que a escolha de um modelo em detrimento de outro é um aspecto muito importante para a identificação de sistemas não lineares. A escolha de uma base e a ordem do modelo são muito importantes para que não sejam acrescentados ao modelo dinâmicas que não existem no sistema real.

Deu-se especial atenção às classes de modelos racionais e polinomiais. Estes conjuntos de modelos conseguem representar uma grande variedade de sistemas não lineares utilizando uma estrutura simples de compreender e que por ser linear nos parâmetros (caso dos modelos polinomiais), faz com que os algoritmos para a

identificação de seus parâmetros sejam mais simples.

Apresentou-se um algoritmo para identificação de modelos racionais, que pode ser utilizado para modelos polinomiais sem dificuldades. Este algoritmo iterativo obteve bons resultados para os experimentos efetuados e estes resultados foram apresentados para diversos tipos de classes de modelos escolhidas.

O algoritmo implementado não contemplou a modelagem do ruído do sistema, sendo este um aspecto para ser desenvolvido em trabalhos futuros, visto que este fato é bastante importante para situações onde $\mathcal{S} \notin \mathcal{M}$, podendo o algoritmo nestas situações convergir para pontos onde os parâmetros identificados não sejam os reais. Esta situação foi apresentada no exemplo do conversor de corrente contínua do tipo Buck (Seção 3.3.2.3).

4 CONTROLE BASEADO EM DADOS

Neste capítulo serão apresentados conceitos básicos sobre projetos de controladores baseados em dados. Iniciando pela Seção 4.1, onde serão abordados brevemente alguns dos mais conhecidos métodos para projetos de controladores baseados em dados e também os conceitos básicos que permeiam sua utilização.

Na Seção 4.2 será apresentado o método VRFT (*Virtual reference feedback tuning*), que utiliza referência virtual para obtenção dos sinais necessários para o projeto do controlador e que será de grande valia para a proposta de trabalho apresentada no Capítulo 5.

Na Seção 4.2.4 serão apresentados alguns exemplos de uso do método VRFT e os resultados obtidos. Ao fim, na Seção 4.3 será apresentado uma breve conclusão sobre o que foi abordado neste capítulo.

4.1 Definições

Desde meados dos anos 1990 têm surgido na literatura uma variedade de projetos de controladores que são construídos diretamente sobre os dados de entrada e saída coletados do sistema. Estes métodos contrastam com os de projetos de controladores baseados em modelos em dois aspectos principais: eles não são baseados no conhecimento do modelo do processo e não têm a intenção de determinar livremente a função de transferência do modelo. Ao invés disso usa-se diretamente o montante de informação carregada pelos dados coletados para ajustar os parâmetros de um conjunto de modelos cuja estrutura é previamente especificada (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). Estes métodos são conhecidos como projetos de controladores baseados em dados ou do inglês *data-driven* ou *data based controller design*.

Como apresentado na Seção 2.6 os dados coletados do sistema podem ser obtidos de diversas formas. Algumas vezes estas informações podem vir da operação normal da planta em malha fechada com a presença de algum controlador. Esta situação tem um grande apelo em plantas industriais, onde a parada do processo

para levantamento de informações é muitas vezes indesejável e por vezes até inviável. Se existe a possibilidade de parar a planta e aplicar sinais predeterminados, o projeto de experimentos pode trazer muitas vantagens como foi apresentado na Seção 2.6.

O projeto de controladores baseados em dados consiste em estimar os parâmetros da estrutura do controlador usando os dados de entrada e saída coletados do sistema. A estimativa é feita de forma direta, ou seja, sem a necessidade de um passo intermediário para identificação da planta do processo (ECKHARD et al., 2009).

Existem na literatura diversos métodos de projetos de controladores baseados em dados que otimizam algum critério de desempenho, com diferentes focos para diferentes critérios. Estes critérios expressam um ou uma combinação de objetivos fundamentais de controle: seguimento de referência, rejeição a ruído e uso econômico da energia de controle. Em (KAMMER; BITMEAD; BARTLETT, 2000) um procedimento iterativo baseado na análise espectral chamado de *Frequency Domain Tuning* (FDT) foi proposto para a minimização do critério H_2 de desempenho para um sistema com referência zero, portanto nenhum objetivo de seguimento de referência é almejado. O método *Virtual Reference Feedback Tuning* (VRFT) (CAMPI; LECCHINI; SAVARESI, 2002; CAMPI; SAVARESI, 2006) é baseado na manipulação de variáveis que transformam um critério H_2 em algo quadrático nos parâmetros. A função custo quadrática resultante pode ser minimizada diretamente, sem que nenhuma iteração seja necessária. Entretanto apenas o objetivo de seguimento de referência é tratado (a não ser que um controlador com dois níveis de liberdade seja usado (LECCHINI; CAMPI; SAVARESI, 2002)) e o mínimo global resultante desta função quadrática coincide com o do critério original somente sob condições ideais. *Correlation-based Tuning* (CbT) (KARIMI; MIŠKOVIĆ; BONVIN, 2003, 2004) não sofre desta segunda limitação, mas é um método iterativo que usa a ideia de variáveis instrumentais para remover o efeito não desejado do ruído enquanto busca seu objetivo de seguimento de referência (BAZANELLA et al., 2008).

Otimização baseada em dados para um critério de desempenho genérico H_2 foi proposto em (HJALMARSSON; GUNNARSSON; GEVERS, 1994), onde um método para obter estimativas não polarizadas diretamente do gradiente da função custo, a partir de dados do sistema em malha fechada, foi elaborado. Este método é chamado de *Iterative Feedback Tuning* (IFT). IFT é discutido a fundo em (HJALMARSSON et al., 1998; HJALMARSSON, 2002) e estendido em (PROCHÁZKA et al., 2005) para critério de desempenhos mais gerais, que contém melhorias em objetivos de robustez (BAZANELLA et al., 2008).

Na Figura 13 é apresentado um sistema com realimentação de saída onde são destacados os blocos do controlador, da planta e da função que modifica o ruído

branco $e(t)$.

O ruído é um processo quasi-estacionário, que pode ser descrito como $\nu(t) = H_0(q)e(t)$, onde $e(t)$ é ruído branco com variância $\sigma_e^2(t)$. Ambas funções de transferência, $G_0(q)$ e $H_0(q)$, são racionais e causais. Assume-se que $H_0(\infty) = 1$, ou seja, a resposta impulsiva do filtro $H_0(q)$ satisfaz $h_0(0) = 1$ (CAMPESTRINI, 2010).

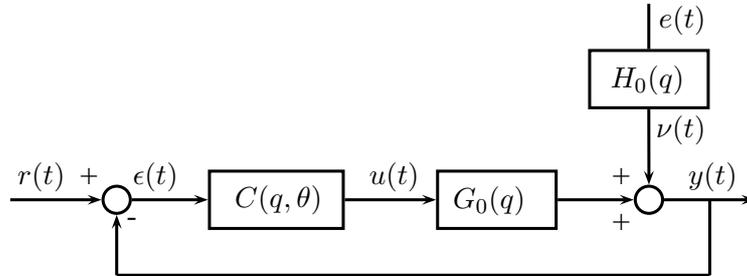


Figura 13: Representação de um sistema de controle em malha fechada, com ruído aditivo na saída. São apresentados também os principais sinais referenciados.

A equação:

$$T(q, \theta) = \frac{C(q, \theta)G_0(q)}{1 + C(q, \theta)G_0(q)} \quad (85)$$

descreve a relação entre entrada e saída do sistema apresentado na Figura 13. Desta forma pode se escrever o sinal de saída do sistema como:

$$y(t) = T(q, \theta)r(t) \quad (86)$$

4.1.1 Critérios de Desempenho

Analogamente ao que foi apresentado para identificação de sistemas no Capítulo 2, a classe de modelos que representa o controlador do sistema, é definido por

$$\mathcal{C} : \{C(q, \theta) \mid \theta \in D_C\} \quad (87)$$

Se existir um conjunto θ que consiga descrever o controlador ideal $C_d(q)$, então este conjunto é denotado por θ_0 , sendo que:

$$C(q, \theta_0) = C_d(q)$$

Existem vários critérios que podem ser utilizados para elencar qual será o melhor conjunto θ do controlador. Alguns dos mais conhecidos são :

- Seguimento de referência,

- Rejeição ao ruído e
- Uso reduzido de esforço de controle.

Para o objetivo de seguimento de referência, o desempenho pode ser avaliado pela norma (BAZANELLA; CAMPESTRINI; ECKHARD, 2012):

$$J_y(\theta) \triangleq E [y(t) - y_d(t)]^2 = \bar{E} [(T(q, \theta) - T_d(q))r(t)]^2 \quad (88)$$

onde $y(t)$ é a saída medida do sistema e $y_d(t)$ é a saída desejada, $T_d(q)$ é o comportamento desejado para o sistema em malha fechada.

Com base nas equações (88) e (85) é possível determinar o equacionamento para o controlador ideal, que é aquele que irá fazer com que o sistema em malha fechada se comporte como o desejado:

$$C_d(q) = \frac{T_d(q)}{G_0(q)(1 - T_d(q))} \quad (89)$$

Outro objetivo fundamental é minimizar o efeito do ruído sobre a saída do sistema. Descreve-se a função sensibilidade $S(q, \theta)$ como:

$$S(q, \theta) = \frac{1}{1 + C(q, \theta)G_0(q)} \quad (90)$$

A saída do sistema em malha fechada referente apenas ao ruído, ou seja, sem levar em conta a parte relativa a referência é dada por (BAZANELLA; CAMPESTRINI; ECKHARD, 2012):

$$y_e(t, \theta) \triangleq S(q, \theta)\nu(t) \quad (91)$$

O critério que determina o desempenho do sistema na rejeição ao ruído é dado por:

$$J_e(\theta) \triangleq \bar{E} [y_e(t)]^2 = \bar{E} [S(q, \theta)\nu(t)]^2 \quad (92)$$

Para fazer com que $J_e(\theta) = 0$ seria necessário que $S(q, \theta) = 0 \quad \forall q$, o que demandaria que $C(q, \theta)G_0(q) \rightarrow \infty \quad \forall q$, tornando desta forma impossível de atingir o objetivo do critério ser zero. Relaxa-se então o critério de desempenho para:

$$J_e(\theta) \triangleq \bar{E} [(S(q, \theta) - S_d(q, \theta))\nu(t)]^2 \quad (93)$$

onde $S_d(q, \theta)$ é a função sensibilidade desejada para o sistema.

Assim como o controlador apresentado em (89) é o controlador ótimo para o critério de desempenho de seguimento de referência, para o critério de rejeição ao ruído o controlador ótimo pode ser obtido. Para minimizar a função custo

(92) escolhe-se $T_e(q)$ como abaixo (detalhes podem ser obtidos em (BAZANELLA; CAMPESTRINI; ECKHARD, 2012)):

$$T_e(q) = 1 - \frac{1}{H_0(q)} \quad (94)$$

Da onde obtém-se o controlador desejado para a rejeição ao ruído $C_e(q)$:

$$C_e(q) = \frac{T_e(q)}{G_0(q)(1 - T_e(q))} = \frac{H_0(q) - 1}{G_0(q)} \quad (95)$$

O controlador apresentado em (95) também é conhecido como controlador de mínima variância.

Neste trabalho os exemplos utilizados usam o critério de seguimento de referência, com o controlador ótimo desejado sendo representado por (89).

4.2 Virtual reference feedback tuning

VRFT do inglês *Virtual reference feedback tuning* é um método direto para identificação de controladores, ou seja, não é necessário o conhecimento ou identificação da planta para que este método seja utilizado. O método baseia-se na utilização de apenas um conjunto de dados, não necessitando de experimentos específicos. O procedimento procura pelo ponto ótimo do critério escolhido para a identificação do controlador (CAMPI; LECCHINI; SAVARESI, 2000).

Diferentemente de métodos iterativos, o VRFT não recai sobre mínimos locais, sempre procurando o mínimo global do critério escolhido, sob condições ideais:

- O sistema não é afetado por ruído, ou seja, $\sigma_e^2(t) = 0$.
- O controlador ideal $C_d(q) \in \mathcal{C}$.
- O controlador é parametrizado linearmente.

O controlador a ser identificado pode ser parametrizado como abaixo:

$$C(q, \theta) = \beta^T(q)\theta \quad (96)$$

onde $\beta(q) = [\beta_1(q) \ \beta_2(q) \ \dots \ \beta_n(q)]^T$ é um vetor linear de funções de transferências de tempo discreto e θ é o vetor de parâmetros a ser estimado.

O problema de identificação do controlador consiste em encontrar um $\hat{\theta}$ que minimize o critério de desempenho (88), replicado abaixo:

$$J_y(\theta) \triangleq \bar{E} [y(t) - y_d(t)]^2 = \bar{E} [(T(q, \theta) - T_d(q))r(t)]^2$$

que é dependente do modelo de referência escolhido.

Apenas alguns métodos genuinamente diretos são encontrados na literatura, dois destes são VRFT e IFT (*Iterative Feedback Tuning*). Mesmo estes dois métodos pertencendo a uma mesma classe, algumas diferenças significativas existem e podem ser ressaltadas (CAMPI; LECCHINI; SAVARESI, 2000):

- *IFT* é um técnica iterativa. Usualmente este método converge para o mínimo local mais próximo das condições iniciais. Ele requer experimentos específicos sobre a planta, com entradas específicas.
- *VRFT* é um procedimento não iterativo que procura pelo mínimo global do critério de desempenho (88). Este método não requer experimentos específicos sobre a planta, podendo inclusive utilizar os dados do funcionamento normal do sistema.

4.2.1 O método

Nesta seção será apresentada uma breve descrição de como funciona o algoritmo para obtenção do controlador utilizando o método VRFT. Para maiores detalhes e discussões aprofundadas é indicada a leitura de (CAMPI; LECCHINI; SAVARESI, 2000, 2002; CAMPESTRINI et al., 2011; BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

Baseado no sinal medido $y(t)$, considera-se um sinal $\bar{r}(t)$ tal que $T_d(q)\bar{r}(t) = y(t)$. Esta referência é conhecida como *virtual* pois ela não existe e não foi utilizada para gerar o sinal $y(t)$. A Figura 14 apresenta o sistema em malha fechada e os sinais presentes.

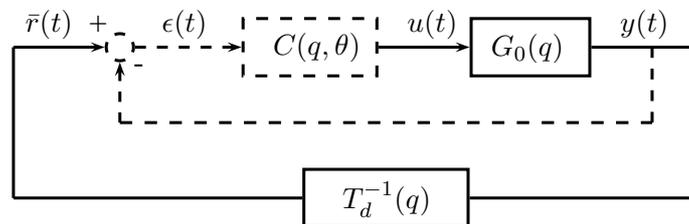


Figura 14: Diagrama de blocos para o método VRFT. São destacados os sinais reais $u(t)$ e $y(t)$ além dos sinais virtuais $\epsilon(t)$ e $\bar{r}(t)$. Em pontilhado está o controlador que quando aplicado sobre a malha, faz com que o sistema se comporte como desejado: $T_d(q)$.

O sinal $\epsilon(t)$ é o erro entre os sinais $y(t)$ e $\bar{r}(t)$. Sabe-se que quando a planta é excitada com o sinal $u(t)$, o sinal $y(t)$ é obtido. Analogamente para um controlador, quando este é excitado com o sinal $\epsilon(t)$, o sinal $u(t)$ é obtido. A tarefa do método VRFT é encontrar este controlador, com os sinais $u(t)$ e $\epsilon(t)$ que são conhecidos, a tarefa reduz-se a um problema de identificação. Comumente, usa-se um pré

filtro nos dados coletados. A ideia principal do uso deste filtro será explicada posteriormente na Seção 4.2.2.

O algoritmo pode ser descrito pelos passos a seguir (CAMPI; LECCHINI; SAVARESI, 2000):

1. Filtram-se os sinais de entrada e saída com algum filtro $L(q)$:

$$y_L(t) = L(q)y(t), \quad u_L(t) = L(q)u(t) \quad (97)$$

$$\epsilon_L(t) = L(q)\epsilon(t) = \bar{r}_L(t) - y_L(t)$$

2. Encontra-se um sinal de referência $\bar{r}_L(t)$ no qual a saída do modelo de referência $T_d(q)$ seja exatamente $y_L(t)$ quando alimentado pelo sinal:

$$y_L(t) = T_d(q)\bar{r}_L(t) \quad (98)$$

3. Seleciona-se o vetor de parâmetros do controlador $\hat{\theta}$ que minimize o critério:

$$J_{VR}^N(\theta) = \frac{1}{N} \sum_{t=1}^N (u_L(t) - \varphi_L^T(t)\theta)^2 \quad (99)$$

$$\varphi_L(t) = \beta(q)\epsilon_L(t)$$

Desde que (99) seja quadrática em θ o vetor de parâmetros $\hat{\theta}$ que minimiza esta função custo pode ser calculado por:

$$\hat{\theta} = \left[\sum_{t=1}^N \varphi_L(t)\varphi_L(t)^T \right]^{-1} \sum_{t=1}^N \varphi_L(t)u_L(t) \quad (100)$$

Com o intuito de exemplificar, na Figura 15 é apresentada uma curva da função custo $J_y(\theta)$ de um sistema hipotético, e a função custo proposta pelo método VRFT $J_{VR}(\theta)$, a qual é quadrática em θ . Ao tentar encontrar o ponto de mínimo da curva $J_y(\theta)$ pode-se recair em algum mínimo local, o que não acontece para a curva $J_{VR}(\theta)$.

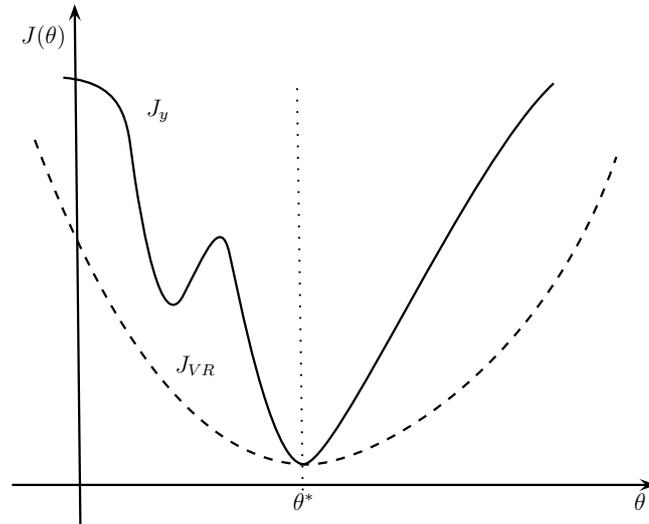


Figura 15: Gráfico da função custo de algum sistema hipotético e a respectiva função custo proposta pelo método VRFT que é mais simples de encontrar o ponto de mínimo pois é quadrática em θ , não recaindo em mínimos locais. O valor θ^* é o ponto de mínimo de ambas as funções custo, logo, minimizando a função custo $J_{VR}(\theta)$ é o equivalente a minimizar $J_y(\theta)$ sob condições ideais.

4.2.2 Utilização do Filtro $L(q)$

Considerando que a função custo $J_y(\theta)$ apresentada em (88) e o critério do método de referência virtual $J_{VR}(\theta)$ apresentado em (99) são diferentes fora de situações ideais, a escolha correta do filtro $L(q)$ propicia que estas duas equações tenham mínimos próximos (CAMPI; LECCHINI; SAVARESI, 2002).

A utilização do filtro é de grande importância em situações onde a escolha do modelo $C(q, \theta)$ não consegue representar exatamente o controlador ótimo ($C_d(q)$), ou seja:

$$C_d(q) \notin \mathcal{C}$$

Aplicando o teorema de Parseval a função custo de seguimento de referência:

$$J_y(\theta) = \bar{E} [(T(q, \theta) - T_d(q))r(t)]^2$$

e usando as relações de malha fechada:

$$T_d(q) = \frac{C_d(q)G_0(q)}{1 + C_d(q)G_0(q)} \quad T(q, \theta) = \frac{C(q, \theta)G_0(q)}{1 + C(q, \theta)G_0(q)}$$

Obtém-se a seguinte expressão no domínio da frequência para a função custo:

$$J_y(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{G_0(e^{j\omega})C(e^{j\omega}, \theta)}{1 + G_0(e^{j\omega})C(e^{j\omega}, \theta)} - \frac{G_0(e^{j\omega})C_d(e^{j\omega})}{1 + G_0(e^{j\omega})C_d(e^{j\omega})} \right|^2 \Phi_r(e^{j\omega}) d\omega$$

Desenvolvendo esta expressão, pode-se descrevê-la mais compactamente como:

$$J_y(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |G_0(e^{j\omega})|^2 |S(e^{j\omega}, \theta)|^2 |S_d(e^{j\omega})|^2 \times |C(e^{j\omega}, \theta) - C_d(e^{j\omega})|^2 \Phi_r(e^{j\omega}) d\omega \quad (101)$$

Adiciona-se o filtro $L(q)$ à função custo do VRFT, $J_{VR}(\theta)$:

$$J_{VR}(\theta) = \bar{E} [L(q)(u(t) - C(q, \theta)\bar{\epsilon}(t))]^2$$

e usando as relações:

$$1 - T_d(e^{j\omega}) = S_d(e^{j\omega})$$

e

$$\frac{T_d(e^{j\omega})}{G_0(e^{j\omega})} = C_d(e^{j\omega})S_d(e^{j\omega})$$

pode-se então reescrever a função custo do VRFT como :

$$J_{VR}(\theta) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |L(e^{j\omega})|^2 \frac{|G_0(e^{j\omega})|^2 |S_d(e^{j\omega})|^2}{|T_d(e^{j\omega})|^2} \times |C_d(e^{j\omega}) - C(e^{j\omega}, \theta)|^2 \Phi_u(e^{j\omega}) d\omega \quad (102)$$

Comparando agora (101) e (102), observa-se que quando $C_d(q) \in \mathcal{C}$ as duas funções custo possuem o mesmo mínimo. Quando $C_d(q) \notin \mathcal{C}$ nenhum dos critérios desaparece e o ponto de mínimo depende de vários fatores dentro da integral. Se os fatores dentro dos dois integrandos são diferentes, não existe motivos para esperar que os dois custos sejam os mesmos (BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

Existe entretanto o parâmetro livre que foi incluído à função custo do VRFT que pode ser escolhido a fim de aproximar estes dois integrandos: o filtro $L(q)$. Para fazer com que $J_{VR}(\theta) = J_y(\theta)$, é suficiente escolher $L(e^{j\omega})$ como:

$$|L(e^{j\omega})|^2 = |T_d(e^{j\omega})|^2 |S(e^{j\omega}, \theta)|^2 \frac{\Phi_r(e^{j\omega})}{\Phi_u(e^{j\omega})}, \quad \forall \omega \in [-\pi; \pi] \quad (103)$$

onde $\Phi_r(e^{j\omega})$ representa o espectro do sinal real da referência $r(t)$ e $\Phi_u(e^{j\omega})$ representa o espectro do sinal aplicado sobre o sistema, medido durante o experimento do VRFT.

O cálculo do filtro apresentado em (103) requer do conhecimento da função $S(q, \theta)$, que não é conhecida. Então a implementação deste filtro irá recair em alguma aproximação desta função de transferência. Quanto melhor for a aproximação, mais próximos serão os mínimos das funções custo $J_y(\theta)$ e $J_{VR}(\theta)$ (BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

Entretanto esta não é a única escolha para a aproximação de $S(q, \theta)$; o método VRFT usa invariavelmente o seguinte:

$$|S(e^{j\omega}, \theta)|^2 \approx |S_d(e^{j\omega})|^2 = |1 - T_d(e^{j\omega})|^2 \quad (104)$$

o que aparenta ser uma aproximação razoável, uma vez que é esperado que as duas funções sensibilidades em (104) sejam próximas ao redor do ponto de mínimo. Usando esta aproximação, a função de transferência do filtro pode ser obtida por:

$$|L(e^{j\omega})|^2 = |T_d(e^{j\omega})|^2 |1 - T_d(e^{j\omega}, \theta)|^2 \frac{\Phi_r(e^{j\omega})}{\Phi_u(e^{j\omega})}, \quad \forall \omega \in [-\pi; \pi] \quad (105)$$

Para o caso onde $C_d \in C(q, \theta)$ a escolha de $L(q)$ não afeta o resultado, usualmente escolhe-se então $L(q) = 1$ (CAMPI; LECCHINI; SAVARESI, 2002).

Com o filtro $L(q)$ calculado como em (105), o valor assintótico de θ pode ser escrito como:

$$\theta_* = \bar{E} [\varphi_L(t) \varphi_L(t)^T]^{-1} \bar{E} [\varphi_L(t) u_L(t)] \quad (106)$$

onde $\varphi_L(t) = L(q)\varphi(t)$ e $u_L(t) = L(q)u(t)$. O cálculo real para obter os parâmetros da estimativa é dado por:

$$\theta = \left[\sum_{t=1}^N \varphi_L(t) \varphi_L^T(t) \right]^{-1} \sum_{t=1}^N \varphi_L(t) u_L(t) \quad (107)$$

e na ausência de ruído, θ é igual ao valor de θ_* apresentado em (106).

Na Figura 16 um exemplo hipotético é apresentado onde o valor de θ que minimiza $J_{VR}(\theta)$ se aproxima do valor de mínimo de $J_y(\theta)$ com a utilização do filtro $L(q)$. Também observa-se que a curva de $J_{VR}(\theta)$ quando o filtro é aplicado apresenta uma redução na sua concavidade, fazendo com que o valor de mínimo na curva encontrado pelo algoritmo seja menos preciso, aumentando a variância das estimativas, enquanto reduz o erro de polarização.

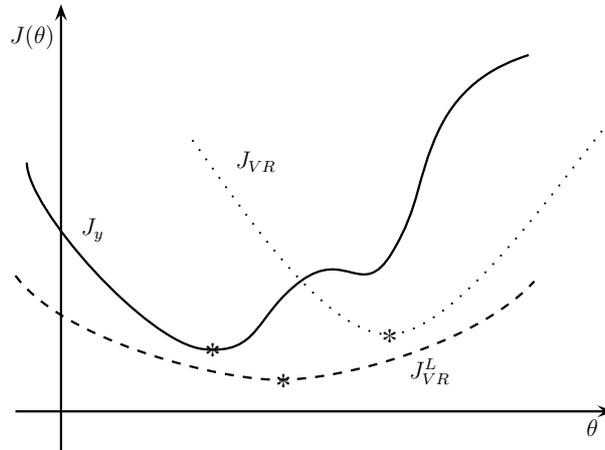


Figura 16: Exemplo hipotético de funções custo na situação onde o controlador ideal não consegue ser representado pelo modelo. O uso do filtro $L(q)$ reduz o erro de polarização.

4.2.3 Dados corrompidos por ruído

Nesta seção será apresentado brevemente o comportamento do método quando o sinal $y(t)$ é corrompido por um ruído aditivo como:

$$\tilde{y}(t) = G_0(q)u(t) + \nu(t) \quad (108)$$

Assume-se que os sinais $u(t)$ e $\nu(t)$ sejam decorrelacionados e também que os dados são coletados com a planta trabalhando em laço aberto (CAMPI; LECCHINI; SAVARESI, 2002; LECCHINI, 2001).

Ao aplicar o algoritmo descrito na Seção 4.2.1 com dados sujeitos a ruídos, o resultado obtido possui erro de polarização. Isso pode ser compreendido analisando a expressão do critério $J_{VR}(\theta)$ quando utiliza-se o sinal $\tilde{y}(t)$:

$$\begin{aligned} J_{VR}(\theta) = & \frac{1}{2\pi} \int_{-\pi}^{\pi} |G_0(e^{j\omega})|^2 |C(e^{j\omega}, \theta) - C_d(e^{j\omega})|^2 |1 - T_d(e^{j\omega})|^2 \frac{|L(e^{j\omega})|^2}{|T_d(e^{j\omega})|^2} \Phi_u(e^{j\omega}) \\ & + \frac{|C(e^{j\omega}, \theta)|^2}{|G_0(e^{j\omega})|^2 |C_d(e^{j\omega})|^2} |L(e^{j\omega})|^2 \Phi_d d\omega \end{aligned} \quad (109)$$

onde Φ_d é a densidade do espectro do ruído.

Quando compara-se a equação (109) com a função custo do método sem a presença de ruído $J_{VR}(\theta)$ (equação (102)) observa-se que há a adição de um termo dentro da integral,

$$\frac{|C(e^{j\omega}, \theta)|^2}{|G_0(e^{j\omega})|^2 |C_d(e^{j\omega})|^2} |L(e^{j\omega})|^2 \Phi_d$$

e que este termo não decai a zero mesmo que $C_d(q) \in \mathcal{C}$. Desta forma é esperado que haja erro de polarização nas estimativas devido ao ruído mesmo quando a classe de controladores consegue descrever o controlador ideal.

Para que haja rejeição a este ruído no método, em (CAMPI; LECCHINI; SAVARESI, 2002) foi sugerido a adição da variável instrumental $\zeta(t)$. Em (110) apresenta-se o regressor deste instrumento:

$$\tilde{\varphi}_L(t) = \beta(q)L(q) (T_d(q)^{-1} - 1) \tilde{y}(t) \quad (110)$$

Os parâmetros do controlador podem então ser calculados como em:

$$\theta_N^{IV} = \left[\sum_{t=1}^N \zeta(t) \tilde{\varphi}_L(t)^T \right]^{-1} \left[\sum_{t=1}^N \zeta(t) u_L(t) \right] \quad (111)$$

São propostas duas alternativas para a escolha da variável instrumental. A primeira garante assintoticamente que $\theta_N^{IV} = \theta_0$, entretanto um experimento adicional é requisitado. O segundo não garante, rigorosamente, $\theta_N^{IV} = \theta_0$ mas o erro esperado é pequeno, além disso um segundo experimento não é necessário (CAMPI; LECCHINI; SAVARESI, 2002).

- *Experimento Repetido:* Executa-se um segundo experimento com a mesma entrada $u(t)$, adquirindo-se a saída $\tilde{y}(t)'$. Como o ruído entre um experimento e outro é independente, $\tilde{y}(t)$ e $\tilde{y}(t)'$ serão diferentes. Obtém-se então a variável instrumental:

$$\zeta(y) = \beta(q)L(q) (T_d(q)^{-1} - 1) \tilde{y}(t)'$$

- *Identificação da planta:* Identifica-se a planta $G_0(q)$ com uma família de modelos como $G(q, \theta_G)$ a partir do conjunto de dados $\{u(t), \tilde{y}(t)\}_{t=1, \dots, N}$ e então gera-se o sinal simulado $\hat{y} = G(q, \theta_G)u(t)$, obtendo a variável instrumental como:

$$\zeta(y) = \beta(q)L(q) (T_d(q)^{-1} - 1) \hat{y}(t)$$

Devido às incertezas na estimativa de $G(q, \theta_G)$, este segundo método não garante que a estimativa tenda assintoticamente a θ_0 .

4.2.4 Exemplos ilustrativos

Nesta seção serão apresentados alguns exemplos ilustrativos da utilização do método do VFRT. Serão utilizados sistemas lineares modelados pelas classes de modelos ARX e BJ quando o controlador ideal $C_d(q)$ faz parte da classe de controladores $C(q, \theta)$ e também um caso onde o controlador ideal não consegue ser apresentado pela classe de controlador escolhida.

Nas identificações apresentadas a seguir será sempre utilizado um sinal PRBS de ordem 7.

4.2.4.1 Controlador PI - sistema Box-Jenkins

Considere o sistema real definido como:

$$G_0(q) = \frac{0.5}{q - 0.85}, \quad H_0(q) = \frac{q}{q - 0.4}$$

Este sistema pode ser compreendido como um sistema *Box-Jenkins* (BJ) (Tabela 1). Deseja-se que o sistema em malha fechada comporte-se de acordo com:

$$T_d(q) = \frac{0.4}{q - 0.6} \quad (112)$$

Tem-se assim que o controlador ideal é dado por:

$$C_d(q) = \frac{T_d(q)}{G_0(q)(1 - T_d(q))} = \frac{0.8(q - 0.85)}{q - 1} \quad (113)$$

Observa-se que este controlador pode ser representado como um controlador *PI* como em:

$$C(q, \theta) = \frac{\theta_1 q + \theta_2}{q - 1} \quad (114)$$

Escolheu-se (114) como sendo a classe de modelos a ser utilizada para a identificação do controlador.

Utilizando o método do VFRT para identificação do controlador quando este está submetido a um ruído com variância $\sigma_e^2 = 0.005$ obtém-se as estimativas dos parâmetros θ_1 e θ_2 apresentados na Figura 17 com um resultado de 100 simulações com $N = 2540$ cada.

Os parâmetros reais esperados para o controlador (equação (113)) e a média de todas as estimativas $\hat{\theta}_N$ (valor representado por uma estrela na Figura 17) não são os mesmos. Em uma situação onde o erro de polarização das estimativas não existe, o aumento de N (número de amostras) implica que esta diferença diminui, tendendo a zero. Em um cenário onde há erro de polarização, se aumentarmos a variância do ruído do sistema, será observado um aumento desta diferença.

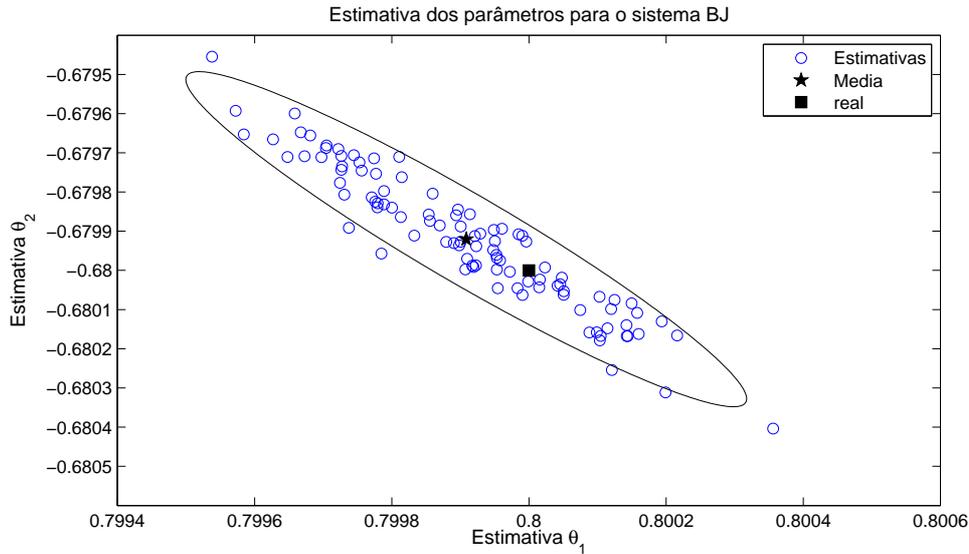


Figura 17: Resultado das 100 estimativas de Monte Carlo dos parâmetros θ_1 e θ_2 para o controlador apresentado em (114). Com $\sigma_e^2 = 0.005$

Na Figura 18 é apresentado o resultado para a estimativa de θ quando a variância do ruído é quadruplicada ($\sigma_e^2 = 0.02$). Observa-se então que o erro de polarização existe na estimativa. Como descrito em (CAMPI; LECCHINI; SAVARESI, 2002) quando o método do VRFT é utilizado com ruído nas amostras, a estimativa é inevitavelmente polarizada. Na Seção 4.2.3 foi sugerido o uso de variáveis instrumentais para que este erro de polarização seja minimizado. Utilizou-se então este método e para um ruído com a mesma variância ($\sigma_e^2 = 0.02$), o resultado obtido é o apresentado na Figura 19.

Observa-se que o erro de polarização foi minimizado e que o resultado obtido possui um custo $J_{VR}^N(\theta) = 5.1242$ e a variância dos parâmetros estimados foi de 0.5064×10^{-6} para θ_1 e de 0.5495×10^{-6} para θ_2 .

A fim de comparar o método VRFT utilizando e não utilizando variáveis instrumentais são apresentados abaixo as Tabelas 5 e 6 onde os custos J_y e J_{VR}^N são apresentados para diferentes valores de variância do ruído para o mesmo sistema BJ.

Utilizando variáveis instrumentais observa-se que o custo $J_y(\theta)$ é significativamente mais baixo quando comparado com o método onde não são utilizadas variáveis instrumentais, corroborando com o que foi apresentado na Seção 4.2.3.

4.2.4.2 Controlador PID - sistema ARX

Considere o sistema real definido pelo modelo ARX:

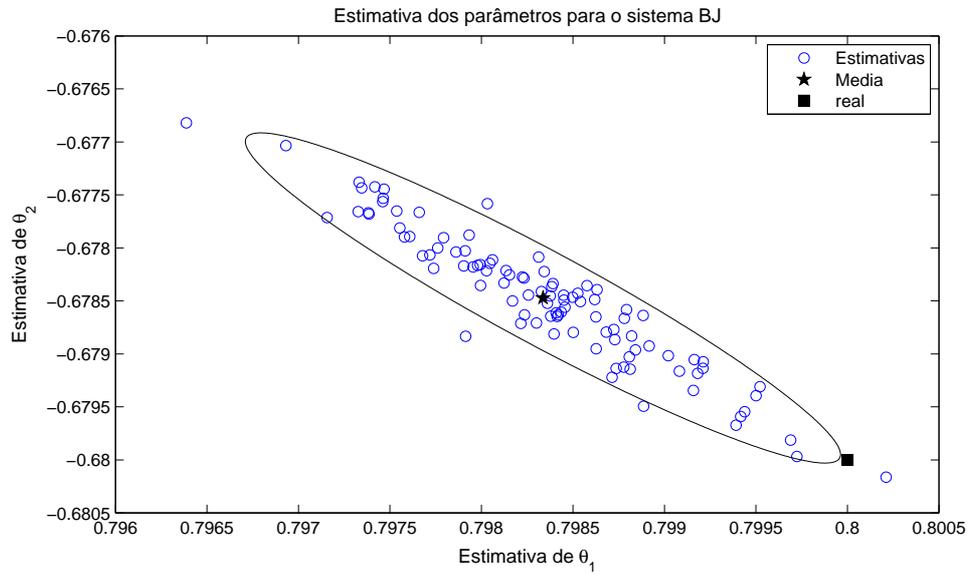


Figura 18: 100 estimativas Monte Carlo dos parâmetros θ_1 e θ_2 para o controlador apresentado em (114) com um ruído de variância de $\sigma_e^2 = 0.02$.

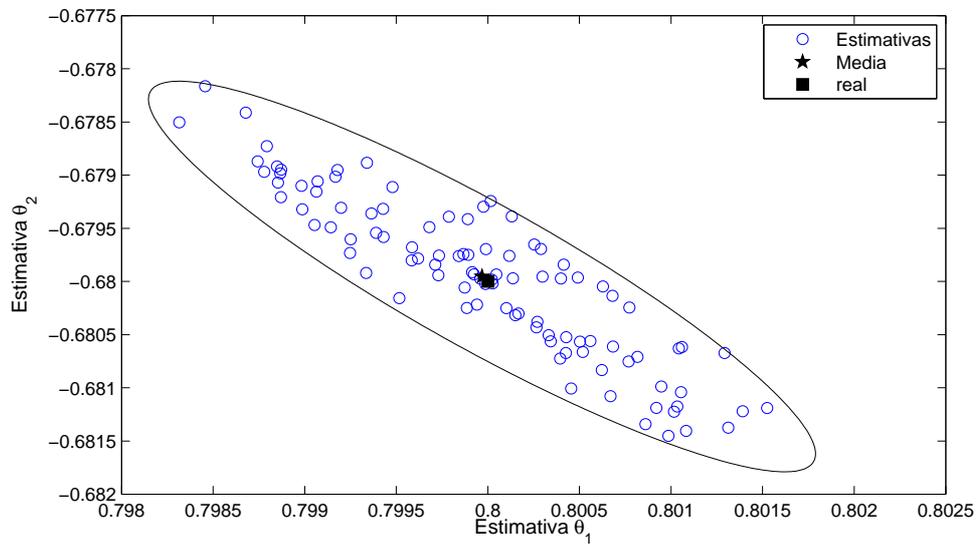


Figura 19: Resultado das 100 estimativas de Monte Carlo dos parâmetros θ_1 e θ_2 para o controlador apresentado em (114) com variância do ruído $\sigma_e^2 = 0.02$ utilizando variáveis instrumentais para estimar os parâmetros.

Tabela 5: Valor dos custos J_{VR}^N e J_y além da variância das estimativas para diferentes valores de σ_e^2 quando o método VRFT não utiliza variáveis instrumentais para a estimativa dos parâmetros θ

Variância σ_e^2	$J_{VR}^N(\theta)$	$J_y(\theta)$	Média θ_1	Média θ_2
0.1	4.87826×10^{-2}	2.15324×10^{-2}	0.76178	0.64466
0.08	3.26789×10^{-2}	1.35688×10^{-2}	0.7759	0.65765
0.06	1.80426×10^{-2}	7.86422×10^{-3}	0.78597	0.66715
0.05	1.19341×10^{-2}	5.52796×10^{-3}	0.79017	0.67088
0.04	7.54289×10^{-3}	3.65088×10^{-3}	0.79351	0.67396
0.01	4.89481×10^{-4}	2.11052×10^{-4}	0.79962	0.67965
0.008	3.25810×10^{-4}	1.83444×10^{-4}	0.79968	0.67969
0.005	1.20816×10^{-4}	5.42193×10^{-5}	0.7999	0.67991
0.003	4.73517×10^{-5}	2.28052×10^{-5}	0.79996	0.67996
0.001	5.02347×10^{-6}	2.77267×10^{-6}	0.8	0.68

Tabela 6: Valor dos custos J_{VR}^N e J_y além da variância das estimativas para diferentes valores de σ_e^2 quando o método VRFT utiliza variáveis instrumentais para a estimativa dos parâmetros θ

Variância σ_e^2	$J_{VR}^N(\theta)$	$J_y(\theta)$	Média θ_1	Média θ_2
0.1	5.19132×10^{-2}	4.14964×10^{-4}	0.79936	0.67971
0.08	3.31577×10^{-2}	2.96000×10^{-4}	0.80037	0.68007
0.06	1.70932×10^{-2}	2.17256×10^{-4}	0.79989	0.67969
0.05	1.27803×10^{-2}	3.14017×10^{-5}	0.80006	0.68005
0.04	7.72145×10^{-3}	9.43578×10^{-5}	0.79996	0.68006
0.01	5.00316×10^{-4}	3.20694×10^{-5}	0.80005	0.68003
0.008	3.44326×10^{-4}	2.93508×10^{-5}	0.80001	0.68004
0.005	1.19855×10^{-4}	6.96916×10^{-6}	0.79999	0.68
0.003	4.22380×10^{-5}	6.33696×10^{-6}	0.80001	0.68001
0.001	5.08709×10^{-6}	1.18016×10^{-6}	0.8	0.68

$$G_0(q) = \frac{q}{(q - 0.9)(q - 0.8)}, \quad H_0(q) = \frac{q^2}{(q - 0.9)(q - 0.8)}$$

Deseja-se que o sistema em malha fechada comporte-se como:

$$T_d(q) = \frac{0.4}{q - 0.6} \quad (115)$$

Tem-se assim que o controlador ideal é definido por:

$$C_d(q) = \frac{T_d(q)}{G_0(z)(1 - T_d(q))} = \frac{0.4(q - 0.9)(q - 0.8)}{q(q - 1)} \quad (116)$$

Observa-se que este controlador pode ser representado como um controlador *PID* como em:

$$C(q, \theta) = \frac{\theta_1 q^2 + \theta_2 q + \theta_3}{q(q - 1)} \quad (117)$$

O vetor de parâmetros θ_0 deste sistema é dado por $\theta_0 = [0.4 \quad -0.68 \quad 0.288]$.

Como já foi observado, a utilização de variáveis instrumentais elimina o erro de polarização existente nas estimativas do método VRFT quando há presença de ruído. Desta forma as informações apresentadas a seguir serão feitas utilizando variáveis instrumentais. Na Figura 20 é apresentada a estimativa dos parâmetros do controlador para um ruído de variância $\sigma_e^2 = 0.05$. Observa-se que não há erro de polarização nas estimativas. O custo para estas e outras, estimativas são apresentados na Tabela 7.

Na Tabela 7 observa-se que o custo $J_y(\theta)$ aumenta com a quantidade de ruído inserido sobre o sistema, mas observa-se que não há erro de polarização nas estimativas. O valor médio das estimativas está em todas as simulações muito próximo ao valor real θ_0 almejado.

4.2.4.3 Controlador não pertence à classe

Até este ponto foram apresentados exemplos de uso do método VRFT quando o controlador que leva o sistema para o comportamento desejado $T_d(q)$ faz parte da classe escolhida para a identificação. Nesta seção será apresentado um exemplo onde a classe de modelos do controlador escolhida não consegue representar o controlador ideal $C_d(q)$. Considere para isso o sistema real descrito por:

$$G_0(q) = \frac{0.2(q - 0.7)}{(q - 0.9)(q - 0.5)}, \quad H_0(q) = \frac{q}{q - 0.3}$$

Deseja-se que em malha fechada ele se comporte como em:

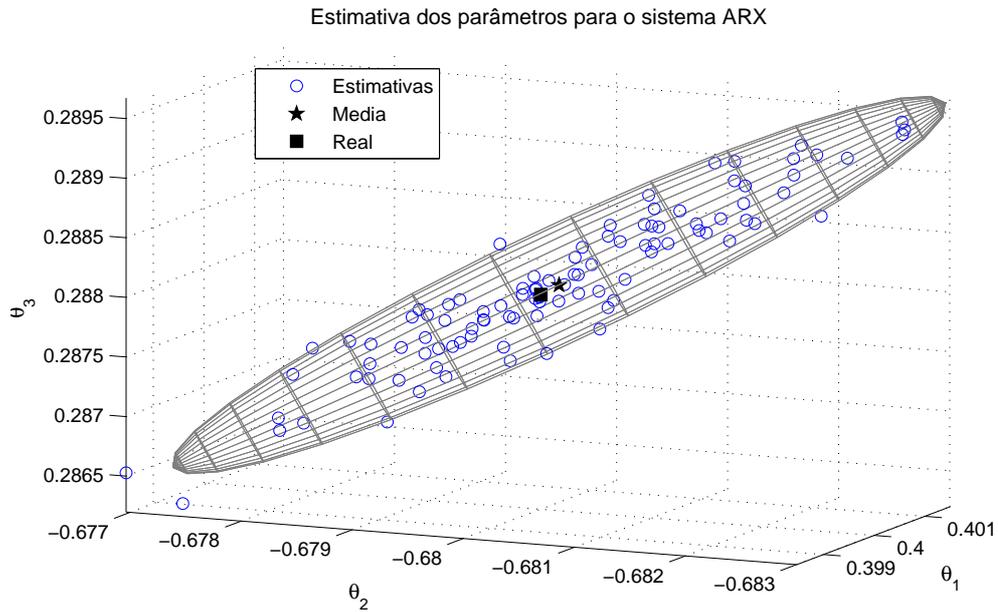


Figura 20: 100 estimativas Monte Carlo dos parâmetros θ_1 , θ_2 e θ_3 para o controlador apresentado em (117) com variância do ruído $\sigma_e^2 = 0.05$ utilizando variáveis instrumentais

Tabela 7: Valor dos custos J_{VR}^N e J_y além da variância das estimativas para diferentes valores de σ_e^2 quando o método VRFT utiliza variáveis instrumentais para a estimativa dos parâmetros θ do controlador (117)

Variância σ_e^2	$J_{VR}^N(\theta)$	$J_y(\theta)$	Média θ		
			θ_1	θ_2	θ_3
0.1	1.02754×10^{-2}	2.21620×10^{-3}	0.39996	-0.67993	0.28796
0.08	6.17981×10^{-3}	2.54891×10^{-3}	0.39998	-0.67996	0.28799
0.06	3.66467×10^{-3}	2.48678×10^{-3}	0.40014	-0.68003	0.28814
0.05	2.66423×10^{-3}	1.13734×10^{-3}	0.40003	-0.68002	0.28802
0.04	1.76864×10^{-3}	1.22744×10^{-3}	0.39991	-0.6798	0.2879
0.01	1.06472×10^{-4}	3.01036×10^{-4}	0.40001	-0.68	0.288
0.008	6.19905×10^{-5}	1.74299×10^{-4}	0.40001	-0.68002	0.28801
0.005	2.47368×10^{-5}	1.02340×10^{-4}	0.4	-0.68001	0.28801
0.003	8.58597×10^{-6}	7.13811×10^{-5}	0.4	-0.68	0.288
0.001	1.07610×10^{-6}	2.72957×10^{-5}	0.4	-0.68	0.288

$$T_d(q) = \frac{0.16q}{(q - 0.6)^2} \quad (118)$$

Neste caso o controlador ideal é definido por

$$C_d(q) = \frac{0.8q(q - 0.9)(q - 0.5)}{(q - 1)(q - 0.36)(q - 0.7)} \quad (119)$$

Para esta identificação optou-se por um controlador do tipo PID como em:

$$C(q, \theta) = \frac{\theta_1 q^2 + \theta_2 q + \theta_3}{q(q - 1)} \quad (120)$$

Utilizando o procedimento descrito na Seção 4.2.3 e o procedimento de experimento repetido, foram feitos 100 experimentos de Monte Carlo e o resultado obtido para a média das estimativas foi:

$$\theta_L = [0.8101 \quad -0.1691 \quad -0.3358]^T$$

onde o índice L indica que este resultado foi obtido utilizando-se o filtro L .

Repetindo a simulação, mas agora sem que o procedimento da utilização do filtro L descrito na Seção 4.2.3, obteve-se o resultado seguinte:

$$\theta = [0.5846 \quad -0.2108 \quad -0.1525]^T$$

Aplicando-se estes resultados ao controlador apresentado em (120) e de posse do comportamento desejado para o sistema em malha fechada ($T_d(q)$) é possível fazer um comparativo da resposta ao salto unitário para o sistema utilizando os dois controladores obtidos. O resultado é apresentado na Figura 21.

Observa-se que para o sistema que utiliza o controlador estimado utilizando-se o filtro L , a resposta ao degrau unitário tem menos erro que o sistema utilizando o outro controlador, ficando este primeiro muito mais próximo da função $T_d(q)$ desejada. Os custos destes dois sistemas são apresentados na Tabela 8.

Tabela 8: Valor dos custos J_{VR}^N e J_y para o sistema controlado por $C(q)$ e $C_L(q)$

Controlador	$J_{VR}^N(\theta)$	$J_y(\theta)$
$C(q)$	0.2877	0.1270
$C_L(q)$	0.4481	0.0542

A fim de comparar as duas estimativas, na Figura 22 é apresentado o diagrama de Bode dos controladores obtidos (utilizando a média das estimativas obtidas).

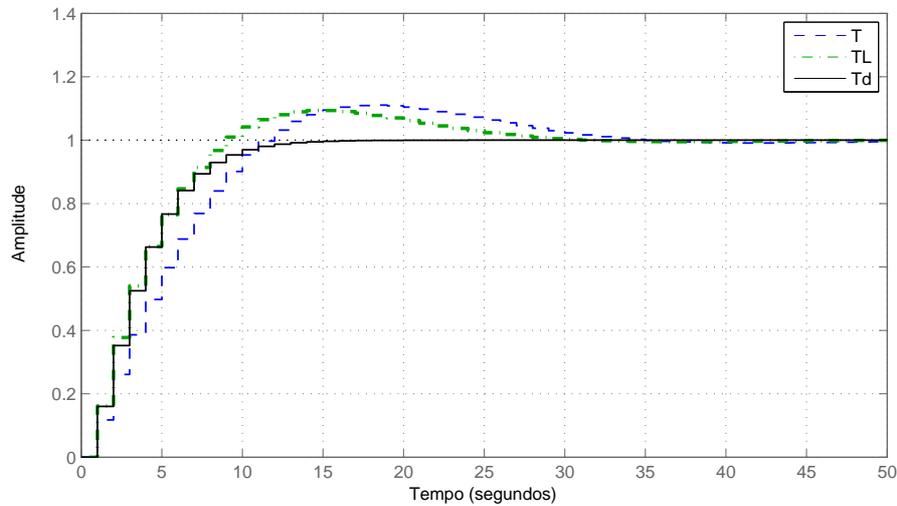


Figura 21: Comparativo da resposta do sistema a um degrau unitário quando o controlador inserido é obtido pelo método VRFT utilizando o filtro L e quando não se utiliza este artifício.

4.3 Considerações Finais

Projeto de controladores que são baseados nos dados obtidos a partir da planta, seja em funcionamento normal ou em experimentos específicos são uma vasta fonte de estudo há mais de 20 anos. Neste período diversos métodos foram propostos e desenvolvidos para se adaptarem as diversas necessidades e finalidades.

Neste capítulo foi apresentada uma breve descrição de alguns destes métodos e indicadas algumas de suas referências na literatura. Apresentou-se um dos princípios básicos que permeiam muitos destes métodos: a função custo baseada em um ou mais dos três objetivos básicos de controle: seguimento de referência, rejeição ao ruído, e uso econômico do esforço de controle.

Deu-se entretanto um enfoque maior ao método VRFT. A razão para a escolha deste em detrimento dos outros métodos citados, não está relacionado com nenhuma vantagem intrínseca que este tenha ou que este seja um método mais completo e abrangente que outros. O motivo foi pelo uso que ele faz de referência virtual para obter os sinais necessários à identificação do controlador ótimo, ideia que será generalizada no próximo capítulo para controladores não lineares.

Apresentou-se também diversos exemplos do uso do método VRFT, situações onde o controlador ótimo faz parte da classe de modelos escolhida, indicando seu desempenho em obter o controlador ótimo para diversos níveis de ruído sobre o sistema. Obteve-se bons resultados, quando utilizado o método das variáveis

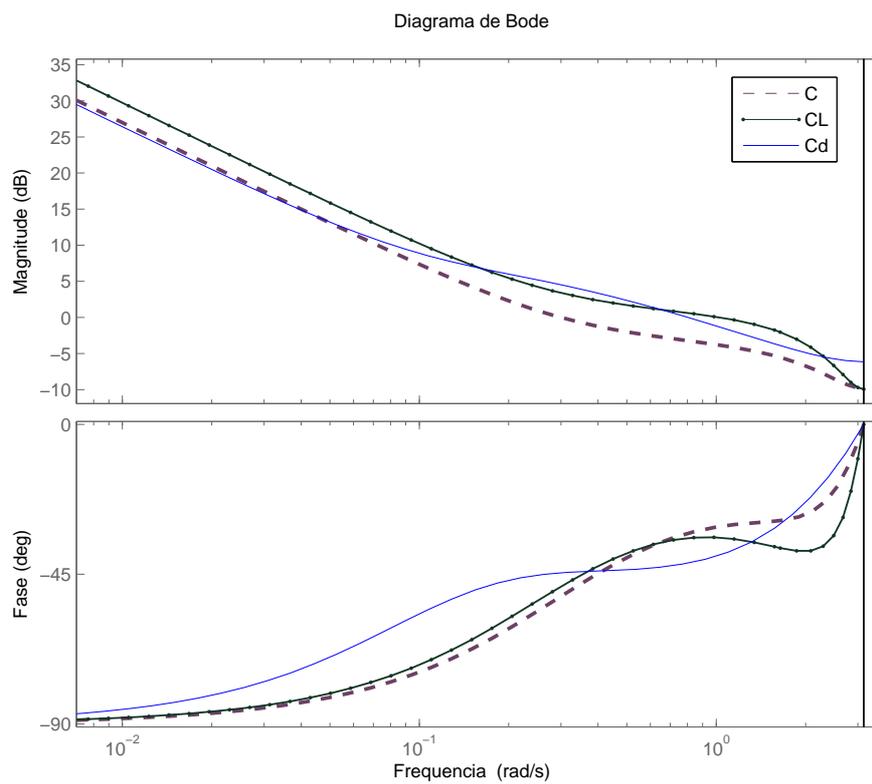


Figura 22: Diagrama de Bode para as funções de transferência dos controladores estimados utilizando VRFT com e sem o filtro $L(q)$ e variáveis instrumentais.

instrumentais, para remover o erro de polarização das estimativas.

A situação onde o controlador ótimo não consegue ser descrito pela classe de modelos também foi apresentada. Observou-se que os resultados obtidos foram bastante satisfatórios, conseguindo-se uma boa aproximação do comportamento esperado para o sistema em malha fechada.

De forma geral este capítulo apresentou a ideia de referência virtual para projetos de controladores baseados em dados e para isso usou-se do método VRFT para descrevê-lo e exemplificá-lo. Em seguida serão apresentadas algumas modificações para que esta ideia de referência virtual possa ser utilizada para projeto de controladores não lineares.

5 IDENTIFICAÇÃO DE CONTROLADORES NÃO LINEARES UTILIZANDO REFERÊNCIA VIRTUAL

Este capítulo apresenta a união das ideias de referência virtual para projetos de controladores e algoritmos de identificação de sistemas não lineares.

No Capítulo 3 apresentou-se diversas estruturas de classes de modelos e em especial a estrutura de modelos polinomiais e racionais. Foi apresentado também (Seção 3.3.1) um algoritmo para identificação deste tipo de classe de modelos. Naquele capítulo deu-se ênfase ao uso deste algoritmo para identificação de sistemas não lineares. Na Seção 5.3 será apresentado o uso deste algoritmo para identificação de controladores.

Para gerar os sinais de entrada e saída do algoritmo de identificação de modelos racionais, será utilizada a ideia de referência virtual, introduzida na Seção 4.2 com o método VRFT.

Serão apresentados nas Seções 5.4 e 5.5 alguns resultados obtidos com a metodologia proposta, quando o controlador possui uma não linearidade estática e quando a não linearidade está na dinâmica do controlador, respectivamente. Os exemplos têm o objetivo de cobrir as situações mais corriqueiras para identificação de controladores não lineares racionais além de prover os resultados para que possa ser validada a utilidade do método.

No Apêndice C são apresentadas as rotinas em Matlab desenvolvidas para os exemplos e simulações apresentadas neste capítulo.

5.1 Introdução

O projeto de controladores descrito no Capítulo 4 tem como base a identificação de controladores lineares. Encontram-se poucos trabalhos na literatura que abordam identificação de controladores não lineares baseados nos dados coletados da planta. Em (GUARDABASSI; SAVARESI, 1997) é apresentado um procedimento para linearização de sistemas não lineares utilizando a ideia de referência virtual. Em (CAMPI; SAVARESI, 2006) é apresentada uma generalização do mé-

todo VRFT que envolve a identificação de controladores não lineares. Seguindo nesta linha, existem diversos trabalhos propondo metodologias e algoritmos para a identificação de sistemas não lineares para inúmeras famílias de modelos. Em (CHEN; BILLINGS, 1989b) é apresentada uma metodologia para a estimativa de parâmetros de sistemas não lineares, de forma recursiva. Os mesmos autores no mesmo ano apresentaram um trabalho sobre identificação de sistemas não lineares utilizando classes de modelos racionais (CHEN; BILLINGS, 1989a). Em (HJALMARSSON; MÅRTENSSON, 2012) é apresentada uma modelagem de ordem finita de um modelo Hammerstein e sua acurácia, modelagem esta que serve como referência para parte do desenvolvimento que será apresentado na Seção 5.4.

Neste trabalho tem-se o intuito de, baseado em referência virtual, determinar os sinais de entrada para a utilização de algum algoritmo de identificação de sistemas não lineares, para com isso determinar qual é o controlador ótimo para que o sistema se comporte em malha fechada como desejado.

5.2 Definições

No Capítulo 2 foram apresentadas as definições para identificação de sistemas lineares. Definiu-se o sistema real pelo símbolo \mathcal{S} , neste este capítulo \mathcal{S}_{nl} é um sistema não linear e para o projeto de controladores, o controlador ideal é definido por \mathcal{C}_d , sendo o controlador que consegue fazer com que o sistema em malha fechada se comporte como desejado.

Define-se a classe de controladores não lineares, parametrizável em θ , como:

$$\mathcal{C} \tag{121}$$

Diz-se que o controlador ideal pode ser representado pela classe de modelos, ou seja, $\mathcal{C}_d \in \mathcal{C}$, se existir um valor de θ_0 que faça com que:

$$\mathcal{C} = \mathcal{C}_d \tag{122}$$

5.3 Método

O método proposto consiste na ideia de utilizar referência virtual para, a partir dos dados coletados do sistema, determinar os sinais de entrada do controlador que fazem com que o sistema em malha fechada se comporte como desejado: $T_d(q)$.

Parte-se do pressuposto de que o processo \mathcal{S}_{nl} é não linear, mas que o comportamento em malha fechada desejado $T_d(q)$ seja linear, portanto o controlador projetado deve, para cumprir com os objetivos de desempenho determinados, cancelar as não linearidades da planta para que possa ser possível uma relação linear entre a entrada $r(t)$ e a saída $y(t)$.

Analogamente ao que foi apresentado na Seção 4.2.1 para controladores lineares, o controlador não linear ideal \mathcal{C}_d quando operando em conjunto com o sistema não linear \mathcal{S}_{nl} resulta um sistema em malha fechada cuja função de transferência é dada por $T_d(q)$. Opta-se que este comportamento em malha fechada seja linear. Desta forma, se $T_d(q)$ for excitado com qualquer sinal $r(t)$, sua saída será $T_d(q)r(t)$.

Baseado no sinal $y(t)$ medido da saída do processo, considera-se um sinal $\bar{r}(t)$ tal que $T_d(q)\bar{r}(t) = y(t)$. Esta referência, como já foi apresentada, é conhecida como *virtual* pois ela não existe e não foi utilizada para gerar o sinal $y(t)$. A Figura 23 apresenta o sistema em malha fechada e os sinais presentes.

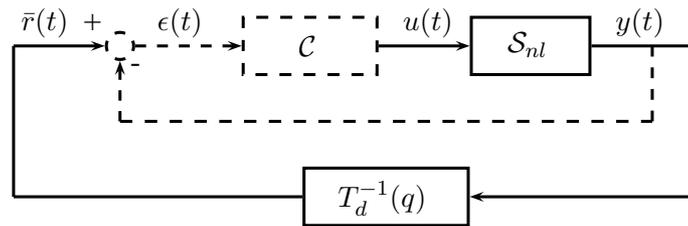


Figura 23: Representação em blocos do sistema em malha fechada para obtenção dos sinais $\epsilon(t)$ e $\bar{r}(t)$ utilizando referência virtual. Tracejado está a realimentação e o controlador, que não estão presentes no procedimento para obtenção dos sinais.

O sinal $\epsilon(t)$ é o erro entre os sinais $y(t)$ e $\bar{r}(t)$. Sabe-se que quando a planta é excitada com o sinal $u(t)$, o sinal $y(t)$ é então obtido. Analogamente para um controlador, quando este é excitado com o sinal $\epsilon(t)$, o sinal $u(t)$ é obtido. A metodologia proposta, analogamente ao método VRFT consiste em encontrar um controlador, com os sinais $u(t)$ e $\epsilon(t)$ que são conhecidos. A tarefa então reduz-se a um problema de identificação de sistemas não lineares.

Fica a critério do projetista, assim como para métodos lineares, determinar a estrutura da classe de modelos a ser utilizada para a identificação do controlador. Para que o comportamento linear $T_d(q)$ possa ser obtido em malha fechada há a necessidade de que o controlador $\mathcal{C}_d \in \mathcal{C}$.

Não é apenas devido à planta ser não linear e o comportamento em malha fechada ser linear que o controlador ideal possuirá um comportamento não linear. Pode-se, a partir de um comportamento em malha fechada não linear e com uma planta linear, ou não, obter-se um controlador que também seja não linear, dependendo das especificações que ao qual o controlador é submetido.

5.3.1 Classe do modelo

A escolha da classe de modelos \mathcal{C} que irá representar o controlador ideal \mathcal{C}_d é de grande importância para que haja sucesso no processo de identificação deste controlador. Neste trabalho optou-se pelo uso da classe de modelos racional apre-

sentada na Seção 3.2.5 pela abrangência desta classe de modelos em representar sistemas reais e pelos bons resultados obtidos na identificação de sistemas não lineares apresentados na Seção 3.3.2.

Aplica-se então o algoritmo proposto na Seção 3.3.1 para a identificação do controlador \mathcal{C} .

É conveniente observar que a escolha da classe está intimamente ligada ao uso do algoritmo de identificação destes modelos ter-se mostrado eficiente, como foi apresentado nos exemplos da Seção 3.3.2, além dos motivos já citados da vasta gama de sistemas reais que modelos racionais conseguem representar. Todavia, a abordagem por referência virtual apresentada poderia ser aplicada para qualquer classe de modelos de sistemas não lineares, onde algum algoritmo para identificação deste tipo de classe pudesse ser utilizado para obtenção dos valores de $\hat{\theta}$ que minimizem algum critério de escolha.

A qualidade das estimativas obtidas é bastante relacionada com a qualidade do algoritmo em estimar os valores de θ que minimizam alguma função custo. Entretanto tudo que foi abordado referente à identificabilidade de um experimento e a ordem de excitação de um sinal, determinam as condições básicas para que o resultado obtido seja livre de erro de polarização.

5.4 Wiener / Hammerstein

Nesta seção serão apresentados os resultados obtidos e os procedimentos para a identificação de controladores não lineares modelados por classe de modelos polinomiais quando a planta \mathcal{S}_{nl} é um sistema não linear do tipo Wiener. Será utilizado o conceito de referência virtual para a obtenção dos sinais necessários para que o projeto de identificação deste controlador possa ser realizado.

Considere que o sistema real possui uma não linearidade estática do tipo Wiener, onde o bloco não linear encontra-se na saída do processo. Como descrito em:

$$z(t) = G_0(q)u(t) + H_0(q)e(t) \quad (123)$$

$$y(t) = \zeta(z(t)) \quad (124)$$

onde $G_0(q)$ e $H_0(q)$ são funções de transferência racionais e o mapa $\zeta(\cdot) : \mathcal{D} \rightarrow \mathcal{D}$, com $\mathcal{D} \subset \mathfrak{R}$, é analítico. A Figura 24 representa este sistema.

Além disso é assumido que a função $\zeta(\cdot)$ é inversível em todo range \mathcal{D} , ou seja, existe uma função $\zeta_R^{-1}(\cdot)$ tal que $\zeta(\zeta_R^{-1}(x)) = x \quad \forall x \in \mathcal{D}$.

Não é difícil de perceber que para este caso o controlador ideal é um sistema Hammerstein, ou seja:

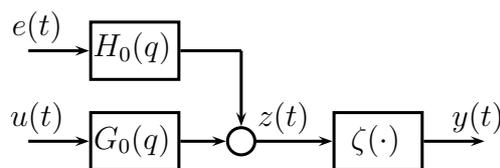


Figura 24: Um sistema do tipo Wiener como descrito por (123) e (124)

$$u(t) = C'_d(q)v(t) \quad (125)$$

$$v(t) = r(t) - \gamma(y(t)) \quad (126)$$

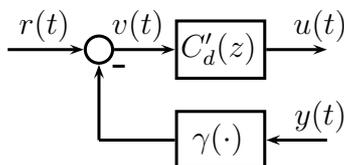


Figura 25: Um controlador do tipo Hammerstein como descrito por (125) e (126)

Realmente, se $\gamma(\cdot) = \zeta_R^{-1}(\cdot)$ e

$$C'_d(q) = \frac{T_d(q)}{G_0(q)(1 - T_d(q))} \quad (127)$$

então o sistema em malha fechada é linear com uma função de transferência $T_d(q)$.

Na prática esta é uma solução padrão de controle, pois controladores comerciais normalmente incluem customizáveis funções de linearização $\gamma(\cdot)$ no laço de realimentação. Fica claro que mesmo neste caso padrão o controlador ideal não é uma função do erro de referência $r(t) - y(t)$, mas sim uma função de $r(t)$ e de $y(t)$ separadamente.

Assumindo que a referência é uma constante conhecida, uma topologia alternativa pode ser elaborada com um controlador que é dependente apenas do erro de referência. A classe de modelos deste controlador pode ser definido como:

$$u(t) = C'(q, \theta)v(t) \quad (128)$$

$$v(t) = \gamma(r(t) - y(t), \eta) \quad (129)$$

onde a função $\gamma(\cdot, \cdot)$ deve ser tal que o sistema com o controlador em malha fechada deve se comportar como se nenhuma das não linearidades estivesse presente, ou seja:

$$\gamma(r(t) - y(t)) = \alpha[r(t) - z(t)] \quad (130)$$

para algum $\alpha \in \mathfrak{R}^+$.

A Figura 26 mostra um sistema com uma não linearidade do tipo Wiener e um controlador descrito como anteriormente.

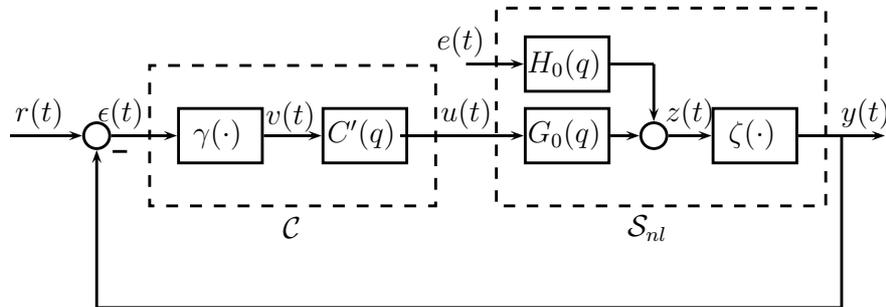


Figura 26: Diagrama de blocos para um sistema não linear do tipo Wiener

Ao fazê-lo, o controlador ideal depende apenas do erro de referência, mas a função $\gamma(\cdot)$ depende do valor de referência, e será uma função diferente para cada valor diferente da referência. Além disso, para valores não constantes, a não linearidade inversa $\gamma(\cdot)$ pode não ser autônoma, violando outra pressuposição da formulação não linear do VRFT proposta em (CAMPI; SAVARESI, 2006).

Agora, entretanto é desejado que o sistema se comporte linearmente em malha fechada como $T_d(q)$. Para obter-se isso o controlador deve cancelar todos os efeitos não lineares de $\zeta(z(t))$. Em outras palavras, é desejado que em malha fechada o sinal $v(t)$ obedeça a equação (130). Entretanto o sistema apresentado na Figura 26 se comporta como em:

$$v(t) = \gamma(r(t) - y(t))$$

e assim pode-se obter a relação entre $\gamma(\cdot)$ e $\zeta(\cdot)$ como:

$$\gamma(r(t) - \zeta(z(t))) = \alpha[r(t) - z(t)]. \quad (131)$$

Fica claro que quando não existe um sinal de referência, $r(t) \equiv 0$, a relação entre os polinômios $\gamma(\cdot)$ e $\zeta(\cdot)$ fica simplificada a:

$$\gamma(\zeta(z(t))) = -\alpha z(t)$$

então neste caso (mas apenas neste caso) $\gamma(\cdot)$ deve ser a inversa de $\zeta(\cdot)$, a menos de uma constante.

5.4.1 Convexidade

O ponto principal do paradigma de referência virtual é de deixar o problema de modelo de referência convexo. Isso é obtido no método VRFT por meio de uma parametrização linear do controlador e o projeto do controlador é obtido por meio da solução de um problema de mínimos quadrados.

Para aproveitar esta virtude computacional é necessário ter um controlador parametrizável linearmente. Desde que a inversa é assumidamente analítica, ela pode ser expandida em serie de Taylor e então truncada:

$$\zeta_R^{-1}(x) \cong \sum_{i=0}^k \frac{1}{i!} \frac{\partial^i \zeta_R^{-1}}{\partial x^i}(0) x^i$$

Se o controlador e a função não linear forem ambos parametrizáveis linearmente, ou seja,

$$C'(q, \rho) = \frac{1}{D(q)} \rho^T N(q) \quad (132)$$

$$\gamma(\epsilon(t)) = \eta^T E(t) \quad (133)$$

com $D(q)$ sendo um polinômio mônico conhecido, $N(q) \triangleq [1 \ q^{-1} q^{-2} \dots q^{-p}]^T$, e $E(t) \triangleq [\epsilon(t) \ \epsilon^2(t) \dots \epsilon^k(t)]^T$. Então, definindo $u_f(t) = D(q)u(t)$ e substituindo as equações (132) e (133) em (125) leva a:

$$u_f(t) = \sum_{i=0}^p \sum_{j=1}^k \theta_{i+j} \epsilon^j(t-i) \quad (134)$$

ou

$$u_f(t) = \theta^T \varpi(t) \quad (135)$$

onde $\theta_{i+j} \triangleq \rho_i \eta_j$ e o vetor $\theta = [\theta_1 \dots \theta_{p+k}]^T$ são os parâmetros a serem identificados. Note que um preço precisa ser pago para obter-se uma parametrização linear: o número de parâmetros que inicialmente era igual a $p+k$, agora é $p \times k$.

O exemplo numérico abaixo visa demonstrar este método proposto para o projeto de controladores do tipo Hammerstein.

5.4.2 Exemplo numérico

Considere uma planta não linear \mathcal{S}_{nl} do tipo Wiener como apresentado na Figura 26. Considere ainda que o bloco linear e não linear desta planta pode ser descrito como:

$$G'(q) = \frac{0.5}{q - 0.9} \quad (136)$$

e

$$\zeta(z(t)) = y(t) = 1.5z(t) + 0.2z^3(t) \quad (137)$$

Deseja-se que esta planta seja controlada para que o comportamento em malha fechada possa ser descrito como em:

$$T_d(q) = \frac{0.4}{q - 0.6} \quad (138)$$

O controlador é definido pela forma apresentada em (128) e (129) com $C(q, \theta)$ como um controlador do tipo proporcional integral (PI).

$$C(q, \theta) = \theta_1 \frac{1}{q - 1} + \theta_2 \frac{q}{q - 1} \quad (139)$$

e a função não linear sendo representada por um polinômio de quarta ordem como:

$$\gamma(\epsilon(t)) = v(t) = a_1\epsilon(t) + a_2\epsilon^2(t) + a_3\epsilon^3(t) + a_4\epsilon^4(t) \quad (140)$$

Considerando entretanto que nenhum dos blocos não lineares existissem no sistema, o controlador ideal poderia ser descrito pela equação (127), resultando em

$$C'_d(q) = \frac{0.8q - 0.72}{q - 1} \quad (141)$$

o qual pode ser exatamente descrito pela classe de controladores apresentada em (139).

A parte não linear do controlador ideal é a inversa da função não linear $\zeta(\cdot)$ dada em (137). Esta função será aproximada por uma série de Taylor truncada como em (140), o que significa, rigorosamente falando, que $C_d(q)$ não pode ser representado por $C(q, \theta)$, ou seja, $C_d(q) \notin C(q, \theta)$. Se por ventura o desempenho do controlador encontrado não for satisfatório, pode-se aumentar a ordem da série de Taylor utilizada para a aproximação, o que pode ser arbitrariamente feito até obter-se um desempenho próximo ao desejado.

O controlador $C'_d(q)$ possui um integrador em sua estrutura. Para evitar problemas de seguimento de referência optou-se por não identificar esta parte do controlador, mantendo o denominador como um integrador e identificando apenas o numerador.

Fazendo-se as substituições necessárias, chega-se à expressão do sinal de saída do controlador:

$$u_f(t) = \theta_1 \epsilon(t) + \theta_2 \epsilon^2(t) + \theta_3 \epsilon^3(t) + \theta_4 \epsilon^4(t) + \theta_5 \epsilon(t-1) + \theta_6 \epsilon^2(t-1) + \theta_7 \epsilon^3(t-1) + \theta_8 \epsilon^4(t-1) \quad (142)$$

onde $u_f(t)$ denota que $u(t)$ foi filtrado pelo denominador de $C'_d(q)$ em (141).

Foram realizados 100 experimentos de Monte Carlo adicionando um ruído com $\sigma_e^2 = 0.05$ e utilizando o algoritmo proposto em conjunto com o procedimento de referência virtual para obter os sinais $r(t)$ e $\epsilon(t)$ e a média das estimativas obtidas foi de:

$$\theta_{\text{média}} = \begin{bmatrix} 0.44717 \\ 2.0216 \times 10^{-3} \\ -6.5181 \times 10^{-4} \\ -1.608 \times 10^{-5} \\ -0.40435 \\ -1.6163 \times 10^{-3} \\ 6.1265 \times 10^{-4} \\ 1.469 \times 10^{-5} \end{bmatrix}^T$$

O custo obtido foi $J_y(\theta_{\text{média}}) = 3.0078 \times 10^{-3}$; isso significa que, a cada amostra, esta é a média da diferença entre o sinal de saída obtido e o desejado. Para as estimativas foram utilizados sinais de entrada do tipo PRBS com 127 pontos para a excitação da planta. Na Figura 27 é apresentado um comparativo entre o sinal de saída do sistema $T_d(q)$ quando submetido a um degrau unitário e o sinal do sistema real quando o controlador identificado com $\theta_{\text{médio}}$ é aplicado sobre a planta em malha fechada.

Na Figura 28 é apresentado o comportamento dos sinais de saída e entrada das não linearidades $\gamma(\epsilon(t))$ e $\zeta(z(t))$ quando o sistema é alimentado por uma referência unitária.

A relação entre os sinais $v(t)$ e $z(t)$ é apresentado na Figura 29. Observa-se que a relação destes sinais é afim, garantido desta forma que a aproximação feita em (140) foi suficiente para representar com acuracidade o que era esperado para $\gamma(t)$ em (131).

5.5 Controladores Racionais

Nesta seção serão apresentados exemplos de projeto de controladores não lineares utilizando a abordagem de referência virtual quando o controlador pode ser descrito por uma classe de modelos racional. Desta forma é possível obter os sinais necessários para identificação do controlador utilizando referência virtual e então aplicar diretamente o algoritmo proposto na Seção 3.3.1.

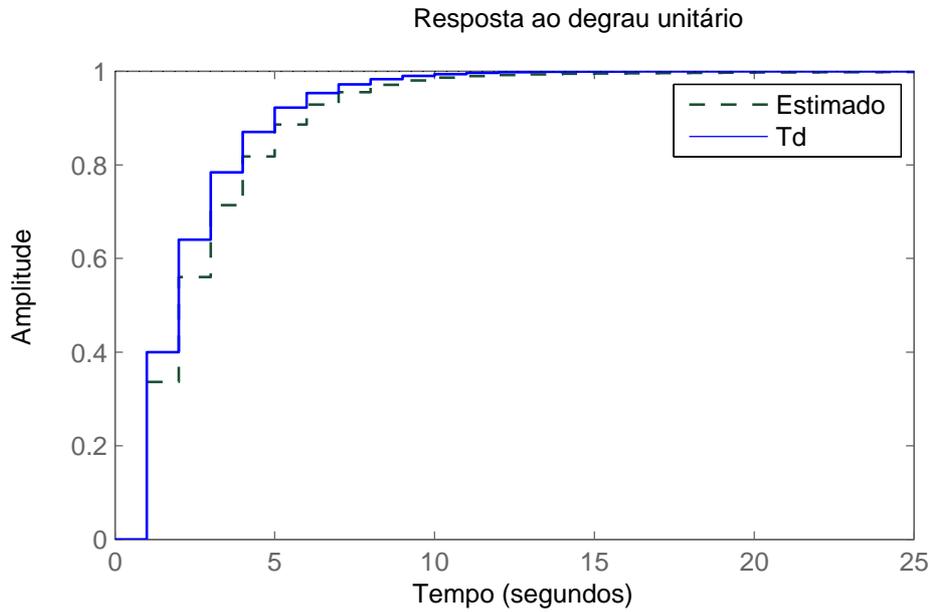


Figura 27: Comparativo da resposta a um degrau unitário do modelo de referência $T_d(q)$ com o sistema real quando o controlador não linear é inserido na planta.

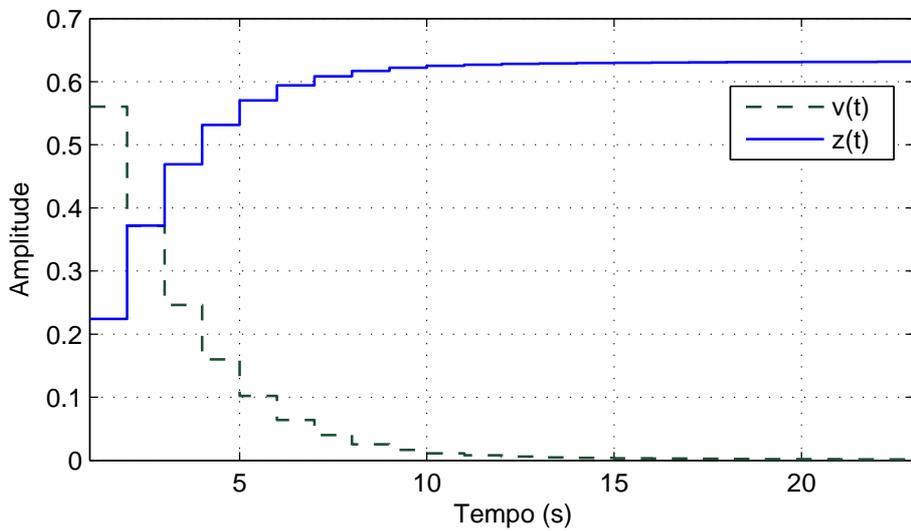


Figura 28: Sinais $v(t)$ e $z(t)$ do sistema apresentado na Figura 26 quando este é excitado por um degrau unitário.

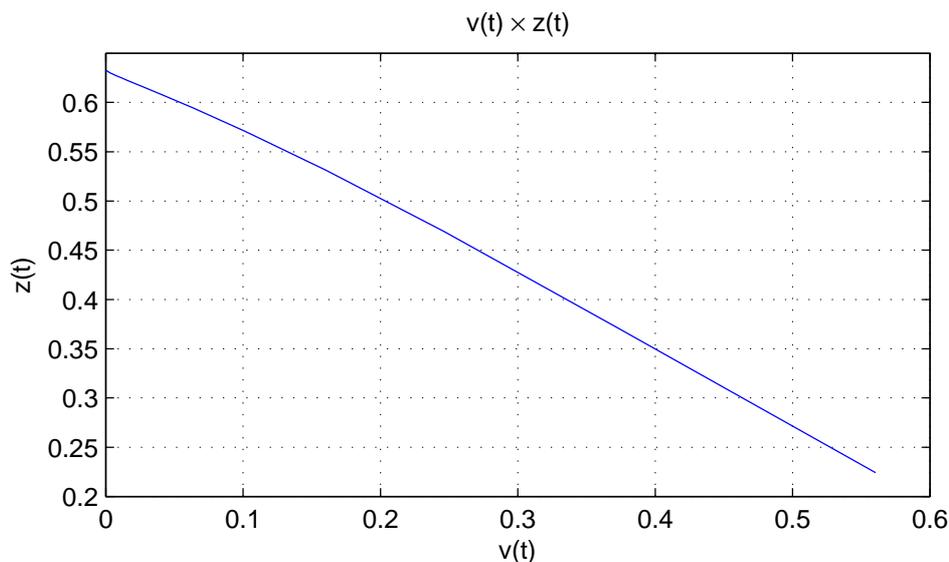


Figura 29: Relação entre os sinais $v(t)$ e $z(t)$ quando o sistema é alimentado por um degrau unitário

Um modelo racional é descrito pela razão entre dois polinômios como:

$$y(t) = \frac{a(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))}{b(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))} + c(e(t-1), \dots, e(t-n_e)) + e(t) \quad (143)$$

onde $a(\cdot)$ e $b(\cdot)$ são polinômios como em:

$$a(\cdot) = \sum_{j=1}^{N_n} p_{nj} \theta_{nj} = \psi_n^T(t-1) \theta_n \quad (144)$$

$$b(\cdot) = \sum_{j=1}^{N_d} p_{dj} \theta_{dj} = \psi_d^T(t-1) \theta_d \quad (145)$$

e onde θ_{nj} e θ_{dj} são os parâmetros dos regressores, possuindo informações da entrada e saída do dos instantes passados de tempo do processo.

Quando o comportamento em malha fechada desejado é linear e a planta é racional, o controlador também será racional. A proposta desta seção é apresentar o paradigma de referência virtual para a identificação deste controlador modelado por uma classe racional. Neste caso a parametrização do controlador será não linear, para isso será utilizado o algoritmo iterativo apresentado na Seção 3.3.1.

Na Seção 3.3.2 foram apresentados alguns resultados da utilização do algoritmo iterativo para a identificação de sistemas racionais. Aqui serão apresentados resultados da utilização deste algoritmo em conjunto com o paradigma de referência virtual para a obtenção do controlador ideal que leva o sistema a se comportar linearmente em malha fechada, como desejado. Serão apresentados exemplos onde a classe de controladores escolhida consegue representar o controlador ideal \mathcal{C}_d e quando isso não é possível.

5.5.1 Controlador ideal representado pela classe de modelos: $\mathcal{C}_d \in \mathcal{C}$

Considere o sistema não linear \mathcal{S}_{nl} descrito por:

$$y(t) = \frac{0.5u(t-1)y(t-1) + u(t-1)}{1 + 0.25y^2(t-2)} \quad (146)$$

Deseja-se que em malha fechada seu comportamento seja linear como em:

$$T_d(q) = \frac{0.4}{q - 0.6} \quad (147)$$

A equação (147) pode ser reescrita em função do tempo como em:

$$y(t) = 0.4r(t-1) + 0.6y(t-1) \quad (148)$$

Ao igualar a equação (146) com (148) e isolar o sinal $u(t)$ tem-se a equação que descreve o comportamento do controlador que levará a planta a ter o comportamento descrito por $T_d(q)$ em malha fechada. Obtém-se desta forma um controlador ideal \mathcal{C}_d descrito como em:

$$u(t) = \frac{0.4r(t) + 0.6y(t) + 0.1y^2(t-1)r(t) + 0.15y(t)y^2(t-1)}{1 + 0.5y(t)} \quad (149)$$

Optou-se por uma classe de modelos que consegue representar a totalidade das dinâmicas presentes em \mathcal{C}_d :

$$u(t) = \frac{\theta_1 r(t) + \theta_2 y(t) + \theta_3 y^2(t-1)r(t) + \theta_4 y(t)y^2(t-1)}{1 + \theta_5 y(t)} \quad (150)$$

Utilizando um sinal PRBS de $N = 254$ e o método de referência virtual para gerar os sinais $r(t)$ e $\epsilon(t)$. Adicionando-se um ruído de variância $\sigma_\epsilon^2 = 0.005$ à saída do sistema, a média das estimativas para 100 experimentos de Monte Carlo foi de:

$$\theta_{\text{média}} = [0.4000 \quad 0.5999 \quad 0.1001 \quad 0.1501 \quad 0.5000]$$

a matriz de covariância obtida neste experimento é descrita por:

$$\theta_{\text{Covariância}} = 1.0 \times 10^{-6} \begin{bmatrix} 0.0498 & 0.0585 & -0.0353 & -0.0362 & -0.0018 \\ 0.0585 & 0.4714 & -0.0449 & -0.3690 & 0.0057 \\ -0.0353 & -0.0449 & 0.0581 & 0.0745 & -0.0040 \\ -0.0362 & -0.3690 & 0.0745 & 0.4647 & -0.0070 \\ -0.0018 & 0.0057 & -0.0040 & -0.0070 & 0.0911 \end{bmatrix}$$

Simulando o sistema com o controlador obtido pela média das estimativas e o sistema em malha fechada desejado, o erro observado para um sinal de entrada do tipo degrau unitário é apresentado na Figura 30.

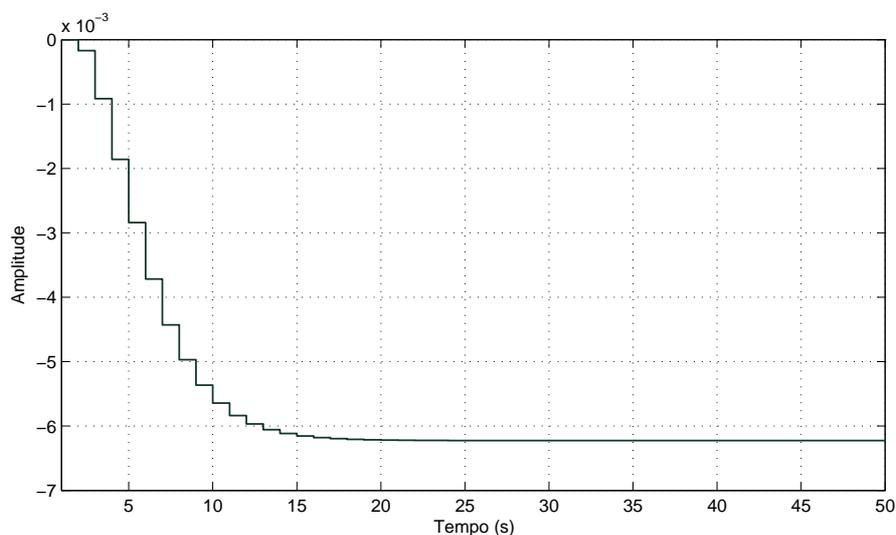


Figura 30: Erro entre a resposta esperada e a obtida para o sistema quando submetido a uma entrada do tipo degrau unitário.

A fim de ilustrar as estimativas obtidas pelo método, na Figura 31 são apresentadas as estimativas para os 100 experimentos realizados, além da elipse de confiança de $\chi^2 = 95\%$ para os parâmetros θ_1 e θ_2 .

O custo $J_{VR}(\theta_{\text{médio}}) = 2.7291 \times 10^{-8}$ foi obtido utilizando-se os sinais de saída do controlador obtido e esperado. Já o custo entre os sinais de saída do sistema em malha fechada estimado e desejado foi de $J_y(\theta_{\text{médio}}) = 0.0033$.

Com o intuito de demonstrar o efeito do ruído sobre as estimativas obtidas o ruído foi aumentado em $4\times$ para $\sigma_e^2 = 0.02$. Obteve-se desta forma os seguintes resultados para as estimativas:

$$\theta_{\text{média}} = [0.3999 \quad 0.5996 \quad 0.1001 \quad 0.1502 \quad 0.4999]$$

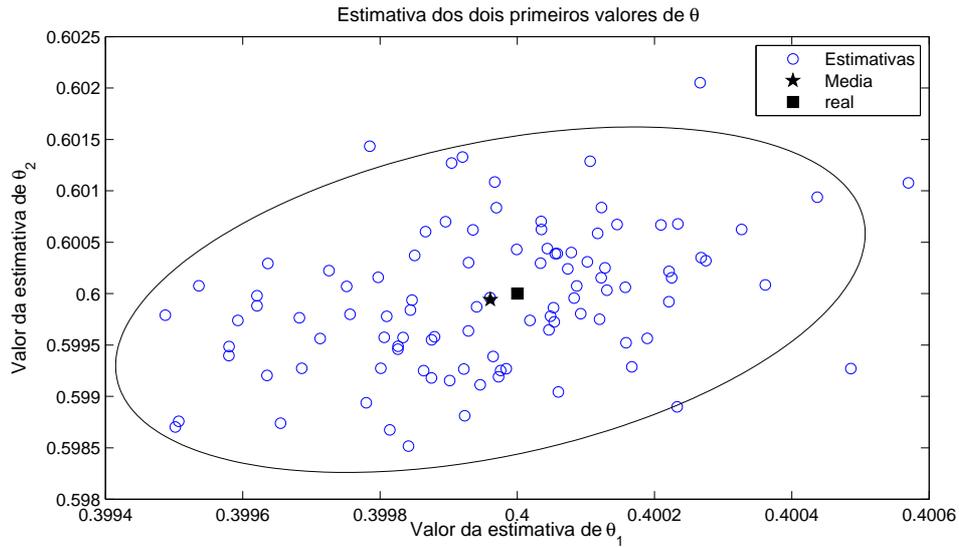


Figura 31: 100 experimentos de Monte Carlo das variáveis θ_1 e θ_2

$$\theta_{\text{Covariância}} = 1.0 \times 10^{-5} \begin{bmatrix} 0.1061 & 0.1207 & -0.0894 & -0.1185 & -0.0310 \\ 0.1207 & 0.7122 & -0.1048 & -0.5357 & 0.0038 \\ -0.0894 & -0.1048 & 0.1344 & 0.1972 & -0.0033 \\ -0.1185 & -0.5357 & 0.1972 & 0.7523 & -0.0571 \\ -0.0310 & 0.0038 & -0.0033 & -0.0571 & 0.1355 \end{bmatrix}$$

Com os custos de $J_{VR}(\theta_{\text{média}}) = 1.0326 \times 10^{-7}$ e $J_y(\theta_{\text{média}}) = 0.0054$, corroborando para o fato de que os custos não foram significativamente afetados pelo aumento do ruído. Observa-se um aumento do erro de variância, pois a matriz de covariância que era da ordem de $\times 10^{-6}$ passou para a ordem de $\times 10^{-5}$, entretanto o valor médio das estimativas permaneceu praticamente inalterado, corroborando a inexistência de erro de polarização.

5.5.2 Controlador ideal não sendo representado pela classe de modelos: $\mathcal{C}_d \notin \mathcal{C}$

Neste exemplo será abordado o caso onde o controlador $\mathcal{C}_d \notin \mathcal{C}$, fazendo com que o sistema em malha fechada não consiga ter a dinâmica esperada. Considere a planta não linear descrita por:

$$y(t) = \frac{0.5u(t-1)y(t-1) + u(t-1)}{1 + 0.25y^2(t-2)} \quad (151)$$

Deseja-se que em malha fechada seu comportamento seja linear como em:

$$T_d(q) = \frac{0.4}{q - 0.6} \quad (152)$$

A equação (152) pode ser reescrita em função do tempo como em:

$$y(t) = 0.4r(t - 1) + 0.6y(t - 1) \quad (153)$$

Ao igualar a equação (151) com (153) e isolar o sinal $u(t)$ tem-se a equação que descreve o comportamento do controlador que levará a planta a ter o comportamento descrito por $T_d(q)$ em malha fechada. Obtém-se desta forma um controlador ideal como em:

$$u(t) = \frac{0.4r(t) + 0.6y(t) + 0.1y^2(t - 1)r(t) + 0.15y(t)y^2(t - 1)}{1 + 0.5y(t)} \quad (154)$$

Escolhe-se uma classe de modelos que não consegue representar completamente (154):

$$u(t) = \frac{\theta_1 r(t) + \theta_2 y(t) + \theta_3 r(t)y(t - 1) + \theta_4 y(t - 1)y(t)}{1 + \theta_5 y(t)} \quad (155)$$

Para as simulações foi adicionado um ruído com $\sigma_\epsilon^2 = 0.05$ e foram realizados 100 experimentos de Monte Carlo onde obteve-se as seguintes estimativas:

$$\theta_{\text{média}} = [0.4696 \quad 0.7011 \quad 0.0083 \quad 0.0063 \quad 0.5013] \quad (156)$$

com uma covariância de:

$$\theta_{\text{Covariância}} = 1.0 \times 10^{-4} \begin{bmatrix} 0.0278 & 0.0396 & 0.0035 & 0.0146 & -0.0137 \\ 0.0396 & 0.3328 & 0.0066 & -0.1651 & 0.0063 \\ 0.0035 & 0.0066 & 0.0382 & 0.0688 & -0.0119 \\ 0.0146 & -0.1651 & 0.0688 & 0.4130 & -0.0562 \\ -0.0137 & 0.0063 & -0.0119 & -0.0562 & 0.0646 \end{bmatrix}$$

Obtém-se um custo para o comportamento do sistema em malha fechada de $J_y(\theta_{\text{médio}}) = 8.66 \times 10^{-3}$. Na Figura 32 é apresentado a resposta do sistema em malha fechada quando submetido a um degrau unitário para o sistema desejado e para o sistema obtido com o controlador parametrizado por (156).

Na Figura 33 é apresentado o erro entre as respostas ao degrau unitário apresentado na Figura 32.

Observa-se que mesmo quando não há uma completa capacidade da classe de modelos em representar o controlador ideal, o erro da estimativa é relativamente

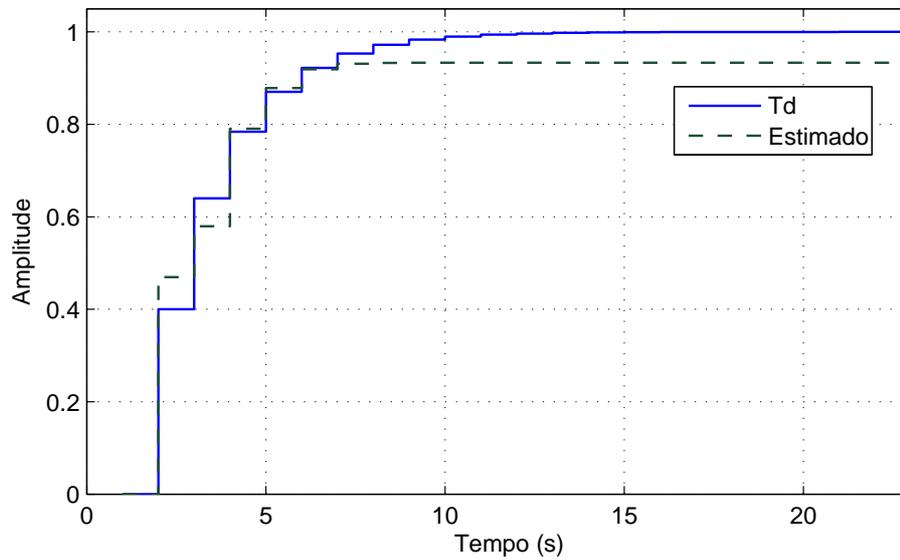


Figura 32: Exemplo onde $\mathcal{C}_d \notin \mathcal{C}$: Resposta do sistema ao degrau unitário para o sistema desejado $T_d(q)$ e o sistema quando o controlador parametrizado por θ como descrito em (156).

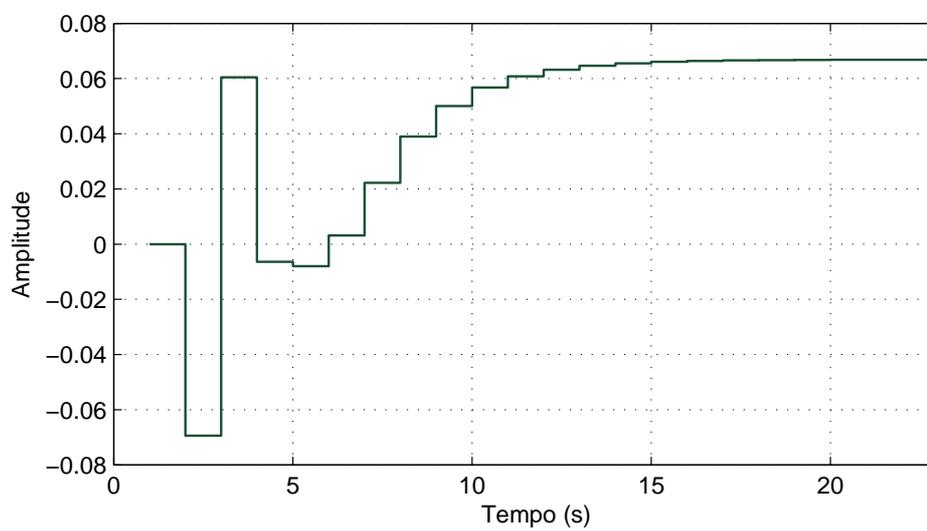


Figura 33: Exemplo onde $\mathcal{C}_d \notin \mathcal{C}$: Erro entre o comportamento do sistema em malha fechada e o comportamento em malha fechada desejado.

baixo, mas não insignificante, ficando em aproximadamente 6% em regime e chegando a atingir quase 10% durante o transitório. Em contrapartida, quando há capacidade do controlador ideal ser representado pela classe do controlador, o erro em regime ficou em 0.02% como apresentado na Figura 30. Este erro poderia ser reduzido, principalmente o erro em regime, que poderia ser reduzido a zero, se a planta possuísse um integrador em sua estrutura, ou se, fosse adicionado um integrador ao controlador e esta parcela da estrutura não fosse identificada em conjunto com o restante do controlador, evitando assim erros de seguimento de referência em regime permanente. Este, entretanto não é o objetivo da apresentação deste exemplo, deseja-se apresentar a qualidade das estimativas utilizando a metodologia quando a classe escolhida para o controlador não é semelhante ao controlador ideal.

5.6 Considerações Finais

Neste capítulo apresentou-se a união das ideias de utilização de referência virtual para a obtenção dos sinais necessários para a determinação do controlador ótimo e a ideia de utilização de algoritmos de identificação de sistemas não lineares para determinação do controlador ótimo quando este faz parte de uma classe de modelos não linear.

Descreveu-se inicialmente a ideia generalizada onde a classe de modelos não linear do controlador precisa estar atrelada apenas ao algoritmo utilizado para sua identificação. Devido aos bons resultados obtidos com a identificação de sistemas racionais apresentados no Capítulo 3, optou-se por utilizar este algoritmo (Seção 3.3.1) para determinar os parâmetros do controlador utilizando-se para isso entretanto dos sinais obtidos por meio do método de referência virtual.

Nas Seções 5.4 e 5.5 apresentou-se alguns exemplos para demonstrar a viabilidade da utilização da metodologia. Foram apresentados exemplos onde a classe de modelos consegue representar a totalidade das dinâmicas de \mathcal{C}_d para que o sistema se comporte em malha fechada linearmente como escolhido por $T_d(q)$. Outro exemplo foi apresentado quando não existe uma completa descrição do controlador ideal pela classe de modelos, ou seja, $\mathcal{C}_d \notin \mathcal{C}$, apresentou-se um comparativo do sistema ideal e do obtido utilizando o controlador obtido quando o sistema é submetido a um degrau unitário.

Também foi apresentado um exemplo onde o controlador é descrito por um modelo polinomial, em um contexto onde a planta do sistema possui uma não linearidade estática na saída do processo, caso este onde o controlador ideal também não se encontra na classe de modelos. A aproximação do modelo escolhido para este controlador se mostrou relativamente próximo ao controlador ideal, visto que o cancelamento da não linearidade da planta foi quase completo, como apresentado

na Figura 29.

De forma geral, apresentou-se neste capítulo a utilização desta metodologia de identificação de controladores não lineares descritos por modelos polinomiais ou racionais e seus resultados. Os resultados obtidos mostram que a metodologia é válida e que em situações onde o controlador ideal pode ser bem representado pela classe de controladores, os resultados obtidos não possuem erro de polarização para experimentos executados em malha aberta.

6 CONCLUSÃO

Neste trabalho diversos assuntos foram abordados para que fosse possível chegar à proposta de projeto de controladores não lineares utilizando referência virtual. Iniciou-se pela teoria básica de identificação de sistemas lineares no Capítulo 2 onde foram apresentados os componentes principais que permeiam toda a teoria de identificação. Foram apresentados alguns dos principais métodos de identificação e quais são suas particularidades e propriedades estatísticas das estimativas obtidas.

No Capítulo 3 foram apresentados as principais classes de modelos para representar sistemas não lineares. Deu-se um enfoque maior para a classe de modelos racionais com a apresentação do algoritmo de identificação proposto para esta classe de modelos. Exemplos de uso foram apresentados e o que ficou evidenciado é que o algoritmo é bastante eficiente ao que ele se propõem. Existem entretanto alguns pontos que podem ser tratados como trabalhos futuros: a convergência do algoritmo está atrelada à convergência do modelos do ruído. Nos exemplos estudados o ruído aplicado foi pouco representativo para fazer com que houvesse qualquer divergência de convergência do algoritmo, mas em situações onde a classe de modelos não consegue representar todas as dinâmicas do sistema real ($\mathcal{S} \notin \mathcal{M}$) o erro de polarização das estimativas era esperado, mas se um modelo de ruído mais complexo fosse adicionado, a percepção que se tem é que este erro seria reduzido e com isso uma melhor estimativa poderia ser obtida. Como nestes exemplos o escopo era apenas demonstrar o funcionamento do algoritmo em situações onde o conhecimento da planta é limitado e que o modelo escolhido não é subdimensionado, não se investiu no desenvolvimento desta melhoria para o algoritmo.

Classes de modelos racionais possuem alguns diferenciais que os destacam de outras classes de modelos para sistemas não lineares: este tipo de classe consegue representar uma grande quantidade de sistemas reais pois relaciona tanto os sinais passados da entrada, saída e do ruído do sistema. Outro ponto de destaque é que este tipo de classe é parametrizável linearmente, no caso de modelo polinomial, podendo para isso ser utilizado o método dos mínimos quadrados para a identificação. Para modelos racionais o algoritmo é iterativo, mas ainda assim simples de

ser desenvolvido.

No Capítulo 4 foi apresentada a ideia básica para projetos de controladores baseados em dados. Foram apresentados brevemente alguns dos principais métodos existentes. O método VRFT foi mais detalhado e alguns exemplos de uso do método foram apresentados. Resultados obtidos para os exemplos apresentados foram satisfatórios, observando-se que o método em sua concepção original possui erro de polarização quando os dados estão corrompidos por ruído, mesmo que este seja branco. Com a proposta de uso de variáveis instrumentais, mostrou-se que este erro de polarização desaparece, ampliando os horizontes de utilização da metodologia.

No Capítulo 5 os resultados centrais deste trabalho foram apresentados. Uniu-se a ideia de referência virtual com o algoritmo de identificação de sistemas não lineares descritos por classe de modelos racional apresentado no Capítulo 3. Para esta situação, o sistema a ser identificado pelo algoritmo é o controlador que é projetado baseado nas necessidades de desempenho do sistema em malha fechada, determinadas pelo projetista.

Um controlador não linear tipicamente será requerido quando a planta tiver comportamento não linear, mas o comportamento do sistema em malha fechada desejado for linear. Desta forma o controlador será encarregado, além de cumprir os requisitos de desempenho do sistema, de cancelar as não linearidades presentes na planta. A escolha de um comportamento linear em malha fechada normalmente é o caso para o método VRFT e neste trabalho todos os exemplos seguiram esta característica.

A escolha de um comportamento não linear para o sistema em malha fechada pode ser o caso de alguma aplicação, mas neste trabalho não foi abordado, poderia entretanto, ser tratado como trabalho futuro, visto que algumas suposições para o uso de referência virtual devam ser modificadas para cobrir esta situação e gerar os sinais necessários para a identificação.

Os exemplos apresentados para sistemas com não linearidades estáticas e com não linearidades na dinâmica do processo apresentados no Capítulo 5 mostraram-se bem completas e em conjunto com situações onde o controlador ideal não consegue ser representado pela classe de modelos ($\mathcal{S} \notin \mathcal{M}$), cobriram grande parte das situações comumente encontradas na prática.

Desta forma é possível afirmar que a metodologia proposta é válida e o estágio atual de desenvolvimento propicia o uso desta metodologia em aplicações práticas.

REFERÊNCIAS

AGUIRRE, L. A. Recovering map static nonlinearities from chaotic data using dynamical models. **Physica D : Nonlinear Phenomena**, Leiden, v.100, n.1-2, p.41 – 57, 1997.

AGUIRRE, L. A. **Introdução à identificação de sistemas, Técnicas lineares e não-lineares aplicadas a sistemas reais**. 2.ed. Belo Horizonte: UFMG, 2004. v.1.

AGUIRRE, L.; JACOME, C. Cluster analysis of NARMAX models for signal-dependent systems. **Control Theory and Applications, IEE Proceedings**, Stevenage, v.145, n.4, p.409 –414, July 1998.

BAZANELLA, A. S.; CAMPESTRINI, L.; ECKHARD, D. **Data-driven Controller Design The H2 Approach**. 1st.ed. Netherlands: Springer, 2012.

BAZANELLA, A. S. et al. Iterative minimization of H2 control performance criteria. **Automatica**, London, v.44, n.10, p.2549 – 2559, 2008.

BILLINGS, S. A.; ZHU, Q. M. Rational model identification using an extended least-squares algorithm. **International Journal of Control**, London, v.54, n.3, p.529–546, 1991.

BILLINGS, S. A.; ZHU, Q. M. Model validation tests for multivariable nonlinear models including neural networks. **International Journal of Control**, London, v.62, n.4, p.749–766, 1995.

CAMPESTRINI, L. **Contribuições para métodos de controle baseados em dados obtidos em apenas um experimento**. 2010. Doutorado em Controle e Automação — Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Programa de Pós-Graduação em Engenharia Elétrica, Porto Alegre, 2010.

CAMPESTRINI, L. et al. Virtual Reference Feedback Tuning for non-minimum phase plants. **Automatica**, London, v.47, n.8, p.1778 – 1784, 2011.

CAMPI, M.; LECCHINI, A.; SAVARESI, S. Virtual reference feedback tuning (VRFT): a new direct approach to the design of feedback controllers. In: **IEEE CONFERENCE ON DECISION AND CONTROL**, Sydney. **Proceedings...** Sydney: IEEE Conference Publications, 2000. v.1, p.623–629.

CAMPI, M.; LECCHINI, A.; SAVARESI, S. Virtual reference feedback tuning: a direct method for the design of feedback controllers. **Automatica**, London, v.38, n.8, p.1337 – 1346, 2002.

CAMPI, M.; SAVARESI, S. Direct nonlinear control design: the virtual reference feedback tuning (vrft) approach. **Automatic Control, IEEE Transactions on**, Notre Dame, v.51, n.1, p.14 – 27, Jan. 2006.

CHEN, S.; BILLINGS, S. A. Representation of non-linear systems: the narmax model. **International Journal of Control**, London, v.49, n.3, p.1012–1032, March 1989.

CHEN, S.; BILLINGS, S. A. Recursive prediction error parameter estimator for non-linear models. **International Journal of Control**, London, v.49, n.2, p.569–594, 1989.

CHEN, S. et al. Practical identification of NARMAX models using radial basis functions. **International Journal of Control**, London, v.52, p.1327–1350, 1990.

CORRÊA, M. V. **Identificação de sistemas dinâmicos não-lineares utilizando modelos NARMAX Racionais**: aplicação a sistemas reais. 1997. Dissertação de mestrado, Programa de Pós Graduação em Engenharia Elétrica — Universidade Federal de Minas Gerais, Minas Gerais, 1997.

ECKHARD, D. et al. Data-Based Control Design for a Process Class with Guaranteed Convergence to the Globally Optimum Controller. In: **EUROPEAN CONTROL CONFERENCE**, Budapest. **Proceedings...** Budapest: EUCA, 2009.

GEVERS, M. A personal view of the development of system identification: a 30-year journey through an exciting field. **Control Systems, IEEE**, Cambridge, v.26, n.6, p.93 –105, Dec. 2006.

GEVERS, M. et al. Identification and the Information Matrix: how to get just sufficiently rich? **Automatic Control, IEEE Transactions on**, Notre Dame, v.54, n.12, p.2828 –2840, Dec. 2009.

- GLAD, S.; LJUNG, L. Model structure identifiability and persistence of excitation. In: IEEE CONFERENCE ON DECISION AND CONTROL, 29, Honolulu. **Proceedings...** Honolulu: IEEE Conference Publications, 1990. v.6, p.3236–3240.
- GUARDABASSI, G. O.; SAVARESI, S. M. Approximate feedback linearization of discrete-time non-linear systems using virtual input direct design. **Syst. Control Lett.**, Amsterdam, v.32, p.63–74, Nov. 1997.
- HJALMARSSON, H. Iterative feedback tuning - an overview. **International Journal of Adaptive Control and Signal Processing**, Malden, v.16, n.5, p.373–395, 2002.
- HJALMARSSON, H. et al. Iterative feedback tuning: theory and applications. **Control Systems, IEEE**, Cambridge, v.18, n.4, p.26–41, Aug. 1998.
- HJALMARSSON, H.; GUNNARSSON, S.; GEVERS, M. A convergent iterative restricted complexity control design scheme. In: IEEE CONFERENCE ON DECISION AND CONTROL, Lake Buena Vista. **Proceedings...** Lake Buena Vista: IEEE Conference Publications, 1994. v.2, p.1735–1740.
- HJALMARSSON, H.; MÅRTENSSON, J. Finite model order accuracy in Hammerstein model estimation. **Automatica**, London, v.48, n.10, p.2640–2646, 2012.
- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the National Academy of Sciences**, Washington, v.79, n.8, p.2554–2558, 1982.
- JANSSON, H. **Experiment Design with Applications in identification for Control**. 2004. 207 f. Doctor thesis in Automatic Control — Royal Institute of Technology (KTH), Stockholm. 2004.
- KAMMER, L. C.; BITMEAD, R. R.; BARTLETT, P. L. Direct iterative tuning via spectral analysis. **Automatica**, London, v.36, n.9, p.1301 – 1307, 2000.
- KARIMI, A.; MIŠKOVIĆ, L.; BONVIN, D. Iterative correlation-based controller tuning with application to a magnetic suspension system. **Control Engineering Practice**, Vienna, v.11, n.9, p.1069 – 1078, 2003. Special Section on Algorithms and Applications of Iterative Feedback Tuning.
- KARIMI, A.; MIŠKOVIĆ, L.; BONVIN, D. Iterative correlation-based controller tuning. **International Journal of Adaptive Control and Signal Processing**, Glasgow, v.18, n.8, p.645–664, 2004.

KHALIL, H. **Nonlinear Systems**. 2nd.ed. New Jersey: Prentice Hall, 1996.

LECCHINI, A. **Virtual reference feedback tuning: a new direct data-based method for the design of feedback controllers**. 2001. Ph.D. thesis in Automatic Control — Università di Brescia, Owensboro, Owensboro, 2001.

LECCHINI, A.; CAMPI, M.; SAVARESI, S. Virtual reference feedback tuning for two degree of freedom controllers. **International Journal of Adaptive Control and Signal Processing**, Malden, v.16, n.5, p.355–371, 2002.

LEONTARITIS, I. J.; BILLINGS, S. A. Input-output parametric models for non-linear systems Part I: deterministic non-linear systems. **International Journal of Control**, London, v.41, n.2, p.303–328, 1985.

LJUNG, L. **System Identification Theory for the User**. 2nd.ed. New Jersey: Prentice Hall PTR, 1999. (Prentice Hall Information And System Sciences Series).

NARENDRA, K.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. **Neural Networks, IEEE Transactions on**, Piscataway, v.1, n.1, p.4 –27, March 1990.

NEUHAUS, T. **Rotinas para identificação de sistemas não lineares racionais**. Disponível em:
<https://github.com/tassianors/rational_models_matlab>. Acesso em: 20 Ago. 2012.

PROCHÁZKA, H. et al. Iterative Feedback Tuning for robust controller design and optimization. In: IEEE CONFERENCE ON DECISION AND CONTROL, AND 2005 EUROPEAN CONTROL CONFERENCE, 44., Seville. **Proceedings...** Seville: IEEE Conference Publications, 2005. p.3602–3607.

SÖDERSTRÖM, T.; STOICA, P. **System identification**. New Jersey: Prentice Hall, 1989. (Prentice-Hall international series in systems and control engineering).

SOLARI, G. E. **Iterative model-free Controller Tuning**. 2005. Doctor thesis — , Louvain-la-Neuve, Belgium, 2005.

TSE, C. K. Chaos from a buck switching regulator operating in discontinuous mode. **International Journal of Circuit Theory and Applications**, Malden, v.22, n.4, p.263–278, 1994.

APÊNDICE A ALGORITMO DE IDENTIFICAÇÃO DE SISTEMAS NÃO LINEARES

Neste capítulo serão apresentados as rotinas criadas na ferramenta Matlab para a identificação de sistemas não lineares, como descrito na Seção 3.3.1. As rotinas aqui apresentadas estão disponíveis em (NEUHAUS, 2012).

Na Seção A.1 é apresentado a rotina principal do algoritmo de identificação de sistemas não lineares. Na Seção A.2 são apresentadas as rotinas auxiliares para a obtenção das matrizes necessárias para a execução do algoritmo.

A.1 Rotina principal

Rotina principal para o algoritmo de identificação de sistemas lineares. Observe-se que o modelo do sistema e os sinais de entrada e saída da planta são alguns dos parâmetros para a estimativa, além de parâmetros de configuração do experimento, como limites de iterações, quantidade de ruído adicionado sobre os dados entre outros.

```

1 function [ret cost] = f_rational_model(simul, model, ic,
   out_sig, in_sig, aux_sig1, aux_sig2)
2 %=====
3 %% Function to perform the identification of a system using
   a recursive algorithm.
4 %
5 % simul :: Simulation parameter. Must have:
6 %     N - size of data array from system
7 %     np - Amount of noise power to inject on each
   simulation
8 %     l - max acceptable number of algorithm iterations.
   This is used as a time out to avoid infinity loop.
9 %     verbose - used as debug to show some useful
   identification information
10 %
11 % model :: Model specification

```

```

12 %           See #f_check_model function for further more
           information
13 %
14 % ic      :: Initial system conditions
15 %
16 % out_sig :: Output signal of system under identification
17 % in_sig  :: Input signal of system under identification
18 % aux_sig1 :: Auxiliary signal one that can be used or not
           in identification procedure
19 % aux_sig2 :: Auxiliary signal two that can be used or not
           in identification procedure
20 %=====
21
22 % start time measurement (used to know how much time has
           been spend in this procedure)
23 tic
24 f_check_model(model);
25
26 if simul.N ~= size(out_sig,1)
27     error('y must be a [1 N] array');
28 end
29
30 % init return variable with zeros
31 ret = zeros(simul.nEstimates, model.dim);
32
33 for m=1:simul.nEstimates
34     clear theta v;
35     %% init some variables at each estimative
36     y = zeros(simul.N, 1);
37     yc = y;
38     model.err_model = 0;
39     l=1;
40
41     % add a random noise with some defined noise power
42     y = f_get_wnoise(out_sig, simul.np);
43
44     psi = f_get_psi(y, yc, in_sig, aux_sig1, aux_sig2,
45                   model);
46     theta(m,:) = (psi'*psi)\(psi'*y);
47
48     % here we got the first estimative, now we start the
           loop until cost function stops to reduce.
49     while (l < simul.l)
50         % get output signal from model last estimative
51         yc = f_y_model(ic, in_sig, aux_sig1, aux_sig2,
52                       theta(l,:), model);
53
54         % only after the first estimative, calc using the
           error model

```

```

53     if l == 2 && model.err_enable == true
54         model.err_model = 1;
55         % enlarge some matrix
56         theta(l, model.dim+model.err_model) = 0;
57         delta(l, model.dim+model.err_model) = 0;
58     end
59
60     %% step 2 - calc covariance
61     v(l)=cov(y-yc);
62
63     % based on user output signal and last estimative,
64     % get new psi
65     psi = f_get_psi(y, yc, in_sig, aux_sig1, aux_sig2,
66         model);
67     [PHI phi] = f_get_phi(y, model);
68
69     % update theta for next estimative, if needed.
70     theta(l+1,:) = (psi'*psi-v(l)*PHI)\(psi'*y-v(l)*phi
71         );
72
73     % Calc Jy cost function
74     costJy(l)=sum(0.5*abs(y-yc).^2)/simul.N;
75     % Check if cost is still decreasing, if not stop
76     if l > 1 && costJy(l)-costJy(l-1) > 0
77         if simul.verbose == true
78             str = sprintf('At iteration %d we checked
79                 that the cost raised from %s to %s', l,
80                 num2str(costJy(l-1), 5), num2str(costJy(
81                 l), 5));
82             disp(str);
83         end
84         break;
85     end
86     if simul.verbose == true
87         str = sprintf('Iteration %d: Cost was %s
88             covariance was %s ', l, num2str(costJy(l),
89             5), num2str(v(l), 5));
90         disp(str)
91     end
92     l=l+1;
93 end
94 % discarding the last estimative, unless we run out by
95 % timeout
96 for o = 1: model.dim
97     ret(m,o) = theta(l-1,o);
98 end
99 % return final cost obtained.
100 cost=costJy(l-1);

```

```

93     % finishing time measurement.
94     total_time=toc;
95     % check timeout occurrence
96     if l >= simul.l
97         error('Procedure diverged and also took %4.2f s to
98             complete',total_time)
99         ret = zeros(simul.nEstimates, model.dim);
100        cost=0;
101        return
102    end
103    if simul.verbose == true
104        str = sprintf('Execution finished in %d iterations
105                    [total time spend: %4.2f seconds]. Final cost
106                    was %s\n', l, total_time, num2str(costJy(l-1),
107                    5));
108        disp(str)
109    end
110 end %m
111 end %function

```

Listagem A.1: Laço principal do algoritmo implementado para identificação de sistemas não lineares descritos por modelos Racionais

A.2 Rotinas auxiliares

Nesta seção são apresentadas as principais rotinas auxiliares para o algoritmo de identificação de sistemas não lineares. A rotina `f_get_phi`, é responsável por retornar as matrizes apresentadas nas equações (74) e (75), enquanto a rotina `f_get_psi` é responsável por calcular a matriz apresentada na equação (73).

Na Listagem A.4 é apresentada a rotina responsável por, baseado na classe de modelo e nos sinais envolvidos na simulação, calcular a saída do sistema.

```

1 function [phi phim] = f_get_phi(y, m)
2 %=====
3 %% Get phi based on model structure
4 % y:: output system data [y1,...,yn]
5 % m:: model structure
6 %
7 % Return phi and PHI
8 %=====
9
10 % Check if model parameter structure is valid
11 f_check_model(m);
12
13 % variables initialization
14 N = max(size(y));

```

```

15 phi = zeros(m.dim + m.err_model, m.dim + m.err_model);
16 phim = zeros(m.dim + m.err_model, 1);
17
18 for i=1:m.dim
19     for j=1:m.dim
20         if i>m.n_dim && j>m.n_dim
21             aux=0;
22             for k=1:N
23                 aux = aux + ((y(k)^m.texp(j))* (y(k)^m.
                    yplus_exp(j))* (y(k)^m.texp(i))* (y(k)^m
                    .yplus_exp(i)));
24             end
25             phi(i,j)=aux;
26         end
27     end
28 end
29
30 for i= m.n_dim+1:m.dim
31     aux=0;
32     for k=1:N
33         aux = aux- ((y(k)^m.texp(i))* (y(k)^m.yplus_exp(i))
            );
34     end
35     phim(i,1) = aux;
36 end
37 end %function

```

Listagem A.2: Função auxiliar do algoritmo de identificação de sistemas não lineares que baseado no sinal de saída do sistema e da classe de modelos calcula a matriz Φ

```

1 function psi = f_get_psi(out_sig, out_prev, in_sig,
    aux_sig1, aux_sig2, m)
2 %=====
3 %% Get Psi based on model structure
4 % out_sig:: output system data
5 % in_sig:: input system data
6 % aux_sig1:: input aux signal
7 % aux_sig2:: input aux signal
8 % m:: model structure
9 %=====
10
11 % check parameters
12 f_check_model(m);
13
14 if m.err_model && max(size(out_sig)) ~= max(size(out_prev))
15     error('out_sig and last out_sig (out_prev) have to have
        the same size');
16 else

```

```

17     err=out_sig-out_prev;
18 end
19
20 N = max(size(out_sig));
21 psi = zeros(N, m.dim+m.err_model);
22
23 for i=max(abs(m.regr))+1:N
24     for j=1:m.dim+m.err_model
25         % err_model
26         if j> m.dim
27             % fill the err_model column.
28             psi(i,j)=err(i);
29             continue;
30         end
31
32         if m.yu(j) == 1
33             yu=out_sig(i-abs(m.regr(j)))^m.texp(j);
34         elseif m.yu(j) == 2
35             yu=in_sig(i-abs(m.regr(j)))^m.texp(j);
36         elseif m.yu(j) == 3
37             yu=aux_sig1(i-abs(m.regr(j)))^m.texp(j);
38         elseif m.yu(j) == 4
39             yu=aux_sig2(i-abs(m.regr(j)))^m.texp(j);
40         else
41             error('not supported option');
42         end
43         % non linearity is yu^a*y^b
44         yu2=1;
45         if m.yplus_yur(j) == 1
46             yu2=out_sig(i-abs(m.yplus_regr(j)))^m.yplus_exp(
47                 j);
48         elseif m.yplus_yur(j) == 2
49             yu2=in_sig(i-abs(m.yplus_regr(j)))^m.yplus_exp(
50                 j);
51         elseif m.yplus_yur(j) == 3
52             yu2=aux_sig1(i-abs(m.yplus_regr(j)))^m.
53                 yplus_exp(j);
54         elseif m.yplus_yur(j) == 4
55             yu2=aux_sig2(i-abs(m.yplus_regr(j)))^m.
56                 yplus_exp(j);
57         end
58         if j<=m.n_dim
59             psi(i,j)=yu*yu2;
60         else
61             % here we should always use y(i) (equation 10.44
62                 aguirre)
63             psi(i,j)=-yu*yu2*out_sig(i);
64         end
65     end
66 end

```

```

61     end
62 end
63 end

```

Listagem A.3: Função auxiliar do algoritmo de indentificação de sistemas não lineares que baseado nos sinais reais e virtuais obtidos do sistema, calcula a matriz Ψ

```

1 function y = f_y_model(ic, in_sig, aux_sig1, aux_sig2,
   theta, m)
2 %=====
3 %% Get output signal based on model structure
4 %
5 % ic:: Initial value for y, can be an array [y(1), y(2),
   ...]
6 % in_sig:: input signal array
7 % theta:: estimative model parameters
8 % m:: model structure
9 %=====
10
11 N = max(size(in_sig));
12 y = zeros(N, 1);
13 % init y array with initial conditions (ic)
14 for j=1: size(ic,2)
15     y(j)=ic(j);
16 end
17 m_dim=size(theta,2);
18
19 % check parameters
20 f_check_model(m);
21 if m.dim+m.err_model ~= m_dim
22     error('size of theta must be the same as model
   dimension');
23 end
24
25 for k=max(abs(m.regr))+1:N
26     num=0;
27     for i=1:m.n_dim
28         if m.yu(i) == 1
29             yu = y(k-abs(m.regr(i)))^m.texp(i);
30         elseif m.yu(i) == 2
31             yu = in_sig(k-abs(m.regr(i)))^m.texp(i);
32         elseif m.yu(i) == 3
33             yu = aux_sig1(k-abs(m.regr(i)))^m.texp(i);
34         elseif m.yu(i) == 4
35             yu = aux_sig2(k-abs(m.regr(i)))^m.texp(i);
36         else
37             error('invalid option, just y(1) u(2) r(3) and
   e(4) are possible')

```

```

38     end
39     % non linearity is  $yu^a*y^b$ 
40     if m.yplus_yur(i) == 1
41         yu = yu*y(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
42     elseif m.yplus_yur(i) == 2
43         yu = yu*in_sig(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
44     elseif m.yplus_yur(i) == 3
45         yu = yu*aux_sig1(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
46     elseif m.yplus_yur(i) == 4
47         yu = yu*aux_sig2(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
48     end
49     num = num+theta(i)*yu;
50 end
51 den = 1;
52 for i=m.n_dim+1:m.dim
53     if m.yu(i) == 1
54         yu = y(k-abs(m.regr(i)))^m.texp(i);
55     elseif m.yu(i) == 2
56         yu = in_sig(k-abs(m.regr(i)))^m.texp(i);
57     elseif m.yu(i) == 3
58         yu = aux_sig1(k-abs(m.regr(i)))^m.texp(i);
59     elseif m.yu(i) == 4
60         yu = aux_sig2(k-abs(m.regr(i)))^m.texp(i);
61     end
62
63     % non linearity is  $yu^a*y^b$ 
64     if m.yplus_yur(i) == 1
65         yu = yu*y(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
66     elseif m.yplus_yur(i) == 2
67         yu = yu*in_sig(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
68     elseif m.yplus_yur(i) == 3
69         yu = yu*aux_sig1(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
70     elseif m.yplus_yur(i) == 4
71         yu = yu*aux_sig2(k-abs(m.yplus_regr(i)))^m.yplus_exp(i);
72     end
73     den=den+theta(i)*yu;
74 end
75 % err=0;
76 % if max(size(theta)) > m.dim
77 %     for h=m.dim+1:max(size(theta))
78 %         err=err+aux_sig2(k)*theta(h);

```

```
79 %         end
80 %     end
81
82     y(k) = num/den;
83 end
84 end
```

Listagem A.4: Função auxiliar que, baseado na classe de modelos e no valor estimado de θ , determina a saída do sistema para uma determinada entrada.

APÊNDICE B ROTINAS PARA IDENTIFICAÇÃO CLASSES DE MODELOS RACIONAIS

Neste capítulo serão apresentados rotinas que utilizam as rotinas de identificação de sistemas não lineares apresentadas no Apêndice A. Serão apresentados aqui as rotinas utilizadas nos exemplos de identificação de sistemas racionais. Na Seção B.1 é apresentada a rotina de identificação de um sistema conversor de corrente contínua extraído de (TSE, 1994).

Na Seção B.2 é apresentado um exemplo onde o sistema é nativamente um sistema não linear racional.

As rotinas aqui apresentadas estão disponíveis em (NEUHAUS, 2012).

B.1 Identificação de um conversor de corrente contínua do tipo Buck

Nesta seção será apresentada a rotina para a identificação do sistema de conversão de corrente contínua do tipo Buck, apresentado na Seção 3.3.2.3.

```

1 %=====
2 %% DC to DC converter
3 % Example from Aguirre book, chapter 10.4
4 %
5 %=====
6 close all; clear all;
7 clc;
8
9 % Add path to find functions
10 f_set_path();
11
12 %% model parameter definition
13 model.n_dim      = 3;
14 model.dim        = 5;
15 model.texp       = [0 3 2 1 2];
16 model.yu         = [1 1 1 1 1];

```

```

17 model.regr      = [1 1 1 1 1];
18 model.yplus_yur = [0 0 0 0 0];
19 model.yplus_exp = [0 0 0 0 0];
20 model.yplus_regr = [0 0 0 0 0];
21 model.err_model  = 0;
22 model.err_enable = true;
23 model.err_size   = 1;
24
25 % number of Experiments
26 exper=100;
27 %% Simulation parameters
28 simul = struct('N', 200, 'nEstimates', 1, 'np', 0.005, '
    verbose', true, 'l', 100);
29
30 % variable init
31 rho_size=size(model.texp,2);
32 theta = zeros(exper, rho_size);
33 y = zeros(simul.N,1);
34 % Initial Condition
35 y(1)=31.5;
36
37 %% Real system - variables, from Aguirre
38 a=2.6204; b=99.875; c=1417.1; d=46.429;
39
40 % Simulation of real system
41 for k=max(abs(model.regr))+1:simul.N
42     y(k)=d*exp(22-y(k-1))+ ((a*y(k-1)^2-b*y(k-1)+c)/y(k-1))
    ;
43 end
44
45 for i = 1: exper
46     [theta(i,:) cost] = f_rational_model(simul, model, [y
    (1)], y, y, 0, 0);
47 end
48
49 % get theta information: mean and covariance
50 m_theta = mean(theta);
51 c_theta = cov(theta);
52 cost;
53
54 % get output signal from model estimative
55 y_estim = f_y_model([y(1)], y, 0, 0, m_theta, model);
56
57 % plot map graphics: y x y+1
58 [a b] = f_aguirre_plot_map(y_estim, 0);
59 [c d] = f_aguirre_plot_map(y, 0);
60 plot(a, b, 'b.');
```

```

63 % Plot covariance matrix to be used in latex
64 f_plot_matrix_std(c_theta);

```

Listagem B.1: Rotina utilizada para a identificação de um conversor Buck. Exemplo de uso das rotinas apresentadas no Apêndice A

B.2 Identificação de uma classe de modelos racionais

Nesta seção será apresentado a rotina para a identificação de um sistema que pode ser descrito por uma classe de modelos racional.

```

1 % Rational Model
2 % example extracted from : rational model identification -
  zhu and billings 1991
3 %=====
4 close all; clear all;
5 clc;
6 % get PATH
7 f_set_path();
8 %start time measurement
9 tic;
10
11 % static non linearity - true system values
12 a1=.2;a2=0.1;a3=1;
13 b1=1;b2=1
14
15 % simulation parameters
16 rho_size=4;
17 % number of prbs signal periods
18 m=5;
19 exper = 100;
20 theta = zeros(exper, rho_size);
21
22 % get PRBS signal
23 [in_sig N]=f_get_prbs(m);
24 %=====
25 % plant simul
26 %=====
27 y = zeros(N, 1);
28 for k=3:N
29     y(k)=(a1*y(k-1)+a2*y(k-1)*in_sig(k-1)+a3*in_sig(k-1)) /
        (in_sig1+b1*y(k-1)^2+b2*y(k-2)^2);
30 end
31 %=====
32 % Controller model definition
33 %=====
34 model.n_dim      = 2;
35 model.dim        = 4;

```

```

36 model.err_model = 0;
37 model.texp      = [1 1 2 2];
38 model.yu       = [1 2 1 1];
39 model.regr     = [1 1 1 2];
40 % tels if there is some non linearity like (y(k-a)^b)*(y(k-
    c)^d)
41 % u = 2 y=1 none =0
42 model.yplus_yur = [0 0 0 0];
43 % tels the d param
44 model.yplus_exp = [0 0 0 0];
45 % tels the C param
46 model.yplus_regr = [0 0 0 0];
47 model.err_enable = true
48 model.err_size   = 1;
49
50 %% Simulation parameters
51 simul=struct('N', N, 'nEstimates', exper, 'np', 0.005, 'l',
    100, 'verbose', true);
52
53 % calc theta #exper times
54 theta = f_rational_model(simul, model, [y(1)], y, in_sig,
    0, 0);
55
56 % get theta information
57 m_theta = mean(theta);
58 v_theta = var(theta);
59 c_theta = cov(theta);
60
61 %% Compare results of real system and simulated
62 y1=zeros(N, 1);
63 for k=3:N
64     y1(k)=(m_theta(1)*y1(k-1)+ m_theta(2)*in_sig(k-1)) / (1+
        m_theta(3)*y1(k-1)^2+ m_theta(4)*y1(k-2)^2);
65 end
66
67 err=0;
68 for k=1:N
69     err = err+abs(y(k)-y1(k))^2;
70 end
71
72 % get step response of the system
73 in_sig = ones(N,1);
74 y1 = zeros(N, 1);
75 y = zeros(N, 1);
76 for k=3:N
77     y(k) = (a1*y(k-1)+ a2*y(k-1)*in_sig(k-1)+ a3*in_sig(k
        -1)) / (1+b1*y(k-1)^2+ b2*y(k-2)^2);
78     y1(k) = (m_theta(1)*y1(k-1) + m_theta(2)*in_sig(k-1)) /
        (1+m_theta(3)*y1(k-1)^2+ m_theta(4)*y1(k-2)^2);

```

```
79 end
80 stairs(y(1:40), 'r');
81 hold;
82 stairs(y1(1:40), 'b')
83 % Stop time measurement
84 elapsed=toc;
```

Listagem B.2: Rotina utilizada para a identificação de um sistema racional

APÊNDICE C ROTINAS PARA IDENTIFICAÇÃO DE CONTROLADORES NÃO LINEARES COM O USO DE REFERÊNCIA VIRTUAL

Neste capítulo serão apresentadas algumas rotinas utilizadas para a identificação de controladores utilizando referência virtual e o algoritmo apresentado na Seção 3.3.1. Inicialmente na Seção C.1 é apresentado um exemplo de identificação de controlador utilizando o método VRFT.

Na Seção C.2 é apresentado como o método proposto funciona quando a planta possui uma não linearidade estática acoplada, sendo assim representada por um modelo de Wiener e o controlador por uma classe de modelos racional.

As rotinas aqui apresentadas estão disponíveis em (NEUHAUS, 2012).

C.1 Rotinas para uso do método VRFT em sistemas lineares

Na Listagem C.1 é apresentado um exemplo de uso do VRFT com variáveis instrumentais onde o controlador ideal $C_d(z)$ não consegue ser descrito pela classe de controladores escolhida $C(z, \theta)$.

```

1 % VRFT method example extracet and adapted from campestrini
   Doc. Thesys page 92
2 % where the ideal controller function isn't inside
   controller class
3 %=====
4
5 close all; clear all;
6 clc;
7 P=path;
8 path(P, '../functions')
9 P=path;
10 path(P, '../functions/signal')
11 P=path;

```

```

12 path(P,'../functions/plots')
13 P=path;
14 path(P,'../functions/vrft')
15
16 %=====
17 % simulation parameters
18 rho_size=3;
19 m=20;
20 exper = 100;
21 cut=1;
22 %=====
23 % variables initialization
24 theta = zeros(exper, rho_size);
25 thetaL = zeros(exper, rho_size);
26 [u N] = f_get_prbs(m);
27
28 % Real system parameters
29 a=0.7; b=0.9; c=0.3; d=0.2; e=0.5;
30 %=====
31 model.a = [1 -(b+e) b*e];
32 model.b = [d -a*d];
33 model.c = [1 -c];
34 model.d = [1 0];
35 model.mn = [0.16 0];
36 model.md = [1 -1.2 0.36];
37 model.TS = 1;
38 model.delay = 1;
39 model.delay_func = tf([1],[1 0], model.TS);
40 model.noise_std = 0.1;
41
42 C_den = [1 -1 0];
43 M = tf(model.mn, model.md, model.TS);
44 L = (1-M)*M;
45 beta = [tf([1 0 0], C_den, model.TS); tf([1 0],C_den ,
      model.TS); tf([1],C_den , model.TS)];
46 %Instrumental Variables
47 IV=beta*L*(inv(M)-1);
48
49 for i = 1: exper
50     [el y] = f_get_vrft_el(model, u);
51     [el2 y2] = f_get_vrft_el(model, u);
52     ul = lsim(L,u);
53     phi2 = lsim(IV, y);
54     instr2 = lsim(IV, y2);
55     phi = phi2(cut: max(size(phi2)),:,1);
56     instr = instr2(cut: max(size(instr2)),:,1);
57     thetaL(i,:) = inv(instr'*phi)*instr'* ul(cut:max(size(
      ul)));
58 end

```

```

59
60 for i = 1: exper
61     [el y] = f_get_vrft_el(model, u);
62     phi2    = lsim(beta, el);
63     phi     = phi2(cut:max(size(phi2)),:,1);
64     theta(i,:) = inv(phi'*phi)*phi'* u(cut:max(size(u)));
65 end
66
67 %=====
68 variance = var(theta);
69 varianceL = var(thetaL);
70
71 % controller with no filter
72 C=tf(mean(theta),C_den, model.TS)
73 % controller with L filter and IV
74 CL=tf(mean(thetaL),C_den, model.TS)
75 % ideal controller
76 Cd=zpk([0 0.9 0.5],[1 0.36 0.7],0.8, model.TS);
77
78 % costs
79 Jvr = f_get_vrft_Jvr(C, el, u)
80 Jmr = f_get_vrft_Jmr(C, model)
81 JvrL = f_get_vrft_Jvr(CL, el, u)
82 JmrL = f_get_vrft_Jmr(CL, model)
83
84 %=====
85 % graphic plotting
86 G = tf(model.b,model.a, model.TS);
87 T = feedback(C*G, 1);
88 TL = feedback(CL*G, 1);
89
90 step(T, TL, M)
91 legend; grid; figure;
92 bode(C, CL, Cd)
93 legend;

```

Listagem C.1: Rotina para identificação de um controlador utilizando VRFT quando o controlador ideal $C_d(z)$ não consegue ser descrito pela classe de modelos escolhida.

C.2 Identificação de um controlador polinomial para um planta do tipo Wiener

Na Listagem C.2 é apresentada a rotina para identificação do controlador proposto na Seção 5.4.2, onde o processo é do tipo Wiener e o controlador é do tipo polinomial.

```

1 % VRFT method using a non-linear model
2 % real system:  $y(k)=0.9y(k-1)+\tanh(u(t-1))$ 
3 %=====
4 close all; clear all;
5 clc;
6 P=path;
7 path(P,'../functions')
8 f_set_path();
9
10 % simulation parameters
11 theta_size = 8;
12 m=1;
13 exper = 100;
14
15 % static non linearity - true system parameters
16 b1 = 1.5; b3 = 0.2; ga1 = 0.5; gb1 = 0.9;
17
18 % model structure
19 model.a      = [1 -gb1];
20 model.b      = [ga1];
21 model.mn     = [0.4];
22 model.md     = [1 -0.6];
23 model.TS    = 1;
24 model.delay  = 1;
25 model.delay_func = tf([1],[1 0], model.TS);
26 model.noise_std = 0.005;
27
28 Td          = tf(model.mn, model.md, model.TS);
29 G           = tf(model.b, model.a, model.TS);
30 Filter     = tf([1],[1 -1], model.TS);
31 theta      = zeros(exper, theta_size);
32 [u N]      = f_get_prbs(m);
33 %=====
34 % plant simul
35 %=====
36 y2=lsim(G, u);
37 %% apply non linearity
38 for k=1:N
39     y(k)=b1*y2(k)+b3*y2(k)^3;
40 end
41 %=====
42 % Controller model definition
43 %=====
44 %% model parameter definition
45 m_rat.n_dim   = theta_size;
46 m_rat.dim     = theta_size;
47 m_rat.texp    = [1 1 2 3 1 1 2 3];
48 m_rat.regr    = [0 0 0 0 1 1 1 1];
49 m_rat.yu      = [3 4 4 4 3 4 4 4];

```

```

50 m_rat.yplus_yur = zeros(1,theta_size);
51 m_rat.yplus_exp = zeros(1,theta_size);
52 m_rat.yplus_regr = zeros(1,theta_size);
53 m_rat.err_enable = true;
54
55 %% Simulation parameters
56 simul=struct('N', N-2, 'nEstimates', 1, 'np', model.
    noise_std, 'l', 100, 'verbose', true);
57
58 %=====
59 % vrft
60 %=====
61 for i = 1: exper
62     % get virtual signals
63     [e y1 rl] = f_get_vrft_e_nl(model, u, y');
64     % Apply filter
65     el = lsim(Filter, e);
66     [theta(i,:) cost] = f_rational_model(simul, m_rat, [u
        (1)], u(1:max(size(u))-2), el(2:max(size(e))), rl(2:
        max(size(e))), y);
67 end
68
69 mtheta = mean(theta);
70 std_theta = std(theta);
71 var_theta = var(theta);
72 cov_theta = cov(theta);
73
74 %=====
75 % Step simulation
76 %=====
77 r = ones(1,N); y = zeros(1,N); e = zeros(1,N); u2 = zeros
    (1,N); w = zeros(1,N);
78
79 for k=2:N
80     % get controller input signal
81     e(k) = r(k)-y(k-1);
82     uu = 0;
83     for kk=1:theta_size
84         if kk<=theta_size/2
85             uu = uu+mtheta(kk)*e(k)^(kk);
86         else
87             uu = uu+mtheta(kk)*e(k-1)^(kk-theta_size/2);
88         end
89     end
90     u2(k) = u2(k-1)+uu;
91     w(k+1) = gb1*w(k)+ga1*u2(k);
92     y(k) = (b1*w(k+1)+b3*w(k)^3);
93 end
94

```

```

95 % get VRFT costs
96 jmr=f_get_vrft_nl_Jmr(Td,r(1:N-1), y(2:N))
97 jvr=f_get_vrft_nl_Jvr(u, u2')
98
99 %=====
100 % Ploting
101 %=====
102 stairs(y(2:N), '-r')
103 hold;
104 step(Td);
105 plotting = step(Td);
106 v=zeros(1,N);
107 v2=zeros(1,N);
108
109 for k=1:N
110     aux=0;
111     for kk=1:theta_size/2
112         aux=aux+(mtheta(kk)/0.8)*e(k)^(kk);
113     end
114     v(k)=aux;
115 end
116 %=====
117 figure;
118 plot(v(2:N), w(3:N+1))
119 grid;
120 xlabel('v(t)');
121 ylabel('\omega(t)');
122 title('v(t) \times \omega(t)');
123 %=====
124 figure;
125 stairs(v(2:max(size(ploting))), 'r')
126 hold;
127 stairs(w(3:max(size(ploting))+1))
128 grid;
129 xlabel('t');
130 title('Resposta de v(t) e \omega(t) a um sinal do tipo
        degrau');
131 legend('v(t)', '\omega(t)');

```

Listagem C.2: Rotina para identificação de um controlador utilizando referência virtual em conjunto com o algoritmo para identificação de sistemas não lineares quando a planta pode ser modelada por uma classe de Wiener.

C.3 Identificação de um controlador racional para uma planta também racional

Na Listagem C.3 é apresentada a rotina para identificação do controlador proposto na Seção 5.5, onde o processo e o controlador são do tipo racional.

```

1 % VRFT method using a non-linear model
2 % real system: y(k)=0.9y(k-1)+tanh(u(t-1))
3 %=====
4 close all; clear all;
5 clc;
6 P=path;
7 path(P,'../functions')
8 f_set_path();
9
10 % simulation parameters
11 theta_size = 5;
12 m          = 2;
13 exper      = 100;
14 cut        = 1;
15 %=====
16 % init variables
17 theta = zeros(exper, theta_size);
18 [u N] = f_get_prbs(m);
19 %=====
20 % Real system parameters
21 a1=.5; a2=1; b1=.25; mn=0.4;
22 theta0 = [mn/a2 (1-mn)/a2 mn*b1/a2 (1-mn)*b1/a2 a1/a2];
23 %=====
24 model.mn      = [mn];
25 model.md      = [1 -(1-mn)];
26 model.TS      = 1;
27 model.delay   = 1;
28 model.delay_func = tf([1],[1 0], model.TS);
29 model.noise_std = 0.1;
30
31 Td = tf(model.mn, model.md, model.TS);
32 %=====
33 % plant simul
34 %=====
35 y = zeros(N, 1);
36 for k=3:N
37     y(k) = (a1*u(k-1)*y(k-1) + a2*u(k-1)) / (1+ b1*y(k-2)
38         ^2);
39 end
40 %=====
41 % Controller model definition
42 %=====
43 c_model.n_dim = 4;

```

```

43 c_model.dim      = 5;
44 c_model.err_model = 0;
45 c_model.texp     = [1 1 1 2 1];
46 c_model.yu       = [4 3 4 3 3];
47 c_model.regr     = [0 0 0 1 0];
48 c_model.yplus_yur = [0 0 3 3 0];
49 c_model.yplus_exp = [0 0 2 1 0];
50 c_model.yplus_regr = [0 0 1 0 0];
51 c_model.err_enable = true;
52 c_model.err_size  = 1;
53
54 %% Simulation parameters
55 simul = struct('N', N-1, 'nEstimates', 1, 'np', model.
    noise_std, 'l', 100, 'verbose', true);
56 %=====
57 % vrft
58 %=====
59 for i = 1:exper
60     [e yl rl] = f_get_vrft_e_nl(model, u, y);
61     [theta(i,:) cost] = f_rational_model(simul, c_model, [u
        (1)], u(1:max(size(u))-1), e, y(1:max(size(y))-1),
        rl);
62 end
63
64 mtheta = mean(theta)
65 vartheta = var(theta)
66 stdtheta = std(theta)
67 covtheta = cov(theta)
68
69 % output signal at ideal controller
70 uc0 = f_y_model([0], e, y, rl, theta0, c_model);
71 uc = f_y_model([0], e, y, rl, mtheta, c_model);
72
73 % VRFT cost of identification
74 Jvr_nl = f_get_vrft_nl_Jvr(uc0, uc)
75 %=====
76 % Plotting
77 y=zeros(N, 1); r=ones(N, 1); e=zeros(N, 1); u=zeros(N, 1);
78
79 for k=3:N
80     e(k) = r(k)-y(k-1);
81     u(k-1) = (mtheta(1)*r(k-1) +mtheta(2)*y(k-1) +mtheta(3)
        *y(k-2)^2*r(k-1) +mtheta(4)*y(k-2)^2*y(k-1)) / (1 +
        mtheta(5)*y(k-1));
82     y(k) = (a1*u(k-1)*y(k-1) +a2*u(k-1))/ (1 + b1*y(k-2)
        ^2);
83 end
84
85 ytd = step(Td, N-2);

```

```
86 stairs(ytd-y(2:N));
87 grid;
88 title('Erro entre a saída desejada e a obtida com o
        controlador estimado')
89 xlabel('t');
90 ylabel('Amplitude');
91
92 f_draw_ellipse(theta(:,1),theta(:,2),theta0(1),theta0(2));
93 f_draw_ellipse(theta(:,3),theta(:,4),theta0(3),theta0(4));
```

Listagem C.3: Rotina para identificação de um controlador racional quando a planta do sistema é também um modelo racional e o comportamento desejado do sistema em malha fechada é linear.