SANTOS DEMETRIO MIRANDA BORJAS

ESTUDO DA IDENTIFICAÇÃO POR SUBESPAÇOS EM MALHA ABERTA E FECHADA E PROPOSTA DE NOVOS ALGORITMOS

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Engenharia.

São Paulo 2009

SANTOS DEMETRIO MIRANDA BORJAS

ESTUDO DA IDENTIFICAÇÃO POR SUBESPAÇOS EM MALHA ABERTA E FECHADA E PROPOSTA DE NOVOS ALGORITMOS

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Engenharia.

Área de Concentração: Engenharia de Sistemas

Orientador: Prof. Dr. Claudio Garcia

São Paulo 2009

FICHA CATALOGRÁFICA

Borjas, Santos Demetrio Miranda

Estudo da identificação por subespaços em malha aberta e fechada e proposta de novos algoritmos / S.D.M. Borjas. -- São Paulo, 2009.

307 p.

Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1. Identificação de sistemas 2. Sistemas discretos (Modelagem; Simulação) 3. Sistemas lineares (Modelos matemáticos) I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e controle II. t.

Aos meus pais Rumaldo Miranda e Petronila Borjas e a minha nova família Alessandra Nayara e Rocio Elizabeth de Miranda.

AGRADECIMENTOS

A Deus, em primeiro lugar, por ter-me permitido concluir este trabalho.

Ao Prof. Dr. Claudio Garcia, que desde o inicio do Mestrado vem me orientando de uma maneira brilhante, fortalecendo meus conhecimentos e direcionando-me nas minhas inquietudes pessoais assim como acadêmicas.

Ao CNPQ – Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio financeiro na forma de bolsa, durante a execução deste trabalho.

Aos professores da Escola Politécnica, com os quais fiz as matérias do Doutorado.

A meus irmãos: Nicolaza Miranda, Juana Miranda, José Miranda, Flor Miranda e Gavidia Borjas pelo seu apoio.

Aos meus amigos peruanos, por sua amizade e companheirismo: em especial a Nestor Castañeda Centurion, por sua ajuda e constante apoio, Luis, Wilfredo, Manfred e tantos outros que direta e indiretamente ajudaram-me a construir este trabalho.

Aos meus amigos brasileiros pela ajuda desinteressada e pela amizade: Silvia, Natalia Butschkau, Fabiana Butschkau, Mariana, Claudia, Amadou, Patrícia, Alexander, Priscila.

A Marina Petroff Garcia, pela ajuda da tradução para Inglês dos artigos apresentados em diferentes congressos.

RESUMO

Neste trabalho se pesquisou a identificação por subespaços para modelos de sistemas lineares invariantes no tempo em espaço de estados operando em tempo discreto, aplicado a sistemas em malha aberta e malha fechada. Dentro da identificação em malha aberta foram estudados os casos determinístico e estocástico – determinístico para os métodos MOESP e N4SID. Como resultado deste estudo se apresentou um novo método, MON4SID, que usa a técnica MOESP para encontrar a matriz de observabilidade estendida e N4SID para recuperar as matrizes do sistema, a partir dos dados de entrada e saída. Duas variantes do método MON4SID são apresentadas para o caso determinístico e uma para determinístico - estocástico. Para verificar seu desempenho, este método foi aplicado a três tipos de processos: processo MIMO, sistema SISO e um processo *benchmark* MIMO da Shell. Foram usados nove algoritmos para identificar os diferentes processos e seus resultados foram comparados. Em todos eles o método MON4SID teve um bom desempenho.

Para o caso de identificação em malha fechada, foram estudados os métodos MOESPC e N4SIDC para malha fechada. Estes métodos surgem como uma extensão dos métodos aplicados para malha aberta, mas a forma de solucionar o problema para malha fechada é diferente, por exemplo, o método MOESP computa um sistema global a partir do qual as matrizes da planta e do controlador são estimadas, através de uma redução de ordem e o método N4SID computa as matrizes da planta através do problema de mínimos quadrados, mas é necessário conhecer os parâmetros de Markov do controlador. Como resultado deste estudo foi apresentado o método MON4SIDC, o qual se baseia no método MON4SID para malha aberta. Este novo método usa a técnica MOESP para computar o sistema global e depois as matrizes da planta são estimadas por meio do método dos mínimos quadrados. Neste método não é necessário ter nenhum conhecimento do controlador. Um sistema simulado é usado para avaliar o desempenho do algoritmo MON4SIDC e seu resultado é comparado com outros algoritmos existentes na identificação para malha fechada PEM, N4SIDC, ARXS e MOESPC. O método MON4SIDC teve um bom desempenho, inclusive para sistemas com ruído, o que não aconteceu para o caso PEM e ARXS.

ABSTRACT

In this work it was researched subspace identification for models of linear time invariant systems in state space operating in discrete time, applied to open and closed loop systems. In the open loop identification the deterministic and stochastic - deterministic cases were studied for the methods MOESP and N4SID. As a result of this study a new method MON4SID is presented, which uses the technique MOESP to find the extended observability matrix and N4SID to compute the system matrices, from a set of input-output measurements. Two variants of the method MON4SID are presented for the case deterministic and one for deterministic - stochastic. To verify its performance, this method was applied to three types of processes: process MIMO, system SISO and a MIMO benchmark process of Shell. Nine algorithms were used to identify the different processes and their results are compared. In all of them the method MON4SID had a good performance.

For the closed loop identification, the methods MOESPC and N4SIDC were studied. These methods appear as an extension of the methods applied for open loop identification, but the form of solving the problem for closed loop is different. For instance, the method MOESP computes a global system from which the plant matrices and controller matrices are computed through an order reduction and the method N4SID computes the plant matrix through the problem of least squares, but it is necessary to know the Markov parameters of the controller. As a result of this study the method MON4SIDC was presented, which uses the method MON4SID for open loop. This new method uses the technique MOESP to compute the global system and the plant matrices are estimated through the problem of least squares. In this method it is not necessary to have any knowledge about the controller. A simulated system is used to evaluate the performance of the algorithm MON4SIDC and its results are compared with other existent algorithms for closed loop identification: PEM, N4SIDC, ARXS and MOESPC. The method MON4SIDC presented a good performance, even with systems with noise, what did not occur with PEM and ARXS.

SUMÁRIO

Lista de Figuras
Lista de Tabelas
Lista de Abreviaturas

1.	INT	RODUÇÃO	1
	1.1	Objetivo	3
	1.2	Justificativa	3
	1.3	Metodologia	4
	1.4	Revisão Bibliográfica	4
		1.4.1 Literatura básica	4
		1.4.2 Literatura avançada	5
	1.5	Estrutura do trabalho	9
2.	MÉ	ODO DE IDENTIFICAÇÃO POR SUBESPAÇO	13
	2.1	Teoria da realização	13
	2.2	Método Ho-Kalman	16
	2.3	Identificação por subespaços	19
		2.3.1 Método MOESP	22
		2.3.2 Algoritmo básico MOESP	22
		2.3.2.1 Teorema (MOESP ORDINÁRIO, caso determinístico)	
		[VERHAEGEN, 1992a]	26
		2.3.3 Algoritmo PIMOESP	28
		2.3.3.1 Teorema 2 (PIMOESP) [VERHAEGEN, 1992b]	34
		2.3.4 Algoritmo POMOESP	36
		2.3.4.1 Teorema 3 (POMOESP) [VERHAEGEN, 1994]	42
3.	IDE	NTIFICAÇÃO DE SISTEMAS PELO MÉTODO N4SID	44
	3.1	Modelos por subespaços	44

3.2 Definição da Projeção Oblíqua.....

47

	3.3	Método N4SID	50
		3.3.1 Método N4SID, Identificação determinística	50
		3.3.1.1 Teorema 1: Identificação determinística	55
		3.3.2 O Método N4SID, Identificação determinística – estocástica	59
		3.3.2.1 Teorema 2. Filtro de Kalman para estado não-estacionário de	
		sistema determinístico – estocástico	65
		3.3.2.2 Teorema 3. Projeção ortogonal	66
		3.3.2.3 Teorema 4: identificação determinística-estocástica	68
4.	ALO	GORITMOS PROPOSTOS	74
	4.1	Algoritmo proposto: MON4SID	74
		4.1.1 Método MON4SID 1 para sistema determinístico usando estados	76
		4.1.2 Método MON4SID 2 para sistema determinístico usando matriz	
		de observabilidade estendida	78
		4.1.3 Método MON4SID 3 para sistema determinístico usando estados	
		totais	79
	4.2	Método MON4SID 4 para sistema determinístico – estocástico	81
		4.2.1 Método MON4SID 5 para sistema determinístico – estocástico.	81
5.	SIM	IULAÇÃO DOS ALGORITMOS DETERMINÍSTICOS E	
	EST	OCÁSTICOS	82
	5.1	Identificação determinística	83
		5.1.1 Identificação puramente determinística	84
		5.1.1.1 Modelo em espaço de estados com matrizes aleatórias	85
		5.1.1.2 Modelo em espaço de estados usado por [VERHAEGEN,	
		1993] e [VAN OVERSCHEE; DE MOOR, 1996]	85
		5.1.1.3 Modelo de um processo <i>benchmark</i> da Shell	91
		5.1.2 Identificação com ruído de baixa (3%) e alta (80%) intensidade	
		para o modelo dado em 5.1.1.1.	95
	5.2	Identificação determinístico - estocástica	99
		5.2.1 Exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]	103

		5 2 2 Madala	usada par IVEDIJAECEN 1002. VAN OVEDSCHEE	104
		5.2.2 Modelo	usado por [verhaegen, 1993; van Overschee;	104
			JR, 1996]	109
		5.2.3 Modelo	de um processo <i>benchmark</i> da Shell	112
(IDE	NTIFICAÇÃO	DE CICTEMAS DOD CUDECDACOS EM MALHA	
0.		NTIFICAÇAU Hada	DE SISTEMAS POR SUBESPAÇOS EM MALHA	120
				120
	0.1		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	120
	6.2	Metodo N4SII		122
		6.2.1 Model	o em espaço de estados em malha fechada	123
		6.2.2 Problem	ma de identificação em malha fechada	123
		6.2.3 Idéia b	ásica para a identificação por subespaços em malha	
		fechad	a	125
		6.2.4 Definio	ção 1 Estado do filtro de Kalman	126
		6.2.5 Teorem	na da projeção para malha fechada	127
		6.2.5.1 Teorem	na 1 Teorema da projeção em malha fehcada	127
		6.2.6 Teorem	na principal para malha fechada	128
		6.2.6.1 Teorem	na 2 Teorema principal para malha fechada	128
		6.2.6.2 Algori	tmo 1: Usando estados	129
		6.2.6.3 Algorit	tmo 2: Usando estados	133
7.	IDE	NTIFICAÇÃO) EM MALHA PELO MÉTODO MOESPC	135
	7.1	Modelo em es	paço de estados em malha fechada método MOESPC	135
	7.2	Problema de id	dentificação em malha fechada (método MOESPC)	136
		7.2.1 Idéia bás	sica para a identificação por subespaços em malha	
		fechada	(método MOESPC)	136
	7.3	Identificação e	em malha fechada pelo método MOESPC	137
		7.3.1 Teorema	1	140
0	N a the			
δ.	ME		SIDU PKUPUSI U PAKA MALHA FECHADA E	144
	SIM	ULAÇUES		144

	3	
8.1	Método MON4SIDC para malha fechada	144
	8.1.1 Problema de identificação em malha fechada	146

		8.1.2 I	dentificação pelo método MON4SIDC	146
	8.2	Simula	ações	148
		8.2.1	Identificação puramente determinística	149
		8.2.2	Identificação com pouco ruído	158
		8.2.3	Identificação com muito ruído	162
9.	CON	NCLUS	ÕES E FUTURAS PESQUISAS	166
	9.1	Para s	istemas em malha aberta	166
	9.2	Para s	istemas em malha fechada	168
	9.3	Contri	buições deste trabalho	169
	9.4	Trabal	lhos futuros	170
		A		
R	EFEF	RENCL	AS BIBLIOGRAFICAS	171
AP	ÊND	ICE A		177
AP	ÊND	ICE B.		218
AP	APÊNDICE C			
AP	APÊNDICE D			
AP	ÊND	ICE E.		276
AP	APÊNDICE F 2			

LISTA DE FIGURAS

Figura 2.1	-	Resposta para um sistema SLIT com entrada zero 10
Figura 2.2	-	Sistema dinâmico com entradas $u(k)$, saídas $y(k)$ e distúrbios $v(k)$ e $w(k)$
Figura 2.3	-	Identificação por subespaços versus identificação clássica 2
Figura 2.4	-	Modelo determinístico
Figura 2.5	-	Modelo com ruído aditivo na saída
Figura 2.6	-	Modelo com ruído corrompido na saída e ruído nos estados 30
Figura 2.7	-	Aplicação f entre a sequência dos dados de entrada – saída eos parâmetros desconhecidos do sistema
Figura 3.1	-	Interpretação da projeção ortogonal no espaço <i>j</i> -dimensional (<i>j</i> =2 neste caso)
Figura 3.2	-	Interpretação da projeção ortogonal no espaço <i>j</i> -dimensional (<i>j</i> =3 neste caso)
Figura 3.3	-	Interpretação da projeção oblíqua no espaço <i>j</i> -dimensional $(j=3 \text{ neste caso e } p = q = r = 1)$.
Figura 3.4	-	Um sistema linear determinístico invariante no tempo
Figura 3.5	-	Diagrama esquemático de obtenção das matrizes do sistema 57
Figura 3.6	-	Sistema linear combinado determinístico-estocástico
Figura 3.7	-	Descrição do procedimento de identificação determinística- estocástica por subespaços
Figura 5.1	-	Modelo dos sistemas caixa preta
Figura 5.2	-	Sistema puramente determinístico
Figura 5.3	-	Sinal de saída $y1$ e entrada $u1$ usados na identificação do modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)
Figura 5.4	-	Sinal de saída y^2 e entrada u^2 usados na identificação do modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)

Figura 5.5	-	Valores singulares para o modelo em espaço de estados com matrizes aleatórias (identificação puramente determinístico)
Figura 5.6	-	Comparação das respostas do processo real (linha continua) versus modelo (linha pontilhada) em espaço de estados com matrizes aleatórias (caso puramente determinístico)
Figura 5.7	-	Comparação através do FIT das respostas do processo real versus modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)
Figura 5.8	-	Comparação da resposta em freqüência do processo versus modelo em espaço de estados com matrizes aleatórias (identificação puramente determinístico)
Figura 5.9	-	Sinal de saída e entrada usados na identificação do modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (caso puramente determinístico)
Figura 5.10	-	Valores singulares para o modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (identificação puramente determinístico)
Figura 5.11		Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) em espaço de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (caso puramente determinístico)
Figura 5.12		Comparação da resposta em freqüência do processo versus modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (identificação puramente determinístico)
Figura 5.13		Modelo <i>benchmark</i> da Shell com entradas D , Q e saídas P , X
Figura 5.14		Valores singulares para o modelo <i>benchmark</i> da Shell (identificação determinístico)
Figura 5.15		Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) <i>benchmark</i> da Shell (caso determinístico)
Figura 5.16		Sistema com ruído na medição

Figura 5.17	Dados de entrada com 3% de ruído para o modelo em espaço de estados com matrizes aleatórias
Figura 5.18	Dados de entrada com 80% de ruído para o modelo em espaço de estados com matrizes aleatórias
Figura 5.19	Dados de saída com 3% de ruído para o modelo em espaço de estados com matrizes aleatórias
Figura 5.20	Dados de saída com 80% de ruído para o modelo em espaço de estados com matrizes aleatórias
Figura 5.21	Sistema determinístico – estocástico na forma inovativa
Figura 5.22	Sinal de saída y1 e entrada u1 usado para identificação do exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]
Figura 5.23	Sinal de saída y2 e entrada u2 usado para identificação do exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]
Figura 5.24	Valores singulares do exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]
Figura 5.25	Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) para o exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]
Figura 5.26	Comparação da resposta em freqüência do processo versus modelo (exemplo apresentado por [OVERSCHEE; DE MOOR, 1996])
Figura 5.27	Sinal de saída <i>y</i> e sinal de entrada <i>u</i> usado na identificação do modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]
Figura 5.28	Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]
Figura 5.29	Comparação da resposta em freqüência do processo real versus modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]
Figura 5.30	Sinal de saída y1 e sinal de entrada u1 usados para identificação do modelo de um processo <i>benchmark</i> da Shell (com 50% de ruído)

Figura 5.31	Sinal de saída y2 e sinal de entrada u2 usados na identificação do modelo de um processo <i>benchmark</i> da Shell (com 50% de ruído)
Figura 5.32	Valores singulares do modelo de um processo <i>benchmark</i> da Shell (com 50% de ruído) 1
Figura 5.33	Espectro de valores AIC do modelo de um processo benchmark da Shell (com 50% de ruído) 1
Figura 5.34	Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) de um processo <i>benchmark</i> da Shell (com 50% de ruído) 1
Figura 5.35	Sistema em malha fechada do modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996]
Figura 5.36	Espectro de valores AIC do modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996]
Figura 5.37	Comparação da resposta em freqüência do processo real versus modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] 1
Figura 6.1	Modelo em malha fechada 1
Figura 7.1	Representação em blocos de um modelo em malha fechada 1
Figura 7.2	Modelo Global em bloco 1
Figura 8.1	Representação de um modelo em malha fechada 1
Figura 8.2	Sistema puramente determinístico em malha fechada 1
Figura 8.3	Sinal de entrada r e saída y usados na identificação em malha fechada (puramente determinística) 1
Figura 8.4	Valores singulares para o sistema em malha fechada (identificação puramente determinística) 1
Figura 8.5	Valores singulares para computar a planta do sistema em malha fechada (identificação puramente determinística) 1
Figura 8.6	Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método MON4SID)

Figura 8.7	Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método MON4SDIDC)
Figura 8.8	Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística método N4SIDC)
Figura 8.9	Valores singulares Valores singulares para computar a planta do sistema em malha fechada (identificação puramente determinística, método N4SIDC)
Figura 8.10	Comparação do diagrama de Bode da planta do sistema em malha fechada (puramente determinística, método N4SIDC) 15
Figura 8.11	Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método N4SIDC)
Figura 8.12	Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método ARXS)
Figura 8.13	Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método ARXS)
Figura 8.14	Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método PEM)
Figura 8.15	Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método PEM)
Figura 8.16	Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método MOESP)
Figura 8.17	Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método MOESPC)
Figura 8.18	Sinal de entrada <i>r</i> e saída <i>y</i> usados na identificação em malha fechada com pouco ruído

Figura 8.19	Comparação do diagrama de Bode e alocação de pólos pelo método MON4SIDC (identificação com pouco ruído)	159
Figura 8.20	Comparação do diagrama de Bode e alocação de pólos pelo método N4SIDC (identificação com pouco ruído)	160
Figura 8.21	Comparação do diagrama de Bode e alocação de pólos pelo método ARXS (identificação com pouco ruído)	160
Figura 8.22	Comparação do diagrama de Bode e alocação de pólos pelo método PEM (identificação com pouco ruído)	161
Figura 8.23	Comparação do diagrama de Bode e alocação de pólos pelo método MOESPC (identificação com pouco ruído)	161
Figura 8.24	Comparação do diagrama de Bode para $n = 9$ pelo método MOESPC (identificação com pouco ruído)	162
Figura 8.25	Sinal de entrada <i>r</i> e saída <i>y</i> usados na identificação em malha fechada com muito ruído	163
Figura 8.26	Comparação do diagrama de Bode e alocação de pólos para método MON4SIDC (identificação muito ruído)	163
Figura 8.27	Comparação do diagrama de Bode e alocação de pólos para método N4SIDC (identificação muito ruído)	164
Figura 8.28	Comparação do diagrama de Bode e alocação de pólos para método ARXS (identificação muito ruído)	164
Figura 8.29	Comparação do diagrama de Bode e alocação de pólos para método PEM (identificação muito ruído	165
Figura 8.30	Comparação do diagrama de Bode e alocação de pólos para método MOESPC (identificação muito ruído)	165
Figura A.1	Vetor v em combinação linear de v_1 e v_2	181
Figura A.2	Espaços complementares, observa-se que $v \in W^{\perp}$	194
Figura A.3	Transformação de Householder	196
Figura A.4	Projeção ortogonal	204
Figura A.5	Projeção sobre uma reta	206
Figura A.6	Projeção ortogonal de um vetor <i>b</i> sobre um subespaço gerado por <i>A</i>	207

Figura A.7	Projeção ortogonal de <i>A</i> no espaço linha de <i>B</i>	209
Figura A.8	Projeção oblíqua	210
Figura D4.1	Projeção oblíqua das saídas futuras $Y_{i 2i-1}$ em $M_{i 2i-1}$ sobre $W_{0 i-1}$	272

LISTA DE TABELAS

Tabela 1	-	Modelo em espaço de estados gerado na forma aleatória	86
Tabela 2	-	Gerando o sinal de entrada e saída	86
Tabela 3	-	Resultados numéricos do desempenho dos algoritmos para o modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)	88
Tabela 4	-	Gerando os dados de entrada e saída	92
Tabela 5	-	Resultados numéricos do desempenho dos algoritmos para o modelo em espaço de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996]	93
Tabela 6	-	Resultados numéricos do desempenho dos algoritmos para o modelo <i>benchmark</i> da Shell (caso determinístico)	97
Tabela 7	-	Dados coletados com ruído	99
Tabela 8	-	Resultados numéricos do desempenho dos algoritmos com ruído de baixa intensidade (3%) aplicada ao modelo em espaço de estados com matrizes aleatórias	102
Tabela 9	-	Resultados numéricos do desempenho dos algoritmos com ruído de alta intensidade (80%) aplicada ao modelo em espaço de estados com matrizes aleatórias	102
Tabela 10	-	Sinal de saída para o sistema determinístico – estocástica.	105
Tabela 11	-	Coletando os dados de entrada e saída para o sistema determinístico – estocástica	106
Tabela 12	-	Resultados numéricos do desempenho dos algoritmos para o exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]	107
Tabela 13	-	Resultados numéricos do desempenho dos algoritmos para o modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]	111
Tabela 14	-	Resultados numéricos do desempenho dos algoritmos para o modelo de um processo <i>benchmark</i> da Shell (com 50% de ruído)	115

LISTA DE ABREVIATURAS

ARXS Algoritmo de Ljung e MacKelvey (1996) CCA Algoritmo de Katayama (2005) para sistemas determinísticos estocásticos **CVA** Canonical Variate analysis IVM Instrumental Variable Methods MNS 1 MON4SID 1 (empregando estados) MNS 2 MON4SID 2 (empregando a matriz de observabilidade estendida) MNS 3 MON4SID 3 (empregando estados totais) MNSE1 Algoritmo MON4SID 4 MNSE2 Algoritmo MON4SID 5 MOESP Multivariable Output-Error State space MOESPC Algoritmo de Verhaegen (1993) apresentado no capítulo 7 Método MOESP implementado por Michael Verhaegen (para **MOESPV** sistema determinístico estocástico) Combina os métodos MOESP e N4SID MON4SID MON4SID 4 Extensão do método MON4SID 4 (para sistema determinístico estocástico) MON4SID 5 Extensão do método MON4SID 5 (para sistema determinístico estocástico) Algoritmo MON4SID para malha fechada MON4SIDC **MPVerh** Método MOESP implementado por Michael Verhaegen N4SID Numerical algorithms for Subspace State Space System Identifcation N4SID Encontra-se implementado no Toolbox do Matlab N4sid 1 Algoritmo determinístico 1 (empregando estados)

N4sid 2	Algoritmo determinístico 2 (empregando matriz de observabilidade estendida)
N4SIDC	Algoritmo 1 de Van Overschee; De Moor, (1997) apresentado no capítulo 6
N4sidE	Algoritmo 1 N4SID (para sistema determinístico estocástico)
N4sidMatlab	Algoritmo N4SID encontra-se implementado no Toolbox do Matlab (para sistema determinístico estocástico)
N4sidO	Algoritmo 2 N4SID (para sistema determinístico estocástico)
PEM	Prediction Error Method
PIMOESP	Past-Input do algoritmo MOESP
POMOESP	Past - Output do algoritmo MOESP
SLIT	Sistema Linear Discreto Invariante no Tempo
SVD	Decomposição em valores singulares
VI	Variáveis Instrumentais

CAPÍTULO 1

1. INTRODUÇÃO

A identificação em malha fechada é de grande interesse para muitas aplicações da engenharia. Por razões de segurança ou restrições de qualidade, é necessário que os experimentos para identificação sejam executados em malha fechada, como indicado por muitos pesquisadores [SÖDERSTROM; STOICA, 1989; LJUNG, 1999].

Os métodos de identificação em malha fechada tiveram seu desenvolvimento nos últimos 30 anos. A análise de sob que condições é possível a identificação em malha fechada, foi tema de muita atenção para vários pesquisadores [BOX; MACGREGOR, 1976; GUSTAVSSON; LJUNG; SÖDERSTROM, 1977; SÖDERSTROM; STOICA, 1989]. Por isto, foram desenvolvidas várias estratégias para esta identificação [SÖDERSTROM; STOICA, 1989; LJUNG, 1999]. Diferentes considerações foram feitas para o problema de identificação em malha fechada, vários esquemas foram sugeridos e diferentes configurações da malha realimentada foram consideradas. Desta forma, muitos sistemas e modelos foram parametrizados de diferentes maneiras. O problema fundamental da identificação em malha fechada é a correlação entre os dados de entrada e os ruídos não medidos. A identificação tradicional em malha fechada está dentro do enfoque do método de predição de erro (PEM). Um estudo nesta área foi feito por Forssell e Ljung (1999). Baseado nessas análises, os métodos de identificação em malha fechada podem ser classificados em três grandes grupos: identificação direta, identificação indireta e identificação conjunta dos dados de entrada - saída. A vantagem do método PEM é que a convergência e variância assintótica dos resultados estão disponíveis, para maiores detalhes ver [LJUNG, 1978; LJUNG, 1985]; e sua desvantagem é que envolve parametrizações complicadas, o que representa uma dificuldade para a identificação de sistemas de múltiplas entradas e múltiplas saídas (MIMO). Como uma alternativa para este problema, surgem os métodos de identificação por subespaços. Dado que sistemas de grandes dimensões são comumente encontrados na indústria, a aplicação dos algoritmos de identificação por subespaços neste campo é bastante promissora [MERCÉRE, et al, 2008; ROBERTO, et al, 2008; VAN WINGERDEN, et al, 2007; VAN WINGERDEN; VERHAEGEN, 2008a; VAN WINGERDEN; VERHAEGEN, 2008b]. Quando se considera um sistema em malha aberta, os métodos de identificação por subespaços têm grande aceitação na indústria, devido a uma parametrização simples e geral para sistemas MIMO. Estes métodos estão dentro do teorema unificado proposto por [VAN OVERSCHEE; DE MOOR, 1995], que engloba vários métodos, dentre os quais tem-se: Canonical Variate Analysis (CVA) [LARIMORE, 1990], Multivariable *Output-Error* (MOESP) State sPace [VERHAEGEN; DEWILDE, 1992], Numerical Algorithms for Subspace State Space (N4SID) [VAN OVERSCHEE; DE MOOR, 1994]. Baseado na unificação do teorema, todos estes algoritmos podem ser interpretados como uma decomposição de valores singulares de uma matriz peso. Propriedades estatísticas tais como consistência e eficiência dos algoritmos foram pesquisadas por [JANSSON; WAHLBERG, 1998; KNUDSEN, 2001; GUSTAFSSON, 2002; BAUER; LJUNG, 2002; BAUER, 2003]. A identificação por métodos de subespaços para sistemas MIMO operando em malha fechada vem sendo tema de pesquisa na última década. Na identificação conjunta dos dados de entrada e saída aplicando métodos por subespaços pode-se citar [VERHAEGEN, 1993]. Para esta identificação, se faz a suposição que o canal da realimentação seja linear. Isto geralmente demanda mais esforço computacional, por isso várias tentativas foram realizadas para identificar diretamente a dinâmica da planta, sem estimar o canal de realimentação. Por exemplo, a modificação do método N4SID (malha aberta) para o método de identificação por subespaços em malha fechada foi apresentada em Van Overschee e Bart De Moor (1997). No entanto, esta identificação requer o conhecimento de um número finito dos parâmetros de Markov do controlador. Vários outros modelos de identificação em malha fechada, baseados em subespaços, foram propostos, ver [LJUNG; MCKELVEY, 1996; JANSSON, 2003; LJUNG, 2003; CHIUSO; PICCI, 2005; KATAYAMA, 2005; VAN WINGERDEN, et al, 2009]. Em geral, estes métodos necessitam de alguma informação do sistema e de suas propriedades estatísticas, para maiores detalhes ver [BAUER, 2004; CHIUSO; PICCI, 2004].

1.1Objetivo

O objetivo deste trabalho é estudar uma nova abordagem para o problema de identificação em malha fechada, aplicando-se técnicas por subespaços, enfatizando-se os seguintes pontos:

- a) Estudar os métodos de identificação por subespaços (MOESP e N4SID) para sistemas em malha aberta que sejam lineares e invariantes no tempo, operando em tempo discreto.
- b) Estudar os métodos de identificação por subespaços (MOESPC e N4SIDC) para sistemas em malha fechada, que sejam lineares e invariantes no tempo, operando em tempo discreto.
- Propor métodos de identificação para malha aberta como resultado da mistura dos métodos MOESP e N4SID.
- d) Comparar estes métodos com um método tradicional de identificação, a saber,
 PEM "prediction error method", mediante exemplos simulados.
- Propor métodos de identificação para malha fechada como resultado da mistura dos métodos MOESPC e N4SIDC.
- f) Comparar estes métodos com um método tradicional de identificação, isto é, PEM, mediante exemplos simulados.

1.2 Justificativa

A principal motivação para a realização deste trabalho reside nos bons resultados obtidos por Peter Van Overschee e Bart De Moor (1997). Este trabalho apresenta dois teoremas para a identificação de sistemas em malha fechada. que podem ser aplicados também (com algumas restrições) ao caso em malha aberta.

Os métodos de identificação por subespaços representam uma modesta carga computacional quando comparados com os métodos tradicionais como PEM, especialmente quando o número de entradas e saídas é grande.

Uma desvantagem destes métodos é que precisam conhecer alguma informação do controlador. Outra desvantagem é que se a ordem do modelo for grande, tem que se aplicar um método para reduzi-la, o que leva a fazer novas iterações no algoritmo. Além disso, existem os problemas de instabilidade e de cancelamento de pólos e zeros, o que restringe a planta a ter fase mínima.

1.3 Metodologia

É necessário o uso de ferramentas da teoria de sistemas, da geometria e da álgebra linear numérica. Estima-se a matriz em blocos de Hankel (M) a partir dos dados de entrada e saída, fatora-se esta matriz (M) para obter estimativas das matrizes de observabilidade e controlabilidade estendida e a ordem do modelo. Por fim, estimam-se as matrizes do sistema a partir das estimativas (M) e da matriz de observabilidade estendida.

1.4 Revisão bibliográfica

As referências bibliográficas para este trabalho são divididas em dois grandes grupos:

- a) Literatura básica: este grupo é composto por livros que descrevem os conceitos básicos da teoria de sistemas lineares, propriedades matemáticas de sistemas dinâmicos em malha fechada e álgebra linear.
- b) Literatura avançada: neste grupo se encontram os trabalhos específicos que descrevem os métodos associados à identificação de sistemas em malha fechada usando métodos por subespaços. Foram retirados de publicações específicas ou de teses relacionadas ao tema.

Seguem algumas indicações das principais referências adotadas para cada um dos grupos descritos acima, bem como uma breve descrição do conteúdo de cada obra citada e comentários relevantes.

1.4.1 Literatura básica

Sistemas lineares

[OGATA,1997] contém uma introdução aos sistemas dinâmicos. Ele prima pelo caráter didático e pela clareza de exposição. Para uma introdução a sistemas em tempo discreto fez-se uso da obra [OGATA, 1995], na qual se encontra no capítulo 5 representações em espaço de estados de sistemas em tempo discreto.

Identificação de sistemas por métodos tradicionais

[GARCIA, 2001; AGUIRRE, 2004], nos quais se encontram definições básicas de identificação de sistemas, modelos matemáticos, modelos de processos e de perturbações, bem como se abordam as técnicas clássicas de identificação de sistemas.

Álgebra linear

[STRANG, 1993] foi usado para se estudar: vetores, matrizes, subespaços, ortogonalidade para subespaços, decomposição QR e a decomposição de valores singulares (SVD).

Processos estocásticos

[PAPOULIS, 1984]: neste livro se encontram definições como processos estacionários e não estacionários, assim como propriedades de processos estocásticos.

[KOVACS, 1996] é um livro muito didático. Nele se revisam conceitos sobre variáveis aleatórias, processos estocásticos, ergodicidade, processos de Markov e processos aleatórios gaussianos.

1.4.2 Literatura avançada

Os seguintes livros foram utilizados no trabalho:

[HEALY, 1986]: onde se pode encontrar conceitos de projeções ortogonais de subespaços e algumas propriedades estatísticas.

[VAN OVERSCHEE; DE MOOR, 1996]: se constitui em uma das referências básicas para a realização deste trabalho. Apresenta a identificação por subespaços para malha aberta de um modo bastante detalhado.

[KAILATH; SAYED, 1999]: onde se pode encontrar a definição de matriz em blocos de Toeplitz e Hankel, as quais foram usadas para o processo de identificação por subespaços.

[LJUNG, 1999]: é um livro básico para a identificação de sistemas clássicos, mas nos últimos capítulos trata um pouco da identificação por subespaços, onde se compreende as definições de matrizes de ponderação.

[PEREIRA DA SILVA, 2003]: é um material didático onde se pode achar definições de sistema observável, não observável, controlável, não controlável, estabilidade, assim

como suas respectivas propriedades e caracterizações.

Os seguintes artigos tratam de identificação por subespaços em malha aberta: [MOONEN *et al.*, 1992]: apresenta algoritmos por subespaços para identificação dos modelos em espaço de estados para sistemas discretos, diretamente dos dados de entrada/saída.

[VAN OVERSCHEE; DE MOOR, 1994a]: trata da identificação por subespaços para sistemas lineares discretos invariantes no tempo, aplicando o algoritmo N4SID para o caso MIMO.

[VAN OVERSCHEE; DE MOOR, 1994b]: descreve como podem ser determinadas as bases do espaço de estados dos modelos identificados com os algoritmos de identificação por subespaços.

[FAVOREEL *et al.*, 2000]: mostra, de forma geral, a identificação por subespaços, faz uma comparação dos algoritmos na literatura como N4SID, IV- 4SID, MOESP, CVA.

[VERHAEGEN, 1994a]: realiza um estudo da identificação em malha aberta, mostra dois algoritmos para tal identificação, gera outra variante do algoritmo MOESP, explica a diferença entre o algoritmo MOESP e o algoritmo N4SID.

[VAN OVERSCHEE; DE MOOR, 1995]: neste artigo se indica a similaridade dos três (MOESP, CVA, N4SID) algoritmos de identificação por subespaços para sistemas combinados (determinístico e estocástico). Mostra-se que os três algoritmos são casos especiais de um teorema geral.

[DE MOOR *et al.*, 1999]: neste artigo se encontram conceitos sobre identificação por subespaços para sistemas discretos, contínuos, lineares e bilineares.

[GUSTAVSSON, 2002]: apresenta uma variante do modelo MOESP em relação a variáveis instrumentais e matrizes peso, para a identificação por subespaços em malha

aberta. Estuda-se a consistência do modelo proposto.

[DE COCK; DE MOOR, 2003]: este artigo dá uma introdução curta sobre os algoritmos de identificação por subespaços para malha aberta. Trata-se dos algoritmos determinístico, estocástico e a combinação de ambos.

[CHUI; MACIEJOWSKI, 2005]: enfoca a identificação por subespaços mediante a aproximação dos primeiros parâmetros de Markov, depois encontra as matrizes do sistema em malha aberta da mesma forma que os métodos encontrados na literatura por subespaços.

Os seguintes artigos tratam de identificação por subespaço em malha fechada: [GUSTAVSSON *et al.*, 1977]: neste artigo se mostra como o método de predição de erro é aplicado para a identificação em malha fechada. Encontra-se também uma classificação dos diferentes tipos de algoritmos com relação à estrutura do modelo, aos métodos aplicados, dentre outros. Expõe-se também um resumo de cada um destes métodos.

[LJUNG; MCKELVEY, 1996]: apresenta uma combinação entre os métodos por subespaços e o método tradicional (ARX). A idéia é encontrar uma estimativa da seqüência de espaço de estados. Para este fim, aplicando-se o método ARX encontra-se a seqüência de saídas com k - passos preditores, aplica-se na seqüência encontrada a decomposição de valores singulares para achar a ordem do sistema. Por último, usando a técnica de mínimos quadrados, se encontram as matrizes do sistema (técnica usada nos métodos por subespaços).

[VAN OVERSCHEE; DE MOOR, 1997]: para obter a identificação por subespaços em malha fechada é preciso coletar os *i* primeiros parâmetros de Markov do controlador. Baseado em dois novos teoremas da projeção ortogonal e oblíqua são obtidas as matrizes do sistema. Apresenta-se uma estratégia de passagem destes teoremas para aqueles utilizados na identificação em malha aberta.

[CHOU; VERHAEGEN, 1997]: a identificação em malha fechada, utilizando o método por subespaços, é baseada na formulação das variáveis de erro (EIV). Este método é similar ao aplicado no método MOESP, mas utiliza uma análise estocástica do processo.

[FORSSELL; LJUNG, 1999]: enfoca a identificação em malha fechada via o método tradicional da predição de erro (PEM). Apresenta também um estudo das propriedades estatísticas relacionadas à função de transferência e aos parâmetros estimados.

[WANG *et al.*, 1999]: apresenta um novo algoritmo baseado no modelo N4SID para a identificação em malha fechada, o qual não precisa conhecer os parâmetros de Markov do controlador e sim o controlador, os sinais de referência e os dados de saída; com estes dados obtêm-se as matrizes da planta.

[KATAYAMA, 2002]: com base nos resultados do artigo [PICCI; KATAYAMA, 1996], apresenta-se um novo método de identificação por subespaços aplicado à malha fechada, usando a decomposição ortogonal do processo em conjunto de entradas e saídas. O problema de identificação é dividido em dois subproblemas: identificação da componente determinística e da componente estocástica. Para isto, pode-se aplicar qualquer dos métodos por subespaços dados na literatura.

[ZHAO; WESTWICK, 2003]: neste artigo apresenta-se uma modificação do método MOESP para sistemas em malha fechada. Usando as entradas das referências passadas como variável instrumental (PR-MOESP), elabora uma comparação deste método com o método EIV-MOESP.

[CHIUSO; PICCI 2004]: neste artigo se analisam as polarizações da identificação por subespaços em malha fechada, dando ênfase ao estudo da condição inicial. Além disso, mostra-se um procedimento alternativo para a construção do vetor de espaço de estados e de um novo método de identificação por subespaços.

[HUANG et al., 2005]: propõe-se uma identificação em malha fechada usando o método da projeção ortogonal e a decomposição em valores singulares, apresenta-se

também um novo modelo baseado no modelo de erro na variável (EIV). Propõe um método para eliminar as polarizações dos modelos por subespaços.

[KATAYAMA *et al.*, 2005]: o objetivo deste artigo é, aplicando qualquer método tradicional por subespaços, obter o modelo em malha fechada em espaço de estados da planta e do controlador, baseado nos dados do sinal de referência, nos dados de entrada, nos dados de saída e em uma entrada adicional exógena denominada sinal determinístico. Este problema é similar àquele tratado em [VERHAEGEN, 1993], mas a forma de encontrar a planta e o controlador é diferente.

[CHIUSO; PICCI, 2005]: neste artigo demonstra-se que se os dados amostrados são infinitos, baseado na idéia do modelo de identificação para predição, a identificação por subespaços aplicada à malha fechada será bem sucedida.

[QIN, 2006]: uma visão geral da identificação por métodos de subespaços é apresentada neste artigo e tratada para o caso de identificação em malha aberta e malha fechada. Esta apresentação é muito didática.

1.5 Estrutura do trabalho

Nesta seção é apresentada a estrutura do trabalho, composta pelos capítulos que são descritos a seguir:

Capítulo 1: Introdução. Fornece uma visão geral do trabalho, apresentando o objetivo que se pretende alcançar, a justificativa para sua execução, a metodologia empregada, a revisão bibliográfica para orientar o leitor interessado em se aprofundar no assunto e o sumário estruturado.

Capítulo 2: Apresenta o problema de identificação por subespaços para malha aberta e solução via o método MOESP. Este método, desenvolvido por Michael Verhaegen, primeiro calcula a matriz de observabilidade a partir de uma matriz em blocos de Hankel, construída a partir dos dados de entrada e saída, depois se calculam as matrizes do sistema. Este método faz uso de projeções ortogonais.

Apresenta-se a identificação para os casos: puramente determinístico (MOESP), estocástica (PIMOESP) e determinística – estocástica (POMOESP). Cada método apresenta a solução do problema através de teoremas e como conseqüência se obtêm os algoritmos, os quais foram implementados no Matlab 6.5, para os diferentes casos.

Capítulo 3: Trata o problema de identificação por subespaços para sistemas em malha aberta através do método N4SID. Este método foi desenvolvido por Van Overschee e De Moor. Este capítulo começa com a definição de projeção oblíqua, a qual é necessária para fazer a identificação por este método. Neste caso, calcula-se a seqüência de estados a partir de uma matriz em blocos de Hankel, construída a partir dos dados de entrada e saída, depois as matrizes do sistema são computadas através do método de mínimos quadrados.

Neste capítulo é apresentada a identificação puramente determinística e determinística
estocástica. A identificação em cada caso é apresentada através de algoritmos, os quais forem implementados no Matlab 6.5.

Capítulo 4: Com base nos métodos estudados MOESP no capítulo 2 e N4SID no capítulo 3, nesta secção, se apresenta o método alternativo MON4SID; que combina os métodos MOESP e N4SID. Os algoritmos propostos determinam as matrizes $A \in C$ de forma similar ao método MOESP, logo, se calcula a nova matriz de observalibilidade estendida. As matrizes do sistema são estimadas em função dos estados do sistema, através das projeções oblíquas.

Se apresentam três algoritmos para o caso determinístico e dois algoritmos para o caso determinístico – estocástico.

Capítulo 5: Neste capítulo se apresentam as simulações para os casos determinístico e determinístico – estocástico. Foram consideradas 3 modelos: um modelo em espaços de estado com matrizes aleatórias, sistema MIMO, um modelo em espaços de estados, sistema SISO, usado por [VERHAEGEN, 1993; VAN OVERSCHEE, DE MOOR, 1996] e um modelo de um processo *benchmark* da Shell, sistema MIMO, [COTT, 1995; ZHU, 1997]. Foram usados 9 algoritmos para avaliar o desempenho dos modelos propostos.

Capítulo 6: Apresenta o problema de identificação em malha fechada, aplicando o método N4SIDC. Este surge como uma extensão do método N4SID aplicado para malha aberta. A base desta identificação é feita por dois teoremas principais, o primeiro mostra a projeção ortogonal das saídas futuras sobre uma matriz auxiliar N (que contém os dados de entradas e saídas passadas). A importância deste teorema é mostrar que a projeção é uma combinação linear dos estados estimados e da matriz auxiliar. O segundo teorema mostra que a seqüência de estados pode ser encontrada diretamente a partir de uma projeção oblíqua das saídas futuras em certa matriz auxiliar sobre os dados de entradas e saídas passadas, sendo que esta projeção é obtida com base no primeiro teorema. Como foi mencionado anteriormente, para este caso é necessário conhecer os parâmetros de Markov do controlador.

Capítulo 7: Mostra uma extensão da família MOESP na identificação em malha fechada. Este método é apresentado por Verhaegen, o qual estima um modelo global a partir do sinal de referência considerado como entrada e como saída considera os sinais de entrada do controlador, da planta e saída da planta. Deste sistema global são computados a planta e o controlador, através de uma redução de ordem.

Capítulo 8: Neste capítulo é apresentado um método alternativo MON4SIDC para a identificação da planta de um sistema em malha fechada e suas simulações são feitas através de um modelo simulado. Este método não precisa ter nenhum conhecimento prévio sobre o controlador. A idéia é a mesma apresentada no capítulo 7, mas a forma de obter as matrizes da planta é diferente. A identificação em malha fechada via o método MON4SIDC é dada para 3 casos: sistema puramente determinístico, sistema com pouco ruído e com muito ruído e seus resultados foram comparados com outros métodos de identificação para malha fechada: PEM, N4SIDC, MOESPC, ARXS.

Capítulo 9: Apresenta as conclusões deste trabalho, as contribuições e trabalhos futuros.

Apêndice A: Apresenta-se a base matemática para poder entender a identificação por

subespaços, a qual reside na teoria matemática de Espaços Vetoriais. Nesse sentido, são estudados conceitos estritamente algébricos, como bases, independência linear e dimensão, assim como conceitos topológicos, como norma e produto interno. Usando as ferramentas desta teoria são explicados e exemplificados conceitos como projeção ortogonal e oblíqua.

Apêndice B: Apresentam-se as demonstrações dos teoremas do capítulo 2.

Apêndice C: Apresentam-se as demonstrações dos teoremas do capítulo 5.

Apêndice D: Apresentam-se as demonstrações dos teoremas do capítulo 6.

Apêndice E: Apresentam-se as demonstrações dos teoremas do capítulo 7.

Apêndice F: Apresentam-se os diferentes algoritmos usados na tese.

CAPÍTULO 2

MÉTODOS DE IDENTIFICAÇÃO POR SUBESPAÇOS

Existem métodos de identificação como "Métodos de Predição do Erro" (Prediction Error Methods - PEM) e os Métodos das Variáveis Instrumentais (Instrumental Variable Methods - IVM), que são muitos populares. Recentemente, os métodos de identificação por subespaços têm emergido como uma alternativa para os métodos tradicionais.

Nos métodos de identificação de sistemas dinâmicos por subespaços são tratados modelos de sistemas lineares invariantes no tempo em espaço de estados operando em tempo discreto. Pelas restrições citadas, pode parecer uma classe altamente restrita de modelos (especialmente por serem lineares), no entanto é bastante surpreendente como muitos processos industriais podem ser descritos com precisão por este tipo de modelo. Por outro lado, existe um grande número de ferramentas disponíveis de projeto de controladores para tais sistemas e modelos. Para empregar tais métodos é necessário o uso de ferramentas de teoria de sistemas, geometria e álgebra linear.

Estes métodos têm sua origem na teoria de espaço de estados desenvolvida nos anos 60/70 [HO; KALMAN, 1966; ZEIGER; McEWEN, 1974; KUNG, 1978].

O problema da identificação por subespaços é apresentado através da teoria da realização.

2.1 Teoria da realização

Considere um sistema linear discreto invariante no tempo (SLIT) dado por:

$$x(k+1) = Ax(k) + Bu(k)$$
 (2.1a)

$$y(k) = Cx(k) + Du(k)$$
(2.1b)

Os vetores $u(k) \in \Re^m$ e $y(k) \in \Re^l$ são, respectivamente os valores medidos das entradas e saídas no instante k dos processos com m entradas e l saídas. O vetor $x(k) \in \Re^n$ é o vetor de estados do processo em tempo discreto no instante k. $A \in \Re^{nxn}$ é a matriz dinâmica do sistema, $B \in \Re^{nxm}$ é a matriz de entrada, $C \in \Re^{lxn}$ é a matriz de saída, $D \in \Re^{lxm}$ é a matriz de transmissão direita.

Tomando-se a transformada Z da resposta ao impulso em (2.1), se tem:

$$G(z) = D + C(zI - A)^{-1}B$$
(2.2)

a qual é chamada *matriz de transferência* do SLIT.

Se no sistema (2.1) é dada uma condição inicial x(0) e as entradas u(k), k = 0,1,..., são conhecidas, então o vetor de estados x(k), k = 1,2,..., é obtido de forma recursiva e assim as saídas y(k), k = 0,1,... são determinadas [SANTOS MIRANDA, 2004] como:

$$y(k) = CA^{k}x(0) + Du(k) + \sum_{i=0}^{k-1} CA^{k-1-i}Bu(k)$$
(2.3)

Na equação (2.3) podem acontecer os seguintes casos:

i) Se u(k) = 0, k = 0,1,..., a equação (2.3) se reduz a

$$y(k) = CA^{k}x(0), \quad k = 0,1,...$$
 (2.4)

então a equação (2.4) é chamada de resposta à entrada zero.

ii) Também, se a condição inicial x(0) = 0, se tem

$$y(k) = Du(k) + \sum_{i=0}^{k-1} CA^{k-1-i} Bu(k), \quad k = 0, 1, \dots$$
(2.5)

a qual é a resposta para uma entrada externa u(k) e é chamada de *resposta para o* estado zero.

Portanto, a resposta para um SLIT pode ser sempre expressa como a soma da resposta à entrada zero com a resposta para o estado zero, isto é, equação (2.3).

Os parâmetros de Markov são muito usados nesta tese, sendo computados a partir da resposta ao impulso do sistema (2.1). Considerando a resposta ao impulso da equação

(2.3), se obtém [vide apêndice B.1]:

$$g(k) = \begin{cases} D & k = 0\\ CA^{k-1}B, & k = 1, 2, \dots \end{cases}$$
(2.6)

onde $g(k) \in \Re^{lxm}$ e (g(0), g(1),...) é chamada de resposta ao impulso ou parâmetros de Markov do sistema (2.1).

Observar que dadas as matrizes (A, B, C, D), a matriz de transferência e a matriz de transferência ao impulso ou parâmetros de Markov podem ser unicamente determinadas pelas equações (2.2) e (2.6), respectivamente.

O problema inverso é: dada G(z) (matriz de transferência) ou g(k) (parâmetros de Markov), encontrar as matrizes (*A*, *B*, *C*, *D*). Este problema é chamado de teoria da realização. Existem duas formas de apresentar este problema.

Problema 1 Suponha que a seqüência da resposta ao impulso $\{g(k), k = 0,1,..\}$ ou a matriz de transferência G(z) do SLIT seja dada. O problema da realização é encontrar a dimensão n e o sistema de matrizes (A, B, C, D).

Problema 2 Suponha que os dados de entrada e saída $\{u(k), y(k), k = 0, 1, ..., N - 1\}$ sejam dados. O problema é identificar a dimensão *n* e o sistema de matrizes (*A*, *B*, *C*, *D*). Este é exatamente o problema de identificação por subespaços para SLIT.

A teoria da realização de sistemas para o caso determinístico pretende, em primeira instância, resolver o seguinte problema: dada a função de transferência, como encontrar um ou mais modelos no espaço de estados, que representem tal sistema. Existem técnicas que permitem solucionar tal problema, sobretudo para o caso SISO. Como a generalização para o sistema MIMO não é trivial, uma alternativa é trabalhar com as matrizes de resposta ao impulso. Existem muitas técnicas para solucionar este problema, porém a teoria da realização, utilizando a decomposição de valores singulares (SVD), tornou-se predominante por ser computacionalmente robusta.
A seguir, apresenta-se o método de Ho-Kalman, pois foi base para o surgimento dos métodos de subespaços.

2.2 Método de Ho-Kalman

O método da realização de Ho-Kalman [H0; KALMAN, 1966] provê uma completa solução para o problema 1.

Observar que se a resposta do sistema (2.1) é dada por (g(0), g(1),..), da equação (2.6) obtém-se que D = g(0), desta forma só resta identificar as matrizes (A, B, C).

Suponha que o sistema (2.1) foi excitado por uma entrada u tal que:

$$u = \{\dots, u(-3), u(-2), u(-1), 0, 0, 0, \dots\}$$
(2.7)

Isto significa que se assume valor diferente de zero para $k \le -1$ e zero para k = 0,1,...

Aplicando-se a entrada (2.7) ao sistema discreto (2.1), observa-se a saída para k = 0,1,... Isto é mostrado na figura 2.1.



Figura 2.1 Resposta para um sistema SLIT com entrada zero

Não é difícil provar que para a seqüência de entradas (2.7), a saída é [vide apêndice B.2]:

$$y(k) = g(k+1)u(-1) + g(k+2)u(-2) + g(k+3)u(-3) + \dots, \quad k = 0, 1, 2, \dots$$
(2.8)
Isto é,

$$y(k) = \sum_{i=-\infty}^{-1} g(k-i)u(i)$$
, $k = 0,1,2,...$ (2.9)

A equação (2.9) é a resposta à entrada zero com estado inicial x(0), o qual é determinado pelas entradas passadas. Isto é mostrado na figura 2.1, em que a linha pontilhada representa a saída y(k) para k = -1, -2, -3, ...

A equação (2.8) pode ser expressa em forma matricial:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \end{bmatrix} = \begin{bmatrix} g(1) & g(2) & g(3) & \cdots \\ g(2) & g(3) & g(4) & \cdots \\ g(3) & g(4) & g(5) & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} u(-1) \\ u(-2) \\ u(-3) \\ \vdots \end{bmatrix}$$
(2.10)

então se define a matriz de Hankel de dimensão infinita como:

$$H_{\infty} = \begin{bmatrix} g(1) & g(2) & g(3) & \cdots \\ g(2) & g(3) & g(4) & \cdots \\ g(3) & g(4) & g(5) & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}$$
(2.11)

A matriz H_{∞} é formada pelos parâmetros de Markov.

A equação (2.10) pode ser expressa como:

$$y_{+} = H_{\infty}u_{-} \tag{2.12}$$

onde y_+ e u_- são vetores de dimensão infinita definidos por:

$$y_{+} = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \end{bmatrix}, \qquad u_{-} = \begin{bmatrix} u(-1) \\ u(-2) \\ u(-3) \\ \vdots \end{bmatrix}.$$

Sejam $\Gamma_{\infty} = \begin{bmatrix} C \\ CA \\ CA^{2} \\ \vdots \end{bmatrix}$ e $\Delta_{\infty} = \begin{bmatrix} B & AB & A^{2}B \cdots \end{bmatrix}$ as matrizes infinitas de observabilidade

e controlabilidade, respectivamente; por simples inspeção observa-se que:

$$H_{\infty} = \Gamma_{\infty} \Delta_{\infty} \tag{2.13}$$

Para um valor finito de índice, se for possível fatorar a matriz $H = H_{\infty}$ como em (2.13), então as matrizes A, B, C podem ser computadas dos primeiros blocos das matrizes de observabilidade e controlabilidade, respectivamente. Pode-se encontrar uma fatoração de *H* por meio da decomposição em valores singulares (SVD), da seguinte forma:

$$H = USV^{T} = \begin{bmatrix} U_{1} & U_{2} \end{bmatrix} \begin{bmatrix} S_{11} & 0 \\ 0 & S_{22} \end{bmatrix} \begin{bmatrix} V_{1}^{T} \\ V_{2}^{T} \end{bmatrix}$$

onde $S_{11} \in \Re^{nxn}$ e $S_{22} = 0$, e definem-se as matrizes Γ e Δ como:

$$\Gamma = (U_1 S_{11}^{1/2}) T^{-1}$$
$$\Delta = T(S_{11}^{1/2} V_1^T)$$

Apresenta-se, a seguir, um teorema básico das propriedades da matriz em blocos de Hankel, o qual tem um importante papel nesta tese.

Teorema 2.1 (propriedades da matriz de Hankel)

Suponha que (A, B, C) seja de fase mínima (isto é, sistema estável e pólos dentro do círculo unitário), então os seguintes enunciados são equivalentes:

- A matriz em blocos de Hankel da equação (2.11) tem posto finito se e somente se a resposta ao impulso tem uma fatorização da forma (2.6).
- ii) A matriz em blocos de Hankel tem posto n, isto é, posto(H) = n. Ademais, a matriz H tem fatorização da forma:

 $H = \Gamma \Delta = \Gamma T T^{-1} \Delta ,$

iii)

onde T é uma matriz não singular.

Seja o vetor de estados para k = 0, dado por $x(0) = \Gamma u_{-}$. Então (2.12) é dado por:

$$y_{+} = \Gamma x(0) \tag{2.14}$$

A prova deste teorema pode ser encontrada em [VAN DER VEEN; DEPRETTERE; SWINDLEHURST, 1993].

Por limitações na computação das matrizes, usa-se um número finito de termos da resposta ao impulso na matriz de Hankel. Deste modo, se pode usar um truncamento da matriz H da forma:

$$H_{k,j} = \begin{bmatrix} g(1) & g(2) & \cdots & g(j) \\ g(2) & g(3) & \cdots & g(j+1) \\ \vdots & \vdots & \ddots & \vdots \\ g(k) & g(k+1) & \cdots & g(k+j-1) \end{bmatrix} \in R^{klxjm}$$

Da mesma forma para as matrizes $\Gamma_k \in \Delta_j$

$$\Gamma_{k} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} \quad e \quad \Delta_{j} = \begin{bmatrix} B & AB & \cdots & A^{j-1}B \end{bmatrix}$$
(2.15)

onde k e j têm que ser maiores que n, a ordem do sistema. Usualmente se usa $n < k \le j$.

A seguir apresenta-se o algoritmo que soluciona o problema 1. Para maiores detalhes pode-se consultar [H0; KALMAN, 1966; ZEIGER; McEWEN, 1974].

Algoritmo da realização determinística

1. Calcule a SVD da matriz H_{kj} , como

$$H_{kj} = \begin{bmatrix} U_s & U_n \end{bmatrix} \begin{bmatrix} \sum_{s} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s^T \\ V_n^T \end{bmatrix} = U_s \sum_{s} V_s^T.$$

2. Encontrar a matriz de controlabilidade estendida e a matriz de observabilidade:

$$\Gamma_k = U_s \sum_{s}^{1/2} T, \quad \Delta_j = T^{-1} \sum_{s}^{1/2} V_s^T$$

onde $T \in \Re^{nxn}$ é uma matriz não singular.

3. Encontrar A, B e C das matrizes Γ_k e Δ_j .

2.3 Identificação por subespaços

Nesta seção apresenta-se a solução do problema 2, enunciado na seção 2.1, baseada no método de identificação por subespaços.

Os métodos lineares de identificação por subespaços estão relacionados com sistemas e modelos da forma:

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$
(2.16a)

$$y(k) = Cx(k) + Du(k) + v(k)$$
 (2.16b)

com

$$E\left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq} \ge 0$$

 $v(k) \in \Re^{l}$ e $w(k) \in \Re^{n}$ são sinais de vetores não medidos, v(k) corresponde ao ruído na medição e w(k) ao ruído do processo. Assume-se que v(k) e w(k) tenham média zero, sejam seqüências vetoriais de ruído branco estacionário e não sejam correlacionados com as entradas u(k). As seqüências de ruído w(k) e v(k) têm matriz de covariância gaussiana $E(w_k w_k^T) = Q \in \Re^{nxn}$, $E(v_k v_k^T) = R \in \Re^{lxl}$, $E(w_k v_k^T) = S \in \Re^{nxl}$, E é o operador esperança matemática e δ é o delta de Kronecker.

A equação (2.16), que representa sistemas dinâmicos, é descrita pictoricamente na figura 2.2, que cobre processos de sistemas físicos, econômicos, biológicos, etc.



Figura 2.2 Sistema dinâmico com entradas u(k), saídas y(k) e distúrbios v(k) e w(k).

Os algoritmos de identificação por subespaços sempre consistem em dois passos. O primeiro passo toma a projeção de certos subespaços gerados a partir dos dados, para encontrar uma estimativa da matriz de observabilidade estendida e/ou uma estimativa dos estados do sistema desconhecido. O segundo passo é recuperar as matrizes do

sistema a partir da matriz de observabilidade estendida ou das estimativas dos estados. Uma das importantes idéias ao se desenvolver a identificação por subespaços foi a reintrodução do conceito *dos estados x(k) de um sistema dinâmico* dentro do sistema de identificação. Em contraste com os algoritmos de identificação "*clássicos*", muitos dos algoritmos por subespaços primeiro estimam / calculam os estados (implícita ou explicitamente), em seguida o modelo em espaço de estados é determinado. Esta é uma importante diferença entre os métodos clássicos, por exemplo PEM e os métodos por sub-espaços, como se ilustra na figura 2.3.



Figura 2.3 Identificação por subespaços versus identificação clássica

A figura 2.3 mostra que a identificação de sistemas tem como objetivo a construção de modelos a partir dos dados de entrada e saída. O lado esquerdo mostra uma aproximação do método por subespaços: os estados de Kalman podem ser estimados diretamente dos dados de entrada e saída, depois é fácil obter as matrizes do sistemas *A*, *B*, *C*, *D*. O lado direito é uma aproximação clássica: primeiro se obtêm as matrizes do sistema com as quais depois podem ser estimados os estados.

Os algoritmos de identificação por subespaços calculam modelos em espaço de estados, a partir de dados de entrada e saída. É uma prática comum distinguir três casos distintos

na identificação de sistemas por subespaços [VAN OVERSCHEE; DE MOOR, 1996]:

- a) O caso puramente determinístico (w(k) = v(k) = 0)
- b) O caso puramente estocástico (u(k) = 0)
- c) O caso combinado determinístico/estocástico.

Existem vários métodos de identificação por subespaços para malha aberta, por exemplo, N4SID, MOESP, CVA etc. O método N4SID para os três casos foi estudado na dissertação de mestrado [SANTOS MIRANDA, 2004], por isso, no capítulo 3 se apresentam os teoremas sem demonstrações.

No capítulo 4 propõe-se um algoritmo de identificação por subespaços para malha aberta MON4SID, para isto precisa-se entender como funciona o método MOESP para o caso determinístico e estocástico.

A notação matricial usadas nesta tese U_p , U_f , Y_p , Y_f ,... etc, é definida no apêndice B.3.

2.3.1 Método MOESP

Esta classe de algoritmo é conhecida como Multivariable Output-Error State sPace (MOESP). Existe o método MOESP básico, do qual um grande número de variações foi criado para diferentes tipos de problemas, tais como o caso sem ruído, ruído branco na saída, o modelo erro de saída e o modelo inovação. O primeiro modelo a ser ilustrado é o MOESP básico, usado para identificação de processos sem ruído. Cabe realçar que se dará uma idéia de como funciona este método, porém maiores detalhes podem ser encontrados em [VERHAEGEN, 1990; VERHAEGEN; DEPRETTERE, 1991; VERHAEGEN; PATRICK, 1992a,b; VERHAEGEN, 1993a,b,c; VERHAEGEN, 1994a].

2.3.2 Algoritmo Básico MOESP

Este algoritmo soluciona o problema de tipo determinístico conforme figura 2.4.



Figura 2.4 Modelo determinístico

Para este caso aborda-se o método de identificação por subespaços para sistemas puramente determinísticos, com ruído de processo w(k) e de medição v(k) nulos, isto é, $w(k) \equiv v(k) \equiv 0$. A implementação deste algoritmo é feita mediante projeções ortogonais de subespaços e a decomposição LQ é feita tomando-se a transposta da decomposição QR, isto é:

% Programa decomposição LQ **function** [R11,R21, R22] = lq(U,Y) km = size(U,1); kp = size(Y,1); [Q,L] = qr([U,Y]',0); Q = Q'; L = L'; R11 = L(1:km,1:km); R21 = L(km+1,km+kp,1:km); R22 = L(km+1:km+kp,km+1:km+kp);

A representação matricial da projeção ortogonal pode ser facilmente computada através da decomposição LQ. Uma justificativa de usar esta decomposição pode ser vista no apêndice B.4.

Formulação do Problema: Dado um conjunto de entradas $u(k) \in \Re^m$ e saídas $v(k) \in \Re^l$ para um sistema desconhecido de ordem *n*:

$$x(k+1) = Ax(k) + Bu(k)$$
(2.17a)

$$y(k) = Cx(k) + Du(k)$$
(2.17b)

Determine: A ordem *n* do sistema e as matrizes (A, B, C, D).

Idéia do Algoritmo MOESP básico

Apresenta-se a seguir uma idéia do funcionamento do método MOESP básico, isto é, conhecendo somente os dados de entrada e saída do sistema (2.17), de que forma são

computadas as matrizes do sistema.

Para o sistema (2.17) se tem a seguinte equação recursiva [vide apêndice B.3, equação (B3.10)]:

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N}$$
(2.18)

Considerando k = 1 na equação (2.18) se obtém:

$$Y_{1,i,N} = \Gamma_i X_{1,N} + H_i U_{1,i,N}$$
(2.19)

onde $[Y_{1,i,N}]_{lixN}$, $[\Gamma_i]_{lixn}$, $[X_{1,N}]_{nxN}$, $[H_i]_{lixmi}$ e $[U_{1,i,N}]_{mixN}$.

A matriz $Y_{1,i,N}$ também pode ser expressa em função das projeções ortogonais como:

$$Y_{1,i,N} = Y_{1,i,N} / U_{1,i,N} + Y_{1,i,N} / U_{1,i,N}^{\perp}$$
(2.20)

onde o lado direito de $Y_{1,i,N}$ na equação (2.20) é a soma da projeção ortogonal do espaço linha de $Y_{1,i,N}$ sobre o espaço de linhas da matriz $U_{1,i,N}$ e a projeção ortogonal do espaço linha de $Y_{1,i,N}$ sobre o complemento ortogonal do espaço de linhas da matriz $U_{1,i,N}$.

A decomposição LQ da matriz $\begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix}$ é denotada por [vide apêndice B.4, equação (B4.11)]: $\begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$ (2.21)

Da equação (2.21) se obtém:

$$U_{1,i,j} = R_{11}Q_1 \tag{2.21a}$$

$$Y_{1,i,j} = R_{21}Q_1 + R_{22}Q_2 \tag{2.21b}$$

onde $[R_{11}]_{mixmi}$, $[R_{21}]_{lixmi}$ e $[R_{22}]_{lixli}$ são as matrizes triangular inferior e $[Q_1]_{mixN}$, $[Q_2]_{lixN}$ são as matrizes ortogonais.

Se o sinal de entrada é persistentemente excitante, então $posto(U_{1,i,N}) = mi$ [vide apêndice B.7, equação (B 7.6)], logo a matriz R_{11} é não singular, assim $Q_1 = R_{11}^{-1}U_{1,i,N}$.

Então (2.21b) pode ser escrito como

$$Y_{1,i,N} = R_{21}R_{11}^{-1}U_{1,i,N} + R_{22}Q_2$$
(2.22)

Observa-se, da equação (2.21b), que se $Q_1 e Q_2$ são ortogonais, então o primeiro termo do lado direito da equação (2.22) é o *span* para os vetores linha da matriz $U_{1,i,N}$ e o segundo termo é ortogonal a este. Isto quer dizer que a matriz $Y_{1,i,N}$ pode ser expressa em termos das projeções ortogonais da matriz $U_{1,i,N}$ e de seu complemento ortogonal $U_{1,i,N}^{\perp}$. Isto é

$$Y_{1,i,N} / U_{1,i,N} = R_{21}Q_1 \tag{2.23}$$

$$Y_{1,i,N} / U_{1,i,N}^{\perp} = R_{22}Q_2$$
(2.24)

Ver apêndice B.4, equações (B 4.5) e (B 4.6).

Logo, substituindo-se (2.23) e (2.24) na equação (2.21b), se obtém:

$$Y_{1,i,N} = Y_{1,i,N} / U_{1,i,N} + Y_{1,i,N} / U_{1,i,N}^{\perp}.$$
(2.25)

Observa-se que $Y_{1,i,N}$ pode ser obtido de duas formas diferentes, isto é, através das equações (2.19) e (2.25), porém cada uma tem significado diferente. Assim , igualando-se as equações (2.19) e (2.25), resulta:

$$\Gamma_i X_{1,N} + H_i R_{11} Q_1 = R_{21} Q_1 + R_{22} Q_2 \tag{2.26}$$

Observa-se na equação (2.26) que o lado direito é uma soma ortogonal, mas o lado esquerdo é soma direta, isto quer dizer que estes termos não são necessariamente ortogonais. Isto implica que $\Gamma_i X_{1,N} \neq R_{22}Q_2$ e $H_i R_{11}Q_1 \neq R_{21}Q_1$.

O objetivo do algoritmo MOESP é computar a matriz de observabilidade estendida Γ_i . Então multiplica-se a equação (2.26) por Q_2^T de ambos os lados, obtendo-se:

$$\Gamma_i X_{1,N} Q_2^T = R_{22}$$

onde se supôs que posto $(X_{1,N}Q_2^T) = n$ e posto $(\Gamma_i) = n$, o qual é igual ao posto $(R_{22}) = n$. Assim, pode-se obter a imagem da matriz de obervabilidade estendida Γ_i . A dimensão n é obtida através de SVD da matriz R_{22} :

$$R_{22} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T$$
(2.27)

onde $n = \dim(\Sigma_1)$, $[U_1]_{lixn} \in [U_2]_{lix(li-n)}$.

Logo:

$$\Gamma_i X_{1,N} Q_2^T = R_{22} = U_1 \Sigma_1 V_1^T$$

Define-se:

$$\Gamma_i = U_1 \Sigma_1^{1/2}$$

Agora se computam as matrizes do sistema, sendo que a matriz C é facilmente obtida de:

$$C = \Gamma_i(1:l,1:n)$$

e a matriz A é obtida de:

$$\Gamma_i(1:l(i-1),1:n)A = \Gamma_i(l+1:il,1:n)$$

Para obter as matrizes *B* e *D*, vide apêndice B.5.

Os teoremas são importantes, pois através deles é possível programar os algoritmos. A seguir apresenta-se o teorema MOESP para o caso determinístico.

2.3.2.1 Teorema (MOESP ORDINÁRIO, caso determinístico) [VERHAEGEN, 1992a]

Hipóteses

1. A entrada u(k) satisfaz a condição:

$$\operatorname{posto}\left[\frac{U_{1,i,N}}{X_{1,N}}\right] = mi + n \tag{2.28}$$

2. i > n

3. A fatoração RQ da matriz de Hankel dos dados de entrada e saída é dada por:

$$\begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
(2.29)

4. A matriz SVD da matriz $R_{22} \in \Re^{lixli}$ é :

$$R_{22} = {}_{li} \left[\begin{matrix} n \\ U_n \end{matrix} \middle| \begin{matrix} l^{i-n} \\ D_n \end{matrix} \right] \left[\begin{matrix} \frac{S_n}{0} \end{matrix} \middle| \begin{matrix} l^{i-n} \\ 0 \\ S_2 \end{matrix} \right] \left[\begin{matrix} V^T \\ (V_n^{\perp})^T \end{matrix} \right]$$

$$posto \left(U_n^{\perp} (l(i-1)+1:li,:) \right) = l$$
(2.30)

5.

Tese

6. O sistema de matrizes A, B, C e D pode ser encontrado das seguintes equações.

$$U_n^{(1)}A_T = U_n^{(2)} \tag{2.31}$$

$$C_T = U_n(1:l,:)$$
 (2.32)

$$\begin{bmatrix} \Xi(:,1:m) \\ \Xi(:,m+1:2m) \\ \vdots \\ \Xi(:,m(i-1)+1:mi) \end{bmatrix} = \begin{bmatrix} U_n^{\perp}(1:l,:)^T & \cdots & U_n^{\perp}(l(i-1)+1:li,:)^T \\ U_n^{\perp}(l+1:2l,:)^T & \cdots & 0 \\ \vdots & \vdots & 0 & 0 \\ U_n^{\perp}(l(i-1)+1:li,:)^T & 0 & 0 \end{bmatrix} \begin{pmatrix} I_l \\ 0 \\ U_n^{(1)} \end{pmatrix} \begin{pmatrix} D \\ B_T \end{pmatrix}$$
(2.33)

onde $\Xi = (U_1^{\perp})^T R_{21} R_{11}^{-1}$.

A prova deste teorema pode ser encontrada no apêndice B.6.

Comentários deste teorema

Para satisfazer a hipótese 1, vários estudos foram realizados em relação ao sinal de entrada u(k). Um estudo do tipo estatístico pode ser visto em [GOPINATH, 1969]. Um estudo do tipo algébrico, o qual é aplicado nesta tese, pode ser encontrado em [DE MOOR, 1988]. Nesta tese se considera u(k) um sinal de entrada persistentemente excitante de ordem 2i [LJUNG, 1999].

A hipótese 5 do teorema acima é necessária para encontrar o sistema de matrizes D e B.

Algoritmo MOESP (caso determinístico)

- 1. Construir as matrizes de Hankel $U_{1,i,N} \in Y_{1,i,N}$.
- Calcular a fatoração RQ (ou LQ) dos dados de entrada e saída, equação (2.32):

$$\begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

- 3. Calcular a SVD da matriz R_{22} dada na equação (2.33) do teorema acima.
- 4. Determinar a ordem do sistema por inspeção dos valores singulares em S_n e particionar o **SVD** para obter U_n .
- 5. Determinar as matrizes $A \in C$ das equações (2.31) e (2.32)

$$U_n^{(1)} A_T = U_n^{(2)}$$

 $C_T = U_n(1:l,:)$

6. Determinar as matrizes $B \in D$ da equação (2.33) do teorema acima.

A implementação deste algoritmo no programa Matlab 6.5 poder ser encontrada no apêndice F.

Como foi mencionado anteriormente, o caso determinístico é puramente acadêmico. A seguir apresenta-se um sistema com ruído na saída, que é mais realista. O algoritmo PIMOESP (Past-Input do algoritmo MOESP) é uma solução para este problema.

A notação matricial usada no algoritmo PIMOESP é dada no apêndice B.7.

2.3.3 Algoritmo PIMOESP

Este algoritmo soluciona o problema de ruído na saída, conforme figura 2.5.



Figura 2.5 Modelo com ruído aditivo na saída

Nesta seção aborda-se o método de identificação por subespaços para sistemas com perturbação externa na saída, que pode ser gerada por erro do instrumento de medição e é representada por v(k), que é ruído branco com média zero [vide apêndice B.7, definição B7.2].

A figura 2.5 pode ser interpretada como a saída de um sistema determinístico perturbado por um ruído v(k), isto é:

$$z(k) = y(k) + v(k)$$

onde y(k) é o sinal de saída do sistema determinístico. Pela equação (2.17b), obtém-se:

$$z(k) = Cx(k) + Du(k) + v(k)$$
(2.29)

Sem perda de generalidade e fazendo abuso da notação, pode-se representar a saída do algoritmo PIMOESP y(k) = z(k), então da equação (2.29), resulta:

$$y(k) = Cx(k) + Du(k) + v(k)$$
(2.30)

O modelo representado na figura 2.5 é aplicado na identificação de sistemas, mediante métodos paramétricos do tipo ARMAX, ARX e OE, modelos definidos em Ljung (1999). Para solucionar este tipo de problema é comum aplicar uma variável instrumental, ver [SÖDERSTROM; STOICA, 1989; LJUNG, 1999].

O método PIMOESP é uma extensão do modelo MOESP ordinário, com aplicação de variáveis instrumentais (VI). Vários tipos de VI podem ser sugeridos, eles podem ser significativamente diferentes na computação numérica, no problema de estimação. Em [VERHAEGEN 1992a] dois tipos de VI são propostos:

- 1. Baseado nas entradas passadas
- 2. Baseado na reconstrução dos estados estimados, denotada por RS

O primeiro leva a estimativas assintóticas sem polarização (*Bias*) quando a saída é perturbada por um processo estocástico independente das entradas, No entanto, uma análise de sensibilidade do cálculo para dados finitos foi melhor para o caso 2. Para maiores detalhes ver [VERHAEGEN 1992a].

Para encontrar estimativas sem polarização existem vários caminhos, geralmente aplicam-se projeções ou se costuma colocar filtros nos dados de entrada e saída. Em qualquer caso é importante que a informação do modelo seja preservada. Para isto, geralmente se faz restrições com relação ao posto dos estados.

Estas duas últimas considerações são impostas no modelo PIMOESP. Primeiro, a operação da variável instrumental é apresentada como a projeção de certas matrizes dentro de um subespaço construído a partir de uma variável instrumental. Segundo, certas matrizes afetadas pela variável instrumental têm que satisfazer as restrições do posto dos estados.

O objetivo do algoritmo PIMOESP é estimar o espaço coluna da matriz observabilidade estendida Γ_i , a partir uma matriz de Hankel construída com os dados de entrada e dados da saída perturbada.

Formulação do problema

Dado: Um conjunto de entradas $u(k) \in \Re^m$ e saídas $y(k) \in \Re^l$ para um sistema desconhecido de ordem *n*, dado por:

$$x(k+1) = Ax(k) + Bu(k)$$
 (2.31a)

$$y(k) = Cx(k) + Du(k) + v(k)$$
 (2.31b)

onde v(k) é um sinal de ruído branco com média zero, não-correlacionado com os dados de entrada.

Determine: A ordem *n* do sistema e as matrizes (A, B, C, D).

Idéia do Algoritmo PIMOESP

Como foi dito anteriormente, o objetivo do método PIMOESP é computar uma estimativa da matriz obervabilidade estendida e a partir dela estimar as matrizes do sistema (2.31). Mas primeiro são apresentadas algumas restrições que são aplicadas neste método.

O algoritmo PIMOESP faz uso das duas seguintes restrições:

- a) v(k) é um sinal do tipo ruído branco com média zero
- b) A variável instrumental considerada é a matriz dos dados passados U_{1iN} .

Uma pergunta natural é porque considerar v(k) do tipo ruído branco com média zero? Para responder esta pergunta, precisa-se do seguinte corolário.

Corolário 1 Seja $[M]_{qxq}$ uma matriz simétrica e $Q = [Q_1 Q_2]_{r q-r} (\operatorname{com} r < q)$ uma matriz ortogonal, tal que posto (Q_1) seja um subespaço invariante de M, isto é, $MQ_1 \subset Q_1$, então Q_1 é um sub-espaço invariante de $M + \sigma^2 I$, para qualquer número real σ .

A prova deste corolário pode ser encontrada em [VERHAEGEN, 1992b].

Da equação (B 8.28), [ver apêndice B.8], tem-se que:

$$\lim_{N \to \infty} \frac{1}{N} R_{22} (R_{22})^T = \Gamma_i P_{x2} \Gamma_i^T + R \nu$$
(2.32)

Logo, se $M = \Gamma_i P_{x2} \Gamma_i^T$ e substituindo-se M por seu SVD, isto é, $SVD(M) = U_{\Gamma} S_{\Gamma} U_{\Gamma}^T$, onde $S_{\Gamma} \in \Re^{nxn}$, então a equação (2.32) é dada por:

$$\lim_{N \to \infty} \frac{1}{N} R_{22} (R_{22})^T = U_{\Gamma} S_{\Gamma} U_{\Gamma}^T + R \nu$$
(2.33)

Foi visto na seção 2.3.2 (algoritmo MOESP ordinário), que se não existir o termo Rv na equação (2.33), o espaço coluna de Γ_i é computado do espaço coluna da matriz U_{Γ} , isto é, existe um subespaço coluna invariante Q_1 da matriz M. Logo, pelo corolário 1, Q_1 é um sub-espaço invariante da matriz $[M + \sigma^2 I]$, mas para garantir esta afirmação temse que impor que a matriz covariância Rv seja proporcional à matriz identidade. A restrição da matriz Rv corresponde a v(k) ser ruído branco com média zero, pois

desta forma se satisfaz a seguinte propriedade estatística: $Rv = E(VV^T) = \sigma^2 I$ [vide apêndice B.7, definição B7.2].

Outra pergunta é porque considerar os dados de entradas passadas como variável instrumental? [vide apêndice B9].

Da equação (B 9.10)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{22} Q_2 W_N^{T} = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i P_{x2} Q_x W_N^{T} + \lim_{N \to \infty} \frac{1}{\sqrt{N}} V_{1,i,N} W_N^{T}$$
(2.34)

Na equação (2.34) se o termo esquerdo é conhecido e o último termo do lado direito é zero, então pode-se encontrar o espaço coluna da matriz Γ_i . Logo, a variável instrumental escolhida tem que satisfazer:

i) $\lim_{N \to \infty} \frac{1}{\sqrt{N}} V_{1,j,N} W_N^T = 0$ ii) posto $\left(\lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i P_{x2} Q_x W_N^T \right) = n$

assim a variável instrumental escolhida é $W_N = U_{1,i,N}$.

A seguir apresenta-se a idéia do método PIMOESP.

A versão **P**ast-Input do algoritmo MOESP, conhecida como PIMOESP, soluciona o problema para o sistema (2.31), usando o método das variáveis instrumentais. Formando a matriz de Hankel de dimensões apropriadas como na secção 2.3.2, não é difícil obter para o sistema (2.31) a seguinte representação [vide apêndice B.7]:

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N} + V_{k,i,N}$$
(2.35)

onde $V_{k,i,N}$ é a matriz de Hankel dos valores do ruído.

Considerando k = 1 na equação (2.35), se obtém:

$$Y_{1,i,N} = \Gamma_i X_{1,N} + H_i U_{1,i,N} + V_{1,i,N}$$
(2.36)

A matriz $Y_{1,i,N}$ é chamada de saídas passadas e o termo $U_{1,i,N}$ de entradas passadas.

Considerando k = i + 1 na equação (2.35), resulta: $Y_{i+1,i,N} = \Gamma_i X_{i+1,N} + H_i U_{i+1,i,N} + V_{i+1,i,N}$ (2.37)

Logo, a matriz $Y_{i+1,i,N}$ é chamada de saídas futuras e o termo $U_{i+1,i,N}$ de entradas futuras.

O primeiro passo é eliminar o termo $H_i U_{i+1,i,N}$ da equação (2.37). Isto é possível multiplicando-se (2.37) pelo operador projeção $\Pi_{U_{i+1,i,N}^{\perp}}$. Agora, para remover a matriz de ruído da equação (2.37), emprega-se uma variável instrumental W, com as restrições i) e ii) citadas acima, isto é, ser correlacionada com os estados mas não-correlacionada com a matriz de ruído $V_{i+1,i,N}$.

Das hipóteses do problema 2.3, o ruído é não correlacionado com os dados de entrada, então o método PIMOESP escolhe como variável instrumental a matriz de entrada dos dados passados, $W = U_{1,i,N}$.

Multiplicando-se ambos lados da equação (2.37) por $\prod_{U_{i+1,i,N}^{\perp}} W$ e com $N \to \infty$, se obtém:

$$\lim_{N \to \infty} \frac{1}{N} Y_{i+1,i,N} \Pi_{U_{i+1,i,N}^{\perp}} W = \lim_{N \to \infty} \frac{1}{N} \Gamma_i X_{i,N} \Pi_{U_{i+1,i,N}^{\perp}} W$$
(2.38)

onde $W = U_{1,i,N}$ e $\Pi_{U_{i+1,i,N}^{\perp}} = I - U_{i+1,i,N}^{T} (U_{i+1,i,N} U_{i+1,i,N}^{T})^{*} U_{i+1,i,N}$, o símbolo ()* indica a pseudo inversa.

A equação (2.38) pode ser eficientemente encontrada através da decomposição LQ, isto é :

$$\begin{bmatrix} U_{i+1,i,N} \\ W \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$
(2.39)

Agora é preciso estimar o espaço coluna da matriz Γ_i .

Como o número de dados tende a infinito, se tem as seguintes relações:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_2)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_{i+1,N}(Q_2)^T$$
(2.40)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_1)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \left(\Gamma_i X_{i+1,N}(Q_1)^T + H_i R_{11} \right)$$
(2.41)

Se a entrada for persistentemente excitante e posto $(\frac{1}{\sqrt{N}}\Gamma_i X_{i+1,N}(Q_2)^T) = n$, então para i > n o espaço coluna de $(1/N)R_{32}$ é uma estimativa consistente para o espaço coluna de Γ_i . Este resultado é obtido da equação (2.39) e (2.40). Para encontrar o espaço coluna da matriz $(1/N)R_{32}$, usa-se SVD:

$$\frac{1}{N}R_{32} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T$$

Define-se:

$$\Gamma_i = U_1 S_n^{1/2}$$

A matriz *C* é facilmente obtida de

$$C = \Gamma_i(1:l,1:n) \tag{2.42}$$

e a matriz A é obtida de

$$\Gamma_i(1:l(i-1),1:n)A = \Gamma_i(l+1:il,1:n)$$
(2.43)

Para obter as matrizes $B \in D$ emprega-se a equação (2.39) e (2.41), para obter:

$$(U_2^{\perp})^T R_{31}(R_{11}^{-1}) = (U_2^{\perp})^T H_i$$
(2.44)

Denotando $\Xi = (U_2^{\perp})^T R_{31} R_{11}^{-1}$, então a equação (2.44) é dada por:

$$\Xi = (U_2^{\perp})^T H_i$$

As matrizes $B \in D$ são encontradas da mesma forma que no algoritmo MOESP ordinário.

A seguir apresenta-se o teorema PIMOESP.

2.3.3.1 Teorema 2 (PIMOESP) [VERHAEGEN, 1992b]

Seja u(k) uma entrada persistentemente excitante para o sistema (2.31) e seja y(k) o sinal de saída expresso na equação (2.30). O sinal v(k) é ruído branco com média zero, independente dos dados de entrada. Dada a fatoração LQ:

$$\begin{bmatrix} U_{i+1,i,N} \\ U_{1,i,N} \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$
(2.45)

Então se tem:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_2)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_{i+1,N}(Q_2)^T$$
(2.46)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_1)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \left(\Gamma_i X_{i+1,N}(Q_1)^T + H_i R_{11} \right)$$
(2.47)

A prova deste teorema pode ser encontrada no apêndice B.10.

Comentários deste teorema

As equações (2.46) e (2.47) formam a base do método PIMOESP, o qual é resumido no algoritmo PIMOESP. As matrizes $A \in C$ são encontradas a partir de R_{32} , vide apêndice B.10. Intuitivamente, $R_{32}Q_2$ é a projeção das saídas dentro das entradas passadas. As matrizes $B \in D$ são computadas a partir de $R_{31} \in R_{11}$, vide apêndice B.10. Intuitivamente, $R_{31}Q_1$ é a projeção das saídas dentro das entradas futuras.

Algoritmo PIMOESP

- 1. Construir as matrizes de Hankel $U_{i+1,i,N}, U_{1,i,N}$ e $Y_{i+1,i,N}$
- 2. Calcular a fatoração LQ, equação (2.45).
- 3. Calcular a SVD da matriz $(1/N)R_{32}$ dada na equação (2.45).
- 4. Determinar a ordem do sistema por inspeção dos valores singulares em S_n e particionar a **SVD** para obter U_1 e $\Gamma_i = U_1 S_n^{1/2}$:

$$\frac{1}{N}R_{32} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T$$

5. As matrizes $A \in C$ são encontradas das equações:

$$\Gamma_{i}(1:l(i-1),1:n)A = \Gamma_{i}(l+1:il,1:n)$$
$$C = \Gamma_{i}(1:l,1:n)$$

6. As matrizes $B \in D$ são encontradas da equação (2.44).

A implementação deste algoritmo no programa Matlab 6.5 poder ser encontrada no apêndice F.

2.3.4 Algoritmo POMOESP

O algoritmo POMOESP (**P**ast - **O**utput do algoritmo MOESP) soluciona o problema de ruído na saída e ruído nos estados, conforme figura 2.6.



Figura 2.6 Modelo com ruído corrompido na saída e ruído nos estados

Nesta seção aborda-se o método de identificação por subespaços para sistemas com perturbação externa na saída, que pode ser erro do instrumento de medição, representada por v(k) e ruído nos estados, representado por w(k). Ambos os sinais são ruído branco com média zero.

A figura 2.6 é um modelo mais realístico, a identificação deste modelo é realizada pelo método direto (modelos tradicionais [LJUNG 1999; SÖDERSTROM; STOICA, 1989]) ou pelo método indireto (modelos MOESP, N4SID, CVA, etc).

Uma aproximação indireta do método de identificação por subespaços é dada pelo modelo POMOESP, que é uma extensão do modelo PIMOESP, pois a diferença entre estes modelos é dada pelo sinal w_k , conforme figuras 2.4 e 2.5. A figura 2.6 representa a equação 2.16.

Para o sistema (2.16) se pode escrever o sistema inovação na forma [VIBERG, 1995]:

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$
(2.48a)

$$y(k) = Cx(k) + Du(k) + v(k)$$
 (2.48b)

onde

w(k) = Ke(k) e v(k) = e(k), K denota o ganho de Kalman.

De forma geral, o problema de identificação por subespaços pode ser visto como encontrar uma aplicação f entre a seqüência dos dados de entrada – saída e os parâmetros desconhecidos do sistema (2.48).



Figura 2.7 Aplicação *f* entre a seqüência dos dados de entrada – saída e os parâmetros desconhecidos do sistema

Para encontrar estes parâmetros a partir dos dados de entrada e saída, existe um passo intermediário, que é a aproximação de uma estrutura de subespaço a partir de um espaço definido pelas matrizes em blocos de Hankel, construída a partir dos dados de entrada – saída.

Na família dos algoritmos MOESP [VERHAEGEN, 1993a] esta estrutura de subespaço é a matriz de observabilidade estendida.

As notações matriciais e definições usadas no método POMOESP são definidas no apêndice B.11.

Formulação do problema

Dado: Um conjunto de entradas $u(k) \in \Re^m$ e saídas $y(k) \in \Re^l$ para um sistema desconhecido de ordem *n*, dado pela equação (2.48), onde $w(k) \in v(k)$ são sinais do tipo ruído branco com média zero, não-correlacionados com os dados de entrada. **Determine**: A ordem *n* do sistema, as matrizes (*A*, *B*, *C*, *D*) e o ganho de Kalman *K*.

A solução deste problema é dada em [VERHAGEN, 1994a], em que se consideram dois tipos de sinais de entrada:

- 1) Sinal de entrada u(k) ruído branco com média zero
- 2) Sinal de entrada u(k) arbitrário (persistentemente excitante)

No método POMOESP, o sinal de entrada é considerada do tipo 2 . A seguir apresenta-se a idéia do método POMOESP.

Idéia do Algoritmo POMOESP

O objetivo neste método é o mesmo dos casos anteriores, isto é, computar a matriz de obervabilidade estendida a partir dos dados de entrada e saída.

Para o sistema (2.25) se tem a seguinte representação [vide apêndice B.11]:

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N} + \Phi_i W_{k,i,N} + V_{k,i,N}$$
(2.49)

Considerando k = i + 1 na equação (2.49) tem-se:

$$Y_{i+1,i,N} = \Gamma_i X_{i+1,N} + H_i U_{i+1,i,N} + \Phi_i W_{i+1,i,N} + V_{i+1,i,N}$$
(2.50)

onde $Y_{i+1,i,N}$ é a matriz em blocos de Hankel das saídas futuras e $U_{i+1,i,N}$ é a matriz em blocos de Hankel das entradas futuras.

Da mesma forma que no algoritmo MOESP e PIMOESP, é preciso obter uma aproximação da matriz de observabilidade estendida Γ_i . Por este motivo é preciso eliminar os dois últimos termos do lado direito da equação (2.50). Isto é feito em dois passos:

Primeiro passo

Para eliminar o termo $H_i U_{i+1,i,N}$, aplica-se o operador projeção $\prod_{iU_f^{\perp}}$, onde f = i + 1, i, N.

Segundo passo

Precisa-se de uma variável instrumental para eliminar os ruídos. A variável instrumental $U_{1,i,N}$ que foi usada no método PIMOESP é uma possível candidata, desde que:

$$\lim_{N \to \infty} \frac{1}{N} W_{k,i,N} U_{1,i,N}^{T} = 0 \qquad e \qquad \lim_{N \to \infty} \frac{1}{N} V_{k,i,N} U_{1,i,N}^{T} = 0$$

Da mesma forma se pode mostrar que:

$$\lim_{N \to \infty} \frac{1}{N} Y_{1,i,N} W_{k,i,N}^{T} = 0 \qquad e \qquad \lim_{N \to \infty} \frac{1}{N} Y_{1,i,N} V_{k,i,N}^{T} = 0$$

Então a variável instrumental no método POMOESP é escolhida como a combinação das entradas e saídas passadas:

$$Z = \begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix}.$$

Portanto, Verhaegen (1994) considera a seguinte projeção ortogonal: $\Pi_{U_{\tau}^{\perp}}Z^{T}$, onde f = i + 1, i, N.

Multiplica-se a equação (2.50) por $\Pi_{U_{\tau}^{\perp}} Z^{T}$ de ambos os lados, resultando:

$$\lim_{N \to \infty} \frac{1}{N} Y_{i+1,i,N} \Pi_{U_{f}^{\perp}} Z^{T} = \lim_{N \to \infty} \frac{1}{N} \Gamma_{i} X_{i+1,N} \Pi_{U_{f}^{\perp}} Z^{T} + \lim_{N \to \infty} \frac{1}{N} H_{i} U_{i+1,i,N} \Pi_{U_{f}^{\perp}} Z^{T} + \lim_{N \to \infty} \frac{1}{N} \Phi_{i} W_{i+1,i,N} \Pi_{U_{f}^{\perp}} Z^{T} + \lim_{N \to \infty} \frac{1}{N} V_{i+1,i,N} \Pi_{U_{f}^{\perp}} Z^{T}$$
(2.51)

Pela projeção ortogonal $U_{i+1,i,N}\Pi_{U_f^{\perp}} = 0$ e por serem os ruídos não correlacionados com os dados de entrada – saída, então os dois últimos termos da equação (2.51) correspondem á matriz nula para $N \to \infty$.

Portanto, à equação (2.51) pode ser expressa como:

$$\lim_{N \to \infty} \frac{1}{N} Y_{i+1,i,N} \Pi_{U_{f}^{\perp}} Z^{T} = \lim_{N \to \infty} \frac{1}{N} \Gamma_{i} X_{i+1,N} \Pi_{U_{f}^{\perp}} Z^{T}$$
(2.52)

A equação (2.52) pode ser eficientemente encontrada através da decomposição LQ, isto é :

$$\begin{bmatrix} U_{i+1,i,N} \\ Z \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} U_{i+1,i,N} \\ U_{1,i,N} \\ Y_{1,i,N} \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix}$$
(2.53)

Agora é preciso estimar o espaço coluna da matriz Γ_i .

Se o sinal de entrada u_k for persistentemente excitante e o número de dados tender a infinito, então se têm as seguintes relações:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_2^T) = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_{i+1,N}(Q_2^T)$$
(2.54)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_3^T) = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_{i+1,N}(Q_3^T)$$
(2.55)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{1,i,N}(Q_1^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_{1,N}(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{21}) \right)$$
(2.56)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{1,i,N}(Q_2^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_{1,N}(Q_2^T) + \frac{1}{\sqrt{N}} H_i(R_{22}) \right)$$
(2.57)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_1^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_{i+1,N}(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{11}) \right)$$
(2.58)

[vide apêndice B.12].

Logo, das equações (2.54) e (2.55) tem-se que:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} \begin{bmatrix} R_{42} & R_{43} \end{bmatrix} = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_{i,N} \begin{bmatrix} Q_2^T & Q_3^T \end{bmatrix}$$
(2.59)

[vide apêndice B.13].

Assim, para encontrar uma estimativa do espaço coluna da matriz Γ_i , toma-se a SVD da matriz $\begin{bmatrix} R_{42} & R_{43} \end{bmatrix}$:

$$\begin{bmatrix} R_{42} & R_{43} \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_n & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

Logo, o espaço coluna de U_1 aproxima consistentemente a matriz de observabilidade estendida Γ_i .

As matrizes A e C podem ser calculadas da mesma forma que no algoritmo MOESP (considerando $\Gamma_i \approx U_1$). A matriz C é facilmente obtida de:

$$C = \Gamma_i(1:l,1:n)$$

e a matriz A é obtida de

$$\Gamma_i(1:l(i-1),1:n)A = \Gamma_i(l+1:il,1:n)$$
(2.60)

Para obter as matrizes $B \in D$ empregam-se as equações (2.56), (2.57) e (2.58) para obter [vide apêndice B.14]:

$$\frac{1}{\sqrt{N}} \begin{bmatrix} R_{31} & R_{32} & R_{41} \end{bmatrix} = \Gamma_i \frac{1}{\sqrt{N}} \begin{bmatrix} X_p(Q_1^T) & X_p(Q_2^T) & X_f(Q_1^T) \end{bmatrix} + H_i \frac{1}{\sqrt{N}} \begin{bmatrix} R_{21} & R_{22} & R_{11} \end{bmatrix} + O(\varsigma)$$
(2.61)

Logo, como o espaço coluna da matriz Γ_i e seu respectivo complemento ortogonal é igual às matrizes $U_1 \in U_1^{\perp}$, então, multiplicando-se a equação (2.61) por $(U_1^{\perp})^T$ de ambos os lados, resulta:

$$\left(U_1^{\perp} \right)^T \frac{1}{\sqrt{N}} \begin{bmatrix} R_{31} & R_{32} & R_{41} \end{bmatrix} = \left(U_1^{\perp} \right)^T H_i \frac{1}{\sqrt{N}} \begin{bmatrix} R_{21} & R_{22} & R_{11} \end{bmatrix} + O(\varsigma)$$
 (2.62)

Como o sinal de entrada é persistentemente excitante, então a matriz $\begin{bmatrix} R_{21} & R_{22} & R_{11} \end{bmatrix}$ tem pseudo-inversa à direita. Multiplicando-se a equação (2.62) por esta inversa, se obtém:

$$(U_1^{\perp})^T \frac{1}{\sqrt{N}} \begin{bmatrix} R_{31} & R_{32} & R_{41} \end{bmatrix} \left(\frac{1}{\sqrt{N}} \begin{bmatrix} R_{21} & R_{22} & R_{11} \end{bmatrix} \right)^T = (U_1^{\perp})^T H_i + O(\varsigma)$$

Esta última equação pode ser reescrita como:

$$\Xi = \left(U_1^{\perp}\right)^T H_i + O(\varsigma) \, .$$

onde:

$$\Xi = \left(U_1^{\perp}\right)^T \frac{1}{\sqrt{N}} \begin{bmatrix} R_{31} & R_{32} & R_{41} \end{bmatrix} \left(\frac{1}{\sqrt{N}} \begin{bmatrix} R_{21} & R_{22} & R_{11} \end{bmatrix}\right)^*.$$

Por último, pode-se recuperar as matrizes (B, D):

$$\begin{bmatrix} \Xi(:,1:m) \\ \Xi(:,m+1:2m) \\ \vdots \\ \Xi(:,m(i-1)+1:mi) \end{bmatrix} = \begin{bmatrix} U_1^{\perp}(l(i-1)+1:li,:)^T & \cdots & U_1^{\perp}(1:l,:)^T \\ 0 & U_1^{\perp}(l(i-1)+1:li,:)^T & \cdots & U_1^{\perp}(l+1:2l,:)^T \\ 0 & 0 & \vdots \\ 0 & \cdots & U_1^{\perp}(l(i-1)+1:li,:)^T \end{bmatrix}$$

$$\begin{pmatrix}
U_{1}((i-2)l+1:(i-1)l,:) & 0 \\
\vdots & \vdots \\
U_{1}(1:l,:) & 0 \\
0 & I_{l}
\end{pmatrix}
\begin{pmatrix}
B \\
D
\end{pmatrix}$$
(2.63)

A seguir apresenta-se o teorema POMOESP.

2.3.4.1 Teorema 3 (POMOESP) [VERHAEGEN, 1994a]

Seja w(k) o processo de ruído e v(k) a medida de ruído do sistema (2.16), sendo ambos ruído branco com média zero, independente da entrada u(j) para todo k, j e do estado inicial x_0 . Suponha que u(k) seja uma entrada persistentemente excitante para o sistema (2.16) e que se tenha a seguinte fatoração LQ:

$$\begin{bmatrix} U_{f} \\ U_{p} \\ Y_{p} \\ Y_{f} \end{bmatrix} = \begin{bmatrix} U_{i+1,i,N} \\ U_{1,i,N} \\ Y_{1,i,N} \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q_{1} \\ Q_{2} \\ Q_{3} \\ Q_{4} \end{bmatrix}$$
(2.64)

Então se satisfaz:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f(Q_2)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_f(Q_2)^T$$
(2.65)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f(Q_3)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_f(Q_3)^T$$
(2.66)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_p(Q_1^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_p(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{21}) \right)$$
(2.67)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_p(Q_2^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_p(Q_2^T) + \frac{1}{\sqrt{N}} H_i(R_{22}) \right)$$
(2.68)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f(Q_1^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_f(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{11}) \right)$$
(2.69)

A prova deste teorema pode ser encontrada no apêndice B.12.

Algoritmo POMOESP

- 1. Construir as matrizes de Hankel $U_f, U_p \in Y_f, Y_p$.
- 2. Calcular a fatoração LQ, equação (2.64).
- 3. Calcular a SVD da matriz $[R_{42} R_{43}]$.
- 4. Determinar a ordem do sistema por inspeção dos valores singulares em S_n e particionar a **SVD** para obter U_1 e U_2 .
- 5. As matrizes *A* e *C* são encontradas das equações:

$$\Gamma_i(1:l(i-1),1:n)A = \Gamma_i(l+1:il,1:n)$$

$$C = \Gamma_i(1:l,1:n)$$

6. As matrizes $B \in D$ são encontradas da equação (2.63).

A implementação deste algoritmo no programa Matlab 6.5 poder ser encontrada no apêndice F

CAPÍTULO 3

IDENTIFICAÇÃO DE SISTEMAS PELO MÉTODO N4SID

Para solucionar o problema da teoria da realização (ver capítulo 2, item 2.1) existem diferentes formas, uma delas é através dos métodos de identificação por subespaços. Como exemplo, o método MOESP, desenvolvido por Michael Verhaegen, primeiro calcula a matriz de observabilidade a partir de uma matriz em blocos de Hankel, construída a partir dos dados de entrada e saída, depois se calculam as matrizes do sistema. Este método faz uso de projeções ortogonais.

Outra forma de solucionar o problema citado acima, é o método N4SID (Numerical algorithms for Subspace State Space System IDentification), apresentado por Peter Van Overschee e Bart de Moor, isto é, calcula-se uma estimativa dos estados do processo a partir de uma matriz em blocos de Hankel, construída a partir dos dados de entrada e saída, depois se determinam as matrizes do sistema. Este método faz uso de projeções oblíquas.

Como foi mencionado anteriormente, os teoremas apresentados neste capítulo não são demonstrados, para demonstrações e maiores detalhes vide [VAN OVERSCHEE; DE MOOR, 1996; SANTOS MIRANDA, 2004].

A seguir apresenta-se a idéia do método N4SID.

3.1 Modelos por subespaços

Os modelos por subespaços citados neste capítulo estão relacionados com sistemas e modelos da forma inovativa (ver capítulo 2)

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$
(3.1a)

$$y(k) = Cx(k) + Du(k) + v(k)$$
 (3.1b)

com

$$E\left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq} \ge 0$$
(3.1c)

onde:

$[u(k)]_{mx1}$	vetor de entrada do processo. São dados conhecidos.
$[y(k)]_{k1}$	vetor de saída do processo. São dados conhecidos.
$[x(k)]_{nx1}$	vetor de estados do processo. São dados não conhecidos.
$[w(k)]_{nx1}$	vetor de ruído do processo, é um sinal não medido.
$[v(k)]_{lx1}$	vetor de ruído na medição, é um sinal não medido.

- [A] _{nxn} matriz dinâmica do sistema, é uma matriz desconhecida
- [B] _{nxm} matriz de entrada do sistema, é uma matriz desconhecida
- $[C]_{lxn}$ matriz de saída do sistema, é uma matriz desconhecida
- $[D]_{lxm}$ matriz de transmissão direta, é uma matriz desconhecida.

 $[Q]_{nxn}$, $[S]_{nxl}$ e $[R]_{lxl}$ são as matrizes de covariância das seqüências de ruído w(k) e v(k). *E* é operador esperança matemática e δ é o delta de Kronecker.

A justificativa de encontrar um modelo inovativo é que a seqüência de estados x(k) seja recuperada dos dados de entrada e saída. Então se procurará um sistema em espaço de estados da forma inovativa.

Por um instante, suponha (idealmente) que seja possível coletar no intervalo de tempo [0, N], as amostras $\{y(k)\}, \{u(k)\} \in \{x(k)\}$ do processo $\{y(k)\}, \{u(k)\} \in \{x(k)\}$. Desde que os processos são gerados pelos dados amostrados, então as matrizes construídas em cada instante de tempo k são finitas, $\{Y(k)\}, \{U(k)\} \in \{X(k)\}, e$ são representadas

pelas fórmulas:

$$U_{k} = [u(k) \quad u(k+1) \quad \dots \quad u(k+N-1)]$$
(3.2)

$$Y_{k} = [y(k) \quad y(k+1) \quad \dots \quad y(k+N-1)]$$
(3.3)

$$X_{k} = [x(k) \quad x(k+1) \quad \dots \quad x(k+N-1)]$$
(3.4)

As equações (3.2), (3.3) e (3.4) também satisfazem o sistema (3.1), isto é:

$$X_{k+1} = AX_k + BU_k + W_k$$

$$Y_k = CX_k + DU_k + V_k$$

$$(3.5)$$

ou, em forma matricial

$$\begin{bmatrix} \boldsymbol{X}_{k+1} \\ \boldsymbol{Y}_{k} \end{bmatrix} = \boldsymbol{\theta} \begin{bmatrix} \boldsymbol{X}_{k} \\ \boldsymbol{U}_{k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{W}_{k} \\ \boldsymbol{V}_{k} \end{bmatrix}$$
(3.6)

onde $\boldsymbol{\theta} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ corresponde aos parâmetros desconhecidos. A equação (3.6) pode

ser interpretada como um modelo de regressão. Se na equação (3.6), as matrizes X_{k+1} , Y_k , X_k e U_k são dadas, então o parâmetro desconhecido θ pode ser calculado pelo método dos mínimos quadrados, isto é:

$$\hat{\theta} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \min_{A,B,C,D} \left\| \begin{bmatrix} X_{k+1} \\ Y_k \end{bmatrix} - \theta \begin{bmatrix} X_k \\ U_k \end{bmatrix} \right\|_F^2$$
(3.7)

onde $\hat{\theta}$ denota a estimativa de θ (as estimativas são denotadas por ($\hat{\circ}$)) e $\|\|_{F}^{2}$ denota a norma de Frobenius de uma matriz.

Da equação (3.7), resulta:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{X}_{k+1} \\ \boldsymbol{Y}_{k} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_{k} \\ \boldsymbol{U}_{k} \end{bmatrix}^{T} \left(\begin{bmatrix} \boldsymbol{X}_{k} \\ \boldsymbol{U}_{k} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_{k} \\ \boldsymbol{U}_{k} \end{bmatrix}^{T} \right)^{-1}$$

As estimativas das matrizes de covariância do processo e do ruído de medição são obtidas de:

$$\begin{pmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{pmatrix} = \frac{1}{N} \begin{pmatrix} \rho w_i & \rho w_{i+1} & \dots & \rho w_{i+N-1} \\ \rho v_i & \rho v_{i+1} & \dots & \rho v_{i+N-1} \end{pmatrix} \begin{pmatrix} \rho w_i & \rho w_{i+1} & \dots & \rho w_{i+N-1} \\ \rho v_i & \rho v_{i+1} & \dots & \rho v_{i+N-1} \end{pmatrix}^T$$
(3.8)

onde $\rho w_k = \hat{x}_{k+1} - \hat{A}x_k - \hat{B}u_k$ e $\rho v_k = y_k - \hat{C}\hat{x}_k - \hat{D}\hat{u}_k$ para k = i,...,i + N - 1 são os resíduos dos mínimos quadrados.

Então em um caso ideal, quando se têm os dados de entrada, saída e a seqüência de estados para dois instantes de tempo sucessivos k e k +1, a identificação do parâmetro $\boldsymbol{\theta}$ no sistema (3.6) é trivial.

Mas infelizmente, na prática, $X_{k+1} e X_k$ não são obtidos e têm que ser estimados dos dados de entrada e saída. Isto é um ponto importante nos métodos de identificação por subespaços, a diferença entre estes métodos reside na forma de como obter a seqüência de estados estimados.

Uma forma de estimar X_{k+1} e X_k é através da projeção oblíqua de certas matrizes em blocos de Hankel, construída a partir dos dados de entrada e saída. Esta projeção oblíqua é computada eficientemente através da decomposição QR (notação dada no método N4SID, para a decomposição LQ).

3.2 Definição da Projeção Oblíqua

Dadas duas matrizes $A \in \Re^{pxj}$ e $B \in \Re^{qxj}$ foi visto que se pode encontrar a projeção ortogonal da matriz A sobre a matriz B, a qual é denotada por A/B. Pode-se encontrar A/B^{\perp} , isto é, a projeção ortogonal da matriz A sobre o complemento ortogonal da matriz B. Pode-se interpretar estas projeções no espaço de dimensão j = 2, conforme mostrado na figura 3.1.



Figura 3.1 Interpretação da projeção ortogonal no espaço *j*-dimensional (*j*=2 neste caso)

Define-se $\Pi_B = B(BB^T)B$ o operador projeção da matriz *B*, e $\Pi_{B^{\perp}} = I - \Pi_B$ o operador projeção do complemento ortogonal da matriz *B*. Assim, as projeções ortogonais podem

ser definidas como: $A/B = A\Pi_B$ e $A/B^{\perp} = A\Pi_{B^{\perp}}$. Então a matriz A pode ser expressa como combinação linear destas projeções ortogonais:

$$A/B^{\perp} = A - A/B$$

Logo:

$$A - A/B = A(I_{j} - B(BB^{T})^{*}B)$$

$$A = A/B + A(I_{j} - B(BB^{T})^{*}B)$$

$$A = A\Pi_{B} + A(I - \Pi_{B})$$

$$A = A\Pi_{B} + A\Pi_{B^{\perp}}.$$
(3.9)
ou

$$A = A/B + A/B^{\perp}$$

Estas projeções podem ser encontradas usando-se decomposições QR (ou LQ), que depois são aplicadas para encontrar os espaços linha ou coluna de certas matrizes.

Ao invés de se decompor A como uma combinação linear de duas matrizes ortogonais, pode-se decompô-la como uma combinação linear de duas matrizes não-ortogonais $B \in \Re^{qxj}$ e $C \in \Re^{rxj}$ e de seus respectivos complementos ortogonais B^{\perp} e C^{\perp} . A idéia é fazer a projeção ortogonal da matriz A sobre a matriz em blocos $\begin{bmatrix} B \\ C \end{bmatrix}$ com a condição de que a intersecção dos espaços fila de B e C seja nula. Esta projeção é mostrada na figura 3.2



Figura 3.2 Interpretação da projeção ortogonal no espaço *j*-dimensional (*j* =3 neste caso).

Pela equação (3.9), a matriz A pode ser expressa por:

$$A = A / \begin{bmatrix} B \\ C \end{bmatrix} + A / \begin{bmatrix} B \\ C \end{bmatrix}^{\perp}$$
(3.10)

Como a matriz $A / \begin{bmatrix} B \\ C \end{bmatrix}$ encontra-se no espaço formado pelas matrizes *B* e *C*, então neste

espaço pode-se encontrar as componentes ortogonais da matriz $A / \begin{bmatrix} B \\ C \end{bmatrix}$, isto é:

$$A / \begin{bmatrix} B \\ C \end{bmatrix} = A / _{C} B + A / _{B} C$$
(3.11)

Na equação (3.11) o lado esquerdo é a projeção ortogonal e o lado direito é a soma direta de subespaços.

Substituindo-se a equação (3.11) na equação (3.10), resulta:

$$A = A/_{C}B + A/_{B}C + A/\begin{bmatrix} B\\ C \end{bmatrix}^{\perp}$$
(3.12)

onde:

 $A/_{C}B$ é a projeção oblíqua de A em B sobre C

 $A/_{B}C$ é a projeção oblíqua de A em C sobre B, [DE COCK; DE MOOR, 2003].

A projeção oblíqua é ilustrada na figura 3.3



Figura 3.3 Interpretação da projeção oblíqua no espaço *j*-dimensional (*j*=3 neste caso e p = q = r = 1).

Definição 3.1: Projeção Oblíqua. A projeção oblíqua do espaço de linhas de $A \in \Re^{pxj}$ no espaço de linhas de $B \in \Re^{qxj}$ sobre o espaço de linhas de $C \in \Re^{rxj}$ é definida como:

$$A/_{B}C = A\begin{bmatrix} C^{T} & B^{T} \end{bmatrix} \cdot \begin{bmatrix} CC^{T} & CB^{T} \\ BC^{T} & BB^{T} \end{bmatrix}^{+} \Big]_{\text{primeiras } r \text{ columas}} \cdot C$$
(3.13)

Muitas vezes os corolários são uma ferramenta para aplicação a casos práticos. Então uma definição equivalente da projeção oblíqua é apresentada a seguir.

Corolário 3.1: projeções oblíquas. A projeção oblíqua do espaço de linhas de $A \in \Re^{pxj}$ no espaço de linhas de $B \in \Re^{qxj}$ sobre o espaço de linhas de $C \in \Re^{rxj}$ pode ser definida como:

$$A/_{B}C = \left[A/B^{\perp}\right] \cdot \left[C/B^{\perp}\right]^{*} \cdot C.$$
(3.14)

A notação matricial usada nesta tese são definidas no apêndice C.1.

3.3 Método N4SID

Este capítulo é baseado no livro de PETER VAN OVERSCHEE e BART DE MOOR Subspace Identification for Linear Systems (1996). Apresenta-se, a seguir, o método N4SID para o caso determinístico e combinado determinístico e estocástico. Para o caso puramente estocástico ver [VAN OVERSCHEE; DE MOOR, 1996] e [SANTOS MIRANDA, 2004].

3.3.1 Método N4SID, Identificação determinística

Nesta sub-seção aborda-se o método de identificação por subespaços para sistemas puramente determinísticos, com ruído de processo w(k) e de medição v(k) nulos, isto é, $w(k) \equiv v(k) \equiv 0$. O objetivo deste capítulo é entender como usar os conceitos e ferramentas da álgebra linear para a identificação determinística, que é depois aplicada a sistemas combinados determinístico e estocástico.

Como, na prática, os dados amostrados são corrompidos por ruído, a identificação de sistemas puramente determinísticos é aplicada de forma acadêmica.

O problema da identificação determinística pode ser postulado como:

Formulação do Problema

Dado: *s* amostras das entradas $u(k) \in \Re^m$ e das saídas $y(k) \in \Re^l$ geradas pelo seguinte sistema determinístico desconhecido, de ordem *n*:

$$x^{d}(k+1) = Ax^{d}(k) + Bu(k)$$
(3.15a)

$$y(k) = Cx^{a}(k) + Du(k)$$
 (3.15b)

onde $x^{d}(k) \in \Re^{n}$ é um vetor de estados, sendo que o superscrito "*d*" denota determinístico.

Determine:

- A ordem *n* do sistema desconhecido.
- As matrizes do sistema $A \in \Re^{nxn}$, $B \in \Re^{nxm}$, $C \in \Re^{lxn}$, $D \in \Re^{lxm}$.

O sistema descrito pelas equações (3.15a) e (3.15b) pode ser representado pelo diagrama de blocos mostrado na figura 3.4.

Sem perda de generalidade, muitas vezes se usara a notação $F_k = F(k)$, onde F representa qualquer seqüência no tempo k.



Figura 3.4 Um sistema linear determinístico invariante no tempo.

Na figura 3.4 o símbolo Δ representa atraso. As entradas e saídas são conhecidas, os estados são desconhecidos, mas são determinados como um resultado intermediário no algoritmo de identificação por subespaços.
Idéia do Algoritmo N4SID, caso determinístico

A idéia é recuperar a sequência de estados. Para isto, tem-se que demonstrar que:

- i) A sequência de estados X_f^d está contida no espaço fila de W_f
- ii) A sequência de estados X_f^d está contida no espaço fila de W_p
- iii) Aplicando-se o teorema da dimensão de Grassmann, obtém-se: dim(espaço $fila[W_p] \cap$ espaço $fila[W_f]) = n$

Para sua demonstração vide apêndice C.2.

Em [KATAYAMA, 2005] demonstra-se que esta interseção é *n* dimensional, e qualquer base para a intersecção entre passado e futuro representa uma seqüência de estado válida.

De i) e ii) vê-se que X_f^d encontra-se em W_f (futuro) e W_p (passado), respectivamente. Isto quer dizer que o vetor de estados serve como uma memória levando informação entre o passado e o futuro. O vetor de estados X_f^d pode ser computado através da projeção oblíqua $Y_f / U_f W_p$.

A equação (C2.1) é dada por:

$$Y_f = \Gamma_i X_f^d + H_i U_f \tag{3.16}$$

A seqüência de estados X_f^d pode ser expressa como a combinação linear das entradas passadas e saídas passadas [vide apêndice C3]:

$$X_f^d = L_p W_p \tag{3.17}$$

Substituindo-se a equação (3.17) na equação (3.16), resulta:

$$Y_f = \Gamma_i L_p W_p + H_i U_f \tag{3.18}$$

Com a finalidade de eliminar o último termo da equação (3.18), multiplica-se ambos os lados da equação (3.18) por $\Pi_{U_{\tau}^{\perp}}$, obtendo-se:

$$Y_{f}\Pi_{U_{f}^{\perp}} = \Gamma_{i}L_{p} W_{p}\Pi_{U_{f}^{\perp}} + H_{i}U_{f}\Pi_{U_{f}^{\perp}}$$
ou

$$Y_f / U_f^{\perp} = \Gamma_i L_p W_p / U_f^{\perp} + H_i \underbrace{U_f U_f^{\perp}}_{=0}$$
(3.21)

$$Y_f / U_f^{\perp} = \Gamma_i L_p W_p / U_f^{\perp}$$
(3.20)

Multiplica-se a equação (3.20) por $\left[W_p / U_f^{\perp}\right]^* W_p$ de ambos os lados, resultando:

$$\left[Y_f / U_f^{\perp}\right] \left[W_p / U_f^{\perp}\right]^* W_p = \Gamma_i L_p \underbrace{W_p / U_f^{\perp} \left[W_p / U_f^{\perp}\right]^* W_p}_{=J1}$$
(3.21)

onde o termo J1 na equação (3.21) é dado por:

$$J1 = W_p / U_f^{\perp} \left[W_p / U_f^{\perp} \right]^* W_p = W_p$$
(3.22)

Substituindo-se a equação (3.22) na equação (3.21), se obtém:

$$\left[Y_f / U_f^{\perp}\right] \left[W_p / U_f^{\perp}\right]^* W_p = \Gamma_i L_p W_p$$
(3.23)

Substituindo-se a equação (3.17) na equação (3.23), resulta:

$$\underbrace{\left[Y_{f} / U_{f}^{\perp}\right]\left[W_{p} / U_{f}^{\perp}\right]^{*} W_{p}}_{=Y_{f} / U_{f}^{\perp} W_{p}} = \Gamma_{i} \underbrace{L_{p} W_{p}}_{=X_{f}^{d}}$$

$$Y_{f} / U_{f}^{\perp} W_{p} = \Gamma_{i} X_{f}^{d}$$
(3.24)

ou

$$\mathcal{G}i = \Gamma_i X_f^d \tag{3.25}$$

onde o símbolo *9i* denota a projeção oblíqua:

$$\mathcal{G}i = Y_f \Big/_{U_f^\perp} W_p \tag{3.26}$$

A projeção oblíqua pode ser eficientemente computada através da fatoração QR (ou LQ). O vetor seqüência de estados pode ser computado através da SVD da matriz \Re , isto é:

$$\mathcal{G}i = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} = U_1 S_1 V_1^T$$
(3.27)

$$X_f^d = \Gamma_i^* \cdot \mathcal{G}_i \tag{3.28}$$

onde $\Gamma_i = U_1 S_1^{1/2} \cdot T$, para alguma matriz não singular *T*.

Da mesma forma, encontra-se \mathcal{G}_{i-1} , projeção oblíqua para a seqüência de estados X_{i+1}^d ,

dada por:

$$\mathcal{G}_{i-1} = Y_f^- /_{U_f^-} W_p^+ \tag{3.29}$$

e a sequência de estados é expressa por:

$$X_{i+1}^{d} = \Gamma_{i-1}^{+} \vartheta_{i-1}$$
(3.30)

Uma vez estimadas, X_i^d e X_{i+1}^d das equações (3.28) e (3.30), respectivamente, o último passo é calcular as matrizes do sistema *A*, *B*, *C* e *D*, pelo método dos minímos quadrados, aplicado ao seguinte conjunto de equações lineares:

$$\begin{pmatrix} X_{i+1}^{d} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_{i}^{d} \\ U_{i|i} \end{pmatrix}$$
(3.31)

Este método é mostrado no Algoritmo 1, usando estados.

OUTRA FORMA DE COMPUTAR AS MATRIZES (A, B, C, D)

Observar a equação (3.28); para estimar a seqüência de estados $X_i^d = X_f^d$, primeiro se calcula a matriz observabilidade estendida $\Gamma_i = U_1 S_1^{1/2} \cdot T$. Conhecida Γ_i , é fácil obter as matrizes $A \in C$.

Determinando as matrizes A e C

A matriz A é determinada pela seguinte equação:

$$A = \underline{\Gamma_i^+} \overline{\Gamma_i}$$

onde $\overline{\Gamma_i}$ denota a matriz Γ_i sem as *l* (número de saídas) primeiras filas e a matriz *C* como as primeiras *l* filas de Γ_i . Agora só falta determinar as matrizes *B* e *D*.

Determinando as matrizes B e D

Multiplica-se a equação (3.16) por Γ_i^{\perp} de ambos os lados, resultando:

$$\Gamma_i^{\perp} Y_f = \Gamma_i^{\perp} \Gamma_i X_f + \Gamma_i^{\perp} H_i U_f$$
(3.32)

onde a matriz $\Gamma_i^{\perp} \in \Re^{(li-n)xli}$ tem posto completo e satisfaz $\Gamma_i^{\perp}\Gamma_i = 0$. Substituindo-se esta última condição em (3.32), se obtém:

$$\Gamma_i^{\perp} Y_f = \Gamma_i^{\perp} H_i U_f \tag{3.33}$$

As matrizes $B \in D$ são encontradas a partir da matriz em blocos H_i . Multiplica-se a equação (3.33) por U_f^* de ambos os lados, resultando:

$$\Gamma_i^{\perp} Y_f U_f^* = \Gamma_i^{\perp} H_i \tag{3.34}$$

com:

1

$$\begin{pmatrix} M_{1} & M_{2} & \dots & M_{i} \end{pmatrix} = \Gamma_{i}^{\perp} \cdot Y_{f} \cdot U_{f}^{+}$$

$$\begin{pmatrix} L_{1} & L_{2} & \dots & L_{i} \end{pmatrix} = \Gamma_{i}^{\perp}$$
Calcula-se *B* e *D* de:
$$\begin{pmatrix} M_{1} \\ M_{2} \\ \vdots \\ M_{i} \end{pmatrix} = \begin{pmatrix} L_{1} & L_{2} & \dots & L_{i-1} & L_{i} \\ L_{2} & L_{3} & \dots & L_{i} & 0 \\ \dots & \dots & \dots & \dots \\ L_{i} & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_{i} & 0 \\ 0 & \underline{\Gamma_{i}} \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix}$$

$$(3.35)$$

Isto é mostrado no Algoritmo 2, usando a matriz de observabilidade estendida.

A seguir apresenta-se o Teorema N4SID para o caso determinístico.

3.3.1.1. Teorema 1: Identificação determinística

Suponha as seguintes hipóteses:

- A sequência de entradas u_k é persistentemente excitante de ordem 2.*i*. 1.
- A intersecção dos espaços de filas de U_f e o espaço de filas de X_p^d é vazia. 2.
- As matrizes de ponderação $W_1 \in \Re^{lixli}$ e $W_2 \in \Re^{jxj}$ são tais que W_1 tem posto 3. completo e W_2 obedece: posto $(W_p) = posto(W_pW_2)$.

 W_1 e W_2 determinam a base no espaço de estados em que o modelo será identificado [VAN OVERSCHEE; DE MOOR, 1996].

Definindo-se a projeção oblíqua ϑ_i como:

$$\Theta_i = Y_f /_{U_f} W_p \tag{3.36}$$

e a decomposição em valores singulares:

$$W_1 \vartheta_i W_2 = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$
(3.37)

$$=U_1S_1V_1^T (3.38)$$

Então tem-se que:

1.
$$\theta_i = \Gamma_i \cdot X_f^d$$
 (3.39)

 A ordem do sistema é igual ao número de valores singulares presentes na equação (3.37) que sejam diferentes de zero.

3.
$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} \cdot T$$
 (3.40)

sendo $T \in \Re^{nxn}$ uma transformação de similaridade arbitrária e não-singular.

A parte da seqüência de estados X^d_f que se encontra no espaço de colunas de W₂ pode ser recuperada a partir de:

$$X_f^d W_2 = T^{-1} \cdot S_1^{1/2} V_1^T \tag{3.41}$$

5.
$$X_f^d = \Gamma_i^+ \cdot \vartheta_i$$
 (3.42)

A prova deste teorema pode ser encontrada em [VAN OVERSCHEE; DE MOOR 1996; SANTOS MIRANDA, 2004].

As principais conseqüências do teorema de identificação determinística são:

- a) A seqüência de estados X^d_f pode ser determinada diretamente a partir dos dados u_k
 e y_k, sem o conhecimento prévio das matrizes do sistema A, B, C e D; e
- b) A matriz de observabilidade estendida Γ_i pode ser determinada diretamente dos dados de entrada e saída.

As matrizes do sistema podem ser extraídas dos resultados intermediários de X_f^d e Γ_i , mediante algoritmos de identificação usando estados e algoritmos de identificação usando a matriz de observabilidade estendida, respectivamente, conforme esquematizado na figura 3.5.



Figura 3.5 Diagrama esquemático de obtenção das matrizes do sistema

A descrição de ambos os algoritmos é feita a seguir [VAN OVERSCHEE; DE MOOR, 1996].

Algoritmo determinístico 1

(empregando estados)

1. Calcule as projeções oblíquas

$$\begin{split} \boldsymbol{\vartheta}_i &= \boldsymbol{Y}_f \: /_{U_f} \: \boldsymbol{W}_p \\ \boldsymbol{\vartheta}_{i-1} &= \boldsymbol{Y}_f^- \: /_{U_f^-} \: \boldsymbol{W}_p^+ \end{split}$$

- 2. Calcule a SVD da projeção oblíqua ponderada $W_1 \vartheta_i W_2 = USV^T$
- 3. Determine a ordem do sistema por inspeção dos valores singulares em S e encontre U_1 e S_1 , através do particionamento adequado da SVD.
- 4. Determine Γ_i e Γ_{i-1} como:

 $\Gamma_{i} = W_{1}^{-1}U_{1}S_{1}^{1/2} \cdot T \quad \Gamma_{i-1} = \underline{\Gamma_{i}}$

onde $\underline{\Gamma_i}$ denota a matriz Γ_i sem as *l* filas (número de saídas) passadas de Γ_i .

5. Determine X_i^d e X_{i+1}^d como:

$$X_i^d = \Gamma_i^* \mathcal{G}_i \qquad X_{i+1}^d = \Gamma_{i-1}^* \mathcal{G}_{i-1}$$

6. Solucione o seguinte conjunto de equações lineares para obter A, B, C e D:

$$\begin{pmatrix} X_{i+1}^{d} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_{i}^{d} \\ U_{i|i} \end{pmatrix}$$

Algoritmo determinístico 2

(empregando a matriz de observabilidade estendida)

- 1. Calcule a projeção oblíqua $\vartheta_{i} = Y_{f} / _{U_{f}} W_{p}$
- 2. Calcule a SVD da projeção oblíqua ponderada $W_1 \vartheta_i W_2 = USV^T$
- 3. Determine a ordem do sistema por inspeção dos valores singulares em S e encontre U_1, U_2 e S_1 , através do particionamento adequado da SVD.

4. Determine
$$\Gamma_i \in \Gamma_i^{\perp}$$
 como:

 $\Gamma_i = W_1^{-1} U_1 S_1^{1/2} \qquad \Gamma_i^{\perp} = U_2^T W_1$

- 5. Determine a matriz do sistema *A* a partir de Γ_i como $A = \underline{\Gamma}_i^+ \overline{\Gamma}_i$, onde $\overline{\Gamma}_i$ denota a matriz Γ_i sem as *l* (número de saídas) primeiras filas.
- 6. Determine *C* como as primeiras *l* filas de Γ_i

Com :

$$\begin{pmatrix} M_1 & M_2 & \dots & M_i \end{pmatrix} = \Gamma_i^{\perp} \cdot Y_f \cdot U_f^+$$
$$\begin{pmatrix} L_1 & L_2 & \dots & L_i \end{pmatrix} = \Gamma_i^{\perp}$$

calcule $B \in D$ de:

$$\begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_i \end{pmatrix} = \begin{pmatrix} L_1 & L_2 & \dots & L_{i-1} & L_i \\ L_2 & L_3 & \dots & L_i & 0 \\ \dots & \dots & \dots & \dots & \dots \\ L_i & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_l & 0 \\ 0 & \underline{\Gamma_i} \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix}.$$

Estes algoritmos estão implementados no apêndice F

Os modelos determinísticos são puramente acadêmicos. A seguir, apresenta-se o método N4SID para um problema mais real, isto é, modelo determinístico - estocástico.

3.3.2 O Método N4SID, Identificação determinística - estocástica

Os métodos de identificação por subespaços estimam seqüências de estados diretamente dos dados de entrada e saída, explicitamente ou implicitamente, com uma projeção ortogonal ou oblíqua dos espaços de filas de certas matrizes em blocos de Hankel de dados (de entrada e saída) nos espaços de filas de outras matrizes em blocos de Hankel, seguido por uma decomposição em valores singulares (SVD) para determinar a ordem, a matriz de observabilidade e/ou a seqüência de estados. A extração do modelo em espaço de estados então se alcança através da solução do problema dos mínimos quadrados. Cada um destes passos pode ser executado usandose algoritmos lineares numéricos bem conhecidos da álgebra, tais como a decomposição em valores singulares e a decomposição QR.

Formulação do problema

Este trabalho se desenvolve sobre um espaço de Hilbert, a identificação determinísticaestocástica se faz para modelos em espaço de estados de tempo discreto, lineares, invariantes no tempo.

O problema de identificação determinística - estocástica pode ser postulado como:

Dados: *s* amostras medidas das entradas u(k) e das saídas y(k) geradas pelo sistema (3.1).

Determine a ordem n do sistema desconhecido, as matrizes do sistema A, B, C, D e as matrizes de covariância de ruído Q, S, R.



O sistema (3.1) na forma de diagrama dee blocos é mostrado na figura 3.6.

Figura 3.6 Sistema linear combinado determinístico-estocástico

Idéia do algoritmo N4SID caso: determinístico - estocástico

A idéia é a mesma apresentada no sistema determinístico, isto é, recuperar a seqüência de estados a partir dos dados de entrada e saída, mas a forma como obtê-lo é diferente, pois agora o sistema tem seqüência de ruído.

O objetivo de um procedimento de identificação é encontrar um modelo cujo comportamento entrada-saída se aproxime daquele do processo real. Este objetivo é resolvido classicamente minimizando-se um "critério do erro de predição", o qual expressa o "desempenho da predição" do modelo no conjunto fornecido de dados. A solução minimizante é designada como o modelo ótimo (veja, por exemplo, [LJUNG, 1999]).

Como indicado acima, deseja-se encontrar um modelo que prediga o comportamento do processo de forma suficientemente exata. Isto pode ser formulado como: prediga as saídas futuras (Y_f) tão exatamente quanto possível, usando toda a informação que possa ser obtida dos dados passados (W_p) e usando o conhecimento das entradas que serão apresentadas ao sistema no futuro (U_f) .

Devido à linearidade do sistema, propõe-se combinar linearmente as entradas passadas

 (W_p) e futuras (U_f) para predizer as saídas futuras (Y_f) . Denotam-se as combinações lineares, respectivamente, por L_p e L_u . A qualidade da predição é medida pela norma de Frobenius, isto é:

$$\min_{\substack{L_p \in \Re^{lix(m+l)i} \\ L_u \in \Re^{lixmi}}} \left\| Y_f - \begin{bmatrix} L_p & L_u \end{bmatrix} \begin{bmatrix} W_p \\ U_f \end{bmatrix} \right\|_F^2$$

Seja:

$$Z_{i} = \begin{bmatrix} L_{p} & L_{u} \end{bmatrix} \begin{bmatrix} W_{p} \\ U_{f} \end{bmatrix} = L_{p}W_{p} + L_{u}U_{f}.$$

A combinação ótima do passado (W_p) para predizer o futuro é $L_p \cdot W_p$, que é exatamente igual à projeção oblíqua $\mathcal{G}_i = L_p \cdot W_p$.

Para conseguir este objetivo, tem-se que:

- i) Expressar Z_i como combinação linear das matrizes W_p e U_f .
- ii) Demonstrar que a combinação ótima do passado (W_p) para predizer o futuro é $L_p \cdot W_p$ e é exatamente igual à projeção oblíqua de $\mathcal{P}_i = L_p \cdot W_p$.

Os itens i) e ii) são realizados através dos três teoremas que são apresentados mais adiante, no entanto, apresenta-se uma seqüência para satisfazer os itens citados acima.

Para satisfazer o item i) considere-se a seguinte sequência:

- a) Escrever os estados estimados pelo filtro de Kalman x̂_k como uma combinação linear das entradas passadas u₀,....,u_{k-1}, saídas passadas y₀,...., y_{k-1} e do estado inicial estimado x̂₀.
- b) Assim, a sequência de estados \hat{X}_i pode ser obtida por:

$$\hat{X}_{i} = \begin{bmatrix} \hat{x}_{i} & \hat{x}_{i+1} & \dots & \hat{x}_{i+j-1} \end{bmatrix} = \begin{bmatrix} A^{i} - \Omega_{i}\Gamma_{i} \\ & \Delta_{i}^{d} - \Omega_{i}H_{i}^{d} \\ \end{bmatrix} \begin{bmatrix} \hat{X}_{0} \\ W_{p} \end{bmatrix}$$

c) Expressar as saídas futuras Z_i como uma combinação linear das entradas passadas W_p e futuras U_f . Isto é feito pela projeção ortogonal:

$$Z_i = L_p W_p + L_u U_f = Y_f \left(\begin{pmatrix} W_p \\ U_f \end{pmatrix} \right) = \Gamma_i \hat{X}_i + H_i^d U_f$$
(3.43)

Se a interseção dos espaços \hat{X}_i e U_f for vazia, então uma estimativa da seqüência \hat{X}_i pode ser encontrada através da projeção oblíqua de Y_f em W_p sobre U_f :

$$\Theta_i \stackrel{def}{=} Y_f / _{U_f} W_p = \Gamma_i \cdot \widetilde{X}_i$$
(3.44)

Desta forma é satisfeita a parte ii).

Apresenta-se, a seguir, duas formas para determinar as matrizes do sistema (3.1), empregando o vetor de estados.

Encontrando as matrizes do sistema via seqüência de estados \widetilde{X}_i

A seqüência de estados \widetilde{X}_i é calculada pela equação (3.44) a partir dos dados de entrada e saída, sem o conhecimento das matrizes do sistema:

$$\widetilde{X}_i = \Gamma_i^* \cdot \mathcal{G}_i \tag{3.45}$$

Analogamente, a seqüência de estado \widetilde{X}_{i+1} pode ser determinada como:

$$\mathcal{G}_{i+1} \stackrel{aeg}{=} Y_f^- /_{U_f^-} W_p^+ = \Gamma_{i-1} \cdot \widetilde{X}_{i+1}$$

$$\widetilde{X}_{i+1} = \Gamma_{i-1}^* \mathcal{G}_{i+1}$$
(3.46)

Então as matrizes do sistema podem ser computadas da seguinte equação linear:

$$\begin{pmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \widetilde{X}_i \\ U_{i|i} \end{pmatrix} + \begin{pmatrix} \rho w \\ \rho v \end{pmatrix}$$
(3.46)

onde ρw e ρv são matrizes residuais. Estas matrizes residuais são perpendiculares a \widetilde{X}_i e $U_{i|i}$, assim a equação (3.46) pode ser solucionada pelo método dos mínimos quadrados.

Este método produz estimativas com polarização, pois a seqüência de estados \hat{X}_i não é calculada diretamente através dos dados de entrada e saída, mas sim por sua estimativa

 \widetilde{X}_i . O problema é que a nova seqüência do filtro de Kalman \widetilde{X}_{i+1} tem estado inicial diferente que a seqüência \widetilde{X}_i , isto é:

Estado inicial para $\widetilde{X}_i = X_p^d /_{U_f} U_p$

Estado inicial para $\widetilde{X}_{i+1} = X_p^d /_{U_t} U_p^+$

Sob estas hipóteses estaria errado escrever a equação (3.46). Pode ser provado (ver [VAN OVERSCHEE; DE MOOR, 1995]) que a diferença entre \hat{X}_i e \tilde{X}_{i+1} é zero quando se satisfaz um dos seguintes enunciados:

- 1. $i \rightarrow \infty$
- 2. O sistema é puramente determinístico
- 3. A entrada determinística u_k é ruído branco.

Quando um dos enunciados acima é satisfeito, então se pode substituir \hat{X}_i e \hat{X}_{i+1} por \widetilde{X}_i e \widetilde{X}_{i+1} , logo as matrizes do sistema são encontradas da equação (3.46).

Encontrando as matrizes do sistema via seqüência de estados \hat{X}_i

A sequência do filtro de Kalman \hat{X}_i está relacionada com a projeção ortogonal dada na equação (3.43):

$$Z_{i} \stackrel{def}{=} Y_{f} \left(\begin{pmatrix} W_{p} \\ U_{f} \end{pmatrix} \right) = \Gamma_{i} \hat{X}_{i} + H_{i}^{d} U_{f}$$
(3.47)

Analogamente, a sequência \hat{X}_{i+1} esta relacionada com:

$$Z_{i+1}^{def} = Y_f^{-} / \begin{pmatrix} W_p^+ \\ U_f^- \end{pmatrix} = \Gamma_{i-1} \hat{X}_{i+1} + H_{i-1}^d U_f^-$$
(3.48)

Os estados iniciais da sequência \hat{X}_i e \hat{X}_{i+1} são iguais e são dados por [VAN OVERSCHEE; DE MOOR, 1996]:

$$\hat{X}_0 = X_p^d / \begin{bmatrix} U_p \\ U_f \end{bmatrix}.$$

Considere o sistema inovação (3.1) (com $w_k = K_k e_k$, $v_k = e_k$) e seus estados estimados

$$\hat{X}_i \in \hat{X}_{i+1}$$
:

$$\hat{X}_{i+1} = A\hat{X}_i + BU_{i|i} + K_i e_i$$
(3.49)

$$Y_{i|i} = C\hat{X}_i + DU_{i|i} + e_i$$
(3.50)

As equações (3.49) e (3.50) podem ser expressas como:

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \hat{X}_i \\ U_{i|i} \end{pmatrix} + \begin{pmatrix} \rho w \\ \rho v \end{pmatrix}$$
(3.51)

onde ρw e ρv são matrizes residuais. Estas matrizes residuais são perpendiculares a \hat{X}_i e $U_{i|i}$, assim as matrizes do sistema podem ser encontradas da equação (3.51) através do método dos mínimos quadrados. Infelizmente, não é possível determinar as seqüências \hat{X}_i e \hat{X}_{i+1} diretamente dos dados de entrada – saída (pois não se tem um teorema que garanta isso, não se podendo então aplicar a solução via equação (3.46), pois como foi dito, os estados iniciais são diferentes).

No entanto, das equações (3.47) e (3.48), pode-se expressar \hat{X}_i e \hat{X}_{i+1} como:

$$\hat{X}_i = \Gamma_i^* \left[Z_i \ H_i^d U_f \right] \tag{3.52}$$

$$\hat{X}_{i+1} = \Gamma_{i-1}^* \begin{bmatrix} Z_{i+1} & H_{i-1}^d U_f^- \end{bmatrix}$$
(3.53)

Nas equações (3.52) e (3.53) os únicos termos desconhecidos são H_i^d e H_{i-1}^d . Z_i e Z_{i+1} são determinados pelas equações (3.47) e (3.48), respectivamente. U_f e U_f^- são matrizes dadas e por último o termo Γ_i^* pode ser calculado da equação (3.44) através da projeção oblíqua.

Das equações (3.51), (3.52) e (3.53) resulta:

/

$$\left(\frac{\Gamma_{i-1}^{*} Z_{i+1}}{Y_{i|i}}\right) = \left(\begin{array}{c}A\\C\\\\Termo 1\end{array}\right) \Gamma_{i}^{*} Z_{i} + \underbrace{T}_{\text{Termo 2}} U_{f} + \left(\begin{array}{c}\rho w\\\rho v\\\\Termo 3\end{array}\right)$$
(3.54)

onde:

$$T = \begin{pmatrix} \begin{bmatrix} B \middle| \Gamma_{i-1}^* H_{i-1}^d \end{smallmatrix} - A \Gamma_i^* H_i^d \\ \begin{bmatrix} D \middle| 0 \end{smallmatrix} - C \Gamma_i^* H_i^d \end{pmatrix}$$
(3.55)

Observar que as matrizes $B \in D$ aparecem linearmente na matriz T. Portanto a equação (3.54) pode ser solucionada através do problema dos mínimos quadrados para as matrizes $A, C \in T$.

Encontradas as matrizes *A* e *C*, as matrizes *B* e *D* podem ser encontradas da equação (3.55), ver [VAN OVERSCHEE; DE MOOR, 1996].

A seguir apresentam-se os teoremas que solucionam os objetivos i) e ii) citados acima.

3.3.2.1 Teorema 2. Filtro de Kalman para estado não-estacionário de sistema determinístico - estocástico

Suponha as seguintes hipóteses:

Dado: a estimativa de estado inicial \hat{x}_0 , a matriz de covariância da estimativa de estado inicial P_0 e as medições de entrada $u_0, ..., u_{k-1}$ e saída $y_0, ..., y_{k-1}$

Então: o estado estimado \hat{x}_k pelo filtro de Kalman de estado não-estacionário pode ser explicitamente escrito como:

$$\hat{x}_{k} = \left(A^{k} - \Omega_{k}\Gamma_{k}\right) \quad \Delta_{k}^{d} - \Omega_{k}H_{k}^{d} \quad \Omega_{k} \left(\begin{array}{c} \frac{\hat{x}_{0}}{u_{0}} \\ \cdots \\ \frac{u_{k-1}}{y_{0}} \\ \cdots \\ y_{k-1} \end{array}\right)$$

onde:

$$\Omega_{k} \stackrel{def}{=} \left(\Delta_{k}^{C} - A^{k} P_{0} \Gamma_{k}^{T} \right) \left(L_{k} - \Gamma_{k} P_{0} \Gamma_{k}^{T} \right)^{-1}$$

A prova deste teorema pode ser encontrada em [VAN OVERSCHEE; DE MOOR, 1996; SANTOS MIRANDA, 2004].

Este teorema mostra como os estados estimados pelo filtro de Kalman \hat{x}_k podem ser escritos como uma combinação linear das entradas passadas $u_0, ..., u_{k-1}$, saídas passadas $y_0, ..., y_{k-1}$ e do estado inicial estimado \hat{x}_0 . Assim, a seqüência de estados \hat{X}_i pode ser obtida por:

$$\hat{X}_{i} = \begin{bmatrix} \hat{x}_{i} & \hat{x}_{i+1} & \dots & \hat{x}_{i+j-1} \end{bmatrix} = \begin{bmatrix} A^{i} - \Omega_{i}\Gamma_{i} \\ & \Delta_{i}^{d} - \Omega_{i}H_{i}^{d} \\ & -\Omega_{i}H_{i}^{d} \\ & -\Omega_{i}\prod_{j=1}^{d} \begin{bmatrix} \hat{X}_{0} \\ & & \\ \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} A^{i} - \Omega_{i}\Gamma_{i} \\ & \Delta_{i}^{d} - \Omega_{i}H_{i}^{d} \\ & & \end{bmatrix} \begin{bmatrix} \hat{X}_{0} \\ & & \\ \end{bmatrix} \begin{bmatrix} \hat{X}_{0} \\ & & \\ \end{bmatrix}$$

$$(3.56)$$

onde \hat{X}_0 corresponde à sequência de estados iniciais.

Esta seqüência de estados é gerada por um banco de filtros de Kalman nãoestacionários, trabalhando em paralelo em cada uma das colunas da matriz em blocos de Hankel das entradas e saídas passadas W_p .

A sequência de estados do filtro de Kalman definida em (3.56) não é única. A sequência depende de como se escolhe a sequência de estados iniciais \hat{X}_0 .

Emprega-se a seguinte notação para indicar a seqüência de estados do filtro de Kalman com estado inicial \hat{X}_0 e matriz inicial P_0 :

$$\hat{X}_i = \hat{X}_{i\left[\hat{X}_0, P_0\right]}$$

3.3.2.2 Teorema 3. Projeção ortogonal

Suponha as seguintes hipóteses:

- 1. As entradas determinísticas u_k não estão correlacionadas com o ruído do processo w_k nem com o ruído de medição v_k .
- 2. A sequência de entradas u_k é persistentemente excitante de ordem 2*.i*.
- 3. O número de dados medidos tende para infinito $N \rightarrow \infty$.
- 4. O ruído de processo w_k e o ruído de medição v_k não são identicamente zero.

Tem-se então que:

$$Z_{i} \stackrel{def}{=} Y_{f} / \begin{pmatrix} W_{p} \\ U_{f} \end{pmatrix} = \Gamma_{i} \hat{X}_{i} + H_{i}^{d} U_{f}$$

com:

$$\hat{X}_i \stackrel{def}{=} \hat{X}_{i\left[\hat{X}_0, P_0\right]}$$

A prova deste teorema pode ser encontrada em [VAN OVERSCHEE; DE MOOR, 1996; SANTOS MIRANDA, 2004].

A importância do teorema 3 é que ele revela uma forma pela qual a seqüência de estados do filtro de Kalman \hat{X}_i se relaciona com os dados de entrada e saída. A matriz projetada Z_i pode ser encontrada a partir dos dados de entrada e saída, sem se conhecer as matrizes do sistema. A projeção Z_i pode ser considerada como uma predição ótima das saídas futuras Y_f , dadas as entradas e saídas passadas W_p e as entradas futuras U_f .

As fórmulas para a sequência do estado inicial \hat{X}_0 e a matriz inicial P_0 podem ser expressas da seguinte forma:

$$\begin{split} \hat{X}_{0} &= X_{p}^{d} / \begin{pmatrix} U_{p} \\ U_{f} \end{pmatrix} \\ P_{0} &= - \Phi_{\left[X_{p}^{d} / \begin{pmatrix} U_{p} \\ U_{f} \end{pmatrix}^{\perp}, X_{p}^{d} / \begin{pmatrix} U_{p} \\ U_{f} \end{pmatrix}^{\perp}\right]} \end{split}$$

Apresenta-se a seguir o teorema principal para o problema da identificação determinística - estocástica. Este teorema permite o cálculo do espaço de filas de uma seqüência de estados do filtro de Kalman \hat{X}_i e do espaço de colunas da matriz de observabilidade estendida Γ_i , diretamente dos dados de entrada e saída, sem conhecimento das matrizes do sistema. As matrizes do sistema podem ser recuperadas da seqüência de estados \hat{X}_i ou a partir da matriz de observabilidade estendida Γ_i . Uma descrição do procedimento de identificação combinada é apresentada na figura 3.7.



Figura 3.7 Descrição do procedimento de identificação determinística-estocástica por subespaços.

3.3.2.3 Teorema 4: identificação determinística-estocástica

Suponha as seguintes hipóteses:

- 1. A sequência de entradas u_k não está correlacionada com o ruído do processo w_k nem com o ruído de medição v_k .
- 2. A sequência de entradas u_k é persistentemente excitante de ordem 2*.i*.
- 3. O número de dados medidos tende para infinito $N \rightarrow \infty$.
- 4. O ruído do processo w_k e o ruído de medição v_k não são identicamente zero.
- 5. As matrizes de ponderação $W_1 \in \Re^{lixli}$ e $W_2 \in \Re^{jxj}$ definidas pelo usuário são tais que W_1 tem posto completo e W_2 obedece: posto $(W_p) = posto(W_pW_2)$.
- 6. Definindo-se a projeção oblíqua ϑ_i como:

$$\vartheta_i \stackrel{def}{=} Y_f /_{U_f} W_p$$

e a decomposição em valores singulares:

$$W_{1} \vartheta_{i} W_{2} = \begin{pmatrix} U_{1} & U_{2} \end{pmatrix} \begin{pmatrix} S_{1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_{1}^{T} \\ V_{2}^{T} \end{pmatrix} = U_{1} S_{1} V_{1}^{T}$$
(3.57)

Então tem-se que:

- 1. $\mathcal{G}_i = \Gamma_i \cdot \widetilde{X}_i$, com: $\hat{X}_i \stackrel{def}{=} \hat{X}_{i[\hat{X}_0, P_0]}$
- A ordem do sistema (3.1) é igual ao número de valores singulares presentes na equação (3.57) que sejam diferentes de zero.

3.
$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2} \cdot T$$

sendo $T \in \Re^{nxn}$ uma transformação de similaridade arbitrária e não-singular.

4. A parte da seqüência de estados \widetilde{X}_i que se encontra no espaço de colunas de W_2 pode ser recuperada a partir de: $\widetilde{X}_i W_2 = T^{-1} \cdot S_1^{1/2} V_1^T$

5. $\widetilde{X}_i = \Gamma_i^+ \cdot \mathcal{G}_i$

A prova deste teorema pode ser encontrada em [VAN OVERSCHEE; DE MOOR 1996] ou [SANTOS MIRANDA, 2004].

Comentários sobre este teorema:

- A hipótese 1 do teorema é satisfeita para sistemas em malha aberta. Em malha fechada o ruído poderá ser realimentado nas entradas através do regulador.
- A hipótese 2 diz que *u_k* deveria ter pelo menos *i* senóides com freqüências distintas.
- A hipótese 3 é necessária somente para uma análise assintótica. Na prática, é suficiente ter uma quantidade grande de dados.
- Pode-se notar que a seqüência de estados X
 _i recuperada deste teorema difere da seqüência de estados X
 _i introduzida no teorema 3, devido a seus estados iniciais X
 ₀ serem diferentes.

Pelo Teorema 3:

$$\hat{X}_0 = X_p^d / \begin{bmatrix} U_p \\ U_f \end{bmatrix}$$

Pelo Teorema 4:

$$\hat{X}_0 = X_p^d /_{U_f} U_p$$

A seguir apresentam-se os algoritmos que são consequências dos teoremas 3 e 4 [VAN OVERSCHEE; DE MOOR, 1996].

Algoritmo 1 N4SID

(Empregando seqüência de estados \widetilde{X}_i)

7. Calcule as seguintes projeções oblíquas:

$$\begin{split} \mathcal{G}_i = Y_f \mid_{U_f} W_p \\ \mathcal{G}_{i+1} = Y_f^- \mid_{U_f^-} W_p^+ \end{split}$$

8. Calcule a SVD da projeção oblíqua ponderada:

 $W_1 \mathcal{G}_i W_2 = USV^T$

- Determine a ordem do sistema por inspeção dos valores singulares em S e particione a SVD para obter U₁ e S₁.
- 10. Determine $\Gamma_i \in \Gamma_{i-1}$ como :

 $\Gamma_i = W_1^{-1} U_1 S_1^{1/2}, \ \ \Gamma_{i-1} = \underline{\Gamma_i}$

onde $\underline{\Gamma_i}$ denota a matriz Γ_i sem as *l* filas.

11. Determine as seqüências de estados:

$$\begin{split} \widetilde{X}_i &= \Gamma_i^+ \cdot \mathcal{G}_i \\ \widetilde{X}_{i+1} &= \Gamma_{i-1}^+ \cdot \mathcal{G}_{i+1} \end{split}$$

12. Solucione o conjunto de equações lineares para encontrar A, B, C e D:

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}$$

13. Determine Q, $S \in R$ dos resíduos, como:

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = E_j \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \cdot \begin{pmatrix} \rho_w^T & \rho_v^T \end{bmatrix}$$

Algoritmo 2 N4SID

(Empregando seqüência de estados \hat{X}_i)

1. Calcule as projeções ortogonal e oblíqua de:

$$\mathcal{G}_{i} = Y_{f} / U_{f} W_{p} \qquad Z_{i} = Y_{f} / \begin{pmatrix} W_{p} \\ U_{f} \end{pmatrix} \qquad Z_{i+1} = Y_{f}^{-} / \begin{pmatrix} W_{p}^{+} \\ U_{f}^{-} \end{pmatrix}$$

- 2. Calcule a **SVD** da projeção oblíqua ponderada: $W_1 \mathcal{G}_i W_2 = USV^T$
- Determine a ordem do sistema por inspeção dos valores singulares em S e particione a SVD para obter U₁ e S₁.
- 4. Determine $\Gamma_i \in \Gamma_{i-1}$ como :

$$\Gamma_i = W_1^{-1} U_1 S_1^{1/2}, \quad \Gamma_{i-1} = \underline{\Gamma_i}$$

onde $\underline{\Gamma_i}$ denota a matriz Γ_i sem as *l* filas.

5. Solucione o seguinte conjunto de equações lineares para se obter A, C e K :

$$\left(\frac{\Gamma_{i-1}^{+} \cdot Z_{i+1}}{Y_{i|i}}\right) = \begin{pmatrix} A \\ C \end{pmatrix} \cdot \Gamma_{i}^{+} \cdot Z_{i} + K \cdot U_{f} + \left(\frac{\rho_{w}}{\rho_{v}}\right)$$

6. Determine *B* e *D* de *K*, *A*, *C*, Γ_i , Γ_{i-1} , através da seguinte equação:

$$\begin{pmatrix} K_{1|1} \\ K_{1|2} \\ \vdots \\ \frac{K_{1|i}}{K_{2|1}} \\ K_{2|2} \\ \vdots \\ K_{2|i} \end{pmatrix} = N \begin{pmatrix} D \\ B \end{pmatrix}$$

com:

$$N = \begin{bmatrix} -L_{1|1} & M_1 - L_{1|2} & \dots & M_{i-2} - L_{1|i-1} & M_{i-1} - L_{1|i} \\ M_1 - L_{1|2} & M_2 - L_{1|3} & \dots & M_{i-1} - L_{1|i} & 0 \\ M_2 - L_{1|3} & M_3 - L_{1|4} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \frac{M_{i-1} - L_{1|i} & 0 & \dots & 0 & 0}{I_i - L_{2|1} & -L_{2|2} & \dots & -L_{2|i-1} & -L_{2|i} \\ -L_{2|2} & -L_{2|3} & \dots & -L_{2|i} & 0 \\ -L_{2|3} & -L_{2|4} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -L_{2|i} & 0 & \dots & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_i & 0 \\ 0 & \Gamma_{i-1} \end{bmatrix}$$

7. Determine Q, $S \in R$ dos resíduos, como:

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = E_j \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \cdot (\rho_w^T & \rho_v^T) \end{bmatrix}$$

Estes algoritmos estão implementados no apêndice F.

Comentários

1. O método N4SID usa a projeção oblíqua das "futuras" saídas (Y_f) dentro das entradas e saídas passadas (W_p) , ao longo das "futuras" entradas (U_f) :

$$\mathcal{G}_i = Y_f /_{U_f} W_p$$
.

Desta projeção se obtém a matriz de seqüência de estados, que serve para calcular as matrizes do sistema (A, B, C, D), pelo método dos mínimos quadrados. Os resíduos do problema dos mínimos quadrados são usados para calcular as matrizes de covariância de ruído e uma estimativa do ganho de Kalman pode ser encontrada.

- 2. A matriz de Hankel dos dados de entrada e saída é da forma $[U_p^T U_f^T Y_p^T Y_f^T]^T$.
- 3. No método POMOESP a matriz de Hankel dos dados de entrada e saída é $[U_f^T U_p^T Y_p^T Y_f^T]^T$.
- 4. O método POMOESP usa as entradas e saídas passadas como uma variável instrumental, para dar uma estimativa consistente das matrizes (A, B, C, D). Isto é

feito por meio de uma projeção ortogonal executada por uma fatorização LQ e SVD. Desta projeção se recupera uma estimativa da matriz de observabilidade estendida Γ_i , da qual as matrizes A e C podem ser estimadas. Segundo, pode-se encontrar uma estimativa da matriz de Toeplitz H_i , da qual as matrizes B e D podem ser estimadas, como uma solução do problema de mínimos quadrados.

 A forma de encontrar as matrizes do sistema dos algoritmos MOESP e N4SID é mostrada na figura 3.8



Figura 3.8 Algoritmos de identificação MOESP e N4SID.

 Existem outros métodos de identificação por subespaços, por exemplo, o método Canonical Variate Analysis CVA [LARIMORE, 1983, 1990]. Estes três métodos MOESP, N4SID e CVA foram unificados em um só teorema em [VAN OVERSCHEE; DE MOOR, 1995].

Inspirado nos métodos MOESP e N4SID, a seguir apresenta-se o método MON4SID, o qual é uma mistura dos métodos MOESP e N4SID.

CAPÍTULO 4

ALGORITMOS PROPOSTOS

Com base no algoritmo MOESP estudado no capítulo 2 e no algoritmo N4SID estudado no capítulo 3, se apresenta o método alternativo MON4SID que combina os métodos MOESP e N4SID.

4.1 Algoritmo proposto: MON4SID

O algoritmo proposto determinam as matrizes $A \in C$ de forma similar ao método MOESP, logo, se calcula a nova matriz de observabilidade estendida Γ_i . As matrizes B,

D, *Q*, *R* e *S* são estimadas em função dos estados do sistema $\hat{X}_i(\tilde{X}_i)$, através das projeções oblíquas, de forma similar aos algoritmos 3 e 4 do algoritmo N4SID.

A forma de ordenar LQ (usada pelo método MOESP) ou QR (usada pelo método N4SID) é diferente para ambos os métodos (ver comentários 2 e 3).

O método MON4SID usa a fatoração LQ, portanto é preciso encontrar as projeções oblíquas a partir desta.

Sejam as matrizes arbitrárias A, B e C com fatoração LQ, dadas por:

$$\begin{bmatrix} B \\ C \\ A \end{bmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix}$$
(4.1)

Afirma-se que a projeção oblíqua $A/_B C$ é dada por:

$$A/_{B}C = L_{32} L_{22}^{*} \begin{bmatrix} L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_{1} \\ Q_{2} \end{bmatrix}$$
(4.2)

Para tal, primeiro se projeta a projeção ortogonal do espaço fila da matriz A sobre o

espaço fila formado pelas matrizes $\begin{bmatrix} B \\ C \end{bmatrix}$, isto é (ver apêndice B):

$$A / \begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
(4.3)

Como foi mencionado anteriormente, a projeção ortogonal também pode ser expressa como a combinação linear dos espaços fila de *B* e *C*:

$$A / \begin{bmatrix} B \\ C \end{bmatrix} = L_b B + L_c C \tag{4.4}$$

Estes espaços fila são as projeções oblíquas da matriz A, isto é:

- $A/_{C}B = L_{b}B$ é a projeção oblíqua de A em B sobre C
- $A/_{B}C = L_{c}C$ é a projeção oblíqua de A em C sobre B

Para se calcular a projeção oblíqua de A em C sobre B, parte-se da relação (4.1) e se obtém:

$$B = L_{11}Q_1 \quad e \quad C = L_{21}Q_1 + L_{22}Q_2 \tag{4.5}$$

Substituindo-se na equação (4.4) a equação (4.5), resulta:

$$A / \begin{bmatrix} B \\ C \end{bmatrix} = L_b B + L_c C = \begin{bmatrix} L_b & L_c \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
(4.6)

Igualando-se as equações (4.3) e (4.5), se obtém:

$$\begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} L_b & L_c \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
$$\begin{bmatrix} L_{31} & L_{32} \end{bmatrix} = \begin{bmatrix} L_b L_{11} + L_b L_{21} & L_c L_{22} \end{bmatrix}$$
(4.7)

Da equação (4.7) resulta:

$$L_{c}L_{22} = L_{32}$$

$$L_{c} = L_{32}L_{22}^{*}$$
(4.8)

Portanto, a projeção oblíqua $A/_B C = L_c C$ é dada por:

$$A/_{B}C = L_{32}L^{*}_{22}C$$

$$A/_{B}C = L_{32}L^{*}_{22}[L_{21} \quad L_{22}].$$
(4.9)

Fazendo uso da equação (4.9), um novo método é apresentado.

4.1.1 Método MON4SID 1 para sistema determinístico usando estados

Este método soluciona o problema apresentado na seção 3.3.1. O método usa a técnica MOESP para calcular a matriz de observabilidade estendida Γ_i e o método N4SID é aplicado para encontrar as matrizes (*A*,*B*,*C*,*D*) através do problema de mínimos quadrados.

A matriz de observabilidade estendida Γ_i pode ser derivada pela simples fatoração LQ de uma matriz construída a partir das matrizes em blocos de Hankel $[U_f^T U_p^T Y_f^T Y_f^T]^T$, na forma:

$$\begin{bmatrix} U_f \\ W_p \\ Y_f \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$
(4.10)

Como foi visto no capitulo 2, uma estimativa da matriz Γ_i , pode ser computada pela SVD da matriz L_{32} , isto é:

$$L_{32} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_n & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U S V^T$$
(4.11)

A ordem *n* do sistema (3.15) é igual ao número de valores singulares mais significativos em *S*. O espaço coluna de U_1 é uma aproximação consistente para Γ_i [VERHAEGEN, 1992a], isto é:

$$\Gamma_i = U_1$$

É claro que o sistema (3.15) pode ser reescrito como:

$$\begin{pmatrix} X_{i+1}^{d} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_{i}^{d} \\ U_{i|i} \end{pmatrix}$$
(4.12)

Logo, para computar as matrizes do sistema (A,B,C,D) é necessário conhecer as matrizes da seqüência dos estados X_i^d e X_{i+1}^d , as quais são dadas pelas equações (3.28)

e (3.30), isto é:

$$X_i^d = \Gamma_i^* \cdot \mathcal{G}_i \tag{4.13}$$

$$X_{i+1}^{d} = \Gamma_{i-1}^{+} \vartheta_{i-1} \tag{4.14}$$

onde as projeções oblíquas $\mathcal{G}_i = Y_f /_{U_f} W_p$ e $\mathcal{G}_{i+1} = Y_f^- /_{U_f^-} W_p^+$ podem ser computadas através da equação (4.9).

Este método pode ser resumido no seguinte algoritmo:

Algoritmo determinístico MON4SID 1

(empregando estados)

- 7. Construir as matrizes de Hankel $U_f, U_p \in Y_f, Y_p$.
- 8. Calcular a fatoração LQ, equação (4.10).
- 9. Calcular a SVD da matriz L_{32} .
- 10. Determinar a ordem do sistema por inspeção dos valores singulares em S e particionar a SVD para obter U_1 .
- 11. Determinar $\Gamma_i \in \Gamma_{i-1}$ como:

$$\Gamma_i = U_1 \qquad \qquad \Gamma_{i-1} = \underline{\Gamma_i}$$

onde Γ_i denota a matriz Γ_i sem as *l* filas (número de saídas) passadas de Γ_i .

- 12. Determine X_i^d e X_{i+1}^d pelas equações (4.13) e (4.14).
- 13. Solucionar o seguinte conjunto de equações lineares para obter *A*, *B*, *C* e *D*:

$$\begin{pmatrix} X_{i+1}^{d} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_{i}^{d} \\ U_{i|i} \end{pmatrix}$$

O algoritmo 2 para sistema determinístico desenvolvido pelo método N4SID, permite apresentar um outro algoritmo, MON4SID 2.

4.1.2 Método MON4SID 2 para sistema determinístico usando matriz de observabilidade estendida

Este método soluciona o problema apresentado na seção 3.3.1. O método usa a técnica MOESP para calcular a matriz de observabilidade estendida Γ_i da qual se obtêm as matrizes *A* e *C*, e o método N4SID é aplicado para computar as matrizes *B* e *D* através do problema de mínimos quadrados.

A matriz de observabilidade estendida Γ_i é computada da mesma forma que no método MON4SID 1, logo as matrizes *A* e *C* podem ser calculadas através de:

$$\Gamma_i(1:l(i-1),1:n)A = \Gamma_i(l+1:il,1:n)$$
(4.15)

$$C = \Gamma_i(1:l,1:n) \tag{4.16}$$

Por último, as matrizes *B* e *D* são computadas da mesma forma que no algoritmo 2 para sistema determinístico N4SID.

Este método pode ser resumido no seguinte algoritmo:

Algoritmo determinístico MON4SID 2 (empregando a matriz de observabilidade estendida)

- 1. Construir as matrizes de Hankel $U_f, U_p \in Y_f, Y_p$.
- 2. Calcular a fatoração LQ, equação (4.10).
- 3. Calcular a SVD da matriz L_{32} .
- 4. Determinar a ordem do sistema por inspeção dos valores singulares em S e particionar a SVD para obter U_1 e U_2 .
- 5. Determinar $\Gamma_i \in \Gamma_i^{\perp}$ como:

$$\Gamma_i = U_1$$
 $\Gamma_i^{\perp} = U_2$

- 6. Determine a matriz do sistema *A* a partir de Γ_i como $A = \underline{\Gamma_i^+} \overline{\Gamma_i}$, onde $\overline{\Gamma_i}$ denota a matriz Γ_i sem as *l* (número de saídas) primeiras filas.
- 7. Determine C como as primeiras l filas de Γ_i

com:

$$\begin{pmatrix} M_1 & M_2 & \dots & M_i \end{pmatrix} = \Gamma_i^{\perp} \cdot Y_f \cdot U_f^+$$
$$\begin{pmatrix} L_1 & L_2 & \dots & L_i \end{pmatrix} = \Gamma_i^{\perp}$$

calcular $B \in D$ de:

$$\begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_i \end{pmatrix} = \begin{pmatrix} L_1 & L_2 & \dots & L_{i-1} & L_i \\ L_2 & L_3 & \dots & L_i & 0 \\ \dots & \dots & \dots & \dots & \dots \\ L_i & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_l & 0 \\ 0 & \underline{\Gamma_i} \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix}.$$

A diferença entre o algoritmo determinístico MON4SID 2 e o algoritmo determinístico 2 N4SID é que o primeiro faz uso da projeção ortogonal e o segundo da projeção oblíqua. A forma de computar a matriz de observabilidade estendida é diferente para ambos os algoritmos, isto é, no algoritmo MON4SID 2 não é necessário computar $S_1^{1/2}$.

Usando o algoritmo MON4SID 1 para sistemas determinísticos é apresentado outro algoritmo, o MON4SID 3, o qual computa uma seqüência de estados totais para depois computar as matrizes do sistema (A,B,C,D).

4.1.3 Método MON4SID 3 para sistema determinístico usando estados totais

Este método soluciona o problema apresentado na seção 3.3.1. O método usa a técnica MOESP para calcular a matriz de observabilidade estendida Γ_i e o método N4SID é aplicado para computar as matrizes (*A*,*B*,*C*,*D*) através do problema dos mínimos quadrados.

A matriz de observabilidade estendida Γ_i é computada da mesma forma que no método MON4SID 1.

Como foi visto anteriormente, para computar as matrizes do sistema (A,B,C,D) a partir da equação (4.12) é necessário conhecer as matrizes da seqüência dos estados X_i^d e X_{i+1}^d , portanto o problema agora é computar as sequências de estados. A projeção oblíqua \mathcal{G}_i dada por:

$$\mathcal{G}_i = Y_f /_{U_f} W_p \tag{4.17}$$

ela pode ser computada a partir de (4.9), isto é:

$$\mathcal{G}_{i} = Y_{f} / _{U_{f}} W_{p} = L_{32} L^{*}_{22} [L_{21} \quad L_{22}] \begin{pmatrix} Q_{1} \\ Q_{2} \end{pmatrix}$$
(4.19)

Uma estimativa da seqüência de estados totais é dada por:

$$X = (\Gamma_i)^* L_{32} L_{22} [L_{21} \quad L_{22}] W_p$$
(4.20)

Logo uma estimativa das matrizes X_i^d e X_{i+1}^d é dada através de:

$$X_i^d = [x_k \ x_{k+1} \ x_{k+2} \dots \ x_{k+N-2}]$$
(4.21)

$$X_{i+1}^{d} = [x_{k+1} \ x_{k+2} \ x_{k+3} \dots \ x_{k+N-1}]$$
(4.22)

Assim, o sistema de matrizes (A,B,C,D) pode ser determinado pela equação (4.12).

Este método pode ser resumido no seguinte algoritmo:

Algoritmo determinístico MON4SID 3

(empregando estados totais)

- 1. Construir as matrizes de Hankel $U_f, U_p \in Y_f, Y_p$.
- 2. Calcular a fatoração LQ, equação (4.10).
- 3. Calcular a SVD da matriz L_{32} .
- 4. Determinar a ordem do sistema por inspeção dos valores singulares em S e particionar a SVD para obter U_1 .
- 5. Determinar Γ_i como: $\Gamma_i = U_1$
- 6. Determinar X pela equações (4.18) e as seqüências de estados X_i^d e X_{i+1}^d .
- 7. Solucionar o seguinte conjunto de equações lineares para obter A, B, C e D:

$$\begin{pmatrix} X_{i+1}^{d} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X_{i}^{d} \\ U_{i|i} \end{pmatrix}$$

A diferença entre o algoritmo determinístico 1 e algoritmo determinístico 3 é que este último não precisa computar duas projeções ortogonais (isto é \mathcal{P}_i e \mathcal{P}_{i+1}) para estimar as seqüências de estados passados e futuros, estas duas seqüências de estados são computadas a partir de uma seqüência de estados totais, a qual tem uma só condição inicial, evitando problemas de polarização.

Sistemas determinísticos não simulam sistemas reais, pois estes sempre se encontram perturbados por ruídos. A seguir se apresentam dois algoritmos para identificação determinística – estocástica.

4.2 Método MON4SID 4 para sistema determinístico - estocástico

Este método soluciona o problema apresentado na seção 3.3.2 O método usa a técnica do método POMOESP para calcular a matriz de observabilidade estendida Γ_i e o método N4SID é aplicado para computar as matrizes (*A*,*B*,*C*,*D*) através do problema de mínimos quadrados.

Este método é uma extensão do método MON4SID 1. Uma estimativa das matrizes de covariância do processo e do ruído de medição são obtidas através da equação (3.8).

Este método pode ser resumido no seguinte algoritmo:

Algoritmo determinístico estocástico MON4SID 4

- 1. Compute os passos de 1 até 6 do algoritmo MON4SID 1.
- 2. Solucione o conjunto de equações lineares para encontrar *A*, *B*, *C* e *D*:

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_{i} \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \rho_{w} \\ \rho_{v} \end{bmatrix}$$

3. Determine Q, $S \in R$ dos resíduos, como:

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = E_j \begin{bmatrix} \rho_w \\ \rho_v \end{pmatrix} \cdot \begin{pmatrix} \rho_w^T & \rho_v^T \end{pmatrix}$$

O algoritmo MON4SID 3 para sistema determinístico apresentado anteriormente,

permite apresentar um outro algoritmo para o caso determinístico - estocástico.

4.2.1 Método MON4SID 5 para sistema determinístico - estocástico

Este método soluciona o problema apresentado na sessão 3.2.3. Este método é uma extensão do método MON4SID 3. Uma estimativa das matrizes de covariância do processo e do ruído de medição são obtidas através da equação (3.8).

Este método pode ser resumido no seguinte algoritmo:

Algoritmo determinístico estocástico MON4SID 5

- 1. Compute os passos de 1 até 6 do algoritmo MON4SID 3.
- 2. Solucione o conjunto de equações lineares para encontrar *A*, *B*, *C* e *D*:

 $\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_{i} \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \rho_{w} \\ \rho_{v} \end{bmatrix}$

3. Determine Q, $S \in R$ dos resíduos, como:

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = E_j \left[\begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \cdot \begin{pmatrix} \rho_w^T & \rho_v^T \end{pmatrix} \right].$$

CAPÍTULO 5

SIMULAÇÃO DOS ALGORITMOS DETERMINÍSTICOS E ESTOCÁSTICOS

Na modelagem fenomenológica do processo, o modelo é desenvolvido a partir de toda informação disponível sobre o processo. Por outro lado, métodos de identificação normalmente não pressupõem qualquer conhecimento prévio do sistema, justificando o nome "identificação caixa-preta". Os algoritmos apresentados nos capítulos 2, 3 e 4 são aqui usados para fazer identificação determinística e estocástica do tipo caixa preta, conforme representado na figura 5.1



Figura 5.1 Modelo dos sistemas caixa preta

Nos modelos caixa - preta não se conhece o sistema, então fornecidos os dados de entrada e saída é preciso encontrar um modelo que represente tal sistema. Os dados de entrada e saída são gerados a partir de três tipos de modelos implementados em Matlab:

* **Modelo em espaço de estados**: estes modelos são dados pelas equações (2.17) e (2.48) para sistemas determinísticos e determinístico – estocásticos, respectivamente. Eles são determinados caso se conheçam as matrizes do sistema, as quais podem ser dadas ou geradas na forma aleatória. São considerados aqui os seguintes modelos em espaço de estados:

- Modelo em espaços de estados com matrizes aleatórias, sistema MIMO.
- Modelo em espaços de estados, sistema SISO, usado por [VERHAEGEN, 1993;
 VAN OVERSCHEE; DE MOOR, 1996].

* Modelo de um processo *benchmark* da Shell, sistema MIMO, [COTT, 1995; ZHU, 1997]

A justificativa para usar estes modelos é:

- Se a ordem de um sistema for conhecida, o modelo obtido tem a mesma ordem.
- Nos sistemas de controle é muito usado o diagrama de Bode para desenhar um controlador digital, portanto o modelo encontrado através dos algoritmos de identificação tem que responder às mesmas freqüências do sistema real.

5.1 Identificação determinística

Os algoritmos para identificação determinística citados nos capítulos 2, 3 e 4 forem implementados no Matlab 6.5, onde N4sid1 denota aqui o algoritmo determinístico 1 (empregando estados), N4sid2 denota aqui o algoritmo determinístico 2 (empregando matriz de observabilidade estendida), MOESP denota aqui o algoritmo MOESP (caso determinístico). Os algoritmos propostos para sistemas determinísticos são: MNS1 denota aqui o algoritmo MON4SID 1 (empregando estados), MNS2 denota aqui o algoritmo MON4SID 2 (empregando a matriz de observabilidade estendida) e MNS3 denota aqui o algoritmo MON4SID 3.

Com o objetivo de ver o desempenho destes algoritmos, foram comparados com os algoritmos N4SID, MPVerh e PEM. O N4SID se encontra implementado no Toolbox do Matlab, MPVerh denota aqui o método MOESP implementado por Michael Verhaegen.

No total foram usados 9 algoritmos para fazer a identificação determinística e os resultados são apresentados através de tabelas.

Comparar a simulação do modelo obtido com dados medidos é provavelmente a forma mais usual de se validar um modelo. Nesse caso, deseja-se saber se o modelo reproduz ao longo do tempo os dados observados. Para a validação, **não se deve usar os dados utilizados para obter o modelo**.

Para se avaliar a qualidade do modelo, aplicam-se indicadores de desempenho. Os indicadores de desempenho mais usados são média relativa do erro quadrático (MRSE), média da variância relativa (MVAF) e o FIT, os quais são definidos como:

$$MRSE(\%) = \frac{1}{l} \cdot \sum_{i=1}^{N} \sqrt{\frac{\sum_{j=1}^{N} (y - \hat{y})^2}{\sum_{j=1}^{N} (y)^2}} \cdot 100$$
$$MVAF(\%) = \frac{1}{l} \cdot \sum_{i=1}^{N} \left(1 - \frac{\operatorname{var}(y - \hat{y})}{\operatorname{var}(y)}\right) \cdot 100$$
$$FIT(\%) = \operatorname{norma}\left(\frac{(y - \hat{y})^2}{\sqrt{N}}\right) \cdot 100$$

onde y é a saída real e \hat{y} é a saída estimada pelo modelo obtido. O índice *MRSE* é muito usado na literatura e o índice *MVAF* é usado pelo SMI toolbox. O índice *FIT* é usado pela função "compare" no Matlab. Estes índices de desempenho são empregados para se avaliar a qualidade do modelo produzido por cada algoritmo.

A identificação neste caso é feita para sistemas puramente determinísticos e sistemas com ruídos de baixa e alta intensidade, os quais não ingressam no processo, mas são considerados ruídos de medição.

5.1.1 Identificação puramente determinística

Um sistema puramente determinístico é apresentado na figura 5.2, onde u representa os dados de entrada e y representa os dados de saída.



Figura 5.2 Sistema puramente determinístico

A seguir apresenta-se, em tabelas, a implementação dos sistemas determinísticos usados para coletar os dados, os quais foram implementados através de funções do Matlab 6.5.

5.1.1.1 Modelo em espaço de estados com matrizes aleatórias

A função "drss" do Matlab 6.5 permite gerar um modelo discreto em espaços de estados, com matrizes do sistema (A,B,C,D) na forma aleatória. A ordem escolhida do modelo é 4, que é igual ao posto (A). Gerou-se um sistema MIMO de duas entradas e

duas saídas. Isto é mostrado na tabela 1.

				Tabela	1
Modelo em	espaço de	estados	gerado	na forma	aleatória

% M = idss(drss(Ordem do modelo, N° de saídas, N° de entradas))				
M = idss(drss(4,2,2));	% modelo de 4 ordem			
[A,B,C,D]=th2ss(M);	% recuperando as matrizes do sistema			

As matrizes do sistema são dadas por:

$$A = \begin{pmatrix} 0.2075 & 0.6037 & 0.0467 & 0.3296 \\ 0.0647 & -0.1354 & 0.8093 & 0.2532 \\ -0.6549 & 0.2657 & -0.1127 & 0.4057 \\ -0.2056 & -0.5368 & -0.0829 & 0.3546 \end{pmatrix} \qquad B = \begin{pmatrix} 0.1542 & 0 \\ -0.3429 & -0.9224 \\ -0.4785 & 1.5295 \\ 0.5955 & -0.0031 \end{pmatrix}$$
$$C = \begin{pmatrix} -0.2141 & 0 & -0.9223 & 0 \\ -2.4532 & 0 & 0.0617 & -0.2908 \end{pmatrix} \qquad D = \begin{pmatrix} 0 & 2.1346 \\ -0.5508 & 0.5446 \end{pmatrix}$$

As matrizes dadas acima foram usadas para coletar os dados de saída, os quais foram salvos para realizar testes com os outros algoritmos, pois do contrário, cada vez que se computar o modelo M, geram-se diferentes matrizes do sistema. Este modelo é observável e controlável.

Obtidas as matrizes do sistema, o passo seguinte é coletar os dados de entrada e saída do sistema determinístico.

O sinal de entrada para o sistema M é um sinal PRBS, o qual é persistentemente excitante e o sinal de saída é gerado através da função "sim" do Matlab, isto é mostrado na tabela 2.

Gerando o sinal de entrada e saída					
% U = IDINPUT(N,TYPE,BAND,LEV	/ELS)				
ut =idinput([1000, 2],'prbs',[0 0.3]);	% sinal de entrada				
yt = sim(M,ut)	% sinal de saída				

Tabela 2 Gerando o sinal de entrada e saída

Para o modelo M com sinal de entrada *ut*, foram coletados 1000 dados, dos quais 700 foram aplicados para identificação e o restante para validação.

Os sinais pré-tratados (feito pelo comando "detrend" do Matlab) usados na identificação são mostrados nas figuras 5.3 e 5.4.



Figura 5.3 Sinal de saída y_1 e entrada u_1 usados na identificação do modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)



Figura 5.4 Sinal de saída *y*2 e entrada *u*2 usados na identificação do modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)

O passo seguinte é encontrar o melhor modelo que simule o processo M, para os algoritmos citados acima. Isto é mostrado na tabela 3. A ordem n = 6 mostrada na figura 5.5 é dada pelos valores singulares mais significativos da matriz *S*, a qual é obtida da SVD de certas projeções oblíquas ou ortogonais, dependendo do algoritmo sendo aplicação. Ver algoritmos do capítulo 2 e 3.


Figura 5.5 Valores singulares para o modelo em espaço de estados com matrizes aleatórias (identificação puramente determinístico)

Tabela 3
Resultados numéricos do desempenho dos algoritmos para o modelo em espaço de
estados com matrizes aleatórias (caso puramente determinístico)

Algoritmog	Tampa (a)	300 dados de validação Yv			
Algoritmos	Tempo (s)	FIT %	MRSE %	MVAF %	
N4sid1	0.328	100	14.25*10 ⁻¹³	100	
N4sid2	0.328	100	9.252*10 ⁻¹³	100	
MOESP	0.266	100	4.946*10 ⁻¹³	100	
MNS1	0.268	100	9.881*10 ⁻¹³	100	
MNS2	0.343	100	$2.808*10^{-13}$	100	
MNS3	0.406	100	$1.07*10^{-13}$	100	
MPVerh	0.297	100	$1.88*10^{-13}$	100	
N4sidMatlab	0.844	99.97	0.0274	100	
PEM	1.844	99.975	0.0274	100	

Analisando os valores da tabela 3, todos os modelos tiveram um bom desempenho em termos de validação. Verifica-se que o tempo de processamento para obtenção do modelo é menor para MOESP, MNS1 e MPVerh. Com o objetivo de visualizar o desempenho do algoritmo proposto, optou-se pelo método MNS1 para identificar o processo.

A figura 5.6 mostra as saídas do processo real e aquelas geradas pelo modelo determinístico identificado (linha vermelha). Pode-se observar que o modelo

identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas.



Figura 5.6 Comparação das respostas do processo real (linha continua) versus modelo (linha pontilhada) em espaço de estados com matrizes aleatórias (caso puramente determinístico)

A figura 5.7 mostra a saída simulada versus a saída real, gerada pela função compare do Matlab 6.5, a qual mostra o valor do indicador FIT.



Figura 5.7 Comparação através do FIT das respostas do processo real versus modelo em espaço de estados com matrizes aleatórias (caso puramente determinístico)

Uma comparação da resposta em freqüência do modelo real com o modelo estimado foi feita para poder determinar se o comportamento de entrada-saída está suficientemente caracterizado, isto é, se a dinâmica do sistema fica suficientemente bem caracterizada.



A figura 5.8 mostra o diagrama de Bode do processo simulado e do processo real.

Figura 5.8 Comparação da resposta em freqüência do processo versus modelo em espaço de estados com matrizes aleatórias (identificação puramente determinístico)

Da figura 5.8 pode-se observar que a resposta em freqüência do modelo obtido é similar à do sistema real, mas se é conhecida a ordem do sistema real, esta não é necessariamente a mesma para o modelo obtido, portanto as matrizes do sistema real e do modelo obtido têm diferentes dimensões. As matrizes do modelo obtido são dadas por:



5.1.1.2 Modelo em espaço de estados usado por [VERHAEGEN, 1993] e [VAN OVERSCHEE; DE MOOR, 1996]

Em [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996] é citado um exemplo para fazer a identificação em malha fechada de uma planta e seu controlador. Esta planta representa um sistema SISO e é usada neste caso para fazer a identificação em malha aberta. As matrizes do sistema que simulam a planta são dadas por:

a = [4.40 1 0 0 0; -8.09 0 1 0 0; 7.83 0 0 1 0; -4 0 0 0 1; 0.86 0 0 0 0];

b = [0.00098; 0.01299; 0.01859; 0.0033; -0.00002];

$$c = [1 \ 0 \ 0 \ 0];$$

Este mesmo exemplo será usado na identificação em malha fechada nos capítulos posteriores.

Dadas as matrizes do sistema, o passo seguinte é coletar os dados de entrada e saída. Isto é mostrado na tabela 4.

Tabela 4

Gerando os dados de entrada e saída

N = 1000;	% numero de dados a coletar
ny = 1	% numero de saídas
u1 =idinput([N, ny],'prbs',[0 0.3]);	
ut = [u1-mean(u1)]	% dados coletados
yt = dlsim(a,b,c,d,ut);	% saidas coletadas

Para este modelo com sinal de entrada u1, foram coletados 1000 dados, dois quais 700 se aplicaram para identificação e o restante para validação.



Os sinais pré-tratados usados na identificação são mostrados na figura 5.9

Figura 5.9 Sinal de saída e entrada usados na identificação do modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (caso puramente determinístico)

O passo seguinte é encontrar o melhor modelo que simule o processo, conforme mostrado na tabela 5. A ordem n = 7 do sistema é dada pelos valores singulares mais significativos, como mostra a figura 5.10.



Ordem do sistema (n)

Figura 5.10 Valores singulares para o modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (identificação puramente determinístico)

Tabela 5 Resultados numéricos do desempenho dos algoritmos para o modelo em espaço de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (identificação puramente determinístico)

Algoritmos	Tompo	300 dados de validação Yv			
Algorithios	rempo	FIT %	MRSE %	MVAF %	
N4sid1	0.234	100	0.0039	100	
N4sid2	0.203	100	0.0046	100	
MOESP	0.203	100	22962*10 ⁻¹⁰	100	
MNS1	0.203	100	79482*10 ⁻¹⁰	100	
MNS2	0.218	100	11719*10 ⁻¹⁰	100	
MNS3	0.203	100	9629800*10 ⁻¹⁰	100	
MPVerh	0.25	100	9.9797*10 ⁻¹⁰	100	
N4sidMatlab	0.359	100	0.0031	100	
PEM	1.187	100	0.0044	100	

Analisando-se os valores da tabela 5, todos os modelos tiveram um bom desempenho em termos de validação.Verifica-se que o tempo de processamento para obtenção do modelo é menor para N4sid2, MOESP, MNS1 e MNS3. Com o objetivo de visualizar o desempenho do algoritmo proposto, optou-se pelo método MNS1 para identificar o processo. A figura 5.11 mostra as saídas do processo real e aquelas geradas pelo modelo determinístico identificado (linha verde). Pode-se observar que o modelo identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas.



Figura 5.11 Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) em espaço de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (caso puramente determinístico)

Foi feita uma comparação da resposta em freqüência do sistema real com o modelo estimado para poder determinar se o comportamento de entrada-saída está suficientemente caracterizado, isto é, se a dinâmica do sistema fica suficientemente bem caracterizada. A figura 5.12 mostra o diagrama de Bode do processo simulado e do processo re



Figura 5.12 Comparação da resposta em freqüência do processo versus modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996] (identificação puramente determinístico)

Da figura 5.12 pode-se observar que o modelo obtido tem uma resposta em freqüência similar à do modelo real.

As matrizes do modelo obtido são dadas por:

$$A = \begin{bmatrix} 1.0042 & -0.3298 & -0.1163 & 0.3923 & 0.4458 & 1.2675 \\ 0.0264 & 0.9657 & 0.7354 & -0.1746 & -0.9898 & -1.6335 \\ -0.0008 & -0.0203 & 0.9270 & -0.8898 & -0.5309 & -2.1901 \\ -0.0001 & -0.0013 & 0.0101 & 0.7757 & 1.3582 & 1.4488 \\ -0.0000 & -0.0004 & -0.0120 & -0.2565 & 0.7279 & 1.8932 \\ 0.0000 & 0.0000 & -0.0000 & 0.0001 & -0.0008 & 0.9970 \end{bmatrix}$$
$$B = \begin{bmatrix} -3.4418 \\ 3.0566 \\ 1.0346 \\ 0.0351 \\ 0.0794 \\ 0.0000 \end{bmatrix}$$
$$C = \begin{bmatrix} -0.2851 & -0.5013 & 0.5450 & 0.4153 & -0.3297 & 0.2598 \end{bmatrix}$$
$$D = \begin{bmatrix} -2.3451e-008 \end{bmatrix}$$

5.1.1.3 Modelo de um processo benchmark da Shell

O processo *benchmark* da Shell [COTT, 1995; ZHU, 1997] é um modelo de uma coluna de destilação com duas entradas e duas saídas.

As entradas do sistema são:	As saídas do sistema são:
D = fluxo de vapor destilado	<i>P</i> = pressão da coluna
Q = carga do refervedor	X = impureza do produto

O modelo da pressão é dado por:

$$P(t) = \frac{-0.6096 + 0.4022q^{-1}}{1 - 1.5298q^{-1} + 0.5740q^{-2}} D(t) + \frac{-0.1055 - 0.0918q^{-1}}{1 - 1.5298q^{-1} + 0.5740q^{-2}} Q(t) + \frac{Ns}{1 - 1.5945q^{-1} + 0.59445q^{-2}} e_p(t)$$

onde $e_p(t)$ é ruído branco que gera perturbações na pressão *P*. O parâmetro *Ns* é usado para o nível de ruído na simulação. Para este caso considera-se Ns = 0.

O modelo da impureza (X) é ligeiramente não linear e é dado por:

$$X(t) = 0.0765 \frac{500000}{Q(t-7) - 1500} + 0.9235X(t-1) + \frac{Ns}{1 - 1.6595q^{-1} + 0.06595q^{-2}} e_x(t)$$

onde $e_x(t)$ é ruído branco que gera perturbações na composição X.

O modelo *benchmark* da Shell implementado no Simulink do Matlab 6.5 é mostrado na figura 5.13.



Figura 5.13 Modelo *benchmark* da Shell com entradas D, Q e saídas P, X.

O processo trabalha em torno dos valores nominais: $D_nom = 20$; $Q_nom = 2500$; P nom = 2800; X nom = 500.

Foram coletados 1000 dados, dos quais 700 se aplicaram para a identificação e o restante para validação.

A ordem n = 11 do sistema é dada pelos valores singulares mais significativos, como mostra a figura 5.14.



Figura 5.14 Valores singulares para o modelo *benchmark* da Shell (identificação determinístico)

O passo seguinte é encontrar o melhor modelo que simule o processo, conforme mostrado na tabela 6.

		· · · ·		,			
	Algoritmos	Tempo300 dados de validação Yv					
Algorithios		(s)	FIT %	MRSE %	MVAF %		
	N4sid1	0.281	100	4.5733*10 ⁻¹⁰	100		
	N4sid2	0.265	100	1.5466*10 ⁻¹⁰	100		
	MOESP	0.188	100	$1.4778*10^{-10}$	100		
	MNS1	0.281	100	3.6195**10 ⁻⁴	100		
	MNS2	0.312	100	0.0046	100		
	MNS3	0.437	100	4.6522*10 ⁻¹¹	100		
	MPVerh	0.297	100	$1.8744*10^{-10}$	100		
	N4sidMatlab	2.329	100	6.8220*10 ⁻⁵	100		
	PEM	7.782	74.75	11.7262	88.5114		

Tabela 6 Resultados numéricos do desempenho dos algoritmos para o modelo *benchmark* da Shell (caso determinístico)

Analisando-se os valores da tabela 6, o método PEM teve o pior desempenho em termos de validação. Verifica-se que o tempo de processamento para obtenção do modelo é menor para o algoritmo MOESP, motivo pelo qual optou-se por este método para identificar o processo, conforme mostrado na figura 5.15.



Figura 5.15 Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) *benchmark* da Shell (caso determinístico)

A figura 5.15 mostra as saídas do processo real e aquelas geradas pelo modelo MOESP determinístico identificado (linha pontilhada). Pode-se observar que o modelo identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas. As matrizes do modelo obtido são dadas por:

-										~
0.981:	0.2289	-0.0198	0.0007	0.0168	0.1709	-0.1317	-0.1049	0.0340	-0.0392	0.0004
0.0042	2 0.9397	-0.0017	-0.0534	-0.3749	-0.2269	0.2243	0.1833	-0.0467	0.0099	-0.1824
0.001	5 -0.0033	0.8950	-0.3620	-0.0085	0.0674	0.0073	-0.0320	0.0124	0.0233	0.0696
0.0002	-0.0002	0.0157	0.6371	-0.2714	0.1768	0.0830	-0.0391	0.0146	0.0980	0.2135
0.000	-0.0001	-0.0005	0.0244	0.9610	-0.3552	0.3003	0.1704	-0.0424	0.0750	0.0012
A= -0.000	0.0001	-0.0009	0.0277	-0.0264	0.6692	0.7192	0.0698	0.0595	0.0981	0.0289
-0.000	0.0001	-0.0002	0.0143	-0.0672	-0.3715	0.3030	-0.6545	0.3011	0.0400	0.4495
-0.000	0.0000	0.0001	-0.0043	0.0061	0.0113	0.2408	-0.4915	-0.4476	-0.4393	-0.3259
-0.000	0.0000	-0.0000	0.0008	0.0007	0.0084	-0.0717	-0.2813	-0.5549	0.7387	-0.1429
0.000	-0.0000	-0.0000	0.0005	-0.0023	-0.0106	0.0073	0.2266	-0.6139	-0.2795	0.6946
0.000	0.0000 0	0.0000	-0.0005	0.0003	-0.0009	0.0521	0.0935	0.0810	0.4034	0.3928
\mathcal{C}		$\overline{)}$								
(0.01	0.04	28								
-0.09	70 -0.06	543								
-1.06	06 -0.95	560								
_0.07	25 0.25									
	15 0.23	12								
$\mathbf{B} = \begin{bmatrix} -0.00 \\ 0.00 \end{bmatrix}$	15 - 0.01	13								
D ⁻ -0.00	29 -0.01	02								
0.00	06 -0.01	22								
0.00	0.00	71								
-0.00	00 -0.00)54								
0.00)2 -0.00	34								
	$\frac{12}{12}$ 0.00	07								

$$C = \begin{bmatrix} -0.0083 & 0.0462 & 0.4490 & 0.6861 & 0.1508 & -0.1258 & -0.0790 & 0.0327 & -0.0179 & -0.0683 & -0.1447 \\ -0.2962 & 0.4683 & -0.0499 & 0.0068 & 0.3721 & 0.4790 & -0.4064 & -0.3334 & 0.0965 & -0.0648 & 0.1808 \end{bmatrix}$$
$$D = \begin{bmatrix} -0.6096 & -0.1055 \\ -0.0000 & 0.0000 \end{bmatrix}$$

A coleta de dados do processo real está sempre alterada por ruídos. Sistemas puramente determinísticos têm interesse apenas acadêmicos. Então se agregaram ruído de processo para os dados de entrada e saída. Neste caso, deseja-se saber como funciona o método de identificação por subespaços para o modelo dado em 5.1.1.1.

5.1.2 Identificação com ruído de baixa (3%) e alta (80%) intensidade para o modelo dado em 5.1.1.1

Sistemas com ruído na medição são mostrados na figura 5.16.



Figura 5.16 Sistema com ruído na medição

Observe-se que o ruído não afeta a planta, somente afeta os sinais coletados de entradas e saídas.

A forma como foram coletados os dados com ruídos de baixa (3%) e alta (80%) intensidade para o sistema dado em 5.1.1.1 é mostrada na tabela 7.

Tabela 7 Dados coletados com ruído

Ruídos de baixa intensidade (3%)	Ruídos de alta intensidade (80%)
[nu,N]=size(u);	[nu,N]=size(u);
Urt=[u+0.03*std(u)*randn(N,nu)];	Urt=[u+0.8*std(u)*randn(N,nu)];
[ny,N]=size(y);	[ny,N]=size(y);
Yrt=[y+0.03*std(y)*randn(N,ny)];	Yrt=[y+0.8*std(y)*randn(N,ny)];

Os sinais de entrada usados na validação para o sistema com ruídos de baixa (3%) e alta (80%) intensidade são mostrados nas figuras 5.17 e 5.18.



Figura 5.17 Dados de entrada com 3% de ruído para o modelo em espaço de estados com matrizes aleatórias.



Figura 5.18 Dados de entrada com 80% de ruído para o modelo em espaço de estados com matrizes aleatórias.

Os sinais de saída usados na validação para o sistema com ruídos de baixa (3%) e alta (80%) intensidade são mostrados nas figuras 5.19 e 5.20. Para poder visualizar o efeito do ruído na saída são amostrados 140 pontos dos dados de validação.



Figura 5.19 Dados de saída com 3% de ruído para o modelo em espaço de estados com matrizes aleatórias.



Figura 5.20 Dados de saída com 80% de ruído para o modelo em espaço de estados com matrizes aleatórias.

Quando o sistema esta perturbado por ruído, muitas vezes, encontrar a ordem n do sistema a partir da decomposição de valores singulares não é fácil, uma alternativa é

dada pelo critério de Akaike. Isto será mostrado com maior detalhe na seção 5.1.3. Neste este caso n = 4 foi obtida pelo critério de Akaike.

Os resultados da simulação são mostrados nas tabelas 8 e 9.

Tabela 8 Resultados numéricos do desempenho dos algoritmos com ruídos de baixa intensidade (3%) aplicada ao modelo em espaço de estados com matrizes aleatórias.

Algoritmos	Tempo (s)	300 dados de validação Yv			
		FIT %	MRSE %	MVAF %	
N4sid1	0.266	95.24	4.75	99.79	
N4sid2	0.250	95.21	4.78	99.79	
MOESP	0.266	95.18	4.82	99.78	
MNS1	0.328	95.32	4.67	99.80	
MNS2	0.359	95.18	4.81	99.78	
MNS3	0.328	95.24	4.76	99.79	
MPVerh	0.281	95.15	4.84	99.78	
N4sidMatlab	1.328	95.21	4.79	99.79	
PEM	1.734	95.22	4.78	99.79	

Analisando-se os valores da tabela 8, o modelo MNS1 é melhor em termos de validação. Verifica-se que o tempo de processamento é menor para o algoritmo N4sid2.

Tabela 9

Resultados numéricos do desempenho dos algoritmos com ruídos de alta intensidade (80%) aplicada ao modelo em espaço de estados com matrizes aleatórias.

Algoritmos	Tompo (s)	300	dados de validaçã	o Yv
	Tempo (s)	FIT %	MRSE %	MVAF %
N4sid1	0.422	21.64	78.3	38.59
N4sid2	0.406	21.90	78.04	38.99
MOESP	0.265	16.6	83.31	30.52
MNS1	0.516	21.44	78.50	38.27
MNS2	0.563	21.86	78.08	38.93
MNS3	0.609	21.59	78.35	38.50
MPVerh	0.203	21.37	78.58	38.15
N4sidMatlab	0.735	21.08	78.8	37.69
PEM	1.828	21.03	78.91	37.62

Analisando-se os valores da tabela 9, o modelo N4sid2 é melhor em termos de validação, verifica-se que o tempo de processamento para obtenção do modelo é menor para o algoritmo MPVerh.

Dos resultados das tabelas 8 e 9 pode-se observar que os valores para todos os algoritmos são quase iguais, pois o ruído não ingressa no processo.

5.2 Identificação determinístico – estocástica

Os algoritmos combinados determinístico e estocástico simulam melhor os processos reais. Foram considerados 9 algoritmos onde N4sidE denota o algoritmo 1 N4SID (empregando seqüência de estados \tilde{X}_i), N4sidO denota o algoritmo 2 N4SID (empregando seqüência de estados \hat{X}_i), POMOESP denota o algoritmo POMOESP dado no capítulo 2. Os algoritmos propostos nesta tese para sistemas determinísticos – estocásticos são: MNSE1 denota o algoritmo MON4SID 4 (empregando estados), MNSE2 denota aqui o algoritmo MN4SID 5, CCA denota o algoritmo de [KATAYAMA, 2005] para sistemas determinísticos estocásticos.

Com o objetivo de ver o desempenho destes algoritmos, foram comparados com os algoritmos N4SIDMatlab, MOESPV e PEM. O N4SIDMatlab se encontra implementado no Toolbox do Matlab, MOESPV denota o método MOESP implementado por Michael Verhaegen.

Os resultados destes algoritmos para fazer a identificação determinístico – estocástica são apresentados através de tabelas.

Para avaliar os algoritmos consideram-se três exemplos:

- a) Exemplo apresentado por Overschee e De Moor, sistema MIMO [OVERSCHEE; DE MOOR, 1996].
- b) Modelo em espaços de estados, sistema SISO, usado por [VERHAEGEN, 1993; VAN OVERSCHEE, DE MOOR, 1996].
- c) Modelo de um processo *benchmark* da Shell, sistema MIMO, [COTT, 1995; ZHU, 1997].

Estes modelos são usados para coletar os dados de entrada e saída, os quais são aplicados na identificação determinístico – estocástica. Os resultados destes algoritmos

são comparados através dos indicadores de desempenho FIT, MRSE e MVAF, os quais são descritos no item 5.1.

5.2.1 Exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]

Este exemplo simula um sistema inovativo da forma:

$$x_{k+1} = ax_{k} + bu_{k} + K(e_{k})$$

$$y_{k} = cx_{k} + du_{k} + e_{k}$$
com
$$E[e_{k} (e_{s}^{T})] = r$$
(5.2)

onde as matrizes do sistema são dadas por:

 $a = [0.603 \ 0.603 \ 0 \ 0; -0.603 \ 0.603 \ 0 \ 0; 0 \ 0 \ -0.603 \ -0.603; 0 \ 0 \ 0.603 \ -0.603];$ $b = [1.1650, -0.6965; 0.6268 \ 1.6961; 0.0751, 0.0591; 0.3516 \ 1.7971];$ c = [0.2641, -1.4462, 1.2460, 0.5774; 0.8717, -0.7012, -0.6390, -0.3600];d = [-0.1356, -1.2704; -1.3493, 0.9846];

A matriz *K* é o ganho de Kalman, sendo dado por:

K = [0.1242, -0.0895; -0.0828, -0.0128; 0.0390, -0.0968; -0.0225, 0.1459]*4;

 e_k na equação (5.1) é ruído branco com média zero e variância 1, tendo matriz de covariância r = [0.0176,-0.0267;-0.0267,0.0497].

A equação (5.2) diz que a matriz de covariância do ruído é *r*, o problema é encontrar um ruído que satisfaça (5.2). Como *r* é uma matriz positiva então, pela fatoração de Cholesk, *r* pode ser decomposta em duas matrizes da forma: $r = R^T R$, onde *R* é uma matriz triangular superior.

Define-se e_k (ruído procurado) como: $e_k = \xi_k R^T$, onde ξ_k é ruído branco com média zero e variância 1, assim e_k é um ruído com média zero, variância 1 e matriz de covariância r. Isto pode ser facilmente comprovado pelo Matlab, através do comando $cov(\xi) = r$. Devido à linearidade do sistema, a saída da equação (5.1) pode ser expressa como a soma de um sistema determinístico mais um sistema estocástico, ver figura 5.21.



Figura 5.21 Sistema determinístico – estocástico na forma inovativa

Na figura 5.21 os blocos da planta determinística e estocástica são determinados pelos sistemas:

Sistema determinístico	Sistema estocástico
$x_{k+1}^d = A x_k^d + B u_k$	$x_{k+1}^s = A x_k^s + K e_k$
$y_k^d = Cx_k^d + Du_k$	$y_k^s = Cx_k^s + e_k$

Dadas as matrizes do sistema (5.1), o passo seguinte é coletar os dados de entrada e saída. O sinal de entrada para este sistema é um sinal PRBS, o qual é persistentemente excitante. O sinal de saída para o sistema determinístico mais estocástico da figura 5.21 é gerado através da função "DLSIM" do Matlab, isto é mostrado nas tabelas 10 e 11.

 Tabela 10

 Sinal de saída para o sistema determinístico - estocástica

Sistema	Sistema estocástico			
N=2000	% N dados coletados			
u1 = prbs(N,0.1);	% gerando a entrada 1			
u2 = prbs(N,0.1);	% gerando a entrada 2	L=2;	% numero	de saídas
ut = [u1-mean(u1)]	u2-mean(u2)]; % entrada	e = rand(N,L	a)*chol(r);	% entrada
yd = dlsim(a,b,c,d,	ut); % saída	ys = dlsim(a,	k,c,eye(L),e);	% saída

 Tabela 11

 Coletando os dados de entrada e saída para o sistema determinístico - estocástica

saída determinística mais estocástica
6 Tratamento dos dados
Dados para identificar
dados para validar

Para o sistema (5.1) com sinal de entrada *ut*, foram coletados 2000 dados, dos quais 1500 se aplicaram para identificação e o restante para validação.

Os sinais pré-tratados usados para a identificação são mostrados nas figuras 5.22 e 5.23.



Figura 5.22 Sinal de saída y1 e entrada u1 usado para identificação do exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]



Figura 5.23 Sinal de saída y2 e sinal de entrada u2 usado para identificação do exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]

A ordem do sistema n = 4 é dada pelos valores singulares mais significativos, como mostra a figura 5.24.



Figura 5.24 Valores singulares do exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]

O passo seguinte é encontrar o melhor modelo que simule o processo, para os algoritmos citados em 5.1.2. Isto é mostrado na tabela 12.

Tabela 12 Resultados numéricos do desempenho dos algoritmos para o exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]

Algoritmos	Tempo (s)	300 dados de validação Yv			
		FIT %	MRSE %	MVAF %	
N4sidE	0.344	94.0518	3.0911	99.8854	
N4sidO	0.391	94.0780	3.0399	99.8880	
POMOESP	0.562	94.0606	3.0736	99.8864	
MNSE1	0.500	94.0814	3.0405	99.8881	
MNSE2	0.578	94.0789	3.0396	99.8879	
CCA	0.516	94.1175	2.9600	99.8904	
MOESPV	1.532	94.0404	3.2039	99.8809	
N4sidMatlab	2.500	94.06	3.4033	99.8708	
PEM	4.265	94.1145	2.9930	99.8890	

Analisando-se os valores da tabela 12, os métodos CCA e PEM tiveram melhor desempenho em termos de validação. Verifica-se que o tempo de processamento para obtenção do modelo é menor para o algoritmo MNS1E. Optou-se pelo método CCA para identificar o processo, porque apresentou melhor desempenho em termos de validação.

A figura 5.25 mostra as saídas do processo real e aquelas geradas pelo modelo identificado (linha vermelha). Pode-se observar que o modelo identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas.



Figura 5.25 Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) para o exemplo apresentado por [OVERSCHEE; DE MOOR, 1996]

Foi feita uma comparação da resposta em freqüência do modelo real com o modelo estimado para poder determinar se a dinâmica do sistema fica suficientemente bem caracterizada. A figura 5.26 mostra o diagrama de Bode do processo simulado e do processo real.



Figura 5.26 Comparação da resposta em freqüência do processo versus modelo (exemplo apresentado por [OVERSCHEE; DE MOOR, 1996])

Da figura 5.12 pode-se observar que a resposta em freqüência do modelo obtido é similar à do modelo real.

5.2.2 Modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]

[VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996] citam como exemplo para fazer a identificação em malha fechada uma planta e seu controlador. Esta planta representa um sistema SISO e é usada neste caso para fazer a identificação em malha aberta de um sistema em espaço de estados na forma inovativa. As matrizes do sistema que simulam a planta são dadas por:

 $a = [4.40\ 1\ 0\ 0\ 0;\ -8.09\ 0\ 1\ 0\ 0;\ 7.83\ 0\ 0\ 1\ 0;\ -4\ 0\ 0\ 0\ 1;\ 0.86\ 0\ 0\ 0\ 0];$ $b = [0.00098\ ;\ 0.01299\ ;\ 0.01859\ ;\ 0.0033\ ;\ -0.00002];$ $c = [1\ 0\ 0\ 0\ 0];$ d=[0];

k=[2.3;-6.64;7.515;-4.0146;0.86336];

$$E[e_k(e_s^T)] = 1/9;$$

Dadas as matrizes do sistema, o passo seguinte é coletar os dados de entrada e saída. O sinal de entrada é uma seqüência de ruído branco com variância 1. Foram coletados 1200 pontos, dois quais 1000 se aplicaram para a identificação e 200 para a validação.

Os sinais pré-tratados usados para identificação são mostrados na figura 5.27.



Figura 5.27 Sinal de saída *y* e sinal de entrada *u* usado na identificação do modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]

O passo seguinte é encontrar o melhor modelo que simule o processo. Para validar o modelo foram usados três critérios de desempenho FIT, MRSE e MVAF, os quais são mostrados na tabela 13. A ordem n = 7 do sistema é dada pelos valores singulares mais significativos.

Com o objetivo de comparar o desempenho de um algoritmo determinístico aplicado a um processo determinístico – estocástico, foi usado o algoritmo N4sid1, conforme é mostrado no final da tabela 13.

Algoritmos		200 dados de validação Yv		
	Tempo (s)	FIT %	MRSE %	MVAF %
N4sidE	0.265	62.2119	0.8599	99.9754
N4sidO	0.250	61.7476	0.8581	99.9755
POMOESP	0.391	62.3795	0.8602	99.9754
MNSE1	0.234	76.2849	0.9596	99.9693
MNSE2	0.281	62.4434	0.8575	99.9755
CCA	0.265	75.8276	0.9026	99.9728
MOESPV	0.422	66.3063	1.6087	99.9182
N4sidMatb	0.766	62.3949	0.8583	99.9755
PEM	1.782	68.2044	0.8248	99.9776
N4sid1	0.218	61.75	37.9048	83.2161

Tabela 13 Resultados numéricos do desempenho dos algoritmos para o modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]

Analisando-se os valores da tabela 13, todos os modelos tiveram um bom desempenho em termos de validação, exceto o algoritmo N4sid1. Isto é uma desvantagem para este algoritmo, que foi implementado somente para sistemas determinísticos. Verifica-se que o tempo de processamento para obtenção do modelo é menor para MNSE1 e N4sidO. Com o objetivo de visualizar o desempenho do algoritmo proposto, optou-se pelo método MNSE1 para identificar o processo.

A figura 5.28 mostra as saídas do processo real e aquelas geradas pelo modelo identificado (linha verde). Pode-se observar que o modelo identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas.



Figura 5.28 Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]

Foi feita uma comparação da resposta em freqüência do processo real com o modelo estimado para poder determinar se a dinâmica do sistema fica suficientemente bem caracterizada. A figura 5.29 mostra o diagrama de Bode do processo simulado e do processo real.



Figura 5.29 Comparação da resposta em freqüência do processo real versus modelo usado por [VERHAEGEN, 1993; VAN OVERSCHEE; DE MOOR, 1996]

Da figura 5.29 pode-se observar que o modelo obtido para baixas freqüências é bom, para altas freqüências se afastam.

5.2.3 Modelo de um processo *benchmark* da Shell

O sinal de entrada é uma seqüência PRBS. Com Ns = 0.5, o qual é chamado de 50% de ruído na pressão. Este ruído tem desvio padrão 1,231. Foram coletados 1200 pontos, dois quais 1000 se aplicaram para a identificação e 200 para validação.

Os sinais pré-tratados usados para identificação são mostrados nas figuras 5.30 e 5.31.



Figura 5.30 Sinal de saída y1 e entrada u1 usados para identificação do modelo de um processo *benchmark* da Shell (com 50% de ruído)



Figura 5.31 Sinal de saída y2 e entrada u2 usados na identificação do modelo de um processo *benchmark* da Shell (com 50% de ruído)

Tentar encontrar a ordem do sistema através dos valores singulares mais significativos, muitas vezes não é possível, como mostra a figura 5.32.



Figura 5.32 Valores singulares do modelo de um processo *benchmark* da Shell (com 50% de ruído)

Outro procedimento é selecionar o valor *n* que minimize os erros de estimação, técnica usada por algoritmos como o PEM, o que requer maior esforço computacional. Existe um critério estatístico que pode ajudar a obter a ordem do modelo, Critério de Informação de Akaike (AIC), definido em [AKAIKE, 1973].

 $AIC(n) = \eta . \ln[\sigma_{error}^2(n)] + 4P_n$

onde η é o número de dados usados na identificação, $\sigma_{error}^2(n)$ é a variância do erro de modelagem para um modelo de ordem $n \operatorname{com} P_n$ parâmetros.

O índice AIC(n) normalmente atinge um mínimo para um determinado número de parâmetros no modelo.

A aplicação do critério *AIC* ao modelo *benchmark* da Shell é mostrada na figura 5.33, onde se pode observar que o valor mínimo ocorre para n=7.



Figura 5.33 Espectro de valores AIC do modelo de um processo *benchmark* da Shell (com 50% de ruído)

O passo seguinte é encontrar o melhor modelo que simule o processo, conforme é mostrado na tabela 14.

Algoritmos	Tempo (s)	200 dados de validação Yv			
		FIT %	MRSE %	MVAF %	
N4sidE	0.578	21.4518	3.9489	99.5655	
N4sidO	0.547	26.9055	3.7871	99.5931	
POMOESP	0.610	3.0649	3.9925	99.5602	
MNSE1	0.766	27.7873	3.7932	99.5895	
MNSE2	0.812	29.5887	3.8393	99.5814	
CCA	0.765	12.2782	3.5051	99.6450	
MOESPV	0.359	4.7341	4.5739	99.4010	
N4sidMatlab	0.812	22.5534	3.5225	99.6430	
PEM	4.625	20.4207	3.4315	99.6644	

Tabela 14 Resultados numéricos do desempenho dos algoritmos para o modelo de um processo *benchmark* da Shell (com 50% de ruído)

Analisando-se os valores da tabela 14, o modelo CCA é o melhor em termos de validação dado pelo critério MVAF. Verifica-se que o tempo de processamento para obtenção do modelo é menor para o modelo MOESPV. Optou-se pelo método CCA para identificar o processo.

A figura 5.34 mostra as saídas do processo real e aquelas geradas pelo modelo identificado (linha azul). Pode-se observar que o modelo identificado reproduz muito bem as principais características do processo. Foram consideradas condições iniciais nulas.



Figura 5.34 Comparação das respostas do processo real (linha contínua) versus modelo (linha pontilhada) de um processo *benchmark* da Shell (com 50% de ruído)

Um processo que opere em malha aberta pode representar um problema. Por exemplo, sistemas instáveis em malha aberta podem ser de operação indesejável ou inconcebível em casos reais, e ainda a presença de integradores no sistema em malha aberta pode representar uma dificuldade para operação sem controle por um período longo de tempo. Além disso, por questões operacionais, muitas vezes não é possível interromper o funcionamento do processo com controlador para realizar o procedimento de identificação.

Um sistema operando em malha fechada é apresentado na figura 5.35. Este sistema é usado em [VAN OVERSCHEE; DE MOOR, 1996] para fazer identificação em malha fechada.

As matrizes do sistema são definidas em 5.1.3.

As matrizes do controlador são dadas por:

Ac=[2.65 -3.11 1.75 -0.39; 1 0 0 0; 0 1 0 0; 0 0 1 0];

Bc=[1; 0; 0; 0];

Cc=[-0.4135 0.8629 -0.7625 0.2521];

Dc=[0.61];



Figura 5.35 Sistema em malha fechada do modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996]

O sinal de referência é uma seqüência de ruído branco com variância 1. Foram coletados 1200 pontos, dos quais 1000 se aplicaram para a identificação. Para fazer a identificação em malha aberta foi usado o modelo N4sidE. A aplicação do critério AIC neste modelo é mostrada na figura 5.36, onde se pode observar que o valor mínimo ocorre para n = 6.



Figura 5.36 Espectro de valores AIC do modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996]

Foi feita uma comparação da resposta em freqüência do modelo real com o modelo estimado. A figura 5.37 mostra o diagrama de Bode dos processos simulado e real.



Figura 5.37 Comparação da resposta em freqüência do processo real versus modelo em espaços de estados usado por [VERHAEGEN, 1993; OVERSCHEE, DE MOOR, 1996]

Da figura 5.37 pode-se concluir que o modelo N4sidE não caracteriza a dinâmica do sistema, portanto aplicar métodos de identificação em malha aberta para identificar processos em malha fechada não é recomendado.

No capítulo seguinte se estuda identificação em malha fechada.

CAPÍTULO 6

IDENTIFICAÇÃO DE SISTEMAS POR SUBESPAÇOS EM MALHA FECHADA

Neste capítulo se lida com a identificação em malha fechada para os métodos em espaço de estados de tempo discreto que foram tratados nos capítulo 2 e 3, onde foram aplicados para malha aberta.

6.1 Introdução

A identificação em malha fechada é de interesse especial para aplicações em casos práticos na indústria. Freqüentemente, é necessário realizar os experimentos de identificação em malha fechada (com realimentação). As razões podem ser que a planta não seja estável em malha aberta, ou pelo fato que a planta deva ser controlada por razões de produção, de segurança ou econômicas.

Vários algoritmos de identificação por subespaços operando em sistemas com realimentação forem apresentados nas últimas décadas. Pode-se citar: [VAN DER KLAUW, *et al*, 1991; VERHAEGEN, 1993; LJUNG; MCKELVEY, 1996; VAN OVERSCHEE; DE MOOR, 1997; CHOU; VERHAEGEN, 1999]. Trabalhos mais recentes são apresentados em [CHIUSO; PICCI, 2002; QIN; LJUNG, 2003; JANSSON, 2003; CHIUSO; PICCI, 2003; LIN, *et al*, 2004; HUANG, *et al*, 2005; KATAYAMA; *et al*, 2005].

Um dos problemas na identificação em malha fechada é referente à correlação entre os dados de entrada e o ruído não medido [SÖDERSTROM; STOICA, 1989; LJUNG, 1999]. Um dos métodos tradicionais na identificação em malha fechada é o Método de Predição de Erro (PEM). Um estudo nesta área é feito por Forssell e Ljung (1999).

Baseado em suas análises, os métodos de identificação em malha fechada podem ser categorizados em três grandes grupos:

Método direto: Se aplica o método PEM usando os sinais de entrada e saída como se o sistema estivesse em malha aberta, sem considerar o sinal de referência para encontrar o modelo. O método funciona independentemente da complexidade do controlador; não necessita de algoritmos ou softwares específicos; sistemas instáveis ou com restrições podem ser tratados sem problemas, desde que o sistema seja estável em malha fechada.

Método indireto: Identificar o sistema em malha fechada a partir do sinal de referência (considerado como sinal de entrada) e do sinal de saída e fazendo uso do conhecimento do controlador, estimar o modelo da planta. A vantagem deste método é que se pode aplicar qualquer método de identificação, pois é um problema em malha aberta. O maior problema é que qualquer erro no conhecimento do controlador se reflete na estimativa do modelo da planta. Não funciona se o controlador apresentar não linearidades.

Método conjunto de entrada e saída: considera tanto os sinais de entrada e saída, como um sistema comandado pelo sinal de referência. A partir destes dados estimar o modelo da planta e o controlador. Fornece estimativa consistente do sistema, independente dos modelos de ruído usados, desde que o controlador tenha realimentação com uma determinada estrutura linear.

Existem outros métodos mais recentes, como a parametrização tailor-made, o método de dois estágios ou a parametrização Youla dual, descrito extensivamente em Van den Hof (2004) e o método das projeções, descrito por Forsell e Ljung (1999).

A vantagem do método PEM é que a convergência e variância assintótica dos resultados estão disponíveis, para maiores detalhes ver [LJUNG, 1978; LJUNG, 1985]; os quais são importantes para aplicações de identificação para controle [HJALMARSSON, 2003]. A desvantagem do método PEM é que ele envolve um complicado conjunto de parâmetros, motivo pelo qual torna-se difícil para identificar modelos de múltiplas entrada e múltiplas saídas (MIMO).

Para solucionar este problema surgem o métodos de identificação por subespaços (SIM), entre os quais pode-se citar: análise da variável canônica (CVA) [LARIMORE, 1990], MOESP [VERHAEGEN; DEWILDE, 1992] e N4SID [OVERSCHEE ; DE MOOR, 1994]. Estes 3 métodos forem unificados num só teorema proposto por Van Overschee e de Moor (1995). Em 1996 Jansson e Wahlberg propuseram um novo método de SIM. Baseado no teorema da unificação, estes métodos podem ser interpretados como a decomposição dos valores singulares de uma matriz peso.

Propriedades estatísticas, tais como consistência e eficiência dos algoritmos foram pesquisadas por [JANSSON; WAHLBERG, 1998; KNUDSEN, 2001; GUSTAVSSON, 2002; LJUNG, 2003; BAUER, 2003].

Como o SIM é uma técnica nova, há muitas dificuldades, citadas em [QUIN; LJUNG, 2003], que propuseram o método SIM parcimonioso (PARSIMs) para aplicação em malha aberta. Na tese de doutorado de um de seus orientados, Weilu Lin (2005) propõe o método SIM parcimonioso (PARSIMs) aplicado a malha fechada.

Verhaegen (1993) propõe um SIM para malha fechada, mas o modelo encontrado é de ordem elevada, assim tem que se reduzir o modelo para obter a representação em espaço de estados da planta e do controlador. Este processo de redução da ordem introduz uma carga computacional extra. Ljung e McKelvey (1996) pesquisaram o SIM através da realização clássica e propõem uma aproximação recursiva baseado no modelo ARX para a identificação em malha fechada. A desvantagem é que a parametrização do modelo ARX não é aplicada para um modelo qualquer, que não tenha estrutura ARX. Em 1997 Van Overschee e De Moor apresentaram o modelo SIM aplicado a um sistema em malha fechada.

6.2 Método N4SIDC

Nesta secção apresenta-se o problema de identificação em malha fechada, aplicando o método N4SIDC. Este surge como uma extensão do método N4SID aplicado para malha aberta.

6.2.1 Modelo em espaço de estados em malha fechada

Um modelo em espaço de estados em malha fechada é mostrado na figura 6.1 [VAN OVERSCHEE; DE MOOR, 1997].



Figura 6.1 Modelo em malha fechada

Na figura 6.1 u(k) é o sinal de entrada, y(k) é o sinal de saída e r(k) é o sinal de referência. v(k) é o ruído não medido e w(k) é o ruído de medição. São distúrbios que atuam na planta linear P(z). O controlador linear é representado por $\Im(z)$.

6.2.2 Problema de identificação em malha fechada

Considerar $u(k) \in \Re^m$, $y(k) \in \Re^l$ e $r(k) \in \Re^m$ como os sinais de entrada, saída e referência gerados pelo sistema em malha fechada da figura 6.1. Os sinais são interconectados através dos seguintes sistemas de espaço de estados.

a) Equações da planta: são descritas pela seguinte equação de diferenças:

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$
(6.1a)

$$y(k) = Cx(k) + Du(k) + v(k)$$
 (6.1b)

com

$$E\left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq} \ge 0$$

A matriz de transferência para o caso puramente determinístico é dada por: $P(z) = D + C(Iz - A)^{-1}B \in \Re^{lxm}$
Da mesma forma que no capítulo 3, o sistema (6.1) é partido em dois subsistemas: parte determinística e parte estocástica.

b) Equações do controlador: são descritas pela seguinte equação de diferenças:

$$x^{c}(k+1) = A_{c}x^{c}(k) + B_{c}y(k)$$
(6.2a)

$$u(k) = r(k) - C_c x^c(k) - D_c y(k)$$
(6.2b)

com $A_c \in \Re^{n_c x n_c}$, $B_c \in \Re^{n_c x l}$, $C_c \in \Re^{m x n_c}$, $D_c \in \Re^{m x l}$. A matriz de transferência do controlador é dada por:

$$\mathfrak{I}(z) = D_c + C_c \left(z I_{n_c} - A_c \right)^{-1} B_c \in \mathfrak{R}^{mxt}$$

Considerar que o problema de identificação em malha fechada seja bem posto, no sentido que as saídas y(k) sejam unicamente determinadas pelos estados da planta e do controlador e pela entrada de referência. Esta condição genérica é satisfeita quando a matriz $(I_l + DD_c)$ é não singular.

Sob as considerações citadas acima, o problema de identificação em malha fechada pode ser enunciado como:

Dado

- Dadas amostras de entradas e saídas de um problema bem posto: u(k), y(k)
 para k = 0,1,.., j + 2i 2 onde j→∞.
- Os primeiros parâmetros de Markov do controlador $\Im(z)$: $\varphi_0,...,\varphi_{i-1}$

Encontrar

- A ordem *n* do sistema
- As matrizes do sistema A, B, C, D por uma transformação de similaridade
- As matrizes Q, S, R

No apêndice D.1 se apresentam as matrizes em blocos de Hankel utilizadas neste capítulo.

A seguir apresenta-se uma idéia básica da identificação por subespaços do modelo N4SIDC. Uma apresentação mais formal é enunciada pelos teoremas de identificação em malha fechada, apresentados a partir da sub-seção 6.24.

Para o sistema (6.1) se tem as seguintes equações matriciais combinadas de estados e de entrada e saída

$$Y_{i|2i-1} = \Gamma_i X_{i|2i-1}^d + H_i^d U_{i|2i-1} + Y_{i|2i-1}^s$$
(6.3)

$$X_{i|2i-1}^{d} = A^{i} X_{0}^{d} + \Delta_{i}^{d} U_{0|i-1}$$
(6.4)

Estas equações são as mesmas dadas no capítulo 3 (teorema 2), só que com outros índices.

Substituindo-se a equação (6.4) na equação (6.3) se obtém:

$$Y_{i|2i-1} = \Gamma_i \left[A^i X_0^d + \Delta_i^d U_{0|i-1} \right] + H_i^d U_{i|2i-1} + Y_{i|2i-1}^s$$
(6.5)

Desta última equação se obtém:

$$Y_{i|2i-1} = T \left[\Gamma_i A^i X_0^d + Y_{i|2i-1}^s + \Gamma_i \Delta_i^d U_{0|i-1} + H_i^d M_{i|2i-1} \right]$$
(6.6)

A prova da equação (6.6) pode ser encontrada no apêndice D.2.

Aplicando-se na equação (6.6) a projeção ortogonal sobre a matriz $\begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$, isto é:

$$Z_{i} = Y_{i|2i-1} / \binom{N_{0|i-1}}{Y_{0|i-1}} = T_{i} \left[\Gamma_{i} \hat{X}_{i} + H_{i}^{d} M_{i|2i-1} \right]$$
(6.7)

onde a matriz \hat{X}_i é interpretada como o estado estimado pelo filtro de Kalman de estado não estacionário e é dada por:

$$\hat{X}_{i} = \left[A - \Omega_{i} \Gamma_{i} \left| \Delta_{i}^{d} - \Omega_{i} H_{i}^{d} \right| \Omega_{i} \right] \left[\frac{\hat{X}_{0}}{W_{p}} \right]$$

e

$$\hat{X}_0 = X_0 / N_{0|2i-1}$$

Maiores detalhes assim como a prova desta projeção dada pela equação (6.7) são apresentados na seção 6.2.4.

A projeção ortogonal da equação (6.7) é importante, pois dela se pode obter a seguinte projeção oblíqua sobre $\begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$ ao longo de $M_{i|2i-1}$:

$$\vartheta_{i} = Y_{i|2i-1} / \binom{U_{0|i-1}}{Y_{0|i-1}} = T_{i} \Gamma_{i} \widetilde{X}_{i}$$
(6.8)

Na equação (6.8) são conhecidas as matrizes $Y_{i|2i-1}$, $Y_{0|i-1} \in U_{0|i-1}$, mas para a matriz $M_{i|2i-1} = U_{i|2i-1} + H_i^c Y_{i|2i-1}$ é necessário ter os *i* parâmetros de Markov do controlador, isto é, conhecer H_i^c . Desta forma é fácil obter ϑ_i .

Aplica-se SVD na matriz ϑ_i para determinar a ordem *n* do sistema e faz-se uma partição adequada na SVD para obter os estados estimados \widetilde{X}_i . Por último, encontram-se as matrizes *A*, *B*, *C* e *D*. Tudo isto é apresentado no algoritmo 1 para malha fechada.

A seguir se enuncia o conceito de seqüência de estados do filtro de Kalman. Para maiores detalhes ver [OVERSCHEE; DE MOOR, 1996].

6.2.4 Definição 1 Estado do Filtro de Kalman

Dada uma matriz simétrica $P_0 \in \Re^{nxn}$ e uma matriz de seqüência de estado inicial $X_0^k \in \Re^{nxj}$, a seqüência de estados do filtro de Kalman X_i^k é definida como:

$$X_{i}^{k} \begin{bmatrix} X_{0}^{k}, P_{0} \end{bmatrix} = \begin{bmatrix} \hat{x}_{i} & \hat{x}_{i+1} & \cdots & \hat{x}_{i+j-1} \end{bmatrix}$$

$$\stackrel{def}{=} (A - \Omega_{i} \Gamma_{i}) X_{0}^{k} + (\Delta_{i}^{d} - \Omega_{i} H_{i}^{d}) U_{0|i-1} + \Omega_{i} Y_{0|i-1}$$
(6.9)

$$\Omega_i^{def} = \left(\Delta_i^G - A^i P_0 \Gamma_i^T\right) \left(L_i - \Gamma_i P_0 \Gamma_i^T\right)^{-1}$$
(6.10)

A seguir se pretende solucionar o problema de identificação em malha fechada enunciado na seção 6.2.2. Depois discutem-se dois teoremas de projeção para malha fechada, que mostram como a seqüência de estados do filtro de Kalman de estado não estacionário e as matrizes relacionadas ao sistema (6.1) podem ser recuperadas de um conjunto de matrizes.

Não se recomenda aplicar os algoritmos de identificação por subespaços [CVA, MOESP, N4SID] para situações em malha fechada, pois os algoritmos por subespaços para malha aberta não funcionam para sinais medidos em malha fechada, tendo em vista que os distúrbios v(k) e w(k) e a entrada u(k) são correlacionados. Este fato é provado em [LJUNG; McKELVEY, 1996].

6.2.5 Teorema da projeção para malha fechada

Dado que os sinais de entrada são correlacionados com os distúrbios, então são necessárias outras matrizes auxiliares $N \in M$, com a propriedade que estas novas matrizes sejam não correlacionadas com estes distúrbios.

6.2.5.1 Teorema 1 Teorema da projeção em malha fechada

Hipóteses

- O sinal de referência r_k é não correlacionado com o ruído de processo w_k e de medição v_k.
- 2. A matriz $N_{0|2i-1}$ tem posto máximo 2mi.
- 3. O número de dados medidos tende para infinito $j \rightarrow \infty$.
- 4. O problema de malha fechada é bem posto, isto é, a matriz $(I_1 + DD_c)$ é invertível.

Tese

$$Z_{i}^{def} = Y_{i|2i-1} / \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \\ M_{0|i-1} \end{pmatrix}$$
$$= Y_{i|2i-1} / \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$$
(6.11)

$$=T_i \left[\Gamma_i \hat{X}_i + H_i^d M_{i|2i-1} \right]$$
(6.12)

com

$$\hat{X}_i = X_i^k [\hat{x}_{0, P_0}], \tag{6.13}$$

$$\hat{X}_0 = X_0 / N_{0|2i-1} \tag{6.14}$$

$$P_0 = -\left[\Sigma_{xx} - \Sigma_{ss} - S_{xn}R_{nn}^{-1}S_{xn}^T\right]$$
(6.15)

A prova deste teorema pode ser encontrada no apêndice D.3.

A hipótese 1 do teorema 3 para malha aberta na secção 3.4.4 é a mesma da hipótese 2 deste teorema, só que ao invés de se usar a matriz de entrada $U_{0|2i-1}$, emprega-se a matriz transformada $N_{0|2i-1}$.

Este teorema é muito parecido com o teorema 3 para malha aberta, dado na secção 3.4.4, no qual é dado que:

$$Z_{i}^{def} = Y_{i|2i-1} / \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \\ U_{i|2i-1} \end{pmatrix} = \Gamma_{i} \hat{X}_{i} + H_{i}^{d} U_{i|2i-1}$$
(6.16)

Comparando-se as equações (6.12) e (6.16) se tem as seguintes diferenças:

* a matriz T_i em (6.12);

* as entradas $U_{i|2i-1}$ da equação (6.16) são substituídas pela matriz $M_{i|2i-1}$ na equação (6.12); e

* o estado inicial dado no teorema 3 para o caso de malha aberta $\hat{X}_0 = X_0 / U_{0|2i-1}$ é substituído para o caso de malha fechada pelo estado inicial $\hat{X}_0 = X_0 / N_{0|2i-1}$ da equação (6.14).

6.2.6 Teorema principal para malha fechada

A importância deste teorema é que ele permite o cálculo do espaço filas de uma seqüência de estados do filtro de Kalman e do espaço colunas do produto T_i e a matriz observabilidade estendida $(T_i\Gamma_i)$, diretamente dos dados de entrada e saída, sem o conhecimento das matrizes do sistema.

6.2.6.1 Teorema 2 teorema principal para malha fechada

Hipóteses

As mesmas dadas no teorema 1 e definindo-se a projeção oblíqua 9 como:

$$\Theta_{i}^{def} = Y_{i|2i-1} / M_{i|2i-1} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$$
(6.17)

E a decomposição de valores singulares:

$$\Theta_{i} = \begin{pmatrix} U_{1} & U_{2} \end{pmatrix} \begin{pmatrix} S_{1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_{1}^{T} \\ V_{2}^{T} \end{pmatrix} = U_{1} S_{1} V_{1}^{T}$$
(6.18)

Teses

1. A matriz ϑ_i é igual ao produto da matriz T_i e da matriz de observabilidade estendida e a seqüência de estados do filtro de Kalman \widetilde{X}_i :

$$\vartheta_i = T_i \Gamma_i(\widetilde{X}_i)$$

com:

$$\begin{split} \widetilde{X}_{i} &\stackrel{def}{=} X_{i}^{k} [\widetilde{x}_{0}, P_{0}] \\ \widetilde{X}_{0} &= \widehat{X}_{0} /_{M_{i} | 2i-1} \begin{pmatrix} U_{0 | i-1} \\ Y_{0 | i-1} \end{pmatrix} \end{split}$$

$$P_0 = -\left[\Sigma_{xx} - \Sigma_{ss} - S_{xn}R_{nn}^{-1}S_{xn}^T\right]$$

- A ordem do sistema dada pelas equações (6.1) é igual ao número de valores singulares presentes na equação (6.18) que sejam diferentes de zero.
- 3. O produto da matriz T_i e da matriz de observabilidade estendida pode ser recuperada de:

$$T_i \Gamma_i = U_1 S_1^{1/2}$$
.

4. A parte da seqüência de estado \widetilde{X}_i pode ser recuperada de:

$$\widetilde{X}_i = S_1^{1/2} V_1^T = \left(T_i \Gamma_i\right)^+ \vartheta_i.$$

A prova deste teorema pode ser encontrada no apêndice D.4.

6.2.6.2 Algoritmo 1: Usando estados

Este algoritmo explica como encontrar uma estimativa das matrizes do sistema $A, B, C \in D$ e das matrizes $Q, R \in S$, a partir dos teoremas 1 e 2.

Observa-se que do teorema 1 se obtém \hat{X}_i (seqüência de estados estimados), equação (6.13). Agora imagine que \hat{X}_i e \hat{X}_{i+1} (onde $\hat{X}_{i+1} = X_{i+1}^k[\hat{x}_{0,P_0}]$) sejam dadas. Ambas são seqüências de estados do filtro de Kalman para o mesmo banco de filtro de Kalman de estados não estacionário [OVERSCHEE; DE MOOR, 1996]. Têm-se as seguintes equações:

$$\hat{X}_{i+1} = A\hat{X}_i + BU_{i|i} + K_i E_{i|i}$$
(6.19)

$$Y_{i|i} = C\hat{X}_i + DU_{i|i} + E_{i|i}$$
(6.20)

ou em forma matricial:

$$\begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} K_i \\ I \end{bmatrix} E_{i|i}$$
(6.21)

Infelizmente, na equação (6.21) não se pode aplicar o método dos mínimos quadrados, pois por hipótese a inovação $E_{i|i}$ não é ortogonal às entradas $U_{i|i}$. Portanto, dados os estados \hat{X}_i e \hat{X}_{i+1} , não é possível determinar o sistema de matrizes a partir das equações (6.19) e (6.20) pelo método dos mínimos quadrados.

Pelo teorema 1 é possível mostrar que o termo inovação $E_{i|i}$ na equação (6.19) e (6.20) é perpendicular ao espaço linha de \hat{X}_i e $M_{i|i}$. Para a prova ver [VAN OVERSCHEE; DE MOOR, 1997].

Então com ajuda de uma nova matriz $M_{i|i}$, as equações (6.19) e (6.20) podem ser apresentadas como:

$$\left[\frac{\hat{X}_{i+1}}{Y_{i|i}}\right] = \left[\frac{A - BD_c T_1 C}{T_1 C} \left| \frac{B(I_m - D_c T_1 D)}{T_1 D} \right] \left[\frac{\hat{X}_{i+1}}{M_{i|i}}\right] + \left[\frac{K_i - BD_c T_1}{T_1}\right] E_{i|i}$$
(6.22)

 $T_1 = (I_i + DD_c)^{-1}$ é não singular, pois o problema é bem posto. Então se são dadas as seqüências \hat{X}_i e \hat{X}_{i+1} , pode-se aplicar o método dos mínimos quadrados na equação (6.22) para estimar as matrizes do sistema, pois os resíduos $E_{i|i}$ são perpendiculares ao

espaço linha de \hat{X}_i e $M_{i|i}$.

Observe que o estado estimado \hat{X}_i é dado para a projeção Z_i (ver teorema 1), da mesma forma se pode provar que:

$$Z_{i+1} = Y_{i+1|2i-1} / \begin{pmatrix} U_{0|i} \\ Y_{0|i} \\ M_{i+1|2i-1} \end{pmatrix} = Y_{i+1|2i-1} / \begin{pmatrix} N_{0|2i-1} \\ Y_{0|i} \end{pmatrix} = T_{i-1} \Big[\Gamma_{i-1} \hat{X}_{i+1} + H_{i-1}^{d} M_{i+1|2i-1} \Big]$$

com

$$\hat{X}_{i+1} = X_{i+1}^k [\hat{X}_{0,P_0}].$$

O ponto aqui é como determinar as sequências de estados \hat{X}_i e \hat{X}_{i+1} a partir das projeções Z_i e Z_{i+1} . Para isto, da equação (6.12) se obtém:

$$Z_{i} = T_{i}\Gamma_{i}\hat{X}_{i} + T_{i}H_{i}^{d}M_{i|2i-1}$$
(6.23)

Como foi mencionado anteriormente, da equação (6.23) se obtém a projeção oblíqua $\vartheta_i = T_i \Gamma_i(\widetilde{X}_i)$, com U_2 da equação (6.18), encontra-se de (6.23)

$$U_2^T Z_i = U_2^T (T_i H_i^d) M_{i|2i-1}$$
(6.24)

O termo desconhecido aqui é $(T_iH_i^d)$, que pode ser considerado da forma:

$$T_i H_i^d = \kappa = \begin{bmatrix} \kappa_0 & 0 & \cdots & 0 \\ \kappa_1 & \kappa_0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \kappa_{i-1} & \kappa_{i-2} & \cdots & \kappa_0 \end{bmatrix}$$

Com $\kappa \in \Re^{lxm}$, então a equação (6.24) fica:

$$U_2^T Z_i = U_2^T \kappa M_{i|2i-1}$$
(6.25)

Para encontrar κ pode-se multiplicar a equação (6.25) por $(M_{i|2i-1})^+$, mas pode ocorrer que esta matriz seja mal condicionada. Em [VAN OVERSCHEE; DE MOOR, 1997] se mostra como obter a matriz κ . Uma vez encontrada κ é muito fácil encontrar:

$$\hat{X}_i = \varphi^+ \left[Z_i - \kappa M_{i|2i-1} \right]$$

$$\hat{X}_{i+1} = \underline{\Phi}^+ \left[Z_{i+1} - \underline{\kappa} M_{i+1|2i-1} \right]$$

onde $\varphi = T_i \Gamma_i = U_1 S_1^{1/2}$ e (\circ) indica a matriz em blocos sem as últimas linhas.

Algoritmo 1 para malha fechada

- 1. Construa as matrizes $M_{i|2i-1} = U_{i|2i-1} + H_i^c Y_{i|2i-1}$ e $M_{i+1|2i-1} = U_{i+1|2i-1} + H_{i-1}^c Y_{i+1|2i-1}$, onde H_i^c e H_{i-1}^c contêm os parâmetros de Markov do controlador.
- 2. Calcule as projeções ortogonal e oblíqua:

$$\vartheta_{i} \stackrel{def}{=} Y_{i|2i-1} / {}_{M_{i}|2i-1} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}, \quad Z_{i} \stackrel{def}{=} Y_{i|2i-1} / \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \\ M_{0|i-1} \end{pmatrix} e \quad Z_{i+1} = Y_{i+1|2i-1} / \begin{pmatrix} U_{0|i} \\ Y_{0|i} \\ M_{i+1|2i-1} \end{pmatrix}$$

3. Calcule as SVD da projeção oblíqua $\vartheta_i = USV^T$.

- 4. Determine a ordem do sistema por inspeção dos valores singulares em S e encontre U_1 , U_2 e S_1 através do particionamento adequado das SVD.
- 5. Determine $\varphi = T_i \Gamma_i = U_1 S_1^{1/2}$ e κ .
- 6. Determine os estados

$$\hat{X}_{i} = \boldsymbol{\varphi}^{+} \left[\boldsymbol{Z}_{i} - \boldsymbol{\kappa} \, \boldsymbol{M}_{i|2i-1} \right],$$
$$\hat{X}_{i+1} = \underline{\boldsymbol{\varphi}}^{+} \left[\boldsymbol{Z}_{i+1} - \underline{\boldsymbol{\kappa}} \, \boldsymbol{M}_{i+1|2i-1} \right].$$

7. Solucione o seguinte conjunto de equações lineares:

$$\begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ M_{i|i} \end{bmatrix} + \tau$$

com $M_{i|i} = U_{i|i} + D_c Y_{i|i}$.

8. Determine as matrizes $A, B, C \in D$, (T_1) como: $B = S_{12} (I_m - C_0 S_{22})^{-1}$, $D = S_{22} (I_m - C_0 S_{22})^{-1}$, $A = S_{11} + BC_0 S_{21}$, $T_1 = (I_1 + DC_0)^{-1}$ e $C = T_1^{-1} S_{21}$.

9. Determine as matrizes Q, $R \in S$ dos resíduos como:

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = \begin{pmatrix} I_n & BC_0 \\ 0 & T_1^{-1} \end{pmatrix} E_j [\tau \tau^T] \begin{pmatrix} I_n & BC_0 \\ 0 & T_1^{-1} \end{pmatrix}^T$$

6.2.6.3 Algoritmo 2 usando estados

No algoritmo 1 mostra-se que a seqüência de estados \hat{X}_i não pode ser encontrada diretamente das projeções e precisa-se encontrar primeiro a matriz κ , para assim poder encontrar as seqüências de estados. No entanto, pelo teorema 2 mostra-se que a seqüência de estados \widetilde{X}_i pode ser encontrada diretamente da projeção oblíqua ϑ_i . Analogamente, a seqüência de estados \widetilde{X}_{i+1} pode ser determinada mediante a projeção oblíqua:

$$\vartheta_{i+1} = Y_{i+1|2i-1} / \int_{M_{i+1}|2i-1} \begin{pmatrix} U_{0|i} \\ Y_{0|i} \end{pmatrix} = T_{i-1} \Gamma_{i-1} \widetilde{X}_{i+1}$$

Desta forma encontram-se as matrizes \widetilde{X}_i e \widetilde{X}_{i+1} . O problema é que ambas seqüências de estado do filtro de Kalman têm diferentes estados iniciais:

Estado inicial para
$$\widetilde{X}_{i} = \hat{X}_{0} / _{M_{i}|_{2i-1}} \begin{pmatrix} U_{0|_{i-1}} \\ Y_{0|_{i-1}} \end{pmatrix}$$

Estado inicial para
$$\widetilde{X}_{i+1} = \hat{X}_0 / \binom{U_{0|i}}{Y_{0|i}}$$

No entanto, justamente como no caso para malha aberta, o efeito desta diferença de estados iniciais desaparece quando converge o filtro de Kalman, isto é, quando $i \rightarrow \infty$. Para maiores detalhes ver [VAN OVERSCHEE; DE MOOR, 1996]. Baseado nesta observação apresenta-se um novo algoritmo.

Algoritmo 2 para caso em malha fechada

- 1. Construa as matrizes $M_{i|2i-1} = U_{i|2i-1} + H_i^c Y_{i|2i-1}$ e $M_{i+1|2i-1} = U_{i+1|2i-1} + H_{i-1}^c Y_{i+1|2i-1}$, onde H_i^c e H_{i-1}^c contêm os parâmetros de Markov do controlador.
- 2. Calcule as projeções ortogonal e oblíqua:

$$\vartheta_{i} \stackrel{def}{=} Y_{i|2i-1} / {}_{M_{i}|2i-1} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} e \quad \vartheta_{i+1} = Y_{i+1|2i-1} / {}_{M_{i+1}|2i-1} \begin{pmatrix} U_{0|i} \\ Y_{0|i} \end{pmatrix}.$$

- 3. Calcule as SVD da projeção oblíqua $\vartheta_i = USV^T$.
- 4. Determine a ordem do sistema por inspeção dos valores singulares em S e encontre U_1 , U_2 e S_1 através do particionamento adequado das SVD. Determine $\varphi = U_1 S_1^{1/2}$.
- 5. Determine os estados:

$$\widetilde{X}_i = \varphi^+ \vartheta_i \qquad e \qquad \widetilde{X}_{i+1} = (\underline{\varphi})^+ \vartheta_{i+1}.$$

Solucione o seguinte conjunto de equações lineares:

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} \widetilde{X}_i \\ M_{i|i} \end{bmatrix} + \tau$$

com $M_{i|i} = U_{i|i} + D_c Y_{i|i}$.

- 6. Determine as matrizes $A, B, C \in D$, (T_1) como: $B = S_{12} (I_m C_0 S_{22})^{-1}$, $D = S_{22} (I_m - C_0 S_{22})^{-1}$, $A = S_{11} + BC_0 S_{21}$, $T_1 = (I_1 + DC_0)^{-1}$ e $C = T_1^{-1} S_{21}$.
- 7. Determine as matrizes Q, $R \in S$ dos resíduos como:

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = \begin{pmatrix} I_n & BC_0 \\ 0 & T_1^{-1} \end{pmatrix} E_j [\tau \tau^T] \begin{pmatrix} I_n & BC_0 \\ 0 & T_1^{-1} \end{pmatrix}^T$$

CAPÍTULO 7

IDENTIFICAÇÃO EM MALHA FECHADA PELO MÉTODO MOESPC

Nesta secção apresenta-se o problema de identificação em malha fechada, aplicando o método MOESP. Este método foi estudado no capítulo 2.

7.1 Modelo em espaço de estados em malha fechada método MOESPC

Um modelo em espaço de estados em malha fechada é mostrado na figura 7.1.



Figura 7.1 Representação em blocos de um modelo em malha fechada

Na figura 7.1 o bloco pontilhado representa um sistema desconhecido, neste bloco assume-se que a planta desconhecida consiste em uma parte determinística, representada pelo bloco \mathbf{P} e uma parte estocástica, representada pelo filtro linear \mathbf{F} e w_k seqüência de ruído branco com média zero.

Este sistema desconhecido está operando em malha fechada com um compensador, o qual é denotado pelo bloco C. Assume-se que o compensador não cause nenhum cancelamento da dinâmica da planta. Baseado na configuração da figura 7.1 apresenta-se o problema de identificação em malha fechada.

7.2 Problema de identificação em malha fechada (método MOESPC)

Dado:

- r_2 é o vetor de valores de referência
- r_1 é uma excitação externa
- e_2 sequência do erro entre a saída do sistema e o sinal r_2
- e_1 seqüência de saída no controlador
- u(k) sequência da saída do controlador adicionada ao ruído r_1
- y(k) saída do sistema

As perturbações atuam sobre os sinais internos e_2 , e_1 , u(k) e y(k). O processo estocástico w(k) não é correlacionado com as seqüências externas r_2 e r_1 , as quais são ruídos não mensuráveis.

O problema é determinar um modelo em espaço de estados para a parte determinística **P** da planta a ser identificada.

7.2.1 Idéia básica para a identificação por subespaços em malha fechada (método MOESPC)

A seguir apresenta-se uma idéia básica da identificação por subespaços do modelo MOESPC. Uma apresentação mais formal é enunciada através do teorema de identificação em malha fechada, enunciada na seção 7.3.

A idéia é fazer uso da função de transferência. Algoritmos usando função de transferência são propostos e aplicados para diferentes situações praticas ver [PHADKE; WU, 1974; CAINES; CHAN, 1975]. Dentro da identificação por subespaços o problema é a identificação para sistemas estocásticos, isto é, a

determinação do modelo de Markov a partir dos dados de entrada e saída. A partir do modelo de Markov se calcula uma representação em espaço de estados da planta **P**, o controlador **C** e o filtro inovativo linear **F**, que descreve as diferentes perturbações que atuam no sistema. Mas [CAINES; CHAN, 1976] demonstraram, através de diferentes experimentos, que para este tipo de aproximação, pequenas variações na matriz de covariância (dos dados de entrada e saída) reproduzem grandes variações para o modelo de Markov obtido.

Este problema é superado, caso se agregue ao sistema uma seqüência de entrada externa conhecida, tal como r_2 e r_1 , ver [GUSTAVSSON, 1977]. Isto é necessário para garantir a identificação em malha fechada.

Verhaegen (1993) também trata o problema de identificação da planta e o controlador quando se usam explicitamente sinais externos r_2 e r_1 em adição a sinais internos. No entanto, contrário aos métodos citados acima, a análise e solução deste problema são dadas em espaço de estados. Isto é usando o método MOESP (aplicado para malha aberta, ver capitulo 2) obtém-se um modelo de estado global, da função de transferência dos sinais r_2 e r_1 para e_2 , e_1 , u(k) e y(k), ver figura 7.2



Figura 7.2 Modelo Global em bloco

Deste modelo em espaço de estados global, um modelo em espaço de estados da planta **P** e do controlador **C** pode ser computado, para isto não se precisa conhecer o controlador.

7.3 Identificação em malha fechado pelo método MOESPC

Como foi dito anteriormente, este método usa estimativas de certas matrizes de ransferência. As relações de entrada e saída na figura 7.1 são representadas em termos

da função matriz de transferência. Por exemplo, para o controlador C, tem-se que:

$$e_1(z) = C(z)e_2(z)$$
(7.1)

onde (z) denota a transformada z usual, por comodidade nas operações se omite a notação explícita (z). Logo, da figura 7.1 se derivam as seguintes relações:

$$e_1 = -CP(I + CP)^{-1}r_1 + C(I + PC)^{-1}r_2 - C(I + PC)^{-1}F_2w$$
(7.2)

$$e_2 = -P(I + CP)^{-1}r_1 + (I + PC)^{-1}r_2 - (I + PC)^{-1}F_2w$$
(7.3)

$$u = (I + PC)^{-1} r_1 + C(I + PC)^{-1} r_2 - C(I + PC)^{-1} F_2 w$$
(7.4)

$$y = P(I + CP)^{-1} r_1 + PC(I + PC)^{-1} r_2 + (I + PC)^{-1} F_2 w$$
(7.5)

Denotando $H_{r_1e_1} = -CP(I + CP)^{-1}$, $H_{r_2e_1} = C(I + PC)^{-1}$ e $v_1 = C(I + CP)^{-1}F_2w$ na equação (7.2), resulta: $e_1 = (H_{r_1e_1})r_1 + (H_{r_2e_1})r_2 - v_1$ (7.6)

Denotando
$$H_{r_1e_2} = -P(I+CP)^{-1}$$
, $H_{r_2e_2} = (I+PC)^{-1}$ e $v_2 = (I+CP)^{-1}F_2w$ na equação (7.3), resulta:

$$e_2 = (H_{r_1 e_2})r_1 + (H_{r_2 e_2})r_2 - v_2$$
(7.7)

Denotando
$$H_{r_1u} = (I + PC)^{-1}$$
, $H_{r_2u} = C(I + PC)^{-1}$ na equação (7.4), resulta:
 $u = (H_{r_1u})r_1 + (H_{r_2u})r_2 - v_1$
(7.8)

Denotando $H_{r_1y} = P(I + CP)^{-1}$, $H_{r_2y} = PC(I + PC)^{-1}$ na equação (7.5), resulta: $y = (H_{r_1y})r_1 + (H_{r_2y})r_2 - v_2$ (7.9)

As equações (7.6)-(7.9) mostram que é possível determinar a função de transferência da planta e do controlador através das relações [VERHAEGEN, 1993]:

$$P = H_{r_1 y} (H_{r_1 u})^{-1} = -H_{r_1 e_2} (H_{r_1 u})^{-1}$$
(7.10)

ou

$$P = H_{r_2 y} (H_{r_2 u})^{-1} = H_{r_2 y} (H_{r_2 e_1})^{-1}$$
(7.11)

$$C = H_{ne_1}(H_{ne_2})^{-1} = -H_{ne_1}(H_{ny})^{-1}$$
(7.12)

e

$$C = H_{r_2 e_1} (H_{r_2 e_2})^{-1} = -H_{r_2 u} (H_{r_2 e_2})^{-1}$$
(7.13)

Assumindo-se que todas as inversas nas equações acima existam, obtém-se um mecanismo para estimar a planta e o controlador. Diferentes possibilidades existem para este propósito, como o método de predição de erro ou a técnica de variável instrumental, descritos em Söderstrom e Stoica (1989) e Ljung (1999), respectivamente. Aparte das dificuldades relacionadas às parametrizações para modelos MIMO (ver Ljung, 1987), não existe um caminho que garanta para as funções de transferências dadas em (7.2) - (7.5) um denominador comum. Para solucionar este problema é considerado um modelo global em espaço de estados, ver figura 7.2. Deste modelo global se pode computar as matrizes de transferências individuais. No entanto, não é necessário fazer isto, pois como é mostrado na seção seguinte, as equações (7.6) até (7.9) podem ser calculadas a partir de um sistema global, desta maneira é possível computar $\mathbf{P} \in \mathbf{C}$.

Para entender isto, considerar w(k) = 0 na figura 7.1 e defina-se o modelo global no espaço de estados como:

$$x(k+1) = Ax(k) + \begin{bmatrix} B_1 & B_2 \end{bmatrix} \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$
(7.14)

$$\begin{bmatrix} e_{1}(k) \\ e_{2}(k) \\ u(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} c_{1} \\ c_{2} \\ c_{3} \\ c_{4} \end{bmatrix} x(k) + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \\ D_{31} & D_{32} \\ D_{41} & D_{42} \end{bmatrix} \begin{bmatrix} r_{1}(k) \\ r_{2}(k) \end{bmatrix}$$
(7.15)

Seja *np* a ordem da planta **P** e *nc* a ordem do controlador **C**, então $x(k) \in R^{np+nc}$. Das equações (7.14) e (7.15), uma representação para a função de transferência H_{r_1y} e H_{r_1u} é dado pelo quádruplo de sistemas de matrizes $[A, B_1, C_4, D_{41}]$ e $[A, B_1, C_3, D_{31}]$, respectivamente. Agora usando regras de concatenação e operações matriciais, se obtém um modelo em espaço de estados para a expressão **P** dada na equação (7.10), para maiores detalhes ver [FRANCIS, 1987]:

$$P = [A, B_{1}, C_{4}, D_{41}] \times [A, B_{1}, C_{3}, D_{31}]^{-1}$$

$$P = [A, B_{1}, C_{4}, D_{41}] \times [A - B_{1}D_{31}^{-1}C_{3}, B_{1}D_{31}^{-1}, -D_{31}^{-1}C_{3}, D_{31}^{-1}]$$

$$P = \begin{bmatrix} A - B_{1}D_{31}^{-1}C_{3} \\ 0 & A - B_{1}D_{31}^{-1}C_{3} \end{bmatrix}; \begin{bmatrix} B_{1}D_{31}^{-1} \\ B_{1}D_{31}^{-1} \end{bmatrix}; (C_{4} - D_{41}D_{31}^{-1}C_{3}); D_{41}D_{31}^{-1} \end{bmatrix}$$
(7.16)

Analogamente, se obtém uma expressão para o controlador C definido em (7.13): $C = [A, B_{2}, C_{3}, D_{32}] \times [A, B_{2}, C_{2}, D_{22}]^{-1}$ $C = [A, B_{1}, C_{4}, D_{41}] \times [A - B_{2}D_{22}^{-1}C_{2}, B_{2}D_{22}^{-1}, -D_{22}^{-1}C_{2}, D_{22}^{-1}]$ $C = \begin{bmatrix} A - B_{2}D_{22}^{-1}C_{2} \\ 0 & A - B_{2}D_{22}^{-1}C_{2} \end{bmatrix}; \begin{bmatrix} B_{2}D_{22}^{-1} \\ B_{2}D_{22}^{-1} \end{bmatrix}; (C_{3} - D_{32}D_{22}^{-1}C_{2}); D_{32}D_{22}^{-1} \end{bmatrix}$ (7.17)

Como foi dito anteriormente, a planta e ou controlador podem ser obtidos diretamente de um sistema global, usando a equação (7.16) e (7.17) respectivamente. Nestas equações se necessita a existência da inversa das matrizes D_{31}^{-1} e D_{22}^{-1} , as quais são dadas pelo seguinte teorema.

7.3.1 Teorema

A planta P dada na figura 7.1 tem representação em espaço de estados da forma:

$$s(k+1) = A_p s(k) + B_p u(k)$$
(7.18)

$$y(k) = C_{p}s(k) + D_{p}u(k)$$
(7.19)

onde $s(k) \in \mathbb{R}^{np}$ e o controlador **C** dado na figura 7.1 tem a representação:

$$v(k+1) = A_c v(k) + B_c e_2(k)$$
(7.20)

$$e_1(k) = C_c v(k) + D_c e_2(k)$$
(7.21)

onde $v(k) \in R^{nc}$. A planta e o controlador se encontram em malha fechada na forma da figura 7.1, estão bem definidos, no sentido que as matrizes $(I + D_p D_c)$ e $(I + D_c D_p)$ têm inversa. Considerando-se w(k) = 0 na figura 7.1, então as matrizes **P** e **C** expressas pelas equações (7.16) e (7.17), respectivamente, existem para todas as possíveis escolhas das matrizes $D_p \in D_c$.

Para a prova deste teorema, ver apêndice E.

Comentários

Com os modelos de espaço de estados individual (7.18), (7.19) e (7.20), (7.21) se pode ter um espaço de estados global da forma (7.14) e (7.15), explicitamente como:

$$\begin{bmatrix} s(k+1)\\ v(k+1) \end{bmatrix} = A \begin{bmatrix} s(k)\\ v(k) \end{bmatrix} + B \begin{bmatrix} r_1(k)\\ r_2(k) \end{bmatrix}$$

$$\begin{bmatrix} e_1(k)\\ e_2(k)\\ u(k)\\ u(k)\\ v(k) \end{bmatrix} = C \begin{bmatrix} s(k)\\ v(k) \end{bmatrix} + D \begin{bmatrix} r_1(k)\\ r_2(k) \end{bmatrix}$$
(7.22)
(7.23)

onde as matrizes do sistema global (A, B, C, D) dadas em (7.22) e (7.23) são definidas em função das matrizes (A_p, B_p, C_p, D_p) e (A_c, B_c, C_c, D_c) , da forma:

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \\ D_{31} & D_{32} \\ D_{41} & D_{42} \end{bmatrix} = \begin{bmatrix} -D_c \Phi D_p & D_c (I - \Phi D_p D_c) \\ -\Phi D_p & I - \Phi D_p D_c \\ I - D_c \Phi D_p & D_c (I - \Phi D_p D_c) \\ \Phi D_p & \Phi D_p D_c \end{bmatrix}$$
(7.24)

onde $\Phi = (I + D_p D_c)^{-1}$. Para as demais matrizes A, B e C ver apêndice E.1.

Logo, o sistema global dado pelas equações (7.14)-(7.15) é o mesmo que o sistema global dado pelas equações (7.22)-(7.23), com as seguintes notações:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; B = \begin{bmatrix} B_1 & B_2 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}; C = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \\ c_{41} & c_{42} \end{bmatrix} e$$

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \\ D_{31} & D_{32} \\ D_{41} & D_{42} \end{bmatrix}$$
(7.25)

Então, se pode obter a planta **P** através da equação (7.16) e a matriz $D_{31} = d_{31}$ é definida por:

$$D_{31} = I - D_c (I - D_p D_c)^{-1} D_p$$
(7.26)

A matriz da equação (7.26) tem inversa e é expressa por [ver apêndice E.2]: $D_{31}^{-1} = I + D_c D_p$. (7.27)

Analogamente, a inversa da matriz D_{22} necessária na equação (7.17) para computar o controlador **C**, é dada por [ver apêndice E.3]:

$$D_{22}^{-1} = I + D_p D_c \tag{7.28}$$

Desta forma, fazendo uso das equações (7.25), (7.26), (7.27) e (7.28) se pode obter a planta **P** e o controlador **C** através das equações (7.16) e (7.17), respectivamente.

Um estudo sobre a influência das matrizes D_p e D_c na existência dos modelos em espaço de estados para **P** ou **C** é dado em [VERHAEGEN, 1993].

A representação em espaço de estados da planta e do controlador obtida pelas equações (7.16) e (7.17), respectivamente, são de fase não mínima. Para ver isto considerar, por exemplo, o modelo da planta dado em (7.16) e aplicando uma transformação linear *T*, resulta [ver apêndice E.4]:

$$P = \begin{bmatrix} \begin{pmatrix} A & 0 \\ 0 & A - B_1 D_{31}^{-1} C_3 \end{pmatrix}; \begin{pmatrix} 0 \\ B_1 D_{31}^{-1} \end{pmatrix}; \begin{pmatrix} C_4 & C_4 - D_{41} D_{31}^{-1} C_3 \end{pmatrix}; D_{41} D_{31}^{-1} \end{bmatrix}$$
(7.29)

Neste caso, tem-se um sistema com np + nc modos não controláveis [VERHAEGEN, 1993]. Descartando estes modos, o modelo da planta se reduz a:

$$P = \left[\left(A - B_1 D_{31}^{-1} C_3 \right); \left(B_1 D_{31}^{-1} \right); \left(C_4 - D_{41} D_{31}^{-1} C_3 \right); D_{41} D_{31}^{-1} \right]$$
(7.30)

Esta redução do modelo da planta (ou controlador) é válida quando se faz uso de um modelo global em espaço de estados, sob as considerações descritas no teorema 1.

ALGORITMO MOESPC PARA MALHA FECHADA

1. Coletar os dados como:

U=[r1 r2] Y=[e1 e2 u y]

- 2. Aplicar qualquer algoritmo MOESP para malha aberta, obtendo as matrizes do sistema global.
- 3. Computar as matrizes da planta e do controlador.
- 4. Aplicar redução para as matrizes da planta e do controlador.

CAPÍTULO 8

MÉTODO MON4SIDC PROPOSTO PARA MALHA FECHADA E SIMULAÇÕES

Existem diversos métodos de identificação por subespaços aplicados a malha fechada, os quais diferem uns dos outros pela configuração do sistema e pela implementação dos algoritmos. Como foi visto nos capítulos 6 e 7, é possível fazer uma extensão dos métodos de identificação por subespaços para malha fechada a partir dos métodos para malha aberta. Por exemplo, para aplicar o método N4SID para malha fechada é necessário ter algum conhecimento sobre o controlador. No caso do método MOESP é necessário conhecer a ordem do controlador, e de um sistema global são obtidas as matrizes da planta e do controlador através de uma redução de ordem.

Neste trabalho se apresenta um método alternativo para a identificação da planta de um sistema em malha fechada, o qual não precisa ter nenhum conhecimento prévio sobre o controlador.

8.1 Método MON4SIDC para malha fechada

O método MON4SID para malha aberta foi apresentado no capítulo 4. A idéia de aplicar este método para identificação em malha fechada foi inspirado no modelo MOESP para malha fechada, o qual foi apresentado no capítulo 7.

O método MON4SIDC identifica um sistema MIMO trabalhando em malha fechada, como mostra a figura 8.1.



Figura 8.1 Representação de um modelo em malha fechada

Da mesma forma como foi visto no capítulo 7, o método MON4SIDC considera como entrada o sinal de referência $[r_1(k) r_2(k)]^T = r(k)$ e como saídas u(k) e y(k), que denotam o sinal de entrada e saída do processo, respectivamente. Os sinais w(k) e v(k)são chamados ruídos do processo e de medição, respectivamente, e são definidos da mesma forma que no capítulo 3.

A planta P dada na figura 8.1 tem representação em espaço de estados da forma:

$$s(k+1) = A_p s(k) + B_p u(k) + w(k)$$
(8.1)

$$y(k) = C_{p}s(k) + D_{p}u(k) + v(k)$$
(8.2)

com $s(k) \in \mathbb{R}^{np}$ e o controlador C dado na figura 8.1, tem a representação:

$$x_{c}(k+1) = A_{c}x_{c}(k) + B_{c}u_{c}(k)$$
(8.3)

$$y_{c}(k) = C_{c}x_{c}(k) + D_{c}u_{c}(k)$$
 (8.4)

A partir das equações (8.1)-(8.4) se pode encontrar o modelo global em espaço de estados do sistema em malha fechada, o qual é dado por:

$$\begin{bmatrix} s(k+1) \\ x_c(k+1) \end{bmatrix} = A \begin{bmatrix} s(k) \\ x_c(k) \end{bmatrix} + B \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} + W(k)$$
(8.5)

$$\begin{bmatrix} u(k) \\ y(k) \end{bmatrix} = C \begin{bmatrix} s(k) \\ x_c(k) \end{bmatrix} + D \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix} + V(k)$$
(8.6)

onde as matrizes (A, B, C, D) têm dimensões apropriadas e são definidas por:

$$A = \begin{bmatrix} A_p - B_p D_c \Phi C_p & B_p C_c - B_p D_c \Phi D_p C_c \\ - B_c \Phi C_p & A_c - B_c \Phi D_p C_c \end{bmatrix} B = \begin{bmatrix} B_p D_c (I - \Phi D_p C_c) & B_p (I - D_c \Phi D_p) \\ B_c (I - \Phi D_p C_c) & - B_c \Phi D_p \end{bmatrix}$$

$$C = \begin{bmatrix} -D_c \Phi C_p & I - D_c \Phi C_c \\ \Phi C_p & \Phi D_p C_c \end{bmatrix} \qquad D = \begin{bmatrix} D_c (I - \Phi D_p C_c) & I - D_c \Phi D_p \\ \Phi D_p C_c & \Phi D_p \end{bmatrix}$$
(8.7)
onde $\Phi = (I + D_p D_c)^{-1}$.

Supondo que a planta e o controlador na forma da figura 8.1 se encontre bem definido, isto quer dizer que a matriz $(I + D_p D_c)$ tem inversa e portanto existe Φ .

A equação (8.7) mostra que as matrizes do sistema em malha fechada ficam determinadas em termos das matrizes da planta e do controlador. Estas matrizes não são únicas, pois é possível obter um sistema equivalente através de uma transformação linear.

8.1.1 Problema de identificação em malha fechada

Considerar $r(k) \in [u(k) \ y(k)]$ como os sinais de entrada e saída gerados pelo sistema em malha fechada da figura 8.1. Os sinais são interconectados através das equações (8.5) e (8.6).

Sob as considerações citadas acima, o problema de identificação em malha fechada pode ser enunciado como:

Dado

Dadas amostras de entradas e saídas de um problema bem posto: r(k) e u(k), y(k) para k = 0,1,..., j + 2i - 2 onde j→∞.

Encontrar

- A ordem *n* do sistema
- As matrizes do sistema A_p, B_p, C_p, D_p por uma transformação de similaridade.

8.1.2 Identificação pelo Método MON4SIDC

Fazendo abuso de notação, o sistema de equações dado por (8.5) - (8.6) pode ser representado por:

$$\overline{X}(k+1) = A \ \overline{X}(k) + Br(k) + W(k)$$
(8.8)

$$\overline{Y}(k) = C\overline{X}(k) + Dr(k) + V(k)$$
(8.9)

onde:

$$\overline{X}(k) = \begin{bmatrix} s(k) \\ x_c(k) \end{bmatrix}, \quad \overline{Y}(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}$$

Aplicando-se o método MON4SID estudado no capitulo 4, encontram-se as matrizes (*A*, *B*, *C*, *D*) dadas pelas equações (8.8) e (8.9).

Dado o sinal de entrada r(k) e as matrizes do sistema em malha fechada (*A*, *B*, *C*, *D*), dadas pelas equações (8.8) e (8.9), é possível obter uma saída estimada de \overline{Y} para o sistema determinístico denotado por:

$$\widetilde{\overline{X}}(k+1) = A \ \widetilde{\overline{X}}(k) + Br(k)$$
(8.10)

$$\widetilde{\overline{Y}}(k) = C\widetilde{\overline{X}}(k) + Dr(k)$$
(8.11)

como o sinal $\tilde{\overline{Y}}(k)$ contém os dados de entrada e saída da planta sem ruído, então é fácil computar uma estimativa das matrizes da planta (A_p, B_p, C_p, D_p) . Isto é feito aplicando-se qualquer método estudado no capítulo 4 para malha aberta.

Algoritmo MON4SIDC para malha fechada

5. Coletar os dados como:

- 6. Aplicar o algoritmo MON4SID para malha aberta, obtendo as matrizes do sistema global.
- 7. Computar as matrizes da planta

8.2 Simulações

Nesta seção se apresenta um sistema simulado para avaliar o desempenho do algoritmo MON4SIDC, o qual é usado para comparar o método proposto com outros algoritmos existentes: PEM, N4SIDC, ARXS, MOESPC. N4SIDC denota aqui o algoritmo 1 de Van Overschee; De Moor, (1997) apresentado no capítulo 6, ARXS, o algoritmo de Ljung and MacKelvey (1996) e MOESPC, o algoritmo de Verhaegen (1993) apresentado no capítulo 7.

Este exemplo simulado foi usado por Verhaegen (1993), Overschee; De Moor (1997), Huang *et al.*, (2005) e Katayama (2005) na identificação em malha fechada.

O sistema é uma planta experimental de laboratório, que consiste em dois pratos circulares girados por um servomotor elétrico com um eixo flexível. Para maiores detalhes ver [HAKVOORT, 1990].

O modelo da planta é dado pelas equações (8.1) e (8.2), onde:

	4.4	1	0	0	0		0.98	[1]	2.3	
	- 8.09	0	1	0	0		12.99		0		- 6.64	
A =	7.83	0	0	1	0	, $B = 10^{-3}$	18.59	, $C^{T} =$	0	, <i>K</i> =	7.515	C D = 0
	- 4	0	0	0	1		3.3		0		- 4.0146	
	0.86	0	0	0	0		- 0.02		0		0.86336	

e(k), é ruído branco, o qual gera os distúrbios na planta.

O controlador tem uma descrição em espaço de estados da mesma forma que (8.3) e (8.4), onde:

 $A_{c} = \begin{bmatrix} 2.65 & -3.11 & 1.75 & -0.39 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{c} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, C_{C}^{T} = \begin{bmatrix} -0.4135 \\ 0.8629 \\ -0.7625 \\ 0.2521 \end{bmatrix} e D_{c} = 0.61.$

Como entrada exógena para o sistema foi considerado o sinal PRBS, o qual é persistentemente excitante. A simulação foi no Simulink do Matlab 6.5.

Foram considerados três tipos de identificação:

- a) Identificação puramente determinística, para o caso e(k)=0.
- b) Identificação com pouco ruído.
- c) Identificação com muito ruído.

8.2.1 Identificação puramente determinística

Para a identificação puramente determinística foi considerado e(k)=0, como mostra a figura 8.2. Foram coletadas 3000 amostras e i=20 (número de blocos linha da matriz de Hankel).



Figura 8.2 Sistema puramente determinístico em malha fechada

Os sinais usados na identificação são mostrados na figura 8.3.



Figura 8.3 Sinal de entrada r e saída y usados na identificação em malha fechada (puramente determinística)

Identificação em malha fechada aplicando o método MON4SIDC

Coletados os dados de entrada r (k) e saída [u(k) y(k)], o passo seguinte é encontrar a ordem do sistema global n = 9, dada pelos valores singulares mais significativos, como mostra a figura 8.4.



Figura 8.4 Valores singulares para o sistema em malha fechada (identificação puramente determinística)

Computadas as matrizes do sistema global, o passo seguinte é computar as matrizes da planta a partir do sistema em espaço de estados, dado pelas equações (8.10) e (8.11). A ordem da planta para o sistema dado pelas equações (8.10) e (8.11) é n = 5, dada pelos valores singulares mais significativos, como mostra a figura 8.5.



Figura 8.5 Valores singulares para computar a planta do sistema em malha fechada (identificação puramente determinística)

A verificação do desempenho do algoritmo foi feita através da comparação do diagrama de Bode do sistema real versus o modelo simulado e alocação de pólos dentro do círculo unitário do sistema real versus o simulado, como mostram as figuras 8.6 e 8.7.



Figura 8.6 Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método MON4SIDC)



Figura 8.7 Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método MON4SDIDC)

Da figura 8.6 pode-se observar que a resposta em freqüência do modelo obtido é similar à do sistema real. Da figura 8.7 pode-se observar que o posicionamento dos pólos do processo real e do modelo obtido são os mesmos. Destas duas figuras pode-se dizer que o modelo simula muito bem o sistema real.

As matrizes da planta simulada são:

$$Ap = \begin{bmatrix} 1.0001 & -0.1442 & -0.0001 & 0.1177 & -0.0225 \\ 0.0032 & 0.9892 & 0.3580 & 0.0457 & 0.0118 \\ -0.0005 & -0.0616 & 0.9312 & -0.2696 & 0.0830 \\ -0.0001 & -0.0007 & -0.0329 & 0.7152 & 0.6415 \\ 0.0000 & 0.0015 & -0.0170 & -0.5701 & 0.7644 \end{bmatrix}$$
$$Bp = \begin{bmatrix} -13.4967 \\ 9.6992 \\ 0.0138 \\ 0.3883 \\ 0.1307 \end{bmatrix}$$
$$Cp = \begin{bmatrix} -0.2193 & -0.3188 & 0.4639 & 0.3383 & -0.0353 \end{bmatrix}$$
$$Dp = \begin{bmatrix} -1.3411 \end{bmatrix} * e - 013$$

A identificação para o método N4SIDC foi feita com a ordem do sistema da planta n = 5. A figura 8.8 mostra o diagrama de Bode para o algoritmo 1 em malha fechada.



Figura 8.8 Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método N4SIDC)

Da figura 8.8 pode-se ver que o algoritmo N4SIDC não acompanhou o sinal real (para altas freqüências), isto não quer dizer que o método seja ruim, pois, considerando-se a ordem da planta n=7, como mostra a figura 8.9, o método teve um ótimo desempenho, como mostra a figura 8.10.



Figura 8.9 Valores singulares para computar a planta do sistema em malha fechada (identificação puramente determinística, método N4SIDC)



Figura 8.10 Comparação do diagrama de Bode da planta do sistema em malha fechada (puramente determinística, método N4SIDC)



Figura 8.11 Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método N4SIDC)

Das figuras 8.10 e 8.11 se pode dizer que o algoritmo N4SIDC teve um bom desempenho.

As matrizes da planta simulada para o método N4SIDC são:

$$An = \begin{bmatrix} 0.9752 & 0.0482 & 0.1507 & -0.0080 & -0.0053 \\ -0.0810 & 1.0565 & 0.1007 & -0.0194 & -0.0192 \\ -0.1797 & 0.0390 & 0.8919 & 0.0570 & -0.1215 \\ -0.0082 & 0.0268 & 0.0820 & 0.7697 & 0.6864 \\ 0.0034 & -0.0343 & -0.1329 & -0.5505 & 0.7095 \end{bmatrix}^{T}$$

$$Bn = [5.4253 - 3.4405 \quad 0.8688 - 1.2632 \quad 2.0890]^{T}$$

$$Cn = [-0.1965 - 0.2179 \quad 0.0556 \quad -0.0394 \quad 0.1032]$$

$$Dn = [0.0069].$$

A identificação pelo método ARXS foi feita com ordem do sistema da planta n=5. As figuras 8.12 e 8.13 mostram o diagrama de Bode e a locação de pólos, respectivamente.



Figura 8.12 Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método ARXS)



Figura 8.13 Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método ARXS)

As matrizes da planta simulada para o método ARXS são:

	1.0440	-0.3521	-0.0614	-0.3772	-0.6156	
	0.0122	1.0320	0.6419	0.2435	1.0468	
Ax =	-0.0035	-0.0311	0.6801	0.8180	0.2355	
	-0.0006	0.0019	-0.0529	0.7883	1.4025	
	0.0015	-0.0037	0.0471	-0.2390	0.8556	
						\mathcal{I}
Bx =[-3.7703	2.7729	0.5933	0.0359	-0.0989] ^T	
Cx =[-0.2429	-0.4569	0.6637	-0.3323	0.3031]	
Dx =	[0].					

Das figuras 8.12 e 8.13 se pode dizer que o algoritmo ARXS teve um bom desempenho.

A identificação para o método PEM foi feita com a ordem da planta n=5. As figuras 8.14 e 8.15 mostram o diagrama de Bode e a locação de pólos, respectivamente.



Figura 8.14 Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método PEM)

Das figuras 8.14 e 8.15 se pode dizer que o algoritmo PEM teve um bom desempenho. As matrizes da planta simulada para o método PEM são:

$$AP = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0.8600 & -4.0000 & 7.8300 & -8.0900 & 4.4000 \\ BP = \begin{bmatrix} 0.0010 & 0.0173 & 0.0868 & 0.2529 & 0.5421 \end{bmatrix}^T$$



Figura 8.15 Comparação da posição dos pólos da planta do sistema em malha fechada (identificação puramente determinística, método PEM)

A identificação para o método MOESPC foi feita com a ordem da planta n= 5. As figuras 8.16 e 8.17 mostram o diagrama de Bode e a locação de pólos, respectivamente.



Figura 8.16 Comparação do diagrama de Bode da planta do sistema em malha fechada (identificação puramente determinística, método MOESPC)





As matrizes da planta simulada para o método MOESPC são:

0.0019 1.0003 -0.0673-0.0086 0.0875 0.0681 0.0086 0.0025 0 9770 -02240Am= -0.00800.00010.9556 0.1852 0.09920.0001 -0.0083 0.0183 0.6943 0.6345 0.0000 -0.0002 0.0026 -0.5740 0.7728 Bm = [-20.8688 10.6452 $(0.0931)^T$ 4.1258 0.4315 Cm = [-0.1778 -0.2245 -0.3611 0.3902 0.0099] Dm = [-1.7126] * e-013

Das figuras 8.16 e 8.17 se pode dizer que o algoritmo MOESPC teve um bom desempenho.

Para este caso particular, sistema determinístico operando em malha fechada, todos os algoritmos tiveram um bom desempenho, mas um sistema sempre esta perturbado por ruídos. Com o objetivo de simular um sistema mais realista, foi adicionado ruído branco no processo da planta.

8.2.2 Identificação com pouco ruído

Foi adicionado ruído branco com média zero e variância 0,001 na planta do processo. Foram coletadas 3000 amostras e i = 20 (número de blocos linha da matriz de Hankel). Os sinais usados na identificação são mostrados na figuras 8.18.



Figura 8.18 Sinal de entrada *r* e saída *y* usados na identificação em malha fechada com pouco ruído

Da figura 8.18 e 8.3 é possível ver a diferença do sistema com pouco ruído e sem ruído.

A ordem do sistema global n = 7 é dada pelos valores singulares mais significativos. Os resultados da identificação são mostrados nas figuras 8.19 a 8.23.



Figura 8.19 Comparação do diagrama de Bode e alocação de pólos pelo método MON4SIDC (identificação com pouco ruído)

Do lado esquerdo da figura 8.19 pode-se observar que o modelo obtido pelo método MON4SIDC é bom em baixas freqüências e em altas se afasta um pouco, mas acompanha o sinal real. O lado direito da figura 8.19 mostra que a planta simulada é estável, pois os pólos do modelo simulado não saem do círculo de raio unitário.


Figura 8.20 Comparação do diagrama de Bode e alocação de pólos pelo método N4SIDC (identificação com pouco ruído)

O diagrama de bode, lado esquerdo da figura 8.20, mostra que o modelo obtido pelo método N4SIDC para baixas freqüências é bom, para altas freqüências se afasta. O lado direito da figura 8.20 mostra que os pólos do modelo estimado se encontram dentro do círculo de raio unitário.



Figura 8.21 Comparação do diagrama de Bode e alocação de pólos pelo método ARXS(identificação com pouco ruído)

O lado direito da figura 8.21 mostra o diagrama de Bode do modelo estimado pelo método ARXS versus o modelo real. Pode-se observar que o modelo obtido para baixas freqüências não acompanha o sinal real, para altas freqüências se afasta um pouco mas acompanha o sinal real. O modelo obtido é estável, como mostra o lado direito da figura 8.21. Observa-se que os pólos estimados não coincidem com os reais.



Figura 8.22 Comparação do diagrama de Bode e alocação de pólos pelo método PEM (identificação com pouco ruído)

Da figura 8.22 pode-se observar que o modelo PEM teve um bom desempenho. O lado direito da figura 8.22 mostra que o modelo obtido pelo método PEM é estável.



Figura 8.23 Comparação do diagrama de Bode e alocação de pólos pelo método MOESPC (identificação com pouco ruído)

O lado esquerdo da figura 8.23 mostra a comparação da resposta em freqüência do processo da planta versus o modelo obtido pelo método MOESPC. Pode-se observar que para este caso o modelo obtido para baixas e altas freqüências se afasta. O lado direito da figura 8.23 mostra que o modelo obtido é estável. Este modelo teve melhor desempenho para ordem do sistema n=9 como mostra a figura 8.24. Esta figura mostra que o modelo obtido para baixas freqüências se afasta.



Figura 8.24 Comparação do diagrama de Bode para n = 9 pelo método MOESPC (identificação com pouco ruído)

Com o objetivo de validar o desempenho do algoritmo MON4SIDC, agregou-se mais ruído na planta do processo.

8.2.3 Identificação com muito ruído

Foi adicionando ruído branco com média zero e variância 0,01 na planta do processo. Foram coletadas 3000 amostras e i = 20. Os sinais usados na identificação são mostrados na figura 8.25.

A ordem do sistema global n = 7 é dada pelos valores singulares mais significativos. Os resultados da identificação são mostrados nas figuras 8.26 - 8.30.



Figura 8.25 Sinal de entrada *r* e saída *y* usados na identificação em malha fechada com muito ruído



Figura 8.26 Comparação do diagrama de Bode e alocação de pólos para método MON4SIDC (identificação muito ruído)

Do lado esquerdo da figura 8.26 pode-se observar que o modelo obtido pelo método MON4SIDC para baixas freqüências é bom, para altas freqüências se afasta. O lado direito da figura 8.26 mostra que o modelo obtido é estável.



Figura 8.27 Comparação do diagrama de Bode e alocação de pólos para método N4SIDC (identificação muito ruído)

O modelo obtido pelo método N4SIDC é instável como mostra o lado direito da figura 8.27. O modelo obtido tenta caracterizar a dinâmica do planta em baixas freqüências, para altas freqüências se afasta.



Figura 8.28 Comparação do diagrama de Bode e alocação de pólos para método ARXS (identificação muito ruído)

O modelo obtido pelo método ARXS não acompanha a dinâmica do sistema em baixas ou altas freqüências. O modelo obtido é instável como mostra a figura 8.28.

Do lado esquerdo da figura 8.29 pode-se observar que o modelo obtido pelo método PEM tenta acompanhar a dinâmica da planta, porém para altas freqüências se afasta. O modelo obtido é instável, pois existe um pólo que não está dentro do círculo de raio unitário, como mostra o lado direito da figura 8.29.



Figura 8.29 Comparação do diagrama de Bode e alocação de pólos para método PEM (identificação muito ruído)



Figura 8.30 Comparação do diagrama de Bode e alocação de pólos para método MOESPC (identificação muito ruído)

O modelo obtido pelo método MOESPC é marginalmente estável como mostra o lado direito da figura 8.30. O lado esquerdo desta figura mostra que o modelo obtido não caracteriza a dinâmica da planta. Das figuras 8.26 a 8.30 se pode ver que o algoritmo MON4SIDC teve o melhor desempenho. Observa-se que os pólos dados pelos algoritmos PEM e ARXS saem do círculo de raio unitário.

CAPÍTULO 9

CONCLUSÕES E FUTURAS PESQUISAS

Neste trabalho se estudou o método de identificação por subespaços para sistemas discretos invariantes no tempo, que trabalham em malha aberta e em malha fechada. Em ambos casos foi feito um estudo teórico dos métodos MOESP e N4SID.

A seguir, na seção 9.1 se apresentam as conclusões para sistemas em malha aberta, na seção 9.2 para sistemas em malha fechada e em 9.3 se apresentam as contribuições desta tese e, por último, na seção 9.4, se enunciam possíveis pesquisas que poderiam ser feitas no futuro.

9.1 Para sistemas em malha aberta

- A teoria da identificação por subespaços foi apresentada em detalhe com as demonstrações dos teoremas principais para cada caso: identificação determinística e combinação determinística - estocástica. Isto foi feito para o método MOESP e método N4SID.
- Com base no item 1 citado acima, observa-se que o método MOESP considera a matriz de Hankel dos dados de entrada e saída da forma [U^T_f U^T_p Y^T_p Y^T_f]^T e para o método N4SID a matriz de Hankel dos dados de entrada e saída da forma [U^T_p U^T_f Y^T_p Y^T_f]^T.
- 3. Tentar apresentar um novo método, mistura dos métodos MOESP e N4SID, usando suas respectivas matrizes de Hankel, levaria a um maior esforço computacional e tempo de processamento. Neste trabalho se apresenta uma alternativa ao problema citado na frase anterior, através do método MON4SID o qual considera a matriz de Hankel dos dados de entrada e saída da forma

 $[U_f^T U_p^T Y_p^T Y_f^T]^T$ (como no método MOESP), a partir da qual são computadas as projeções oblíquas (como no método N4SID).

- 4. No método N4SID é necessário computar duas projeções oblíquas para recuperar as matrizes do sistema, portanto, é necessário ter duas condições iniciais, isto pode levar a problemas de polarização. Para solucionar este problema, no método proposto MON4SID se computa uma projeção oblíqua, chamada aqui de seqüência de estados totais, a qual tem uma só condição inicial. Desta seqüência são computadas as seqüências de estados passados e de estados futuros. As matrizes do sistema (*A*,*B*,*C*,*D*) são computadas através do método de mínimos quadrados.
- 5. O número de linhas *i* da matriz de Hankel dos dados de entrada e saída é dado pelo usuário e está diretamente relacionado com o número de dados coletados. Idealmente, se os dados são infinitos, o sistema identificado simula muito bem o sistema real, mas na prática isso não acontece. Então, considerando diferentes valores para *i*, é possível obter um melhor modelo.
- 6. Com o objetivo de comparar o desempenho dos algoritmos apresentados neste trabalho, foram usados nove algoritmos em três processos diferentes e para se avaliar a qualidade dos modelos, aplicam-se indicadores de desempenho. Os indicadores de desempenho mais usados são média relativa do erro quadrático (MRSE), médio da variância relativa (MVAF) e o FIT. De acordo com FIT, os modelos obtidos pelo método MON4SID tiveram um bom desempenho.
- 7. Querer fazer uma comparação das matrizes do sistema real com o sistema simulado para o caso MIMO, não é recomendável, pois um mesmo sistema pode ter muitas representações em espaços de estados, além disso, nem sempre o melhor modelo simulado tem que ter a mesma ordem do modelo real.

9.2 Para sistemas em malha fechada

- Tentar aplicar métodos de identificação em malha aberta para identificar processos em malha fechada não é recomendado, pois os dados de entrada da planta estão correlacionados com os dados de saída.
- 2. Existem diversos métodos de identificação por subespaços aplicados a malha fechada, estes métodos diferem uns dos outros pela configuração do sistema e pela implementação dos algoritmos. Como foi visto nos capítulos 6 e 7, é possível fazer uma extensão dos métodos de identificação por subespaços para malha fechada, a partir dos métodos para malha aberta. Por exemplo, para aplicar o método N4SID para malha fechada é necessário ter algum conhecimento sobre o controlador. No caso do método MOESP é necessário conhecer a ordem do controlador, logo de um sistema global são obtidas as matrizes da planta e do controlador através de uma redução de ordem.
- A teoria da identificação por subespaços para sistemas operando em malha fechada foi apresentada em detalhes com as demonstrações dos teoremas principais para o caso de MOESP e N4SID.
- 4. Para solucionar o problema citado no item 2, apresentou-se o método MON4SIDC, o qual usa o método MON4SID para identificar a planta de um sistema operando em malha fechada e seus resultados são comprados com outros algoritmos existentes na literatura: PEM, N4SIDC, ARXS e MOESPC. O algoritmo proposto tem vantagem sobre o método N4SIDC no sentido que não precisa ter algum conhecimento do controlador e sobre o modelo MOESP, fazer redução de ordem para computar a planta do sistema em malha fechada.
- 5. O sistema de uma planta experimental de laboratório, que descreve dois pratos circulares girados por um servomotor elétrico com um eixo flexível é muito usado por diversos pesquisadores como: Verhaegen (1993), Overschee; De Moor (1997), Huang, *et al.*, (2004), Katayama (2005), entre outros. Este exemplo é usado neste trabalho para fazer a identificação de um sistema em

malha fechada através do método proposto. Para este sistema, foram considerados três tipos de identificação: identificação puramente determinística (sistema sem ruído), identificação com pouco ruído (ruído com variância 0,001) e identificação com muito ruído (ruído com variância 0,01). Para verificar o desempenho dos algoritmos foi feita a comparação do diagrama de Bode do sistema real versus o modelo simulado e a alocação de pólos do sistema real versus o simulado. No caso do sistema puramente determinístico, considerando ordem da planta do sistema n = 5, todos os modelos tiveram um ótimo desempenho, exceto para o método N4SIDC. Isto não quer dizer que o modelo seja ruim, pois pode acontecer que para outra ordem o modelo tenha um bom desempenho (neste caso n = 9). No caso de pouco ruído, com ordem da planta n= 7, não tiveram bom desempenho os algoritmos ARXS e MOESPC, mas com n=9 o método MOESPC teve melhor desempenho. Na identificação com muito ruído, os métodos N4SIDC, ARXS e PEM apresentaram pólos instáveis (fora do círculo unitário). Para este caso particular, o método MON4SIDC apresentou um bom desempenho em relação à planta do sistema real em malha fechada.

6. A desvantagem do método MON4SIDC é que não estima diretamente a ordem da planta, mas sim uma ordem do sistema global. A ordem da planta é estimada a partir do sistema simulado dos dados de entrada e saída da planta.

9.3 Contribuições deste trabalho

- 1. Os métodos MOESP e N4SID foram apresentados de forma didática para os diferentes tipos de identificação.
- 2. Foram feitas demonstrações dos diferentes teoremas, para os métodos de identificação por subespaços para os métodos MOESP e N4SID. Cabe ressaltar que alguns teoremas se encontram demonstrados nos artigos destes métodos de forma rápida, no entanto, nesta tese, estes mesmos teoremas são demonstrados de forma detalhada. Outros teoremas que somente são enunciados nos artigos sem demonstração, foram aqui demonstrados, como por exemplo: o teorema principal de identificação para malha fechada pelo método N4SIDC, teorema do sistema global usado na identificação em malha fechada pelo método MOESPC, etc.

- 3. Apresentação dos métodos MON4SID (três algoritmos) como mistura dos métodos MOESP e N4SID para sistemas puramente determinísticos.
- 4. Apresentação dos métodos MON4SID (dois algoritmos) como mistura dos métodos MOESP e N4SID para um sistema determinístico estocástico.
- 5. Apresentação do modelo MON4SIDC como mistura do modelo MOESP e N4SID para identificar a planta de um sistema operando em malha fechada.
- 6. Apresentação da implementação em Matlab 6.5 dos diferentes algoritmos citados na tese.

9.4 Trabalhos futuros

- Os algoritmos de identificação por subespaços descritos neste trabalho são do tipo caixa preta, no sentido que nenhum conhecimento "a priori" do processo pode ser incluído. Entretanto, na prática, muitas características do processo são freqüentemente conhecidas (integrador, estabilidade, ganho em regime estacionário, etc). A possibilidade para incluir estes conhecimentos nos algoritmos de identificação por sub-espaços realçaria extremamente seu valor prático.
- 2. A maioria dos sistemas tem um comportamento não linear, exceto em uma determinada faixa de operação, onde pode ser considerado linear. Modelos lineares aproximam sistemas não lineares ao redor de um ponto de operação. O desempenho do modelo linear (isto é, suas características preditivas) é deteriorado, caso se varie o ponto de operação do sistema. Para descrever globalmente o comportamento do sistema deve-se recorrer a modelos não lineares. Seria muito interessante poder fazer um estudo de identificação de sistemas usando-se o método de identificação por subespaços em malha aberta para sistemas não lineares.
- 3. Estudar a identificação por subespaços em malha fechada com os métodos de identificação usados para sistemas não lineares.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1. AGUIRRE, A. L. Introdução à Identificação de sistemas técnicas lineares e não lineares aplicadas a sistemas. Belo Horizonte, Editora UFMG, 2004.
- AKAIKE, H. Information theory and an extension of the maximum likelihood principle. In: Second International Symposium on Information Theory, Budapest, Hungary. Petrov, B.N.; Csaki, F.; Eds., p. 267-281, 1973.
- 3. BACHMAN, G; NARICE, L. Funtional Analysis, Canada, Ed. General Publishing Company, Ltd., 30 Lesmill Road, Don Mills, Toronto, Ontario.2000.
- 4. BAUER D. Subspace algorithms. In Proceedings of the 13th IFAC SYSID Symposium, p. 1030-1041, Rotterdam, NL, Aug 2003.
- 5. BAUER, D. Asymptotic properties of subspace estimates. Preprint, Technical University, Wien, 2004.
- 6. BAUER, D.; LJUNG, L. Some facts about the choice of the weighting matrices in larimore type of subspace algorithms. Automatica, v. 38, p. 763–773. 2002.
- 7. BOTTURA, P. Análise Linear de Sistemas. Editora Guanabara dois, 1982.
- 8. BOX, P.; MACGREGOR, J. Parameter estimation with closed loop operating data, Technometrics v. 18, p. 371-380. 1976.
- CAINES, E.; CHAN, W. Estimation, Identification and Feedback. In Advances in Systems Identification (R. K. Mehra, D.G. Lainiotis, Eds), Academic Press, New York. 1976
- CAINES, E.; CHAN, W. Feedback betweem stationary stochastic processes. IEEE Trans AC-20. p. 498-508. 1975
- 11. CAINES, L. Linear Stochastic Systems. Jhon Wiley and Sons, New York, 1998.
- CHIUSO, A.; PICCI, G. Geometry of oblique splitting, minimality and hankel operators. Lecture notes in control and Information Sciences, v. 286, p. 85–124. Berlin: Springer, 2002.
- 13. CHIUSO, A.; PICCI, G. Consistency analysis of some closed-loop subspace identification methods. Proceedings of the 43rd IEEE Conference on Decision & Control. Paradise Island, the Bahamas, 2004.
- 14. CHIUSO, A.; PICCI, G. Consistency analysis of some closed-loop subspace identification methods. Automatica, v. 41, n. 3, p. 377-391. March 2005.
- 15. CHIUSO, A.; PICCI, G. Subspace identification random processes with feedback. In Proceedings of the IFAC international symposium on system identification (SYSID), Rotterdam, 2003.
- 16. CHIUSO, A.; PICCI, G. The asymptotic variance of subspace estimates. J. Econometrics, v. 118, p. 257-291, 2004.
- 17. CHOU, C; VERHAEGEN, M. Closed-loop identification using canonical correlation analysis. In Proceedings of the European control conference, Karlsruhe (pp. CD-ROM, F162), 1999.

- CHOU, C; VERHAEGEN, M. Subspace algorithms for the identification of multivariable dynamic errors-in variables models. Automaica, v. 33, n. 10, p. 1857-1869, 1997.
- 19. CHUI, C.; MACIEJOWSKI. M. Subspace identification a Markov parameter approach, International Journal of Control, v. 78, n. 17, p. 1412 1436, December 2005.
- COOT, B. Introduction to the Process Identification Workshop at the 1992 Canadian Chemical Engineering Conference, Journal of Process Control, v. 5, n. 2, p. 67-69. 1995
- DE COCK, K.; DE MOOR, B. Subspace identification methods, in Contribution to section 5.5, Control systems robotics and automation of EOLSS, UNESCO Encyclopedia of Life Support Systems, Unbehauen H.D., v. 1 of 3, Eolss Publishers Co., Ltd. Oxford, UK, p. 933-979, 2003.
- 22. DE MOOR B. Mathematical concepts and techniques for modeling of static and dynamic systems. PhD thesis, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium, 1988.
- 23. DE MOOR, B., Total linear least squares with inequality constraints, Internal Report 90-02, ESAT-SISTA, K.U.Leuven, Belgium, 1990.
- 24. DE MOOR, B.; VAN OVERSCHEE, P.; FAVOREEL, W. Numerical algorithms for subspace state space system identification an overview", Birkhauser Book series on Applied and Computational Control, signals and Circuits, Biswa Datta Editor, p. 247-311, 1999.
- FAVOREEL, W; DE MOOR, B., VAN OVERSCHEE, P. Subspace state space system identification for industrial processes, Journal of Process Control, v. 10, p. 149-155. 2000
- FORSSELL, U.; LJUNG, L. Closed-loop identification revisited. Automatica, v. 35, p. 1215-1241, 1999.
- 27. FRANCIS, B. A. A course in H_{∞} control theory. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin. 1987.
- 28. GARCIA, C. 2001 Identificação de sistemas, PEE material da disciplina pósgraduação, 2001.
- 29. GEORGE, B.; LAWRENCE, N. Functional Analysis, Canada, Ed. General Publishing Company, Ltd., 30 Lesmill Road, Don Mills, Toronto, Ontario.2000.
- 30. GOPINATH, B. On the identification of linear time-invariant systems from inputoutput data, Bell Syst. Tech. J., v. 48, n. 5, p. 1101–1113, 1969
- 31. GUSTAVSSON, T. Subspace-based system identification: Weighting and prefiltering of instruments. Automatica, v. 38, p. 433–443. 2002.
- 32. GUSTAVSSON, T.; LJUNG. L.; SÖDERSTROM, T. Identification of processes in closed loop-Identifiability and accuracy aspects, Identification and Systems parameter Estimation, p 41-77.1977.
- HAKVOORT, R. Approximate identification in the controller design problem, Master Thesis, Delft University of Technology, Measurement and control theory section. Mech. Eng., A-538, 1990

- 34. HEALY, M. J. R. Matrices for statistics, Clarendon Press, New York, Oxford University Press, 1986.
- 35. HJALMARSSON, H. From experiments to closed loop control. In Proceeding of the 13th IFAC SYSID Symposium, pages 1-14, Rotterdam, NL, Aug 2003.
- 36. HO B. L; KALMAN, R. E. Efficient construction of linear state variable models from input/output functions. Regelungstechnik, v. 14, p. 545-548,1966.
- 37. HUANG, B.; STEVEN X.; QIN, J. Closed-loop subspace identification: an orthogonal projection approach, Journal of Process Control, v. 15, n. 1, p. 53-66, February 2005.
- JANSSON, M. Subspace identification and arx modelling. In Proceedings of SYSID 2003, Rotterdam, 2003.
- 39. JANSSON, M.; WAHLBERG, Bo. A linear regression approach to state-space subspace system identification. Signal Processing, v. 52, p.103-129, 1996.
- 40. JANSSON, M.; WAHLBERG, Bo. On consistency of subspace methods for system identification. Automatica, v. 34. p. 1507-1519, 1998.
- 41. KAILATH, T; SAYED, A. H. Fast realizable algorithms for matrices with structure, society of Industrial and Applied mathematics, 1999.
- 42. KATAYAMA, T. Subspace Identification of Closed-Loop Systems, SICE 2002. Proceedings of the 41st SICE Annual Conference, v. 3, p. 1517-1520. Aug, 2002.
- 43. KATAYAMA, T.; KAWAUCHI, H.; PICCI, G. Subspace identification of closed loop systems by the orthogonal decomposition method. Automatica, v. 41, n. 5, p. 863-872., 2005.
- 44. KATSUHIKO, O. Modern control engineering, 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 1997.
- 45. KLAUW, A; VERHAEGEN, M.; BOSCH, P. State space identification of closed loop systems. In Proceedings of the 30th IEEE conference on decision & control, Brighton, UK, p. 1327–1332, 1991.
- 46. KNUDSEN, T. Consistency analysis of subspace identification methods based on linear regression approach. Automatica, v. 37, p. 81-89, 2001.
- KOVÁCS, Z. L. Teoria da probabilidade e processos estocásticos Aplicações em Engenharia de Sistemas e Processamento de Sinais, DEDALUS-Acervo-EPEL, 1996.
- KUNG, S. Y. A. New identification method and model reduction algorithm via singular value decomposition, 12th Asilomar Conf. on Circuits, Systems and Comp., p. 705714, Asilomar, CA, 1978.
- 49. LARIMORE, W. Canonical variate analysis in identification, filtering and adaptive control. In IEEE Conference on Decision and Control, p. 596-604, Dec 1990.
- LARIMORE, W. System Identification, Reduced-Order Filtering and Modeling Via Canonical Variate Analysis. Proceedings of the American Control Conference, ACC, San Francisco, p. 445–451.1983.

- LIN, W.; QIN, J.; LJUNG, L. A Framework for Closed-loop Subspace Identification with Innovation Estimation. Texas Wisconsin Modeling and Control Consortium, Tech. Rep. 07, 2004.
- 52. LIPSCHUTZ, S. Algebra linear, 3 ed., São Paulo : Pearson Makron Books, 2004.
- LJUNG, L. Aspects and experiences of user choices in subspace identifications methods, In 13th IFAC Symposium on System Identification, 2003. p. 1802-1807, 2003.
- 54. LJUNG, L. Asymptotic variance expression for identified black-box tranfer function models. IEEE Trans. Auto. Cont., v. 30, p. 834-844, 1985.
- 55. LJUNG, L. Convergence analysis of parametric identification methods. IEEE Trans. Automatica. Cont., 23: p.770-783, 1978.
- 56. LJUNG, L. System identification Theory for the User", Englewood Cliffs, Prentice Hall, NJ, c 1999.
- 57. LJUNG, L.; MCKELVEY, T. Subspace identification from closed loop data. Signal Processing, v. 52, n. 2, p. 209–216, 1996.
- MERCÈRE, G., BAKO, L.,; LECŒUCHE, S. Propagator-based methods for recursive subspace model identification, Signal Processing, v. 88, n. 3, p. 468-491. 2008
- 59. MIRANDA, S. Estudo de identificação por sub-espaços e aplicação a um processo industrial, Dissertação de Mestrado, Laboratório de Automação e Controle- Depto. De Engenharia de Telecomunicações e Controle- Escola Politécnica da Universidade de São Paulo, 2004.
- 60. MOONEEN, MAC; DE MOOR, B.; RAMOS, J., TAN, S. A subspace identification algorithm for descriptor systems, System & Control Letters, v. 9, p. 47-52, 1992.
- 61. OVERSCHEE, P., DE MOOR, B. Subspace algorithms for the stochastic identification problem. Automatica, Vol. 29, no. 3, pp. 649-660, 1993.
- 62. PAPOULIS, A. Probability, random variables, and stochastic processes, McGrawHill, NY, 1984.
- 63. PEREIRA DA SILVA, P. S. Controle linear multivariável, Apostila da disciplina controle multivariável de pós-graduação, 2003.
- 64. PHADKE, M. S.; S. M. Wu. Identification of multiinput-multioutput transfer function and noise model of a blast furnace from closed-loop data. IEEE Trans. Aut.Control, v. 19, p. 944-951. 1974.
- 65. QIN, J. Computers & Chemical Engineering, v. 30, n. 10-12, p. 1502-1513, 2006.
- 66. QIN, S.; LJUNG, L. Closed-loop subspace identification with innovation estimation. In Proceedings of SYSID 2003. Rotterdam, 2003.
- 67. ROBERTO, P., KURKA, G., CAMBRAIA, H. Application of a multivariable input-output subspace identification technique in structural analysis, Journal of Sound and Vibration, v. 312, n. 3, p. 461-47. 2008
- 68. SÖDERSTROM, T.; STOICA, P. System identification, International Series in Systems and Control Engineering, Prentice Hall, New York, 1989.

- 69. STRANG, GILBERT. Introduction to linear algebra, Wellesley-Cambridge Press, 1993.
- 70. VAN DEN HOF System Identification. Lecture notes. Delft University of technology, 2004.
- 71. VAN DER KLAUW, C.; VERHAEGEN, M; VAN DE BOSCH, J. State space identification of closed loop systems. Proceedings of the 30th IEEE Conference on Decision & Control, p. 1327-1332, Brighton, 1991.
- VAN DER VEEN, J.; DEPRETTERE, F; SWINDLEHURST, A. Subspace based signal analysis using singular value decomposition, Proc. IEEE, v. 81, n. 9, p. 1277-1308, 1993.
- 73. VAN OVERSCHEE, P.; DE MOOR B. Subspace identification for linear systems, Kluwer Academic Publishers, 1996.
- 74. VAN OVERSCHEE, P.; DE MOOR, B. A unifying theorem for three subspace system identification algorithms. Automatica, Special Issue on Trends in System Identification, v. 31, n. 12, p. 1853-1864, 1995.
- 75. VAN OVERSCHEE, P.; DE MOOR, B. Closed loop subspace systems identification. In Proceedings of 36th IEEE conference on decision & control, San Diego, CA, p. 1848–1853, 1997.
- 76. VAN OVERSCHEE, P.; DE MOOR, B. N4SID: Subspace algorithms for the identification of combined deterministic stochastic systems. Automatica, special issue on statistical signal processing and control, v. 30, n. 1, p. 75-93, 1994a.
- 77. VAN OVERSCHEE, P.; DE MOOR, B. N4sid: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, v. 30, p. 75-93, 1994.
- 78. VAN OVERSCHEE; DE MOOR, B. About the choice of State Space Basis in Combined Deterministic-Stochastic Subspace Identification, Automatica, Special Issue on Trends in System Identification, v. 31, n. 12, p. 1877-1883, 1994b.
- 79. VAN WINGERDEN, J. W., & VERHAEGEN, M. Subspace identification of MIMO LPV systems: The PBSID approach. In Proceedings of the 47th IEEE conference on decision and control. 2008a
- 80. VAN WINGERDEN, J. W., & VERHAEGEN, M. Subspace identification of multivariable LPV systems: A novel approach. In Proceedings of the IEEE multi-conference on systems and control. 2008b.
- 81. VAN WINGERDEN, J. W., FELICI, F., & VERHAEGEN, M. Subspace identification of MIMO LPV systems using piecewise constant scheduling sequence with hard/soft switching. In Proceedings of the European control conference. 2007
- 82. VAN WINGERDEN, J. W., HOUTZAGER, I., FELICI, F., & VERHAEGEN, M. Closedloop identification of the time-varying dynamics of variable-speed wind turbines [Special issue on Wind turbines: New challenges and advanced control solutions]. International Journal of Robust and Nonlinear Control, v. 19, n. 1, p. 4_21. 2009
- 83. VERHAEGEN, M. Application of a subspace model identification technique to identifylti systems operating in closed-loop. Automatica, v. 29, p. 1027–1040, 1993.

- 84. VERHAEGEN, M. Application of a Suspace Model Identification Technique to identify LTI systems operating in clossed loop, Automatica, v. 29, n. 4, p. 1027-1040. 1993c.
- VERHAEGEN, M. Identification of MIMO output-error state space models, Techincal Report N90.08, Delft University of Technology, Network Theory Section. 1990.
- 86. VERHAEGEN, M. Identification of the deterministic and stochastic part of MIMO state space model under the presence of process and measurement noise, In Proc. Of the European Control Conference 1993, Groningen, The Netherlands. 1993b.
- 87. VERHAEGEN, M. Identification of the deterministic of MIMO state space models give in innovations form input-output data, In the Special Issue on Statistical Signal Processing and Control of Automatica, v. 30, n. 1, p. 61-74. 1994a
- VERHAEGEN, M. Subspace Model Identification. Part III: Analysis of the ordinary Output-Error State Space Model identification algorithms. Int. J. Control, v. 58, n. 3, p. 555-586. 1993a.
- 89. VERHAEGEN, M.; DEPRETTERE, E. Subspace Model Identification, In, E. F. Deprettere and A. J. van der Veen (Eds): Algorithms and Parallel VLSI architectures, v. B, p. 13-32. Elservier, Amsterdam, 1991.
- 90. VERHAEGEN, M.; DEWILDE, P. Subspace model identification. part i: the output-error state-space model identification class of algorithms. International Journal of Control, v. 56 p. 1187-1210, 1992.
- 91. VERHAEGEN, M.;PATRICK DEWILDW P. Subspace Model Identification. Part I: The Output-Error State Space Model identification class of algorithms. Int. J. Control, v. 56, n. 5, p. 1187-1210. 1992a.
- 92. VERHAEGEN, M.;PATRICK DEWILDW P. Subspace Model Identification. Part II: Analysis of the elementary Output-Error State Space Model identification algorithms. Int. J. Control, v. 56, n. 5, p. 1211-1241,1992b.
- 93. WANG, J.; JIYA, J.; LIU, X.; YAO, Y; WANG, G; XU, S. Vehicle Electronics Conference, (IVEC'99) Proceeding of the IEEE international, v. 1, p. 342-351, 999
- 94. WEILU LIN, B. Closed loop Subspace Identification and Fault Diagnosis with Optimal Structured Residuals. University of Texas at Austin, 2005.
- 95. ZEIGER, H.; MCEWEN, A. Approximate Linear Realizations of Given Dimension via Ho's Algorithm. IEEE Transactions on Automatic Control, v. 19, p. 153, 1974.
- ZHAO, Y.; WESTWICK, DT. Closed-loop system identification using subspacebased methods, Proc. IEEE Canadian ECE Conf. Montreal, Quebec, p. 1727-1730, 2003.
- 97. ZHU, Y. Multivariable process identification for MPC: The asymptotic method and its applications. Journal of Process Control, v. 8, n. 2, p. 101–115. 1997

APÊNDICE A

INTRODUÇÃO A ESPAÇOS VETORIAIS

O objetivo deste capítulo não é se aprofundar em conceitos matemáticos de análise funcional, demonstrações de teoremas etc, mas sim, apresentar os conceitos fundamentais necessários para poder entender os capítulos desta tese.

Neste capítulo define-se o seguinte: dado um conjunto F quaisquer que satisfaça certas propriedades de adição e produto, então o conjunto F é um corpo. Dado um conjunto de vetores V que satisfaça oito propriedades sobre um corpo F, então V é chamado de espaço vetorial. Sendo V um espaço vetorial, pode-se perguntar que conjunto gera o espaço V; suponha que seja B tal conjunto (B subconjunto de V), então o conjunto B, cujos elementos são linearmente independentes, chama-se de base do espaço vetorial V. Além disso, seja n o número de elementos da base, então a dimensão de $V \notin n$. Se no espaço vetorial V define-se uma norma $\|$ $\|$, então V é chamado de espaço vetorial normado. Isto permite calcular a magnitude de um vetor. Dados os elementos do espaço vetorial normado, deseja-se saber se estes elementos são ortogonais, então é preciso definir a função produto interno denotada por (•,•). Assim, dois elementos do espaço vetorial são ortogonais se seu produto interno é igual a zero. Sejam Agora dois espaços vetoriais V e W define-se transformação linear à aplicação T que leva vetores de V a W via T. Se T está definido de V em V, então T é chamado de operador (exemplos: matriz, derivada, projeções ortogonais, projeções oblíquas etc). Todos estes conceitos são apresentados a seguir.

A.1 Corpo

Definição A.1.1 Um *Corpo* consiste em um conjunto F de elementos chamados escalares e duas operações chamadas adição e multiplicação definidas sobre F que satisfazem as seguintes propriedades.

Adição ou soma

$$+: FxF \to F$$

(\alpha, \beta) \dots +(\alpha, \beta) = \alpha + \beta

A todo par de escalares $\alpha e \beta$ em *F* corresponde um escalar $\alpha + \beta$ em *F* chamado soma de $\alpha e \beta$ tal que:

- C.1 $\alpha + \beta = \beta + \alpha$
- C.2 $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$

C.3 Existe um elemento neutro denotado por 0, tal que $\alpha + 0 = 0$ C.4 Existe um único elemento $\beta = -\alpha \in F$ tal que $\alpha + \beta = 0$

Produto

•:
$$FxF \rightarrow F$$

 $(\alpha, \beta) \mapsto \bullet(\alpha, \beta) = \alpha\beta$

Para quaisquer escalares $\alpha \in \beta \in F$ corresponde um escalar $\alpha\beta \in F$ chamado produto de $\alpha \in \beta$ tal que:

C.5 $\alpha\beta = \beta\alpha$

C.6
$$(\alpha\beta)\gamma = \alpha(\beta\gamma)$$

C.7 Existe um elemento não nulo 1 tal que: $1\alpha = \alpha$

C.8 $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$

Para todo $\gamma \in F$.

Exemplo A.1.1 Considere o conjunto de números reais \Re ou o conjunto dos números complexos, isto é \aleph seja $F = \Re$ (ou $F = \aleph$) com as definições usuais de soma e produto, então \Re (ou \aleph) é um corpo.

A.1.2 Espaço Vetorial

Definição A.1.2 Um *Espaço Vetorial* ou espaço linear ou espaço vetorial linear sobre um corpo F consiste em um conjunto não vazio, V, de elementos chamados vetores e de duas operações chamadas adição vetorial e multiplicação por um escalar, definidas sobre $V \in F$, conforme indicado a seguir:

Multiplicação por um escalar $V: VxV \rightarrow V$ $V: FxV \rightarrow V$ $(v, w) \mapsto V(v, w) = v + w$ $(\alpha, v) \mapsto V(\alpha, v) = \alpha v$ Se $v, w \in V$, então $v + w \in V$; Se $v \in V$ e $\alpha \in F$, então $\alpha v \in V$

O conjunto V é um espaço vetorial sobre F se satisfaz os seguintes axiomas:

- V.1 v + w = w + v
- V.2 v + (w + u) = (v + w) + u
- Existe um elemento $\overline{0} \in V$, tal que $v + \overline{0} = \overline{0} + v = v$, V.3
- Para cada $v \in V$, existe um elemento $-v \in V$ tal que $v + (-v) = (-v) + v = \overline{0}$ V.4

V.5
$$\alpha(\beta)v = (\alpha\beta)v$$

- V.6 $\alpha(v+w) = \alpha v + \alpha w$
- V.7 $(\alpha + \beta)v = \alpha v + \beta v$

V.8 1v = v

Para todo v, w e $u \in V$; $\alpha, \beta \in F$.

Um espaço vetorial V sobre um corpo F será denotado por (V, F) ou simplesmente por V, sempre que possível.

Exemplo A.1.2 O espaço (\Re^n, \Re) é um espaço vetorial sobre \Re , com a adição de vetores e a multiplicação por escalar definida como:

$$(x_1, x_2, x_3, \dots, x_n) + (y_1, y_2, y_3, \dots, y_n) = (x_1 + y_1, x_2 + y_2, x_3 + y_3, \dots, x_n + y_n) \quad e$$

$$\alpha(x_1, x_2, x_3, \dots, x_n) = (\alpha \ x_1, \alpha \ x_2, \alpha \ x_3, \dots, \alpha \ x_n) \quad \text{onde } \alpha, x_i \ e \ y_i \in \Re.$$

onde $v = (x_1, x_2, x_3, \dots, x_n) \ e \ w = (y_1, y_2, y_3, \dots, y_n) \in \Re^n.$

Exemplo A.1.3 Seja o espaço de matrizes $M_{m,n}$, que está formado pelo conjunto de todas as matrizes mxn sobre um corpo arbitrário F. Então M_{mn} é um espaço vetorial sobre F em relação às operações usuais de adição de vetores e multiplicação por um escalar.

Observe que os exemplos A.1.2 e A.1.3 são espaços vetoriais, pois satisfazem as oito

Adição vetorial

propriedades desde V1 até V8.

A.1.3 Subespaço Vetorial

Definição A.1.3 Seja (V, F) e W um subconjunto de V $(W \subset V)$.

W é um *Subespaço vetorial* de V, se W é ele próprio um espaço vetorial sobre F em relação à adição de vetores e à multiplicação por escalar em V.

O seguinte teorema e corolário gera um critério simples para identificar subespaços.

Teorema A.1.1: Um subconjunto W não vazio, de um espaço vetorial V é um subespaço vetorial de V se estiverem satisfeitas as seguintes condições.

i) Para todo $u \in w \in W \implies u + w \in W$

ii) Para todo $\alpha \in F$ e $u \in W \implies \alpha u \in W$.

Corolário A.1.1 se V é um espaço vetorial e $W \subset V$, então W é subespaço vetorial de V se e somente se:

i)
$$0 \in W$$
 (ou $W \neq 0$) e

ii) $\alpha u + \beta t \in W$ para todo $u, t \in W$ e $\alpha, \beta \in F$

Observação A.1.1

Todo espaço vetorial *V* admite pelo menos dois subespaços vetoriais: o subespaço nulo e o próprio espaço vetorial *V*. Estes subespaços são chamados de subespaços triviais. Os demais subespaços, se existirem, são chamados *subespaços próprios*.

Exemplo A.1.4 Considere o sistema homogêneo AX = 0, onde $A \in \Re^{3x3}$ e $X \in \Re^3$. Seja $G = \{X^T = [x_1 \ x_2 \ x_3]$, $x_1, x_2 \in x_3 \in R\}$ o conjunto de todas as soluções homogêneas. Então G é um subespaço vetorial de \Re^3 .

Exemplo A.1.5 O conjunto \Re^2 , não é um subespaço de \Re^3 , pois \Re^2 não é um subconjunto de \Re^3 .

Exemplo A.1.6 Seja $V=M_{m,n}$ o espaço de matrizes m x n. Então o subconjunto W_1 , de matrizes triangulares (superior) e o subconjunto W_2 de matrizes simétricas são subespaços de V, pois não são-vazios e são fechados em relação à adição matricial e à multiplicação por escalar.

A.1.4 Combinação Linear

Definição A.1.4 Sejam $\{v_1, v_2, ..., v_n\}$ vetores em um espaço vetorial *V*. Diz-se que um vetor $v \in V$ é *combinação linear* de $v_1, v_2, ..., v_n$ se existem escalares $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n \in F$ tal que

$$v = \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n = \sum_{i=1}^n \alpha_i v_i$$

Exemplo A.1.6 Considere os seguintes vetores de \Re^2 , $v_1 = (3,1)$ e $v_2 = (2,0)$. Então $v=(1,2) \in \Re^2$ é uma combinação linear de v_1 e v_2 . com efeito:

 $v = \alpha_1(3,1) + \alpha_2(2,0) = (\alpha_1 3 + \alpha_2 2, \alpha_1 1 + \alpha_2 0) \quad \text{mas} \quad v = (1,2)$ (1,2) = (\alpha_1 3 + \alpha_2 2, \alpha_1 1) \quad do qual se obtêm \alpha_1 = -1 \quad e \alpha_2 = 2 Assim \vert v = (-1)\vert v_1 + 2\vert v_2 \quad entire or v \u00e9 \u00e9 \u00e9 uma combinação linear de \vert v_1 e \vert v_2.



Figura A1 Vetor v em combinação linear de $v_1 e v_2$.

A.1.5 Soma e Soma Direta

Sejam U e W subespaços de V. Então S = U + W é o conjunto de todos os vetores u + w tais que $u \in U$ e $w \in W$.

Teorema A.1.2 A soma S de dois subespaços vetoriais $U \in W$ de V é um subespaço

vetorial. A intersecção $U \cap W$ é também um subespaço de V.

Exemplo A.1.7 Sejam $U = \{(a,b,0); a, b \in \Re\}$ e $W = \{(0,0,c); c \in \Re\}$. Então

$$S = U + W = \{(a, b, c); a, b \in c \in \mathfrak{R}\} = \mathfrak{R}^3.$$

Sejam $U \in W$ subespaços de V. Diz-se que V é a *soma direta* de $U \in W$ e representa-se por :

 $V=U\oplus W$

se V = U + W e $U \cap W = \{0\}$.

Exemplo A.1.8 Sejam
$$U = \{(a,0,b,0); a, b \in R\}$$
 e $W = \{(0,c,0,d); c, d \in \Re\}$. Então
 $S = U + W = \{(a,b,c,d); a,b,c \in d \in \bot\} = \Re^4 \in U \cap W = \{(0,0,0,0)\} = \{0\}, \log O$
 $S = U \oplus W.$

Foi visto que a soma de dois espaços vetoriais é outro espaço vetorial, isto é gera outro espaço vetorial. Uma definição mais formal é dada a seguir.

2.4 Subespaços gerados

Seja *V* um espaço vetorial e $A = \{v_1, v_2, ..., v_n\} \subset V$ com $A \neq \phi$. O conjunto *S* de todos os vetores de *V* que sejam combinações lineares dos vetores de *A* é um subespaço vetorial de *V*. Então o conjunto *A* gera *S* (*S* = ger(*A*)).

Observação 1.2

i) Os vetores $v_1, v_2, ..., v_n$ são os geradores do espaço S. S = ger(A)

ii) Span (S) = { $\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n$ tais que $\alpha_i \in F$ }

iii) A é o conjunto gerador

Exemplo A.1.9 Seja o espaço vetorial $V = \Re^3 e A = \{v_1, v_2, v_3, v_4\} \subset V$, onde $v_1 = (1,1,1)$, $v_2 = (1,1,0), v_3 = (1,0,0), v_4 = (0,0,2)$. Qual é o espaço gerado pelos vetores de *A*? Solução: seja v = (x, y, z) um elemento dos S = ger(A), então:

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + \alpha_4 v_4$$
$$v = (x, y, z) = (3\alpha_1, 2\alpha_2, \alpha_3, 2\alpha_4)$$

Se $\alpha_4 = a \in \Re \implies \alpha_1 = z - 2a$; $\alpha_2 = y - z + 2a$ e $\alpha_3 = x - y$

$$\Rightarrow v = (x, y, z) = (z - 2a)v_1 + (y - z + 2a)v_2 + (x - y)v_3 + av_4$$

Logo os vetores $\{v_1, v_2, v_3, v_4\}$ geram $V=S=\Re^3$.

Observa-se no exemplo A.1.9 que com quatro vetores se gera \Re^3 , mas se $A = \{e_1, e_2, e_3\}$, onde e_i para i=1,2,3 são os vetores canônicos, também geram \Re^3 . Então como caracterizar um conjunto de vetores que geram um espaço vetorial?. Para responder esta pergunta é preciso definir quando dois vetores são linearmente independentes (LI).

A.1.7 Dependência e independência linear

Definição A.1.5 Um conjunto de vetores $\{v_1, v_2, ..., v_n\}$ de um espaço (V,F) é chamado *linearmente dependente* sobre o corpo *F* se e somente se existirem escalares $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n \in F$ não todos nulos tais que:

$$\sum_{i=1}^{n} \alpha_{i} v_{i} = \alpha_{1} v_{1} + \alpha_{2} v_{2} +, \dots, + \alpha_{n} v_{n} = 0$$
(A.1)

Caso contrario, o conjunto de vetores $\{v_1, v_2, ..., v_n\}$ é chamado *linearmente independente* em (V,F).

Exemplo A.1.10 Seja $V = \Re^2$ e $v_1 = (1,2)$, $v_1 = (0,1)$, então $\{v_1, v_2\}$ é LI.

Teorema 1.2 As linhas não-nulas de uma matriz em forma escalonada são linearmente independentes. [LIPSCHUTZ, 2004].

A seguir caracterizam-se os conjuntos geradores, cujos vetores são LI. Formam uma base do espaço vetorial. O número de elementos do conjunto dá a dimensão do espaço vetorial.

A.1.8 Base e dimensão

Definição A.1.6 Um conjunto $B = \{v_1, v_2, ..., v_n\} \subset V$ é uma base do espaço vetorial V se:

i) $B \notin LI$ ii) $B \text{ gera } V \Leftrightarrow \text{ ger}(B) = V$.

Exemplo A.1.11 O conjunto $B = \{e_1, e_2\} \subset \Re^2$ é uma base do espaço vetorial \Re^2 , onde $e_1 = (1,0)$ e $e_2 = (0,1)$, pois *B* é LI e *B* gera \Re^2 . *B* é chamada base canônica.

Diz-se que um espaço vetorial V tem dimensão finita n ou que é n-dimensional e se escreve dim V = n.

Teorema A.1.3 Se o conjunto $B = \{v_1, v_2, ..., v_n\}$ é uma base de um espaço vetorial *n*dimensional então todo vetor $x \in V$ pode ser escrito de uma única maneira na forma:

$$x = \sum_{i=1}^{n} \alpha_i \, v_i \qquad \qquad \alpha_i \in \mathfrak{R}$$

Os α_i são as componentes de *x* com respeito à base *B*.

Prova

A prova será feita por contradição. Suponha que exista uma outra combinação linear dos vetores da base tal que:

$$x = \sum_{i=1}^{n} \beta_i v_i$$

Então pode-se escrever:

$$\sum_{i=1}^{n} (\beta_i - \alpha_i) v_i = 0$$

Como os vetores de uma base são LI isto implica que $\beta_i = \alpha_i$.

Teorema A.1.4 Se *V* um espaço vetorial de dimensão finita, então toda base de *V* tem o mesmo número de elementos [LIPSCHUTZ, 2004].

Este teorema diz que a escolha de base num espaço vetorial não é única, mas o número de elementos de cada uma das bases é sempre o mesmo.

Até agora se mostra que é possível representar cada vetor de um espaço vetorial V, por meio de uma ênupla, desde que se tenham escolhido uma base B de V. Ocorre então a

pergunta: como se modifica esta representação do vetor, caso seja expressado em outra base (pois *V* tem várias bases). O seguinte enunciado aclara essa dúvida.

Escolha duas bases $B = \{v_1, v_2, ..., v_n\} W = \{w_1, w_2, ..., w_n\}$ de *V*. Como *B* é uma base, cada vetor em *W* pode ser escrito de maneira única (Teorema A.1.3) como combinação linear dos elementos de *B*:

$$w_{1} = c_{11}v_{1} + c_{12}v_{2} + \dots + c_{1n}v_{n}$$

$$w_{2} = c_{21}v_{1} + c_{22}v_{2} + \dots + c_{2n}v_{n}$$

$$\dots \dots \dots \dots \dots \dots \dots$$

$$w_{n} = c_{n1}v_{1} + c_{n2}v_{2} + \dots + c_{nn}v_{n}$$

Seja *P* a transposta da matriz de coeficientes acima:

$$P = \begin{bmatrix} c_{11} & c_{21} & \dots & c_{n1} \\ c_{12} & c_{22} & \dots & c_{n2} \\ \dots & \dots & \dots & \dots \\ c_{1n} & c_{2n} & \dots & c_{nn} \end{bmatrix}$$

P é chamada matriz de mudança de base (ou matriz de Transição) da velha base *B* para a nova base *W*. Como *B* é LI então a matriz *P* é invertível. De fato, sua inversa P^{-1} é a matriz de transição da base *W* de volta à base *B*.

Seja $B = \{v_1, v_2, ..., v_n\}$ uma base de um espaço vetorial *V* sobre um corpo *F* e suponha que $P = (p_{ij})$ seja uma matriz arbitrária não singular sobre *F*. Então os *n* vetores

$$w_i = p_{1i}v_1 + p_{2i}v_2 + \dots + p_{ni}v_n$$
 para todo $i = 1, 2, \dots, n$

são LI e assim formam outra base W de V.

Exemplo A.1.12 Considere um espaço vetorial V no qual se define o sistema:

$$x_{k+1} = Ax_k + Bu_k \tag{A.1.2}$$

$$y_k = Cx_k + Du_k \tag{A.1.3}$$

Seja uma base $B = \{x_1, x_2, ..., x_n\}$ (dos vetores de estado) do espaço V.

Define-se um novo vetor de estado

$$x(k) = P\hat{x}(k) \tag{A.1.4}$$

onde P é uma matriz não singular (isto é, existe a inversa de P).

Substituindo-se (A.1.4) nas equações (A.1.3) e (A.1.4) se obtém:

$$P\hat{x}(k+1) = AP\hat{x}(k) + Bu(k)$$

$$y(k) = CP\hat{x}(k) + Du(k)$$
(A.1.5)

Multiplicando-se ambos lados da equação (A.1.5) por P^{-1} , se obtém

$$\hat{x}(k+1) = P^{-1}AP\hat{x}(k) + P^{-1}Bu(k)$$

Definindo-se $P^{-1}AP = \hat{A}$, $P^{-1}B = \hat{B}$, $CP = \hat{C}$ e $D = \hat{D}$ se obtém um novo sistema:

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}u(k)$$

$$y(k) = \hat{C}\hat{x}(k) + \hat{D}u(k)$$
(A.1.6)

Na nova base $W = \{ \hat{x}_1, \hat{x}_2, ..., \hat{x}_n \}.$

Os dois sistemas são equivalentes e os vetores de estado x(k) e $\hat{x}(k)$ estão relacionados pela matriz *P*. Dado que a matriz *P* seja qualquer matriz não singular, então existe uma quantidade infinita de espaço de estado.

Definição A.1.7 A **dimensão** de um espaço vetorial V não nulo é o número de vetores de uma base para B. Denota-se por: dim(V).

Se *B* não possui base, $\dim(V) = 0$.

Exemplo A.1.13 Seja a matriz A_{mxn} , então dim(A)=mxn

Observação A.1.2 Seja V um espaço vetorial de dim(V) = n. Se $S \subset V$ é um subespaço de V, então dim $(S) \le n$. Se dim(S) = n, então S=V.

Teorema A.1.5 Sejam $U \in W$ subespaços de dimensão finita de um espaço vetorial V. Então U + W tem dimensão finita e

A.1.9 Espaço linha e espaço coluna de uma matriz

Seja *A* uma matriz arbitrária m x n sobre um corpo $F(F=\Re)$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

As linhas de A,

$$R_1 = (a_{11}, a_{12}, \dots, a_{1n}), \dots, R_m = (a_{m1}, a_{m2}, \dots, a_{mn})$$

podem ser encaradas como vetores em \Re^n e, daí, geram um subespaço de \Re^n , chamado *espaço linha* de *A*. Isto é

esplin
$$A=ger(R_1, R_2, ..., R_m)$$

Analogamente, as colunas de *A* podem ser encaradas como vetores em \Re^n e, daí, geram um subespaço de \Re^m chamado *espaço coluna* de *A*, denotado espcol *A*.

Alternativamente, espcol A= esplin A^{T} .

Teorema A.1.6 Toda matriz é equivalente por linhas a uma única matriz em forma canônica reduzida por linhas. A prova pode ser vista em [LIPSCHUTZ, 2004].

Estamos familiarizados com funções ordinárias, tais como a função f definida pela equação $f(x) = x^2$. Essa função transforma um número real em outro número real, no caso, no seu quadrado. Por exemplo, o número 2 é transformado em 4, isto é, f(2) = 4. Enunciam-se agora funções que transformam vetores em vetores. Se A é uma matriz m xn, então a equação T(x) = Ax define uma função T, que transforma um vetor x em \Re^n em um vetor Ax, em \Re^m . Assim, T é uma função de \Re^n para \Re^m .

A.1.10 Aplicações lineares

Sejam V e W espaços vetoriais sobre o mesmo corpo F. Uma aplicação $T:V \rightarrow W$ é

chamada de aplicação linear (ou transformação linear ou homomorfismos de espaços vetoriais) e se verifica que:

$$T(\alpha_1 v_1 + \alpha_2 v_2) = \alpha_1 T(v_1) + \alpha_2 T(v_2)$$
(A.1.7)

para quaisquer v_1 e v_2 em V e quaisquer escalares α_1 e α_2 em F.

A equação (A.1.7) mostra que uma transformação linear satisfaz o princípio da superposição.

Sendo T uma função, cada vetor $v \in V$ tem um só vetor imagem $w \in W$, que será indicado por w = Tv.

Exemplo A.1.14 $T: \mathfrak{R}^2 \to \mathfrak{R}^3$ definida por T(x,y) = (3x, -2y, x - y) é uma transformação linear.

Observações A.1.3

i) Se $T(0) \neq 0$, T não é linear.

ii) Quando em T: $V \rightarrow W$, V = W, então a transformação linear é chamada de Operador Linear.

iii) Uma matriz A_{mxn} sempre determina uma transformação linear $T: \mathfrak{R}^n \to \mathfrak{R}^m$ onde

$$T(v) = Av.$$

iv) Seja uma transformação linear $T: V \to W$ seja $B = \{v_1, v_2, ..., v_n\}$ uma base de V. Então para todo $v \in V$ se pode expressar $v = \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n$, assim sua imagem é

$$T(v) = T(\alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n) = \alpha_1 T(v_1) + \alpha_2 T(v_2) + ... + \alpha_n T(v_n)$$

Se $T(v_1), T(v_2), ..., T(v_n)$ são conhecidas, então sempre é possível obter a imagem de T(v).

A.1.11 Núcleo de uma transformação linear

Definição A.1.7 Chama-se de núcleo de uma transformação linear $T: V \rightarrow W$, ao conjunto de todos os vetores $v \in V$ tal que T(v) = 0. Indica-se esse conjunto por

$$N(T) = \ker(T) = \{v \in V; T(v) = 0\}$$

Exemplo A.1.15

Seja a transformação linear $T: \mathfrak{R}^2 \to \mathfrak{R}^3$ tal que T(x, y, z) = (x - y + 4z, 3x + y + 8z). O núcleo

$$N(T) = \{(x, y, z) \in \mathfrak{R}^3; \ T(x, y, z) = (0, 0)\} \Leftrightarrow (x - y + 4z, 3x + y + 8z) = (0, 0)$$

Resolvendo, para z = a, obtem-se x = -3a e y = a.

Assim $N(T) = \{a(-3,1,1); a \in R\}$; o vetor (-3,1,1) gera o espaço N(T).

A.1.12 Imagem de uma transformação linear

Definição A.1.8 Chama-se de imagem de uma transformação $T: V \to W$, ao conjunto de vetores $w \in W$ que são imagens de pelo menos um vetor $v \in V$. Indica-se este conjunto por Im(T), ou seja

Im $(T) = \{w \in W; T(v) = w, \text{ para algum } v \in V\}.$

Im $(T) \subset W$ é um subconjunto, um subespaço.



O seguinte teorema relaciona a noção de dimensão com a de aplicação linear $T:V \rightarrow W$. No caso de *V* ter dimensão finita, subsiste a seguinte relação fundamental.

Teorema A.1.5 Seja V de dimensão finita e T: $V \rightarrow W$ uma aplicação linear. Então: $\dim(V) = \dim(\ker(T)) + \dim(\operatorname{Im}(T))$

Exemplo A.1.16 Seja $T: \mathfrak{R}^2 \to \mathfrak{R}^3$ tal que T(x, y, z) = (x, y, 0). Então a $\operatorname{Im}(T) = \mathfrak{R}^2$ e $\ker(T) = \mathfrak{R}$, assim $3 = \dim(\mathfrak{R}^3) = 1+2$.

Observação A.1.4 Seja $T: V \rightarrow W$ uma aplicação linear. Define-se o posto de T como a dimensão de sua imagem e nulidade de T como a dimensão de seu núcleo. Assim, pelo

teorema A.1.5 tem-se que:

posto
$$T$$
 + nulidade T = dim V

O posto de uma matriz foi definido como a dimensão de seu espaço coluna e espaço linha. Note também que, encarando a matriz A como uma aplicação linear, então ambas as definições se correspondem, pois a imagem de A é precisamente seu posto coluna.

Exemplo A.1.17

Considere um sistema de *m* equações lineares com *n* incógnitas sobre um corpo *F*:

 $c_{11}\mathbf{v}_{1} + c_{12}\mathbf{v}_{2} + \dots + c_{1n}\mathbf{v}_{n} = \mathbf{b}_{1}$ $c_{21}\mathbf{v}_{1} + c_{22}\mathbf{v}_{2} + \dots + c_{2n}\mathbf{v}_{n} = \mathbf{b}_{2}$ $c_{n1}\mathbf{v}_{1} + c_{n2}\mathbf{v}_{2} + \dots + c_{nn}\mathbf{v}_{n} = \mathbf{b}_{n}$

que é equivalente à equação matricial

$$Av = b$$

onde $A=(c_{ij})$ é a matriz de coeficientes e $v = (v_i)$ e $b = (b_i)$ são os vetores coluna das incógnitas e das constantes, respectivamente. Ora, a matriz *A* pode ser encarada como uma aplicação linear

$$A: \mathfrak{R}^n \to \mathfrak{R}^m$$

Assim, a solução da equação Av=b pode ser vista como a pré-imagem de $b \in \Re^m$ sobre a aplicação linear $A: \Re^n \to \Re^m$. Além disso, a solução da equação homogênea associada Av=0 pode ser vista como o núcleo da aplicação linear A.

O Teorema A.1.5 sobre aplicações lineares gera a seguinte relação:

 $\dim(\ker A) = \dim \mathfrak{R}^n - \dim(\operatorname{Im} A) = n - \text{posto A}$

Mas *n* é precisamente o número de incógnitas no sistema homogêneo Av=0.

A.1.13 Distância, Norma e Produto Interno em um espaço vetorial

Para poder realizar operações algébricas em espaços vetoriais é necessária uma medida (métrica), mas isto ocorre da de forma natural quando se define uma norma, da qual se obtém um produto interno. Com estas definições se pode tratar tópicos conhecidos em \Re , como limite, derivada, integral, projeções etc. Cabe aclarar que um espaço vetorial

V pode ter várias métricas e normas e se pode definir diferentes produtos internos.

A.1.13.1 Métrica

Seja *V* um conjunto arbitrário, chama-se espaço métrico a um par (*V*, *d*), quando existe uma função $d: VxV \rightarrow \Re$ que obedece as propriedades:

- d.1 $d(x, y) \ge 0$ e igual a zero se e somente se x = y.
- d.2 d(x, y) = d(y, x)
- d.3 $d(x,z) \le d(x,y) + d(y,z)$ para qualquer x, y e $z \in V$ (desigualdade do triângulo)

Exemplo A.1.18 Seja o espaço (\Re^n, d) , define-se a métrica usual:

$$d((x_1,...,x_n),(y_1,...,y_n)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}$$

Introduzindo a idéia de origem (fazendo y = 0) e fazendo a métrica ter um comprimento definido, independente de *x*, define-se a seguir, norma ou comprimento de um vetor $x \in V$ representado por ||x||.

A.1.13.2 Norma

Chama-se espaço normado a um par (V, || ||), onde V é um espaço vetorial sobre um corpo $F(\Re \text{ ou } \aleph)$ e $|| ||: V \to \Re^+$ é uma função que verifica as seguintes condições, N.1 ||x|| = 0 se e somente se x = 0,

- N.2 $\|\lambda x\| = |\lambda| \|x\|$
- N.3 $||x + y|| \le ||x|| + ||y||$
- para $x, y \in V \in \lambda$ um escalar ($\lambda \in F$):

À função $\| \|$ chama-se de norma.Todo espaço normado é em particular um espaço métrico, com a métrica $d(x, y) = \|x - y\|$. Neste caso a norma é recuperada da métrica d através de $\|x\| = d(x,0)$. Mas nem toda métrica em um espaço vetorial é definida por uma norma.

Exemplo A.1.19 Para $1 \le p \le \infty$, o espaço l_p consiste no conjunto das sucessões $x = (x_1, ..., x_n)$ tais que

$$\left(\sum_{i=1}^{\infty} \left|x_{i}\right|^{p}\right)^{\frac{1}{p}} < \infty$$

A norma de um elemento de $x \in l_p$ é $||x||_p = \left(\sum_{i=1}^{\infty} |x_i|^p\right)^{\frac{1}{p}}$.

A prova poder ser vista em [BACHMAN; NARICE, 2000].

Exemplo A.1.20 Seja $A \in R^{mxn}$. A definição da norma de uma matriz pode ser compatível com a norma de um vetor. A norma mais popular de uma matriz é a norma de Frobenius, dada por:

$$||A||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}$$

A.1.13.3 Espaços com produto interno

Suponha que V seja um espaço vetorial real ou complexo sobre um campo F (\Re ou \aleph). Um produto interno em V é uma aplicação

$$(,): VxV \to F$$
$$< x, y > \mapsto (x, y)$$

onde (x,y) denota o produto interno dos dois vetores e $\langle x,y \rangle$ representa somente a ordem dos vetores em VxV. Esta aplicação é um produto interno se satisfaz:

p.1 (x,y)= $\overline{(x, y)}$; x,y \in V, onde a barra denota o complexo conjugado

p.2
$$(\alpha x + \beta y, z) = \alpha(x, z) + \beta(y, z)$$
; $\alpha, \beta \in F$ e $x, y, z \in V$

p.3 $(x, x) \ge 0$ para todo $x \in V$ e $(x, x) = 0 \Leftrightarrow x = 0$

Os espaços com produto interno são chamados de espaços pré-Hilbert.

Se V é um espaço de Banach (vide [BACHMAN; NARICE, 2000]) que satisfaz as propriedades de produto interno, então V é um espaço de Hilbert.

Por p.3 (*x.x*) é não negativo e então sua raiz quadrada positiva existe. Utiliza-se a notação $||x|| = \sqrt{(x,x)}$ ou em forma equivalente

$$\|x\|^2 = (x, x) = x^T x$$
 (A.1.8)

Se ||x|| = 1, então x é chamado vetor unitário e se diz normalizado.

Dois vetores de um espaço vetorial V são ortogonais, se (x,y)=0.

O ângulo θ entre dois vetores x e y é definido por $\cos \theta = \frac{(x, y)}{\|x\| \|y\|}$ para $0 \le \theta \le \pi$

A.1.13.4 Conjunto ortogonal de vetores

Seja *V* um espaço vetorial e $B = \{v_1, v_2, ..., v_n\} \subset V$ é ortogonal se dois vetores quaisquer, distintos são ortogonais, isto é $(v_i, v_j) = 0$ para $i \neq j$.

Teorema A.1.6 Um conjunto ortogonal de vetores $A = \{v_1, v_2, ..., v_n\}$ não nulos é LI.

A.1.13.5 Base Ortogonal e Base Ortonormal

Diz-se que uma base $B = \{v_1, v_2, ..., v_n\}$ de V é ortogonal se o conjunto de vetores é um conjunto ortogonal.

Diz-se que uma base $B = \{v_1, v_2, ..., v_n\}$ de V é ortonormal se B é uma base ortogonal e todos seus vetores são unitários.

Seja *B* uma base de *V*, então existe uma base ortogonal \widetilde{B} para *V*, isto é possível mediante o processo de ortogonalização de Gram – Schmidt.

A.1.14 Complementos ortogonais

Seja $W \subset V$, V é um espaço vetorial com produto interno, o complemento ortogonal de W, denotado por W^{\perp} consiste dos vetores em V que são ortogonais a todo vetor $w \in W$: $W^{\perp} = \{v \in V: (v, w) = 0 \text{ para todo } w \in W \}.$



Figura A.2 Espaços complementares, observa-se que $v \in W^{\perp}$.

É fácil mostrar que W^{\perp} é um subespaço de V.

No exemplo A.1.17 se b=0 então Av=0, assim W é o conjunto solução formado pelos vetores v. Mas pode-se dar outra interpretação a W, aplicando-se a noção de ortogonalidade. Especialmente cada vetor solução v é ortogonal a cada linha de A e, consequentemente, W é complemento ortogonal do espaço linha de A.

Teorema A.1.7 Seja W um subespaço de V. então V é soma direta de W e W^{\perp} , isto é $V = W \oplus W^{\perp}$.

A.1.15 Matrizes Ortogonais

As matrizes com o mesmo número de linhas e colunas são chamadas de matrizes quadradas. Umas das propriedades destas matrizes é a comutatividade AB=BA. Se $A=(a_{ij})$ e $B=(b_{ij})$ são matrizes quadradas de ordem n, a diagonal (ou diagonal principal) de A consiste nos elementos $a_{11}, a_{22}, ..., a_{nn}$. O traço de A (trace A) é a soma dos elementos das diagonais. As potências de A são definidas como: $A^2 = AA$, ..., $A^{n+1} = A^n A$ e $A^0 = I$, onde I é a matriz identidade. A matriz quadrada A é *invertível* (ou não singular) se existe uma matriz B tal que AB=BA=I, tal matriz B é única e é chamada de inversa de A, e se denota por A^{-1} . Também se o determinante de A e' diferente de zero então a matriz A é invertível. Se $A_1, A_2, ..., A_k$ são invertíveis, então seu produto é invertível e $(A_1, A_2, ..., A_k)^{-1} = A_k^{-1} A_{k-1}^{-1} ... A_1^{-1}$. Se A é uma matriz quadrada com a propriedade $A^T = A$, então A é simétrica e A é anti-simetrica se $A^T = -A$.

Uma matriz real A é ortogonal se $AA^{T} = A^{T}A = I$. Note que uma matriz ortogonal A é

necessariamente quadrada e invertível, com inversa $A^{-1} = A^{T}$.

Teorema A.1.8 Seja *A* uma matriz quadrada. Então as seguintes afirmações são equivalentes: a) *A* é ortogonal, isto é, $A^T = A^{-1}$; b) as linhas de *A* formam um conjunto ortonormal; c) as colunas de *A* formam um conjunto ortonormal.

O teorema é válido para o produto interno usual em \Re^n . Não é válido se atribuímos outro produto interno em \Re^n .

Vários problemas algébricos envolvendo matrizes podem ser resolvidos deixando-as em alguma forma especial (diagonal, triangular, entre outras); isto é possível através de uma seqüência de transformações ortogonais elementares. Por exemplo, para solucionar um problema linear AX=B uma forma de atacar este problema é pelo método de eliminação gaussiana, este método é equivalente a fatorar a matriz A em produto LU, onde L é uma matriz triangular inferior e U é uma matriz triangular superior, a partir de daí podemos calcular o determinante de A e sua inversa $A^{-1} = U^{-1}L^{-1}$. Este método é aplicado quando a matriz A é quadrada, mas se A é uma matriz mxn existem outras formas de solucionar este problema pelo método de Cholesky, Fatoração QR mediante Householder, decomposição em valores singulares. Quando o posto de A é menor que n, o sistema é indeterminado. Nestes casos se diz que A é de posto deficiente. Sabe-se que tal problema não tem solução única e habitualmente se exige a condição de norma mínima.

O algoritmo completamente estável para estes casos é a decomposição dos valores singulares (DVS). Uma alternativa é a fatoração QR, que é estável quase sempre, a qual pode ser calculada no Matlab pelo comando [Q R]=qr (A,0) e a solução do problema inicial é X=R\(Q'*B).

A.1.16 Fatoração QR

Para considerar a decomposição QR, é necessária a transformação de Househlder.

Lema A.1.1 Considere dois vetores $x \neq y \in \Re^n$ com ||x|| = ||y||. Então existe um vetor
$u \in \Re^n$ tal que

$$y = (I - 2uu^T)x$$
, $||u|| = 1$ (A.1.9a)

O vetor u é unicamente definido por

$$u = \pm \frac{x - y}{\|x - y\|}$$
(A.1.9b)
$$Span \{u\}^{\perp}$$

$$Span \{u\}^{\perp}$$

Px=y

Figura A.3 Transformação de Householder

Prova

Se $x \neq y$ então $x - y \neq 0$ assim $||x - y|| \neq 0$. Defina-se um vetor $u = \pm \frac{x - y}{||x - y||}$ o qual

existe $e \in \Re^n$, com norma

$$||u|| = \left|\frac{x-y}{||x-y||}\right| = \frac{||x-y||}{||x-y||} = 1.$$

Agora prova-se que o vetor *u* satisfaz a condição $y = (I - 2uu^T)x$

Observe que

$$(I - 2uuT)x = x - 2uuTx$$
(A1.10)

Substituindo-se a equação (A.1.9b) em (1.10)

$$(I - 2uu^{T})x = x - 2uu^{T}x = x - 2\left[\pm \frac{(x - y)}{\|x - y\|} \cdot \left(\pm \frac{(x - y)}{\|x - y\|}\right)^{T}\right]x$$
$$(I - 2uu^{T})x = x - 2\left[\frac{(x - y)(x - y)^{T}}{\|x - y\|^{2}}\right]x$$
(A.1.11)

Se (x,y)=(y,x) então $x^T y = y^T x$ e pela equação (A.1.8) aplicada em (A.1.11)

$$(I - 2uu^{T})x = x - 2 \left[\frac{(x - y)(x - y)^{T}}{\|x - y\|^{2}} \right] x = x - 2 \frac{(x - y)(x^{T}x - y^{T}x)}{(x - y)^{T}(x - y)}$$
$$(I - 2uu^{T})x = x - 2 \frac{(x - y)(x^{T}x - y^{T}x)}{x^{T}x - x^{T}y - y^{T}x + y^{T}y} = x - 2 \frac{(x - y)(x^{T}x - y^{T}x)}{2(x^{T}x - y^{T}x)} = x - x + y$$
$$(I - 2uu^{T})x = y$$

Por último, prova-se a unicidade de *u*. Para isso supor que exista outro vetor *v* em \Re^n que também satisfaça o Lema 1.1 com ||v|| = 1.

Então o vetor *y* pode ser achado mediante o vetor *u* e o vetor *v*:

 $y = (I - 2uu^T)x = (I - 2vv^T)x$ então $x - 2uu^Tx = x - 2vv^Tx \implies u(u^Tx) = v(v^Tx), \forall x$

Como x é arbitrário então seja
$$x = u$$
 (ou $x = v$) então
 $u(u^{T}u) = v(v^{T}u) \implies u||u|| = v(v^{T}u) \qquad \text{mas} \quad ||u|| = 1$
 $u = v(v^{T}u) \implies \pm \frac{x - y}{||x - y||} = \pm \frac{x - y}{||x - y||} \quad (v^{T}u) \quad \text{assim}$
 $\pm 1 = v^{T}u \qquad \text{mas usando o fato que} \quad ||u||^{2} = 1^{2} = 1$
 $\pm ||u||^{2} = v^{T}u \implies \pm (u, u) = (v, u) \implies \pm u = v$.

Portanto *u* é único.

A matriz $P = (I - 2uu^T)$ do Lema 1.1 é chamada da transformação de Householder, a qual é simétrica e satisfaz

$$P^{2} = (I - 2uu^{T})^{2} = I - 4uu^{T} + 4u(u^{T}u)u^{T} = I \implies P^{2} = PP = I$$

Então a matriz P é simétrica $(P = P^T)$ e $P^2 = PP = PP^T = I \implies P^{-1} = P = P^T$, isto implica que P é uma transformação ortogonal e se pode interpretar como um operador que faz Px = y ou Py = x.

Apresenta-se a seguir, uma idéia para depois encontrar a matriz QR. Seja o vetor $a \in \Re^n$. O problema é transformar o vetor a para o vetor b ($b \in \Re^n$, b é construído), com as restrições ||a|| = ||b||, isto é:

Dado

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_n \end{bmatrix} \text{ transformar } a \text{ para o vetor } b = \begin{bmatrix} b_1 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \text{ isto } e (Pa = b), \quad ||a|| = ||b|| = |b_1|$$

Como ||a|| = ||b||, se tem $||a|| = |b_1|$ então $b_1 = \pm ||a||$.

Assim define-se

$$\widetilde{a} = a - b = \begin{bmatrix} a_1 - b_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} e$$

$$\|\widetilde{a}\|^2 = (\widetilde{a}, \widetilde{a}) = (a_1 - b_1)^2 + a_2^2 + \dots + a_n^2 = a_1^2 - 2a_1b_1 + b_1^2 + a_2^2 + \dots + a_n^2$$

$$\|\widetilde{a}\|^2 = (-2a_1b_1 + b_1^2) + a_1^2 + a_2^2 + \dots + a_n^2 = -2a_1b_1 + b_1^2 + \|a\|^2 = -2a_1b_1 + b_1^2 + b_1^2$$

$$\|\widetilde{a}\|^2 = 2b_1(b_1 - a_1) = -2b^T\widetilde{a}$$
(A.1.12)

Observe que se o sinal de b_1 é igual ao sinal de a_1 , então existe a possibilidade que $|a_1 - b_1|$ tenha valor pequeno. Esta dificuldade pode ser solucionada escolhendo-se o sinal de b_1 oposta a a_1 .

Da mesma forma que no Lema 1.1 define-se o operador

$$P = I - \frac{2\widetilde{a} \ \widetilde{a}^{T}}{\|\widetilde{a}\| \|\widetilde{a}\|}$$
(A.1.13)

Substituindo-se (A.1.12) na equação (A.1.13) se obtém

$$P = I - \frac{2\widetilde{a} \ \widetilde{a}^{T}}{\left\|\widetilde{a}\right\| \left\|\widetilde{a}\right\|} = I - \frac{2\widetilde{a} \ \widetilde{a}^{T}}{\left\|\widetilde{a}\right\|^{2}} = I + \frac{\widetilde{a} \ \widetilde{a}^{T}}{b^{T}a} = I + \frac{\widetilde{a} \ \widetilde{a}^{T}}{b_{1}\widetilde{a}_{1}}$$
(A.1.14)

Agora aplica-se o operador *P* ao vetor *a* para obter o vetor *b*, mas antes observe que se $||a|| = |b_1|$ é fácil notar que $a^T a = ||a||^2 = b_1^2$ (pois $|b_1| = \sqrt{b_1^2}$), então $a^T a = b_1^2$. E da definição de *a* e *b* se obtém $b^T a = b_1 a_1$.

$$Pa = \left[I + \frac{\widetilde{a} \ \widetilde{a}^{T}}{b_{1}\widetilde{a}_{1}}\right]a = a - \frac{(a-b)(a-b)^{T}}{b_{1}(a_{1}-b_{1})}a = a - \frac{(a-b)(a^{T}a-b^{T}a)}{b_{1}(a_{1}-b_{1})} = a - \frac{(a-b)(b_{1}^{2}-b_{1}a_{1})}{b_{1}(a_{1}-b_{1})}$$

$$Pa = a - \frac{(a-b)b_{1}(b_{1}-a_{1})}{b_{1}(a_{1}-b_{1})} = a - a + b = b$$
(A.1.15)
$$Pa = b$$

Portanto conhecendo $\tilde{a} = a - b$ e b_1 , o vetor *a* pode ser transformado no vetor *b* com as especificações dadas acima. Sabe-se que este método é muito eficiente e computável, pois só as primeiras componentes de *a* são modificadas na computação.

Com estas idéias básicas apresenta-se, a seguir, a decomposição QR, a qual é muito aplicado na álgebra linear numérica.

Lema 1.2 Toda matriz retangular $A \in \mathbb{R}^{m \times n}$, m > n é decomposta no produto de duas matrizes:

$$A = QR \tag{A.1.16}$$

donde $Q \in \Re^{mxn}$ é uma matriz ortogonal com $Q^T Q = I_n$ e $R \in \Re^{nxn}$ é uma matriz triangular superior. O lado direito da equação (A.1.16) é chamado de decomposição QR da matriz A.

Prova

A decomposição A = QR é equivalente a $Q^T A = R$, então a matriz Q^T é uma transformação ortogonal que transforma a matriz A em uma matriz triangular superior. Isto é feito pelo método de transformação de Householder.

Seja $a^{(1)} = A(:,1)$ a primeira coluna do vetor A. Defina $u^{(1)} = \tilde{a} e b_1$ da forma:

$$a^{(1)} = \begin{bmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{m1} \end{bmatrix} \implies u^{(1)} = a^{(1)} - b^{(1)} = \begin{bmatrix} a_{11} - b_1 \\ a_{21} \\ \dots \\ a_{m1} \end{bmatrix}, \text{ com } b^{(1)} = \begin{bmatrix} b_1 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

donde $b_1^{(1)} = \pm \|a^{(1)}\|$. De acordo com (A.1.13), se obtém o operador

$$P^{(1)} = I + \frac{u^{(1)} (u^{(1)})^T}{(b^{(1)})^T u^{(1)}}$$

Defina $P^{(1)}A = A^{(1)}$, (considere $A = [a^{(1)} \vdots a^{(2)} \vdots \dots a^{(m)}]_{mxm}$), assim:

$$P^{(1)}A = A^{(1)} = \left[I + \frac{u^{(1)}(u^{(1)})^{T}}{(b^{(1)})^{T}u^{(1)}}\right]A = A + \frac{(a^{(1)} - b^{(1)})(a^{(1)} - b^{(1)})^{T}}{(b^{(1)})^{T}u^{(1)}}A$$

$$P^{(1)}A = \left[a^{(1)} \vdots a^{(2)} \vdots \dots a^{(m)}\right]_{mxm} + \frac{(a^{(1)} - b^{(1)})_{mx1}(a^{(1)} - b^{(1)})_{1xm}}{((b^{(1)})^{T}u^{(1)})_{1x1}} \left[a^{(1)} \vdots a^{(2)} \vdots \dots a^{(m)}\right]_{mxm}}$$

$$P^{(1)}A = \left[a^{(1)} + \frac{(a^{(1)} - b^{(1)})_{mx1}(a^{(1)^{T}} - b^{(1)^{T}})_{1xm}}{((b^{(1)})^{T}u^{(1)})_{1x1}}a^{(1)} \vdots \dots \vdots a^{(m)} + \frac{(a^{(1)} - b^{(1)})_{mx1}(a^{(1)^{T}} - b^{(1)^{T}})_{1xm}}{((b^{(1)})^{T}u^{(1)})_{1x1}}a^{(m)}\right]_{mxm}$$
Observa-se que $A^{(1)}(\vdots 1)$ é o mesmo que a equação (A 1 15) motivo pelo qual se obtém

$$P^{(1)}A = \left[Pa^{(1)} \vdots Pa^{(2)} \dots \vdots Pa^{(m)}\right]_{nxm}$$

$$P^{(1)}A = \left[b^{(1)} \vdots a^{(2)} + \frac{(a^{(1)} - b^{(1)})_{mx1}(a^{(1)^{T}} - b^{(1)^{T}})_{1xm}}{((b^{(1)})^{T}u^{(1)})_{1x1}}a^{(2)} \dots \vdots a^{(m)} + \frac{(a^{(1)} - b^{(1)})_{mx1}(a^{(1)^{T}} - b^{(1)^{T}})_{1xm}}{((b^{(1)})^{T}u^{(1)})_{1x1}}a^{(m)}\right]_{mxm}$$

$$P^{(1)}A = \begin{bmatrix} b_1 & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & a_{m2}^{(1)} & a_{m3}^{(1)} & \cdots & a_{mn}^{(1)} \end{bmatrix}$$

Assim o vetor na primeira coluna de $A^{(1)}$ é o vetor $b^{(1)}$ e os demais vetores colunas de $A^{(1)}$ encontram-se afetados por $P^{(1)}$. Na transformação seguinte o vetor $b^{(1)} = A^{(1)}(:,1)$ não é modificado, e assim é colocado como a primeira coluna de *R*.

Agora encontra-se a segunda coluna do vetor $A^{(1)}$. Para isso define-se $a^{(2)}$, $u^{(2)}$ e $b^{(2)}$ como:

$$a^{(2)} = \begin{bmatrix} 0 \\ a_{22}^{(1)} \\ a_{32}^{(1)} \\ \cdots \\ a_{m2}^{(1)} \end{bmatrix} \implies u^{(2)} = a^{(2)} - b^{(2)} = \begin{bmatrix} 0 \\ a_{22}^{(1)} - b_2 \\ a_{32}^{(1)} \\ \cdots \\ a_{m2}^{(1)} \end{bmatrix} , \qquad b^{(2)} = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$$

donde $b_2 = \pm \|a^{(2)}\|$. E $P^{(2)}$ é definida por

$$P^{(2)} = I + \frac{u^{(2)} (u^{(2)})^{T}}{(b^{(2)})^{T} u^{(2)}} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & p_{22}^{(2)} & \cdots & p_{2m}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & p_{m2}^{(2)} & \cdots & p_{mm}^{(2)} \end{bmatrix}$$
(A.1.17)

 $P^{(2)}$ é uma matriz ortogonal, em efeito (aplicando a formula A.1.12)

$$\begin{bmatrix} P^{(2)} \end{bmatrix}^2 = I + 2 \frac{u^{(2)} (u^{(2)})^T}{(b^{(2)})^T u^{(2)}} + \frac{u^{(2)} (u^{(2)})^T u^{(2)} (u^{(2)})^T}{(b^{(2)})^T u^{(2)}} = I - 4 \frac{u^{(2)} (u^{(2)})^T}{\|u^{(2)}\|^2} + 4 \frac{u^{(2)} \|u^{(2)}\|^2}{\left[\|u^{(2)}\|^2\right]^2} = I$$

$$P^{(2)} P^{(2)} = I$$

Assim multiplicando $A^{(1)}$ por $P^{(2)}$ se obtém $P^{(2)}A^{(1)} = P^{(2)}P^{(1)}A = A^{(2)} \text{ isto } \acute{e}:$ $P^{(2)}A^{(1)} = \begin{bmatrix} P^{(2)}\begin{bmatrix} b_{1}\\0\\\vdots\\0\end{bmatrix} \\ P^{(2)}\begin{bmatrix} a_{12}^{(1)}\\a_{22}^{(1)}\\\vdots\\a_{m2}^{(1)} \end{bmatrix} \\ \vdots \\ \cdots \\ \vdots \\ P^{(2)}A^{(1)} = \begin{bmatrix} b^{(1)} \\ \vdots \\ 0 \end{bmatrix} \\ + P^{(2)}a^{(1)} \\ \vdots \\ 0 \end{bmatrix} + P^{(2)}a^{(2)} \\ \vdots \\ 0 \end{bmatrix} \\ \vdots \\ \cdots \\ \vdots \\ P^{(2)}A^{(1)} = \begin{bmatrix} b^{(1)} \\ \vdots \\ 0 \end{bmatrix} \\ = P^{(2)}A^{(1)} = \begin{bmatrix} b^{(1)} \\ \vdots \\ 0 \end{bmatrix} \\ = \begin{bmatrix} a^{(1)}_{12}\\a_{1n}^{(1)}\\a_{2$

$$P^{(2)}A^{(1)} = \begin{bmatrix} b_1 & a_{12}^{(1)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & b_2 & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{m3}^{(2)} & \cdots & a_{mn}^{(2)} \end{bmatrix} = A^{(2)}$$
(A.1.18)

Note que a primeira linha e coluna de $A^{(2)}$ são as mesmas de $A^{(1)}$.

Repetindo-se este procedimento até a *n*-coluna, se obtém a matriz triangular inferior $A^{(n)}$ da forma:

$$P^{(n)}P^{(n-1)}\cdots P^{(1)}A = A^{(n)} = \begin{bmatrix} R\\0 \end{bmatrix}$$
(A1.19)

Como cada componente de $P^{(j)}$ para j = 1,..,n é ortogonal e simétrica, então

$$A = P^{(1)}P^{(2)} \cdots P^{(n)}A^{(n)} = Q\begin{bmatrix} R\\ 0 \end{bmatrix}$$

onde $R \in \Re^{nxn}$ é triangular superior e $Q \in \Re^{mxn}$ é ortogonal, isto completa a prova.

A decomposição em QR é aplicada para achar uma base ortonormal para um conjunto de vetores.

A.1.17 Projeções

Definição A.1.2 Suponha que \Re^n seja dado pela soma direta de subespaços $V \in W$, isto é $\Re^n = V \oplus W$, $V \cap W = \{0\}$.

Então o vetor $x \in \Re^n$ pode ser unicamente expresso como

$$x = v + w \quad , \qquad v \in V \quad e \quad w \in W \tag{A.1.20}$$

donde v é a projeção de x sobre V ao longo de W e w é a projeção de x sobre W ao longo de V.

Como a projeção é um operador linear em \Re^n , então ela pode ser expressa como uma matriz.

Para definir a projeção ortogonal (isto é, a projeção entre dois espaços ortogonais, por exemplo, $V \in V^{\perp}$) são necessárias algumas propriedades da matriz projeção. Assim se

enunciará a seguir algumas propriedades de uma matriz P idempotente, para depois demonstrar que toda matriz projeção é idempotente.

Lema A.2 Suponha que a matriz $P \in \Re^{nxn}$ seja idempotente (isto é $P^2 = P$) se e somente se,

$$Ker(P) = Im(I_n - P) \tag{A.1.21}$$

Prova

⇒] Suponha que $P^2 = P$ então demonstrar que $Ker(P) = Im(I_n - P)$ é provar que $Ker(P) \subset Im(I_n - P)$ e $Ker(P) \supset Im(I_n - P)$ Seja $x \in Ker(P)$ então Px = 0, observa-se que x = (I - P)x resulta que $x \in Im(I - P)$, assim $Ker(P) \subset Im(I_n - P)$.

Seja $x \in \Re^n$, tem-se que $0 = Px - Px = Px - P^2x = P(I - P)x$, então $(I - P)x \in Ker(P)$ significa que $Im(I - P) \subset Ker(P)$. Isto prova (A.1.21).

⇐] Se $Ker(P) = Im(I_n - P)$ então provar que P é idempotente

Para qualquer vetor $z \in \mathbb{R}^n$, seja sua imagem x = (I - P)z, então pela hipótese se tem que $x \in Ker(P)$, então $0 = Px = P(I - P)z = P(x - Px) = Px - P^2x$ para qualquer vetor $z \in \mathbb{R}^n$, implica que $P^2 = P$.

Corolário A.1.1 Suponha que se a matriz $P \in \Re^{n \times n}$ seja idempotente, então $\Re^n = \text{Im}(P) + Ker(P)$

$$\mathfrak{R}^n = \operatorname{Im}(P) \oplus Ker(P) \tag{A.1.22}$$

Prova

Para qualquer vetor $x \in \mathbb{R}^n$ pode ser expressado como x = Px + (I - P)x então por (A.1.21),

$$\mathfrak{R}^{n} = \operatorname{Im}(P) + \operatorname{Im}(I - P) = \operatorname{Im}(P) + Ker(P)$$
(A.1.23)

Agora seja $x \in \text{Im}(P) \cap Ker(P)$. Então se tem $x \in \text{Im}(P)$ e $x \in Ker(P)$, assim $x \in \text{Im}(P)$ então x = Py para $y \in \Re^n$ e para $x \in Ker(P)$ então 0 = Px. Da hipótese

 $P^2 = P$ se obtém que $0 = Px = P^2x = P(Px) = Py = x$. Como *x* é um vetor arbitrário então $\{0\} = \text{Im}(P) \cap Ker(P)$, assim o lado direito de (A.1.23) é expresso como a soma direita, isto é $\Re^n = \text{Im}(P) \oplus Ker(P)$.

Lema A.1.3 A matriz $P \in \Re^{nxn}$ é a matriz projeção sobre Im(P) ao longo de Ker(P) se e somente se P é uma matriz idempotente.

Prova

⇒] Se *P* é a matriz projeção sobre Im(*P*) ao longo de *Ker*(*P*), então se provará que *P* é uma matriz idempotente. Pela definição A.1.2 (considerando *V*=Im (*P*) e *W*= *Ker* (*P*)) então para qualquer $x \in \Re^n$, x = v + w, $v \in V$ e $w \in W$ onde v é a projeção de x sobre *V* ao longo de *W*.

Para qualquer
$$x \in \mathfrak{R}^n$$
, $v = Px \in \text{Im}(P)$ e $0 = Px \in Ker(P)$.
Seja $P(Px) = Pv = v$ (A.1.24)

pois a projeção de $v \in V$ sobre V ao longo de W é o mesmo vetor v, isto é v = v + 0, pelo qual v = Pv. Mas v = Px, substituindo-se na equação (A.1.24)

P(Px) = Pv = v = Px então $P^2 = P$ para qualquer $x \in \mathbb{R}^n$.

 \Leftarrow] *P* é uma matriz idempotente então se provará que *P* é a matriz projeção sobre Im(*P*) ao longo de *Ker*(*P*)

Define-se $V = \{v/v = Px, x \in \mathbb{R}^n\}$ e $W = \{w/w = (I - P)Px, x \in \mathbb{R}^n\}$. Pelo corolário A.1.1 e lema A.2 implica que $x \in \mathbb{R}^n$ é decomposto unicamente como

x = Px + (I - P)x = v + w, $v \in V$ e $w \in W$ com $V \cap W = \{0\}$

Pela definição A.1.2 P é a matriz projeção sobre V = Im(P) ao longo de W = Ker(P).

A.1.18 Projeção ortogonal

Definição A.1.5 Suponha que $V \subset \Re^n$ então para qualquer $x \in \Re^n$ pode ser unicamente expresso como

$$x = v + w \quad , \qquad v \in V \quad e \quad w \in V^{\perp} \tag{A.1.25}$$

Isto é um caso particular da definição A.1.2 com $W = V^{\perp}$ e v é chamado de projeção ortogonal de x sobre V. Isto é mostrado na figura 1.4.



Figura A.4 Projeção ortogonal

Existe uma condição necessária e suficiente que caracteriza uma projeção ortogonal e é dada por:

Lema A.1.5 A matriz $P \in R^{nxn}$ é a projeção ortogonal sobre Im(P) se e somente se satisfaz as duas condições

i) $P^2 = P$ ii) $P^T = P$

prova é só aplicar o lema A.1.3

Apresenta-se a seguir, uma forma de encontrar a matriz projeção *P*. Inicialmente apresenta-se para uma reta e depois para um subespaço.

A.1.18.1 Projeção sobre uma reta

Dado um ponto $b = (b_1, b_2, ..., b_m) \in \Re^m$ e uma reta que passe pelo origem na direção do vetor $a = (a_1, a_2, ..., a_m) \in \Re^m$. Seja *p* um ponto sobre a reta. Imagine que se esteja olhando sobre o ponto *b* para encontrar o ponto *p* que fecha a *b*. Esta é a idéia de ortogonalidade. Assim a reta que une *b* até *p* é perpendicular ao vetor *a*.

A projeção p é múltipla do vetor a (isto é $p=\tilde{x} a$). Caso se conheça \tilde{x} então encontra-se a projeção p e assim encontra-se a matriz projeção P.

Observe que a reta b-p é ortogonal ao vetor a, então:

$$(a,b-p) = (a,b-\tilde{x}a) = 0 \implies (a,b)-(a, \tilde{x}a) = 0 \implies a^{T}b-\tilde{x}a^{T}a = 0 \implies \tilde{x} = \frac{a^{T}b}{a^{T}a}$$
(A.1.26)
$$p = \tilde{x} a \quad (\tilde{x} \text{ é um escalar})$$

Τ.



Figura A.5 Projeção sobre uma reta

Matriz projeção

Seja a projeção $p = a\tilde{x} = a\frac{a^{T}b}{a^{T}a} = Pb$ onde a matriz P é definida como $P = \frac{aa^{T}}{a^{T}a}$. A matriz P é *mxm* mas o posto (P) = 1, pois se esta projetando dentro do espaço unidimensional uma reta que é a.

Pelo Lema A.1.3 outra matriz projeção é (I-P), a qual produz o outro lado de e.

A.1.18.2 Projeção sobre um subespaço

Sejam *n* vetores $a_1, a_2, ..., a_n$. Supostos que sejam LI. Generalizando a idéia para o caso unidimensional, encontra-se uma combinação linear $\tilde{x}_1a_1 + \tilde{x}_2a_2 + ... + \tilde{x}_na_n$ que feche o vetor *b*. O problema é projetar qualquer vetor *b* sobre o espaço *n*-dimensional gerado pelos vetores $\{a_1, a_2, ..., a_n\}$.

Se n=1 (só um vetor) é a projeção sobre uma reta. A reta é o espaço coluna de uma matriz A. Em geral a matriz A tem n colunas $a_1, a_2, ..., a_n$. Suas combinações em \Re^m são os vetores $A\tilde{x}$ no espaço coluna. Então procura-se $p = A\tilde{x}$ (a projeção) que é o espaço coluna.

Procura-se \tilde{x} para achar a projeção *p* e assim encontramos a matriz projeção *P*.



Figura A.6 Projeção ortogonal de um vetor b sobre um subespaço gerado por A.

O vetor erro $e = b - A\tilde{x}$ é perpendicular ao subespaço gerado pela combinação linear dos vetores $a_1, a_2, ..., a_n$ do espaço coluna de A (isto é Im(A)).

Como o espaço nulo de A^T é ortogonal ao espaço coluna em \mathfrak{R}^m , cada vetor y do espaço nulo de A^T é perpendicular a cada coluna de A (isto é Im(A)).

Então *e* tem que pertencer $a N(A^T) = Kert(A)$. Assim, observa-se que tomando-se apropriadamente os vetores

$$a_{1}^{T}(b - A\widetilde{x}) = 0$$

$$\vdots$$
 ou
$$\begin{bmatrix} a_{1}^{T} \\ \vdots \\ a_{n}^{T} \end{bmatrix} \begin{bmatrix} b - A\widetilde{x} \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
 em forma equivalente $A^{T}e = 0$ (A.1.27)

$$a_{n}^{T}(b - A\widetilde{x}) = 0$$

Então $e \in N(A^T) = Kert(A)$.

Da equação (A.1.27) se tem $A^{T}(b - A\tilde{x}) = 0 \implies \tilde{x} = (A^{T}A)^{-1}A^{T}b$. Observe que $(A^{T}A)^{-1}$ existe pois a matriz $A^{T}A$ é nxn e é invertivel (tem colunas LI). Assim a *projeção* de *b* sobre o subespaço é dada por:

$$p = A\widetilde{x} = A(A^T A)^{-1} A^T b$$

Esta fórmula mostra a *matriz projeção* que produz p = Pb $P = A(A^T A)^{-1} A^T$

A.1.18.3 Projeção oblíqua

A seguir apresenta-se a matriz projeção ortogonal e matriz projeção oblíqua.

Seja *B* uma matriz *mxn*,
$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{1mn} \end{bmatrix}_{mxn}$$
, as linhas da matriz *B* podem ser

expressas na forma $b_i = [b_{i1} \ b_{i2} \ \dots \ b_{in}]^T$ para $i = 1, 2, \dots, m$. Então se tem um conjunto de vetores $\{b_1, b_2, \dots, b_m\}$. Se um elemento *b* pertence ao espaço linha de *B*, então pode-se expressa-lo como uma combinação linear dos vetores linhas da matriz *B*:

$$b = \alpha_1 b_1 + \alpha_2 b_2 + \ldots + \alpha_m b_m = [b_1 \ b_2 \ \ldots \ b_m] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \sum_{i=1}^m \alpha_i b_i = B^T \alpha_i$$

onde $\alpha = \begin{bmatrix} \alpha_1 & \dots & \alpha_m \end{bmatrix}^T$.

Seja b = x/B a projeção ortogonal de um vetor arbitrário x no espaço linha de B que pode ser escrita como $b = B^T \alpha$; então b satisfaz $(x - b)^T b = 0$ (por projeção ortogonal). Então $(x - b)^T b = (x - B^T \alpha)^T B^T \alpha = 0$, assim: $x^T B^T \alpha - \alpha^T B B^T \alpha = 0 \Rightarrow (x^T B^T - \alpha^T B B^T) \alpha = 0 \Rightarrow Bx - B B^T \alpha = 0$ (transpondo) $B B^T \alpha = Bx$ (A.1.28)

Na equação (A.1.28) a solução de α depende de BB^T : Se BB^T é não singular, então : $\alpha = (BB^T)^{-1}Bx$. Se BB^T é singular, utiliza-se a pseudo – inversa de BB^T : $\alpha = (BB^T)^+Bx$

Portanto a projeção b = x/B é dada por: $b = B^T \alpha = B^T (BB^T)^+ Bx = \Pi_B x$

onde

$$\Pi_B = B^T (BB^T)^+ B \tag{A.1.28}$$

é o operador projeção ortogonal no espaço linha de B

Seja a_i um vetor linha de A, a projeção ortogonal de a_i no espaço linha de B é:

 $a / B = \prod_{B} a_{i}$ ou $a / B = a_{i}^{T} \prod_{B}$

Então pode-se descrever matematicamente a projeção ortogonal do espaço linha de *A* sobre o espaço linha de *B* por:

$$A/B = A\Pi_B = AB^T (BB^T)^+ B \tag{A.1.29}$$

Pode-se decompor a matriz *A* como a combinação linear de duas matrizes ortogonais é dizer *B* e B^{\perp} conforme mostrado na figura 1.6.



Figura A.7 Projeção ortogonal de *A* no espaço linha de *B*.

$$A = A\Pi_B + A\Pi_{B^{\perp}} \tag{A.1.30}$$

Das equações (1.39) e (1.40) pode-se escrever

$$A = A / B + A / B^{\perp} = A \Pi_{B} + A \Pi_{B^{\perp}}$$
 (A.1.31)

onde

$$\Pi_B^{\perp} = I - \Pi_B \tag{A.1.32}$$

e A/B^{\perp} é a projeção do espaço linha de *A* no complemento ortogonal do espaço de linhas da matriz *B*.

Analogamente, pode-se decompor *A* como combinação linear de duas matrizes *B* e *C* não ortogonais e de seus complementos ortogonais através das projeções ortogonais $\Pi_{[C,B]}A \in \Pi^{\perp}_{[C,B]}A$:

$$A = A / [C B] + A / [C B]^{\perp} = \prod_{[C,B]} A + \prod_{[C,B]} A$$
(A.1.33)

Cuja representação aparece na figura 1.7.



Figura A.8 Projeção obliqua

A matriz *A* foi projetada ortogonalmente no espaço [C B] (denotada por A/[C B]) e não seu complemento ortogonal $[C B]^{\perp}$ (denotado por $A/[C B]^{\perp}$), depois a matriz A/[C B] foi projetada no espaço *C*, esta projeção é denotada por $P_{C,B}A$. De forma análoga a matriz A/[C B] foi projetada no espaço *B*, esta projeção é denotada por $P_{B,C}A$. Desta forma de obtém que $A/[C B] = P_{C,B}A + P_{B,C}A$. Então a formula (A.1.43) pode ser expressa como:

 $A = A / [C B] + A / [C B]^{\perp} = P_{C,B} A + P_{B,C} A + A / [C B]^{\perp}$

Ao termo $A/_{C} B = P_{B,C}A$ se chama de projeção oblíqua sobre B ao longo de C.

Definição A.1.6 A projeção oblíqua do espaço de linhas de $A \in \Re^{pxj}$ no espaço de linhas de $B \in \Re^{qxj}$ sobre o espaço de linhas de $C \in \Re^{rxj}$ é definida como:

$$A/_{B}C = A\begin{bmatrix} C^{T} & B^{T} \end{bmatrix} \cdot \begin{bmatrix} (CC^{T} & CB^{T}) \\ BC^{T} & BB^{T} \end{bmatrix}^{+} \Big]_{prime iras \ r} \cdot C$$

Muitas vezes os corolários são uma ferramenta para aplicação a casos práticos. Então uma definição equivalente da projeção obliqua é apresentada a seguir:

Corolário A.1.2 A projeção oblíqua do espaço de linhas de $A \in \Re^{pxj}$ no espaço de linhas de $B \in \Re^{qxj}$ sobre o espaço de linhas de $C \in \Re^{rxj}$ pode ser definida como:

$$A/_{B} C = \left[A/B^{\perp}\right] \cdot \left[C/B^{\perp}\right]^{+} \cdot C .$$

A.1.19 Métodos dos mínimos quadrados e SVD

Como foi dito anteriormente, uma das aplicações das projeções ortogonais é na solução do problema de mínimos quadrados. Antes de enunciar esta aplicação, apresenta-se primeiro algumas idéias.

Seja $A \in \Re^{nxr}$ com rank(A) = r e $Im(A) = A \subset \Re^n$. Seja a decomposição QR da matriz A dada por

$$\mathbf{A} = \begin{bmatrix} Q_A & Q_A^T \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_A R , \qquad Q_A \in \mathfrak{R}^{mxn}$$
(A.1.26)

Como *R* é não singular se tem $A = \text{Im}(A) = \text{Im}(Q_A)$, isto é o vetor coluna de Q_A forma uma base ortonormal da Im(A) = A e Q_A^T forma uma base ortonormal do complemento ortogonal $(\text{Im}(A))^{\perp}$ (isto é A^{\perp}).

Agora define-se a matriz P_A como

$$P_A = Q_A Q_A^T \in \mathbb{R}^{n \times n} \tag{A.1.27}$$

É claro que a matriz P_A é uma projeção sobre a Im (P_A) (aplicar lema A.1.5). Então qualquer vetor $z \in \Re^n$ é decomposto de forma única como

 $z = x + y \operatorname{com} x \in A$ e $y \in A^{\perp}$

Assim tem-se que

$$x = P_A z$$
 e $y = (I - P_A) z$ (A.1.28)

Então P_A e $(I - P_A)$ são projeções ortogonais sobre A (=Im(A)) e A^{\perp} , respectivamente.

Isto permite minimizar a norma de um vetor. Isto é enunciado no seguinte Lema:

Lema A.1.6 Suponha que A seja um subconjunto de \Re^n . Então para qualquer $z \in \Re^n$, $P_A z$ é o único vetor que satisfaz o seguinte:

 $\min_{x \in A} \|z - x\| = \|z - P_A z\|$

Prova

Se $z \in \Re^n$ e como $A \subset R^n$, então pode ocorrer que $z \in A$ ou $z \notin A$. Se $z \in A$ então z = z + y com $z \in A$ e $y \in A^{\perp}$, onde $P_A z = z$ é único. Se $z \notin A$ então, para qualquer $x \in A$ se tem pela equação (A.1.28) que $x - P_A z \in A$, mas $(I - P_A) z$ é ortogonal a A, assim $x - P_A z \perp (I - P_A) z$, por tanto, $||z - x||^2 = ||z - P_A z + P_A z - x|| = ||(I - P_A) z - (x - P_A z)||^2 = ||(I - P_A) z||^2 + ||(x - P_A z)||^2$ O lado direito da equação acima é mínimo para $P_A z = x$, o qual é único (pois $||z - x|| = ||z - P_A z||$).

A.1.19.1 Decomposição em valores singulares (SVD)

Lema A.1.7 Suponha que o posto de $A \in \Re^{mxn}$ seja $r \le \min(n,m)$. Então existem matrizes ortogonais $U \in R^{mxm}$ e $V \in R^{nxn}$ tais que

$$A = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^{T} , \qquad \Sigma = \begin{bmatrix} \sigma_{1} & & \\ & \sigma_{2} & \\ & & \ddots & \\ & & & \sigma_{r} \end{bmatrix}$$
(A.1.29)

onde $U^T U = I_m$, $V^T V = I_n$ e $\sigma_1 \ge \sigma_2 \ge ..\sigma_r \ge \sigma_{r+1} = .. = \sigma_p = 0$, $p = \min(m, n)$

Diz-se que $\sigma_1, \sigma_2, ..., \sigma_p$ são os autovalores da matriz *A* e (A.1.29) é sua decomposição em valores singulares (SVD).

Prova

Hipóteses: dada a matriz A, posto (A) = r, demonstra-se que existem matrizes ortogonais U e V tais que $A = U\Sigma V^T$.

Observe-se que $A^T A \in \Re^{nxn}$ é definida não negativa. Se a matriz $A^T A$ tiver *n* vetores LI, então se pode diagonalizar mediante uma transformação de similaridade. Caso contrario se diagonalizará por uma transformação ortogonal $V \in \Re^{nxn}$.

Seja $\lambda_1, \lambda_2, ..., \lambda_n$ os autovalores da matriz $(A^T A)$ e seus correspondentes auto vetores

denotados por $v_1, v_2, ..., v_n \in \Re^n$. Então se tem

$$A^T A v_i = \lambda_i v_i$$
 para $i = 1,..,n$

Mas como o posto(A) = posto($A^T A$) = r então se tem:

$$\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_r > \lambda_{r+1} = \ldots = \lambda_n = 0$$

Define-se:

$$\sigma_i^2 = \lambda_i$$
 para $i = 1,...,n$ e $V = \begin{bmatrix} V_r & \widetilde{V}_r \end{bmatrix}$

onde

$$V_r = [v_1 \ v_2 \cdots v_r] \ \mathbf{e} \ \widetilde{V}_r = [v_{r+1} \ v_{r+2} \cdots v_n]$$

Desta forma estamos demonstrando a existência da matriz V, agora provar que é ortogonal

$$V^{T}V = \begin{bmatrix} V_{r}^{T} \\ \widetilde{V}_{r}^{T} \end{bmatrix} \begin{bmatrix} V_{r} & \widetilde{V}_{r} \end{bmatrix} = \begin{bmatrix} V_{r}^{T}V_{r} & V_{r}^{T}\widetilde{V}_{r} \\ \widetilde{V}_{r}^{T}V_{r} & \widetilde{V}_{r}^{T}\widetilde{V}_{r} \end{bmatrix} = \begin{bmatrix} I_{r} & 0 \\ 0 & I_{n-r} \end{bmatrix} = I_{n}$$

Obseve-se que os auto vetores v_i para i = 1,..,n formam uma base ortogonal do qual se pode ter uma base ortonormal.

E que:

$$A^{T}Av_{i} = \sigma^{2}v_{i}$$
 para $i = 1,..,r$ (A.1.30)

$$A^{T}Av_{i} = 0v_{i} = 0$$
 para $i = r + 1,..,n$ (A.1.31)

Agora por último precisamos definir a matriz U para isto observe da equação (A.1.30) se tem

$$A^T A V_r = V_r \Sigma_+^2 \tag{A.1.32}$$

onde $\Sigma_{+}^{2} = diag(\sigma_{1}^{2}, \sigma_{2}^{2}, ..., \sigma_{r}^{2})$, multiplicando na equação (A.1.32) pela esquerda pela matriz *A* e por Σ_{+}^{-1} pela direita e considerando que $\Sigma_{+}^{-1}\Sigma_{+} = I_{r}$, temos

$$AA^{T}AV_{r} = AV_{r}\Sigma_{+}^{2}$$

$$AA^{T}AV_{r}\Sigma_{+}^{-1} = AV_{r}\Sigma_{+}^{2}\Sigma_{+}^{-1}$$
(A.1.33)

Da equação (A.1.33) se define

$$U_r = AV_r \Sigma_+^{-1} \in \mathfrak{R}^{mxr} \tag{A.1.34}$$

então substituindo a equação (A.1.34) na equação (A.1.33) se tem

$$AA^T U_r = U_r \Sigma_+^2 \tag{A.1.35}$$

Claramente observamos que $(\sigma_1^2, \sigma_2^2, ..., \sigma_r^2)$ são autovalores de AA^T , então os vetores de

 U_r formam uma base ortonormal, isto é:

$$U_{r}^{T}U_{r} = (\Sigma_{+}^{-1}V_{r}^{T}A^{T})AV_{r}\Sigma_{+}^{-1} = \Sigma_{+}^{-1}V_{r}^{T}(A^{T}AV_{r})\Sigma_{+}^{-1} = \Sigma_{+}^{-1}V_{r}^{T}(V_{r}\Sigma_{+}^{2})\Sigma_{+}^{-1} = \Sigma_{+}^{-1}\Sigma_{+}^{2}\Sigma_{+}^{-1} = I_{r}$$
$$U_{r}^{T}U_{r} = I_{r}$$

Como as *r* das n colunas de A^T são LI, podemos encontrar uma matriz $\widetilde{U}_r \in \Re^{mx(m-r)}$ tal que

$$A^T \widetilde{U}_r = 0 \tag{A.1.36}$$

Onde $\widetilde{U}_r^T \widetilde{U}_r = I_m$.

Desta forma se define a matriz $U = \begin{bmatrix} U_r & \widetilde{U}_r \end{bmatrix} \in \Re^{mxm}$. Esta matriz satisfaz $U^T U = I_m$, isto é

$$U^{T}U = \begin{bmatrix} U_{r}^{T} \\ \widetilde{U}_{r}^{T} \end{bmatrix} \begin{bmatrix} U_{r} & \widetilde{U}_{r} \end{bmatrix} = \begin{bmatrix} U_{r}^{T}U_{r} & U_{r}^{T}\widetilde{U}_{r} \\ \widetilde{U}_{r}^{T}U_{r} & \widetilde{U}_{r}^{T}\widetilde{U}_{r} \end{bmatrix} = \begin{bmatrix} I_{r} & 0 \\ 0 & I_{m-r} \end{bmatrix} = I_{m}$$

Observe que

$$U_{r}^{T}\widetilde{U}_{r} = (\Sigma_{+}^{-1}V_{r}^{T}A^{T})\widetilde{U}_{r} = \Sigma_{+}^{-1}V_{r}^{T}(A^{T}\widetilde{U}_{r}) = \Sigma_{+}^{-1}V_{r}^{T}(0) = 0$$

Agora somente nos resta demonstrar $A = U\Sigma V^T$, isto é

$$U^{T}AV = \begin{bmatrix} U_{r}^{T} \\ \widetilde{U}_{r}^{T} \end{bmatrix} A \begin{bmatrix} V_{r} & \widetilde{V}_{r} \end{bmatrix} = \begin{bmatrix} U_{r}^{T}AV_{r} & U_{r}^{T}A\widetilde{V}_{r} \\ \widetilde{U}_{r}^{T}AV_{r} & \widetilde{U}_{r}^{T}A\widetilde{V}_{r} \end{bmatrix}$$
(A.1.37)

Observe de $U_r^T U_r = I_r$, se tem

 $U_r^T A V_r \Sigma_+^{-1} = I_r \Longrightarrow U_r^T A V_r \Sigma_+^{-1} \Sigma_+^1 = \Sigma_+^1 \Longrightarrow U_r^T A V_r = \Sigma_+^1$

Da equação (A.1.31) se tem $A\widetilde{V}_r = 0$, por isto $U_r^T A\widetilde{V}_r = 0$ e $\widetilde{U}_r^T A\widetilde{V}_r = 0$ e como \widetilde{U}_r é ortogonal a U_r , é dizer $(\widetilde{U}_r, U_r) = 0 \Rightarrow (\widetilde{U}_r, U_r) = \widetilde{U}_r^T U_r = \widetilde{U}_r^T A V_r \Sigma_+^{-1} = 0$.

Então a equação (A.1.37) se pode expressar como:

$$U^{T}AV = \begin{bmatrix} \Sigma_{+} & 0 \\ 0 & 0 \end{bmatrix} = \Sigma$$

Do qual se obtém

 $A = U\Sigma V^T$.

É claro que a equação (A.1.29) pode ser expressada como

 $A = U\Sigma V^{T} = U_{r}\Sigma_{+}V_{r}^{T}$ Onde $U_{r} \in \Re^{mxr}$ e $V_{r} \in \Re^{nxr}$.

A.1.19.2 Mínimos quadrados

Considere o sistema da equação

$$Y = \psi \Theta + E \tag{A.1.38}$$

O qual pode ser representado na forma b = Ax onde $A = \psi$, b = Y e $x = \Theta$ (observe que o erro *E* nesta representação não foi considerado).

A solução do problema Ax = b é achar o vetor de parâmetros x. Surgem perguntas de forma natural como: sob que condições o problema tem solução, se existe uma única, solução, se não existe solução se pode obter um vetor x que minimize a norma ||Ax - b||. Observe que se a solução não é exata, pode-se definir um vetor de erro dado por $\xi = Ax - b$ onde se considera $\xi = E$.

Um método para solucionar Ax = b é aplicar a eliminação gaussiana.

A matriz A_{nxn} pode ser considerada como uma transformação linear. Se $A = [a_1 \ a_2 \ ... a_n]$, onde $\{a_1, a_2, ..., a_n\}$ são vetores colunas da matriz A e formam uma base de \Re^n (posto(A) = n), então ao encontrar a combinação linear dessa base com os respectivos pesos $x^T = [x_1 \ x_2 \ ... x_n]$ que geram o vetor b, resolve o problema. Mas o problemas surgem se:

- 1. A matriz A tenha posto r = n
- 2. A matriz A tenha posto r = m
- 3. A matriz A tenha posto r < min(n, m)

Exemplo A.1.21 Considerar o problema Ax=b, onde a matriz A é singular de posto r < n.

$$\begin{bmatrix} 6 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Com m=n=2 > r=1. Sua solução não é exata, mas pode-se determinar uma solução

aproximada. Como A é singular, a inversa de A, A^{-1} , não existe, então uma tentativa é multiplicar à esquerda ambos os membros da equação pela matriz A^{T} para obter:

$$A^T A x = A^T b \tag{A.1.39}$$

Se o termo $A^T A$ for não singular (isto é: det $(A^T A) \neq 0$) então existe a inversa. Multiplica-se então ambos membros da equação (A.1.39) pela inversa obtendo-se:

$$(A^{T}A)^{-1} A^{T}Ax = (A^{T}A)^{-1} A^{T}b$$

onde

$$Ix = (A^T A)^{-1} A^T b = A^* b$$

onde $A^* = (A^T A)^{-1} A^T$ é chamada matriz pseudo- inversa de A à esquerda.

Desta forma, para o exemplo dado, se tem a solução aproximada

$$x = A^*b = \begin{bmatrix} 6 & 3 \\ 2 & 1 \end{bmatrix}^* \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0,33 \\ 0,16 \end{bmatrix}$$

A seguir enuncia-se uma idéia do problema de mínimos quadrados. Seu objetivo é minimizar a norma $\min_{x \in \Re^n} ||Ax - b||$, $A \in \Re^{mxn}$ e $b \in \Re^m$, onde $m \ge n$. Suponha que posto (A) = n e a decomposição QR de A seja dada por:

$$A = Q\begin{bmatrix} R\\0\end{bmatrix}, \qquad Q \in \mathfrak{R}^{mxm}, R \in \mathfrak{R}^{nxn}$$

Como a norma é invariante sob a transformação ortogonal, se tem

$$\left\|Ax - b\right\|^{2} = \left\|Q^{T}(Ax - b)\right\|^{2} = \left\|\begin{bmatrix}R\\0\end{bmatrix}x - \begin{bmatrix}b_{1}\\b_{2}\end{bmatrix}\right\|^{2}, \quad \text{onde} \quad Q^{T}b = \begin{bmatrix}b_{1}\\b_{2}\end{bmatrix} \quad \text{com} \quad b_{1} \in \Re^{n} \quad \text{e}$$

$$b_2 \in \Re^{m-n}$$
.

Assim tem-se que:

$$||Ax - b||^2 = ||Rx - b_1||^2 + ||b_2||^2$$

Como o termo $\|b_2\|^2$ não depende de x, então o problema de mínimos quadrados se reduz a

 $Rx = b_1$ ou

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & r_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Assim a solução de $x_{n,x_{n-1},...,x_1}$ se obtém em forma recursiva.

Como foi mencionado anteriormente, se posto (*A*) é menor que *n*, alguns elementos da diagonal de *R* são zeros. Então a solução do problemas de mínimos quadrados não é única. Mas adicionando-se uma restrição de norma ||x|| mínima, pode-se obter uma única solução.

APÊNDICE B

B.1 Prova da equação (2.6)

Supor que o sistema (2.1) tenha como entrada um sinal de impulso unitário discreto dado por:

$$u_{k} = u(k) = \delta_{k} = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$
(B 1.1)

Com o estado inicial sendo nulo, $x_0 = x(0) = 0$, a partir do instante em que ocorre o impulso tem-se (na equação 2.1)

Para o instante
$$n = 0 \Rightarrow u_0 = u(0) = 1$$
 (isto é por (B 1.1))
 $x_1 = Ax_0 + Bu_0 \Rightarrow x_1 = Bu_0 = B$
 $y_0 = Cx_0 + Du_0 \Rightarrow y_0 = Du_0 = D = g_0$ (B 1.2)

Para o instante $n = 1 \Rightarrow u_1 = u(1) = 0$ (isto é por (B 1.1)) $x_2 = Ax_1 + Bu_1 \Rightarrow x_2 = AB$ $y_1 = Cx_1 + Du_1 \Rightarrow y_1 = CB = g_1$

Para o instante $n = 2 \Rightarrow u_2 = u(2) = 0$ (isto é por (B 1.1)) $x_3 = Ax_2 + Bu_2 \Rightarrow x_3 = A(AB) = A^2B$ $y_2 = Cx_2 + Du_2 \Rightarrow y_1 = C(AB) = CAB = g_2$

Suponha que é verdadeiro para $n = k - 1 \Rightarrow u_{k-1} = u(k-1) = 0$ (isto é por (B 1.1)) $x_k = Ax_{k-1} + Bu_{k-1} \Rightarrow x_k = A(A^{k-2}B) = A^{k-1}B$ $y_{k-1} = Cx_{k-1} + Du_{k-1} \Rightarrow y_{k-1} = C(A^{k-2}B) = CA^{k-2}B = g_{k-1}$

Prova-se agora que se cumpre para todo $n = k \Rightarrow u_k = u(k) = 0$ (isto é por (B 1.1)) $x_k = Ax_{k-1} + Bu_{k-1} \Rightarrow x_k = A(A^{k-2}B) = A^{k-1}B$

$$y_{k-1} = Cx_{k-1} + Du_{k-1} \implies y_{k-1} = C(A^{k-2}B) = CA^{k-2}B = g_{k-1}$$
 (B 1.3)

Portanto da equação (B 1.2) e (B 1.3) temos que a resposta do sistema (2.1) ao impulso é dada por:

$$g_{k} = \begin{cases} D & k = 0 \\ CA^{k-1}B, & k = 1, 2, \dots \end{cases}$$

B.2 Prova da equação (2.8)

Para provar a equação (2.8), precisamos primeiro provar a seguinte equação $x_0 = \dots + A^{k-1}u_{-k-1} + \dots + A^3Bu_{-4} + A^2Bu_{-3} + ABu_{-2} + Bu_{-1}$ (B 2.1)

Seja a sinal de entrada u no sistema (2.1), é dada pela equação (2.7)

Para o instante $n = -1 \implies u_{-1} = u(-1)$ $x_0 = Ax_{-1} + Bu_{-1}$ Para o instante $n = -2 \implies u_{-2} = u(-2)$ $x_{-1} = Ax_{-2} + Bu_{-2} \implies x_0 = Ax_{-1} + Bu_{-1} = A(Ax_{-2} + Bu_{-2}) + Bu_{-1}$ $x_0 = A^2x_{-2} + ABu_{-2} + Bu_{-1}$ Para o instante $n = -3 \implies u_{-3} = u(-3)$

$$x_{-2} = Ax_{-3} + Bu_{-3} \implies x_0 = A^2 x_{-2} + ABu_{-2} + Bu_{-1}$$
$$x_0 = A^3 x_{-3} + A^2 Bu_{-3} + ABu_{-2} + Bu_{-1}$$

Supor que é valido para $n = -k \implies u_{-k} = u(-k)$ $x_{-k+1} = Ax_{-k} + Bu_{-k} \implies x_0 = A^k x_{-k} + A^{k-1} u_{-k} + ... + A^{-3} Bu_{-4} + A^{-2} Bu_{-3} + ABu_{-2} + Bu_{-1}$

Então para
$$n = -k - 1 \implies u_{-k-1} = u(-k - 1)$$

 $x_{-k} = Ax_{-k-1} + Bu_{-k-1}$
 $\implies x_0 = A^k x_{-k-1} + A^{k-1} u_{-k-1} + ... + A^{-3} Bu_{-4} + A^{-2} Bu_{-3} + A Bu_{-2} + Bu_{-1}$ (B 2.2)

A equação (B 2.2) depende sempre do estado anterior e para $k = -1, -2, -3, \dots$ pode-se

expressar como:

$$x(0) = x_0 = \dots + A^{k-1}u_{-k-1} + \dots + A^3Bu_{-4} + A^2Bu_{-3} + ABu_{-2} + Bu_{-1}$$

Prova-se agora (2.8), da formula (2.4) temos que:

$$y(k) = CA^{k}x(0)$$
substituindo (B 2.2) na equação (2.4)

$$y(k) = CA^{k}(... + A^{k-1}u_{-k-1} + ... + A^{3}Bu_{-4} + A^{2}Bu_{-3} + ABu_{-2} + Bu_{-1})$$

$$y(k) = ... + \underbrace{CA^{2k-1}B}_{g_{2k-1}}u_{-k-1} + + \underbrace{CA^{k+3}B}_{g_{k+4}}u_{-4} + \underbrace{CA^{k+2}B}_{g_{k+3}}u_{-3} + \underbrace{CA^{k+1}B}_{g_{k+2}}u_{-2} + \underbrace{CA^{k}B}_{g_{k+1}}u_{-1}$$
(B 2.3)

Substituindo (2.6) na expressão anterior:

$$y(k) = g_{k+1}u_{-1} + g_{k+2}u_{-2} + g_{k+3}u_{-3} + g_{k+4}u_{-4} + \dots, \quad k = 0, 1, 2, \dots, s$$

B.3 Notações usadas na tese

Por motivos de clareza e compreensão, considere na equação (2.16) o caso $w_k = 0 = v_k$, logo a equação (2.16) é dada por

$$x_{k+1} = Ax_k + Bu_k \tag{B 3.1a}$$

$$y_k = Cx_k + Du_k \tag{B 3.1b}$$

O sinal de saída para o instante k = s é expressado por:

$$y_s = [C] x_s + [D] u_s$$
 (B 3.2)

para o instante
$$k = s + 1$$

 $y_{s+1} = Cx_{s+1} + Du_{s+1} = C(Ax_s + Bu_s) + Du_{s+1}$
 $y_{s+1} = CAx_s + CBu_s + Du_{s+1}$
 $y_{s+1} = [CA] x_s + [CB \quad D] \begin{bmatrix} u_s \\ u_{s+1} \end{bmatrix}$
(B 3.3)

para o instante
$$k = s + 2$$

 $y_{s+2} = Cx_{s+2} + Du_{s+2} = C(Ax_{s+1} + Bu_{s+1}) + Du_{s+2} = CA(Ax_s + Bu_s) + CBu_{s+1} + Du_{s+2}$
 $y_{s+2} = CA^2x_s + CABu_s + CBu_{s+1} + Du_{s+2}$

$$y_{s+2} = [CA^{2}] x_{s} + [CAB \quad CB \quad D] \begin{bmatrix} u_{s} \\ u_{s+1} \\ u_{s+2} \end{bmatrix}$$
(B 3.4)

para o instante k = s + 3

$$y_{s+3} = Cx_{s+3} + Du_{s+3} = C(Ax_{s+2} + Bu_{s+2}) + Du_{s+3} = CA(A^2x_s + ABu_s) + CABu_{s+1} + CBu_{s+2} + Du_{s+3}$$

$$y_{s+3} = CA^{3}x_{s} + CA^{2}Bu_{s} + CABu_{s+1} + CBu_{s+2} + Du_{s+3}$$
$$y_{s+3} = [CA^{3}]x_{s} + [CA^{2}B \quad CAB \quad CB \quad D]\begin{bmatrix}u_{s}\\u_{s+1}\\u_{s+2}\\u_{s+3}\end{bmatrix}$$
(B 3.5)

continuando até o instante k = s + i - 1 a equação B11 pode ser expressada

$$y_{s+i-1} = [CA^{i-1}] x_s + [CA^{i-2}B \quad CA^{i-3}B \quad \cdots \quad CB \quad D] \begin{bmatrix} u_s \\ u_{s+1} \\ \vdots \\ u_{s+i-2} \\ u_{s+i-1} \end{bmatrix}$$
(B 3.6)

O sinal de saída, é expressada na forma matricial através das equações (B 3.2), (B 3.3), (B 3.4),...e (B 3.6)

$$\begin{bmatrix} y_{s} \\ y_{s+1} \\ \vdots \\ y_{s+i-2} \\ y_{s+i-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-2} \\ CA^{i-1} \end{bmatrix} x_{s} + \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3}B & CA^{i-4}B & \cdots & D & 0 \\ CA^{i-2}B & CA^{i-3}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u_{s} \\ u_{s+1} \\ \vdots \\ u_{s+i-2} \\ u_{s+i-1} \end{bmatrix}$$
(B 3.7)

Observe-se que o índice *s* pode assumir diferentes valores fixos portanto se terá diferentes equações matriciais da forma (B 3.7), isto é:

$$\begin{bmatrix} y_0 & y_1 & \cdots & y_s & \cdots & y_{N-1} \\ y_1 & y_2 & \cdots & y_{s+1} & \cdots & y_N \\ \vdots & \vdots & \ddots & \vdots \\ y_{i-1} & y_i & \cdots & y_{s+i} & \cdots & y_{N+i-2} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-2} \\ CA^{i-1} \end{bmatrix} \begin{bmatrix} x_0 & x_1 & \cdots & x_s & \cdots & x_{N-1} \end{bmatrix} +$$

$$+\begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3}B & CA^{i-4}B & \cdots & D & 0 \\ CA^{i-2}B & CA^{i-3}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u_0 & u_1 & \cdots & u_s \\ u_1 & u_2 & \cdots & u_{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \cdots & u_{s+i} \\ \end{bmatrix}$$
(B 3.8)

A equação (B 3.8) pode-se expressar em forma mais general como:

$$\begin{bmatrix} y_{k} & y_{k+1} & \cdots & y_{k+s} & \cdots & y_{k+N-1} \\ y_{k} & y_{k+2} & \cdots & y_{k+s+1} & \cdots & y_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{k+i-1} & y_{k+i} & \cdots & y_{k+s+i} & \cdots & y_{k+N+i-2} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-2} \\ CA^{i-1} \end{bmatrix} [x_{k} & x_{k+1} & \cdots & x_{k+s} & \cdots & x_{k+N-1}] + \\ \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3}B & CA^{i-4}B & \cdots & D & 0 \\ CA^{i-2}B & CA^{i-3}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u_{k} & u_{k+1} & \cdots & u_{k+s} & \cdots & u_{k+N-1} \\ u_{k} & u_{k+2} & \cdots & u_{k+s+1} & \cdots & u_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{k+i-1} & u_{k+i} & \cdots & u_{k+s+i} & \cdots & u_{k+N+i-2} \end{bmatrix}$$
(B 3.9)

A equação (B 3.9) pode ser expressa a traves da forma compacta como:

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N}$$
(B 3.10)

Onde as matrizes em blocos de Hankel dos dados de entrada e saída são representadas por:

$$U_{k,i,N} = \begin{bmatrix} u_{k} & u_{k+1} & \cdots & u_{k+N-1} \\ u_{k+1} & u_{k+2} & \cdots & u_{k+N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k+i-1} & u_{k+i} & \cdots & u_{k+N+i-2} \end{bmatrix}$$

$$Y_{k,i,N} = \begin{bmatrix} y_{k} & y_{k+1} & \cdots & y_{k+N-1} \\ y_{k+1} & y_{k+2} & \cdots & y_{k+N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k+i-1} & y_{k+i} & \cdots & y_{k+N+i-2} \end{bmatrix}$$
(B 3.12)

os sub índices (k, i, j) indicam: k é o índice que representa o primeiro elemento da matriz, i é o número de linhas blocos da matriz e N representa o número de colunas da matriz.

Matriz de dados de entradas passadas e Futuros

Considerando k = 0 na equação (B 3.11) se tem $U_{0,i,N}$

$$U_{0,i,N} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{N-1} \\ u_1 & u_2 & \cdots & u_N \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \cdots & u_{N+i-2} \end{bmatrix} = U_{0|i-1} = U_p$$
(B 3.13)

A matriz definida em (B 3.13) é chamada de matriz de Hankel dos dados de entrada passadas. A notação $U_{0,i,N-1}$ é usada no método MOESP por Verhaegen e a notação $U_p = U_{0|i-1}$ é usada no método N4SID por Van Overschee e Bar De Moor.

Considerando k = i na equação (B 3.11) se tem $U_{i,2i,N}$

$$U_{i,2i,N} = \begin{bmatrix} u_i & u_{i+1} & \cdots & u_{i+N-1} \\ u_{i+1} & u_{i+2} & \cdots & u_{i+N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & \cdots & u_{N+2i-2} \end{bmatrix} = U_{i|2i-1} = U_f$$
(B 3.14)

A matriz definida em (B 3.14) é chamada de matriz de Hankel dos dados de entrada futuras. A notação $U_{i,2i,N-1}$ é usada no método MOESP por Verhaegen. E a notação $U_f = U_{i|2i-1}$ é usada no método N4SID Van Overschee e Bar De Moor.

Matriz de dados de saídas passadas e Futuros

Da mesma forma que em (B 3.13) e (B 3.14) são definidas as matrizes de hankel das saídas passadas e futuras, denotada respectivamente por

$$Y_{0,i,N-1} \in Y_{i,2i,N-1}$$
 (B 3.15)

notação usada em Verhaegen e $Y_p = Y_{0|i-1}$, $Y_f = Y_{i|2i-1}$ notação usada por Van Overschee e Bar De Moor.

A matriz estendida de observabilidade Γ_i , onde o i (i > n) denota o número de linhas de blocos, é definida como:

Assumindo-se que o par $\{A, C\}$ seja observável, então o posto $(\Gamma_i) = n$.

Seqüência de estados

Assim como as matrizes em blocos de Hankel, as seqüências de estados desempenham um importante papel nos algoritmos de identificação por sub-espaços. Define-se a seqüência de estados X_i como:

$$X_{k,N} \stackrel{def}{=} (x_k \quad x_{k+1} \quad \dots \quad x_{k+N-2} \quad x_{k+N-1}) \in \Re^{n \times N}$$
(B 3.17)

onde o subscrito "k" denota o primeiro elemento da seqüência de estados.

Como no caso anterior, se define a seqüência de estados passados e futuros como:

$$X_{p} = X_{0} = \begin{bmatrix} x_{0} & x_{1} & \dots & x_{N-2} & x_{N-1} \end{bmatrix}$$
 seqüência de estados passados. (B 3.18)
$$X_{f} = X_{i} = \begin{bmatrix} x_{i} & x_{i+1} & \dots & x_{i+N-2} & x_{i+N-1} \end{bmatrix}$$
 seqüência de estado futuros. (B 3.19)

Notação usada no método N4SID. Para o caso MOESP estes estados passados são definidos $X_{0,N-1}$ e estados futuros $X_{i,N-1}$

A matriz em blocos triangular inferior de Toeplitz H_i é definida como:

$$H_{i} \stackrel{def}{=} \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{bmatrix} \in \Re^{lixmi}$$
(B 3.20)

B.4 Justificativa de usar a decomposição LQ

No apêndice A foi definido a projeção ortogonal do espaço linha de A sobre o espaço

linha de B denotado por

$$A/B \stackrel{def}{=} AB^T (BB^T)^* B \tag{B 4.1}$$

Onde $(\bullet)^*$ denota a matriz pseudo-inversa de Moore- Penrose de (\bullet) .

E a projeção da matriz *A* no complemento ortogonal do espaço de linhas da matriz *B*, é expressada por

$$A/B^{\perp} \stackrel{def}{=} A - A/B \tag{B 4.2}$$

A representação matricial destas projeções (B 4.1) e (B 4.2), pode ser facilmente computada através da decomposição LQ da matriz $\begin{bmatrix} B \\ A \end{bmatrix}$.

Seja *A* e *B* matrizes de posto completo e LQ a decomposição da matriz $\begin{bmatrix} B \\ A \end{bmatrix}$, denotada

por:

$$\begin{bmatrix} B\\ A \end{bmatrix} = LQ^T = \begin{bmatrix} L_{11} & 0\\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T\\ Q_2^T \end{bmatrix}$$
(B 4.3)

Donde $L \in \Re^{(p+q)x(p+q)}$ é uma matriz triangular inferior, com $L_{11} \in \Re^{qxq}$, $L_{21} \in \Re^{pxq}$, $L_{21} \in \Re^{pxp}$ e $Q \in \Re^{Nx(p+q)}$ é uma matriz ortogonal, isto é,

$$Q^{T}Q = \begin{bmatrix} Q_{1}^{T} \\ Q_{2}^{T} \end{bmatrix} \begin{bmatrix} Q_{1} & Q_{2} \end{bmatrix} = \begin{bmatrix} I_{q} & 0 \\ 0 & I_{p} \end{bmatrix}$$
(B 4.4)

Então a representação matricial das projeções ortogonais (B 4.1) e (B 4.2) são expressadas como

$$A/B = L_{21}Q_1^T$$
 (B 4.5)

$$A/B^{\perp} = L_{22}Q_2^T$$
 (B 4.6)

Prova

Prova-se inicialmente (B 4.5)

Da equação (B 4.3) temos:

$$B = L_{11}Q_1^T \tag{B 4.7}$$

$$A = L_{21}Q_1^T + L_{22}Q_2^T$$
(B 4.8)

Substituindo-se (B 4.7) e (B 4.8) na equação (B 4.1), se obtém

$$A / B = (L_{21}Q_1^T + L_{22}Q_2^T)(L_{11}Q_1^T)^T (L_{11}Q_1^T (L_{11}Q_1^T)^T)^* L_{11}Q_1^T$$

$$A / B = (L_{21}Q_1^T + L_{22}Q_2^T)Q_1L_{11}^T (L_{11}Q_1^T (L_{11}Q_1^T)^T)^* L_{11}Q_1^T$$

$$A / B = L_{21}L_{11}^T (L_{11}Q_1^T (L_{11}Q_1^T)^T)^* L_{11}Q_1^T$$

$$A / B = L_{21}L_{11}^T (L_{11}L_{11}^T)^* L_{11}Q_1^T$$

Pela definição da matriz pseudo-inversa de Moore- Penrose, ver apêndice A

$$A / B = L_{21} L_{11}^{T} \left(\left(L_{11}^{T} L_{11} L_{11} L_{11}^{T} \right)^{-1} L_{11}^{T} L_{11} \right) L_{11} Q_{1}^{T}$$

$$A / B = L_{21} \underbrace{L_{11}^{T} \left(L_{11}^{T} \right)^{-1} \left(L_{11} \right)^{-1} \left(L_{11}^{T} \right)^{-1} \left(L_{11}^{T} \right)^{-1} L_{11}^{T} L_{11} L_{11} Q_{1}^{T}}$$

$$A / B = L_{21} \underbrace{\left(L_{11} \right)^{-1} \left(L_{11} \right)^{-1} L_{11} L_{11} Q_{1}^{T}}_{I}$$

$$A / B = L_{21} Q_{1}^{T}.$$
(B 4.9)

Agora se prova (B 4.2)

Substituindo-se (B 4.8) e (B 4.9), na equação (B 4.2)

$$A / B^{\perp} \stackrel{def}{=} A - A / B = A - L_{21}Q_{1}^{T}$$

$$A / B^{\perp} = L_{21}Q_{1}^{T} + L_{22}Q_{2}^{T} - L_{21}Q_{1}^{T}$$

$$A / B^{\perp} = L_{22}Q_{2}^{T}$$
(B 4.10)

Portanto se queremos encontrar a projeção ortogonal $Y_p = A$ saídas passadas sobre o espaço linha $U_p = B$ entradas passadas, denotada por $Y_p / U_p = A / B$, então a representação matricial através da composição LQ é dada por

$$\begin{bmatrix} B\\ A \end{bmatrix} = \begin{bmatrix} U_p\\ Y_p \end{bmatrix} = LQ^T = \begin{bmatrix} L_{11} & 0\\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T\\ Q_2^T \end{bmatrix}$$
(B 4.11)

Logo

 $A/B = Y_p/U_p = L_{21}Q_1^T.$

B.5 computando as matrizes B e D

Para a estimação das matrizes *B* e *D*. Da equação (2.27) se tem $U_2^T R_{22} = 0$ e $U_2^T \Gamma_i = 0$ Multiplica-se a equação (2.26) por U_2^T de ambos lados, resulta:

$$U_2^T H_i R_{11} Q_1 = U_2^T R_{21} Q_1$$

Multiplica-se Q_1^T pela direita nesta última equação, obtém-se:

$$U_2^T H_i = U_2^T R_{21} R_{11}^{-1}$$
 (B 5.1)

esta equação é linear com respeito às matrizes $B \in D$, então pode-se usar o método dos mínimos quadrados para eles. Defina-se

 $U_2^T = [L_1 \ L_2 \ \dots \ L_i]$, $U_2^T R_{21} R_{11}^{-1} = [M_1 \ M_2 \ \dots \ M_i]$

da equação (B 5.1) se tem:

$$\begin{split} L_1 D + L_2 CB + ... + L_{i-1} CA^{i-3} B + L_i CA^{i-2} B &= M_1 \\ L_2 CB + L_3 CB . + .. + L_i CA^{i-3} B &= M_2 \\ L_{i-1} D + L_i CB &= M_{i-1} \\ L_i D &= M_i \end{split}$$

do qual se obtém as matrizes *B* e *D*.

B.6 Prova do Teorema (MOESP ORDINÁRIO, caso determinístico)

A demonstração é feita através, da seguinte seqüência.

- a) Provar, $post(\Gamma_i X_{1,N}) = n$
- b) Provar $\Gamma_i R_{x2}Q_x = R_{22}Q_2$
- c) Aplicar SVD à matriz R_{22}
- d) Recuperar as matrizes $(A, B, C, D) \approx (A_T, B_T, C_T, D_T)$

Aplicando a decomposição RQ (ou LQ) na hipótese 1, se obtém

$$\begin{bmatrix} U_{1,i,N} \\ X_{1,N} \end{bmatrix} = \frac{mi}{n} \begin{bmatrix} \frac{mi}{R_{11}} & 0 \\ R_{x1} & R_{x2} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_x \end{bmatrix}$$
(B 6.1)

Como a sinal de entrada u(k) é persistentemente excitante de ordem 2*i* então, $posto(U_{1,i,N}) = mi$ e pela decomposição LQ, R_{11} e R_{x2} são matrizes quadradas invertivel, isto é $posto(R_{11}) = mi$ e $posto(R_{x2}) = n$.

Da equação (B 6.1) se obtém

$$U_{1,i,N} = R_{11}Q_1 \tag{B 6.2}$$

$$X_{1,N} = R_{x1}Q_1 + R_{x2}Q_x \tag{B 6.3}$$

Para provar a parte a) : $post(\Gamma_i X_{1,N}) = n$, é preciso o seguinte lema

Lema B.1. Desigualdade de Sylvester, para maiores detalhes ver [KAILATH 1980, p.655]:

Seja
$$M_1 \in \mathbb{R}^{mxn}$$
 e $M_2 \in \mathbb{R}^{nxp}$ então
 $post(M_1) + post(M_2) - n \le post(M_1M_2) \le \min\{post(M_1), post(M_2)\}.$

Multiplica-se a equação (B 6.2) por Γ_i de ambos lados, resulta:

$$\Gamma_i X_{1,N} = \Gamma_i (R_{x1}Q_1 + R_{x2}Q_x)$$
(B 6.4)

$$\Gamma_i X_{1,N} = \Gamma_i R_{x1} Q_1 + \Gamma_i R_{x2} Q_x \tag{B 6.5}$$

Multiplica-se a equação (B 6.5) por Q_x^T , se obtém:

$$\Gamma_i X_{1,N} Q_x^T = \Gamma_i R_{x2} \tag{B 6.6}$$

Multiplica-se a equação (B 6.6) por Q_x , se obtém:

$$\Gamma_{i} X_{1,N} = \Gamma_{i} R_{x2} Q_{x}$$
Logo
$$post(\Gamma_{i} X_{1,N}) = post(\Gamma_{i} R_{x2} Q_{x})$$
(B 6.7)

aplicando Lema B.1 para a matriz $\Gamma_i R_{x2}Q_x$, e fazendo uso de $post(\Gamma_i) = n$,

 $posto(R_{x2}Q_x) = n$, temos:

$$post(\Gamma_1) + post(R_{x2}Q_x) - n \le post(\Gamma_1R_{x2}Q_x) \le \min\{post(\Gamma_1), post(R_{x2}Q_x)\}$$
$$n + n - n \le post(\Gamma_1R_{x2}Q_x) \le \min\{n, n\}$$
$$n \le post(\Gamma_1R_{x2}Q_x) \le n$$

de donde se conclui que:

$$post(\Gamma_i X_{1,N}) = post(\Gamma_i R_{x2}Q_x) = n$$
(B 6.8)

Prova-se agora parte b) $\Gamma_i R_{x2}Q_x = R_{22}Q_2$

Da hipótese 3 do teorema (2.3.3) temos

$$\begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix} = \begin{bmatrix} mi \\ li \\ R_{21} \\ R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
(B 6.9)

da equação B47 obtém-se

$$Y_{1,i,N} = R_{21}Q_1 + R_{22}Q_2 \tag{B 6.10}$$

a equação (219) é expressada por

$$Y_{1,i,N} = \Gamma_i X_{1,N} + H_i U_{1,i,N} , \text{ ou}$$

$$\Gamma_i X_{1,N} = Y_{1,i,N} - H_i U_{1,i,N}$$
(B 6.11)

substituindo-se(B 6.10) e (B 6.2) na equação (B 6.11), se obtém:

$$\Gamma_i X_{1,N} = R_{21} Q_1 + R_{22} Q_2 - H_i R_{11} Q_1$$

Ou em forma matricial

$$\Gamma_{i}X_{1,N} = \begin{bmatrix} R_{21} - H_{i}R_{11} | R_{22} \end{bmatrix} \begin{bmatrix} Q_{1} \\ Q_{2} \end{bmatrix}$$
(B 6.12)

igualando as equações (B 6.5) com (B 6.12), resulta:

$$\Gamma_{i} X_{1,N} = \Gamma_{i} R_{x1} Q_{1} + \Gamma_{i} R_{x2} Q_{x} = \begin{bmatrix} R_{21} - H_{i} R_{11} | R_{22} \end{bmatrix} \begin{bmatrix} Q_{1} \\ Q_{2} \end{bmatrix}$$

$$\Gamma_{i} R_{x1} Q_{1} + \Gamma_{i} R_{x2} Q_{x} = (R_{21} - H_{i} R_{11}) Q_{1} + R_{22} Q_{2}$$
 (B 6.13)

_ _

multiplica-se a equação (B 6.13) por Q_1^T de ambos lados e fazendo uso das propriedades da matriz ortogonal $Q_1Q_2 = 0$, $Q_xQ_1 = 0$ e $Q_1Q_1^T = I_{mi}$, se obtém:

$$\Gamma_i R_{x1} = R_{21} - H_i R_{11} \tag{B 6.14}$$

da mesma forma multiplica-se na equação (B 6.13) por Q_2^T de ambos lados e fazendo uso das propriedades da matriz ortogonal, se obtém:

$$\Gamma_i R_{x2} Q_x Q_2^T = R_{22} \quad \text{ou,}$$

$$\Gamma_i R_{x2} Q_x = R_{22} Q_2 \quad (B \ 6.15)$$

da equação (B 6.8) temos que: $post(\Gamma_i R_{x2}Q_x) = n$, substituindo-se (B 6.8) na equação (B 6.15), resulta:

$$post(1, R_{x2}Q_x) = post(R_{22}Q_2) = n$$

 $post(R_{22}Q_2) = n$ (B 6.16)

a equação (B 6.16) quer dizer que o espaço coluna da matriz $R_{22}Q_2$ é igual ao espaço coluna da matriz Γ_i , esta informação é importante para a identificação a traves do método MOESP.

Pela hipótese 1 do teorema 2.3.3, R_{22} é uma matriz ivertivel, e da equação (B 6.16) conclui-se que $post(R_{22}) = n$.

Aplicando-se SVD à matriz R_{22} , se obtém

$$R_{22} = {}_{li} \begin{bmatrix} n \\ U_n \end{bmatrix} {}_{n}^{li-n} \end{bmatrix} \begin{bmatrix} \frac{n}{0} \\ 0 \end{bmatrix} {}_{22}^{li-n} \end{bmatrix} \begin{bmatrix} V^T \\ (V_n^{\perp})^T \end{bmatrix}$$
$$R_{22} = U_n S_n V^T$$
(B 6.17)

Pelo citado acima temos que o espaço coluna de R_{22} é igual ao espaço coluna de Γ_i . da equação (B 6.17), o espaço coluna de R_{22} é igual a U_n , então existe uma matriz $T \in R^{nxn}$ não singular chamada matriz de transformação, tal que:

$$\Gamma_i T = U_n \tag{B 6.18}$$

da definição Γ_i matriz oservabilidade estendida, temos que:

$$\begin{bmatrix} CT\\ (CT)(T^{-1}AT)\\ \vdots\\ (CT)(T^{-1}A^{i-1}T) \end{bmatrix} = \begin{bmatrix} C_T\\ C_TA_T\\ \vdots\\ C_TA_T^{i-1} \end{bmatrix} = U_n$$

agora é fácil recuperar as matrizes $A \in C$ ou $A_T \in C_T$

$$C_T = U_n(1:l,:)$$

$$U_n^{(1)}A_T = U_n(1:l(i-1),:)A_T = U_n(l+1:il,:) = U_n^{(2)}.$$

Por último computa-se as matrizes $B \in D$ a equação (2.19) e (B 6.10) é dada por:

$$\left. \begin{array}{l} Y_{1,i,N} = \Gamma_i X_{1,N} + H_i U_{1,i,N} \\ Y_{1,i,N} = R_{21} Q_1 + R_{22} Q_2 \end{array} \right\} \implies \Gamma_i X_{1,N} + H_i U_{1,i,N} = R_{21} Q_1 + R_{22} Q_2$$
 (B 6.19)

substituindo-se as equações (B 6.2), (B 6.17) e (B 6.18) na equação (B 6.19), resulta:

$$(U_n T^{-1})X_{1,N} + H_i(R_{11}Q_1) = R_{21}Q_1 + (U_n S_n V_n^T)Q_2$$
(B 6.20)

como $(U_n^{\perp})^T U_n = 0$, multiplica-se (B 6.20) por $(U_n^{\perp})^T$ de ambos lados, resulta:

$$\left((U_n^{\perp})^T H_i R_{11} - (U_n^{\perp})^T R_{21} \right) Q_1 = 0$$

Ou

$$(U_n^{\perp})^T H_i R_{11} - (U_n^{\perp})^T R_{21} = 0$$
(B 6.21)

pelo argumento da equação (B 6.1) temos que, R_{11} é uma matriz invertivel, multiplicase (B 6.21) por $(R_{11})^{-1}$ de ambos lados, se obtém:

$$(U_n^{\perp})^T H_i - (U_n^{\perp})^T R_{21}(R_{11}^{-1}) = 0$$

ou

$$(U_n^{\perp})^T R_{21}(R_{11}^{-1}) = (U_n^{\perp})^T H_i$$
(B 6.22)

Denota-se $\Xi = (U_n^{\perp})^T R_{21} R_{11}^{-1}$ e substituindo-se esta notação na equação (B 6.22) resulta: $\Xi = (U_n^{\perp})^T H_i$ (B 6.23)
Como as matrizes A_T e C_T foram computadas em função da matriz U_n a matriz H_i e a equação (B 6.23) pode ser expressada como:

$$\begin{bmatrix} \Xi(:,1:m) \\ \Xi(:,m+1:2m) \\ \vdots \\ \Xi(:,m(i-1)+1:mi) \end{bmatrix} = \begin{bmatrix} U_n^{\perp}(1:l,:)^T & \cdots & U_n^{\perp}(l(i-1)+1:li,:)^T \\ U_n^{\perp}(l+1:2l,:)^T & \cdots & 0 \\ \vdots & \vdots & 0 & 0 \\ U_n^{\perp}(l(i-1)+1:li,:)^T & 0 & 0 \end{bmatrix} \begin{pmatrix} I_l \\ 0 \\ U_n^{(1)} \end{pmatrix} \begin{pmatrix} D \\ B_T \end{pmatrix}$$
(B 6.24)

da equação (B 6.24) é fácil recuperar as matrizes B_T e D.

B.7 Notações matriciais usadas no algoritmo PIMOESP

Os sinais nesta teses, são realizações de processos estocásticos ergódigos [OVERSCHEE DE MOOR, 1996]. Isto é, para $N \rightarrow \infty$ existe um processo estocástico ergódigo $U_j \in \Re^m$ e $V_k \in \Re^l$ tais que

$$[u_j \ u_{j+1} \dots u_{N+j-1}]$$
 e $[v_k \ v_{k+1} \dots v_{N+k-1}]$

são realizações de U_j e V_k respectivamente, e pela ergocidade, temos que

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} u_{j+i-1} v_{k+i-1}^{T} = E \left[U_{j} V_{k}^{T} \right]$$
(B 7.1)

onde E[] é o operador expectância.

Outra forma de representar a equação (B 7.1) é:

$$\frac{1}{N} \sum_{i=1}^{N} u_{j+i-1} v_{k+i-1}^{T} + O_{N}(\varsigma) = E \left[U_{j} V_{k}^{T} \right]$$
(B 7.2)

onde $O_N(\varsigma)$ é uma matriz limitada de dimensão apropriada de norma ς , a qual é zero para $N \to \infty$.

Da mesma forma da equação (B 3.10), pode-se obter

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N} + V_{k,i,N}$$
(B 7.3)

onde $V_{k,i,N}$ é uma matriz de Hankel cujos elementos v_k são os ruídos, e é construída da mesma forma que a equação (B 3.11).

Definição B7.1 Uma sequência ergódiga $u_k \in \mathbb{R}^m$ é persistentemente excitante de ordem *i* desde o instante *k*, se e sô se

$$posto\left(\lim_{N \to \infty} \frac{1}{N} U_{k,i,N} U_{k,i,N}^{T}\right) = mi \quad . \tag{B 7.4}$$

A condição (B 7.4) pode ser denotada como:

$$posto\left(\frac{1}{N}U_{k,i,N}U_{k,i,N}^{T}+O_{N}^{k}(\varsigma^{k})\right)=mi$$
(B 7.5)

Para N suficientemente grande, a definição de persistência de excitação da seqüência u_k pode ser interpretada da mesma forma que para dados finitos, isto é [VERHAEGEN 1992b],

$$posto(U_{k,i,N}) = mi$$
 (B 7.6)

Definição B7. 2 A sequência $U_k \in \mathbb{R}^m$ é uma sequência de ruído branco, se tem media zero e satisfaz as seguintes condições

$$E[U_k U_j^T] = \begin{cases} \sigma_u^2 I_m & , \ k = j \\ 0 & , \ k \neq j \end{cases}$$
(B 7.7)

Ou

$$E\left(\begin{bmatrix} U_{k} \\ U_{k+1} \\ \vdots \\ U_{k+i-1} \end{bmatrix} \begin{bmatrix} U_{k}^{T} & U_{k+1}^{T} & \cdots & U_{k+i-1}^{T} \end{bmatrix} \right) = \sigma_{u}^{2} \begin{bmatrix} I_{m} & 0 & \cdots & 0 \\ 0 & I_{m} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & I_{m} \end{bmatrix}.$$
(B 1.700)

Baixo a hipóteses da ergocida, a equação (B 7.7) se pode expressar como:

$$\lim_{N \to \infty} \frac{1}{N} U_{k,i,N} U_{k,i,N}^{T} = \sigma_{u}^{2} I_{mi}$$
(B 7.11)

Ou

$$\frac{1}{N}U_{k,i,N}U_{k,i,N}^{T} = \sigma_{u}^{2}I_{mi} + O_{N}^{2}(\varsigma)$$
 (B 7.12)

B.8 Teorema 8.1

Suponha as seguintes hipóteses

1. Seja u(k) uma sinal persistentemente excitante, tal que

$$posto\left[\frac{U_{1,i,N}}{X_{1,N}}\right] = mi + n \tag{B 8.1}$$

$$2. \qquad i > n$$

3. A fatoração LQ das matrizes

$$\left[\frac{U_{1,i,N}}{Y_{1,i,N}}\right] e \left[\frac{U_{1,i,N}}{X_{1,N}}\right]$$

é dada por:

$$\begin{bmatrix} U_{1,i,N} \\ Y_{1,i,N} \end{bmatrix} = \begin{bmatrix} mi \\ li \\ R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
(B 8.2)

$$\begin{bmatrix} \underline{U}_{1,i,N} \\ \overline{X}_{1,N} \end{bmatrix} = \begin{bmatrix} mi & n \\ R_{11} & 0 \\ R_{x1} & R_{x2} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_x \end{bmatrix}$$
(B 8.3)

4. Dado os limites

$$\lim_{N \to \infty} \frac{1}{N} U_{1,i,N} (U_{1,i,N})^T = R_u$$
(B 8.4)

$$\lim_{N \to \infty} \frac{1}{N} V_{1,i,N} (V_{1,i,N})^T = R_v$$
(B 8.5)

$$\lim_{N \to \infty} \frac{1}{N} R_{x1} (R_{x1})^T + \lim_{N \to \infty} \frac{1}{N} R_{x2} (R_{x2})^T = P_{x1} + P_{x2}$$
(B 8.6)

$$\lim_{N \to \infty} \frac{1}{N} U_{1,i,N} (V_{1,i,N})^T = 0$$
(B 8.7)

$$\lim_{N \to \infty} \frac{1}{N} X_{1,N} (V_{1,i,N})^T = 0$$
(B 8.8)

Então tem-se que

$$\lim_{N \to \infty} \frac{1}{N} R_{22} (R_{22})^T = \Gamma_i P_{x2} \Gamma_i^T + Rv$$
 (B 8.9)

Prova

Por comodidade considerar a seguinte notação $U_p = U_{1,i,N}$, $Y_p = Y_{1,i,N}$, $V_p = V_{1,i,N}$ e

$$X_p = X_{1,N}$$
.
Da equação (B 8.2) se obtém:
 $U_p = R_{11}Q_1$

$$Y_p = R_{21}Q_1 + R_{22}Q_2 \tag{B 8.11}$$

Na equação (B 7.3), para o caso k = 1, é dada por:

$$Y_{1,i,N} = \Gamma_i X_{1,N} + H_i U_{1,i,N} + V_{1,i,N}$$

ou em forma equivalente

$$Y_p = \Gamma_i X_p + H_i U_p + V_p \tag{B 8.12}$$

Igualando as equações (B 8.11) e (B 8.12) resulta:

$$R_{21}Q_1 + R_{22}Q_2 = \Gamma_i X_p + H_i U_p + V_p .$$
 (B 8.13)

Da equação (B 8.3) temos que:

_

$$X_{p} = R_{x1}Q_{1} + R_{x2}Q_{x}$$
(B 8.14)

Substituindo-se (B 8.14) e (B 8.10) na equação (B 8.13), se obtém:

$$\Gamma_{i}(R_{x1}Q_{1} + R_{x2}Q_{2}) + H_{i}(R_{11}Q_{1}) + V_{p} = R_{21}Q_{1} + R_{22}Q_{2}$$

ou
$$\Gamma_{i}R_{x1}Q_{1} + \Gamma_{i}R_{x2}Q_{2} + H_{i}R_{11}Q_{1} + V_{p} = R_{21}Q_{1} + R_{22}Q_{2}$$
(B 8.15)

-

da hipóteses (B 8.7) temos que :

$$\lim_{N \to \infty} \frac{1}{N} V_p (U_p)^T = 0$$
(B 8.16)

a equação (B 8.16) pode ser expressada na forma [vide apêndice B.7, equação (B 7.2)]

$$\frac{1}{N}V_p(U_p)^T = O(\varsigma) \tag{B 8.17}$$

Substituindo-se a equação (B 8.10) na equação (B 8.16), obtém-se:

$$\frac{1}{N}V_p(R_{11}Q_1)^T = O(e)$$

(B 8.10)

$$\frac{1}{\sqrt{N}} V_p \frac{1}{\sqrt{N}} Q_1^T R_{11}^T = O(e)$$

$$\frac{1}{\sqrt{N}} V_p Q_1^T = O(e) \left(\frac{1}{\sqrt{N}} R_{11}^T\right)^{-1}$$
(B 8.18)

multiplica-se a equação (B 8.14) por V_p^T de ambos lados resulta:

$$X_p V_p^T = R_{x1} Q_1 V_p^T + R_{x2} Q_x V_p^T$$

aplicando limite para $N \rightarrow \infty$, nesta última equação, obtemos:

$$\lim_{N \to \infty} \frac{1}{N} X_p V_p^T = \lim_{N \to \infty} \frac{1}{N} R_{x1} Q_1 V_p^T + \lim_{N \to \infty} \frac{1}{N} R_{x2} Q_x V_p^T$$
(B 8.19)

pela hipótese (B 8.8), a equação (B 8.19), resulta:

$$0 = \lim_{N \to \infty} \frac{1}{N} R_{x1} Q_1 V_p^T + \lim_{N \to \infty} \frac{1}{N} R_{x2} Q_x V_p^T$$
(B 8.20)

ou

$$0 = \frac{1}{\sqrt{N}} R_{x1} \left(\frac{1}{\sqrt{N}} Q_1 V_p^T \right) + O_2(e) + \frac{1}{N} R_{x2} Q_x V_p^T + O_3(e)$$
(B 8.21)

Substituindo-se a equação (B 8.18) na equação (B 8.21), resulta:

$$O_4(e) = \frac{1}{\sqrt{N}} R_{x1} \left(\frac{1}{\sqrt{N}} R_{11} \right)^{-1} O(e)^T + \frac{1}{N} R_{x2} Q_x V_p^T$$
(B 8.22)

onde $O_4(e) = -(O_2(e) + O_3(e))$

Como os limites da hipóteses (B 8.4) e (B 8.6) existem, então os limites dos fatores $\lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} R_{x1} \right) e \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} R_{11} \right) também existem, portanto existe o limite do produto$ $\lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} R_{x1} \right) \left(\frac{1}{\sqrt{N}} R_{11} \right)^{-1}$

Então da equação (B 8.223) pode-se definir uma nova sequência de números reais de matrizes limitadas $O_5(e)$, tais que

$$\frac{1}{N}R_{x2}Q_{x}V_{p}^{T} = O_{5}(e)$$
(B 8.23)

onde
$$O_5(e) = \left(O_4(e) - \frac{1}{\sqrt{N}} R_{x1} \left(\frac{1}{\sqrt{N}} R_{11}\right)^{-1} O(e)^T\right)$$
 tende a zero quando $N \to \infty$.

A equação (B 8.23) expressada em termos do limite é:

$$\lim_{N \to \infty} \frac{1}{N} R_{x2} Q_x V_p^T = 0.$$
 (B 8.24)

Da equação (B 8.15) se obtém:

$$\Gamma_i R_{x1} Q_1 + \Gamma_i R_{x2} Q_x + H_i R_{11} Q_1 + V_p - R_{21} Q_1 = R_{22} Q_2$$
(B 8.25)

Multiplica-se a equação (B 8.25) por Q_2^T de ambos lados, resulta

$$\Gamma_{i}R_{x1}\underbrace{Q_{1}(Q_{2}^{T})}_{=0} + \Gamma_{i}R_{x2}\underbrace{Q_{x}(Q_{2}^{T})}_{=I} + H_{i}R_{11}\underbrace{Q_{1}(Q_{2}^{T})}_{=0} + V_{p}(Q_{2}^{T}) - R_{21}\underbrace{Q_{1}(Q_{2}^{T})}_{=0} = R_{22}\underbrace{Q_{2}(Q_{2}^{T})}_{=I}$$

$$\Gamma_{i}R_{x2}Q_{x}Q_{2}^{T} + V_{p}Q_{2}^{T} = R_{22}.$$
(B 8.26)

Por último calcular o produto $R_{22}(R_{22})^T$, e substituindo-se a equação (B 8.23) neste produto, se obtém:

$$\frac{1}{N}R_{22}(R_{22})^{T} = \frac{1}{N} \Big(\Gamma_{i}R_{x2}Q_{x}Q_{2}^{T} + V_{p}Q_{2}^{T} \Big) \Big(\Gamma_{i}R_{x2}Q_{x}Q_{2}^{T} + V_{p}Q_{2}^{T} \Big)^{T} \\ \frac{1}{N}R_{22}(R_{22})^{T} = \frac{1}{N} \Big(\Gamma_{i}R_{x2}Q_{x}Q_{2}^{T} + V_{p}Q_{2}^{T} \Big) \Big(Q_{2}Q_{x}^{T}R_{x2}^{T}\Gamma_{i}^{T} + Q_{2}V_{p}^{T} \Big) \\ \frac{1}{N}R_{22}(R_{22})^{T} = \frac{1}{N} \Bigg(\Gamma_{i}R_{x2}R_{x2}^{T}\Gamma_{i}^{T} + \Gamma_{i}\underbrace{R_{x2}Q_{x}V_{p}^{T}}_{=O_{5}(e)} + \underbrace{V_{p}^{T}Q_{x}^{T}R_{x2}^{T}}_{=O_{5}(e)} \Gamma_{i}^{T} + V_{p}V_{p}^{T} \Big) \\ \frac{1}{N}R_{22}(R_{22})^{T} = \frac{1}{N}\Gamma_{i}R_{x2}R_{x2}^{T}\Gamma_{i}^{T} + \Gamma_{i}O_{5}(e) + O_{5}(e)\Gamma_{i}^{T} + \frac{1}{N}\Big(V_{p}V_{p}^{T}\Big)$$
(B 8.27)

tomando limite para $N \rightarrow \infty$, a equação (B 8.27) resulta:

$$\lim_{N \to \infty} \frac{1}{N} R_{22} (R_{22})^T = \lim_{N \to \infty} \frac{1}{N} \Gamma_i R_{x2} R_{x2}^T \Gamma_i^T + \Gamma_i \lim_{N \to \infty} O_5(e) + \lim_{N \to \infty} O_5(e) \Gamma_i^T + \lim_{N \to \infty} \frac{1}{N} (V_p V_p^T)$$

$$\lim_{N \to \infty} \frac{1}{N} R_{22} (R_{22})^{T} = \Gamma_{i} \lim_{N \to \infty} \frac{1}{N} R_{x2} R_{x2}^{T} \Gamma_{i}^{T} + \lim_{N \to \infty} \frac{1}{N} (V_{p} V_{p}^{T})$$
$$\lim_{N \to \infty} \frac{1}{N} R_{22} (R_{22})^{T} = \Gamma_{i} P_{x2} \Gamma_{i}^{T} + R_{v} \quad .$$
(B 8.28)

B.9 Prova de: Outra pergunta é porque considerar os dados de entradas passadas como variável instrumento?

Suponha que a variável instrumento é $W \in \Re^{ixN}$.

A equação (B 8.25) é dada por:

$$\Gamma_i R_{x1} Q_1 + \Gamma_i R_{x2} Q_x + H_i R_{11} Q_1 + V_p - R_{21} Q_1 = R_{22} Q_2$$

ou

$$\left(\Gamma_{i}R_{x1}Q_{1} + H_{i}R_{11}Q_{1} - R_{21}Q_{1}\right) + \Gamma_{i}R_{x2}Q_{x} + V_{p}$$
(B 9.1)

Multiplica-se a equação (B 8.25) por Q_1^T de ambos lados, resulta:

$$(\Gamma_{i}R_{x1}Q_{1} + \Gamma_{i}R_{x2}Q_{x} + H_{i}R_{11}Q_{1} + V_{p})Q_{1}^{T} = R_{21}Q_{1}Q_{1}^{T} + R_{22}Q_{2}Q_{1}^{T}$$

$$\Gamma_{i}R_{x1} + H_{i}R_{11} + V_{p}Q_{1}^{T} = R_{21}$$

$$\Gamma_{i}R_{x1}Q_{1} + H_{i}R_{11}Q_{1} + V_{p} = R_{21}Q_{1}$$
(B 9.2)

Substituindo-se a equação (B 9.2) na equação (B 9.1) se obtém:

$$R_{22}Q_{2} = \left(\Gamma_{i}R_{x1}Q_{1} + H_{i}R_{11}Q_{1} - \left(\Gamma_{i}R_{x1}Q_{1} + H_{i}R_{11}Q_{1} + V_{p}\right)\right) + \Gamma_{i}R_{x2}Q_{x} + V_{p}$$

$$R_{22}Q_{2} = -V_{p} + \Gamma_{i}R_{x2}Q_{x} + V_{p}$$

$$R_{22}Q_{2} = \Gamma_{i}R_{x2}Q_{x} - V_{p} + V_{p}$$
(B 9.3)

Da equação (B 8.18) se obtém que:

$$\frac{1}{\sqrt{N}}V_{p} = O(e) \left(\frac{1}{\sqrt{N}}R_{11}^{T}\right)^{-1} Q_{1}^{T}$$
(B 9.4)

Substituindo-se a equação (B 9.4) na equação (B 9.3), resulta:

$$\frac{1}{\sqrt{N}}R_{22}Q_2 = \frac{1}{\sqrt{N}}\Gamma_i R_{x2}Q_x - O(e) \left(\frac{1}{\sqrt{N}}R_{11}^T\right)^{-1}Q_1 + \frac{1}{\sqrt{N}}V_p$$
(B 9.5)

1

Agora o uso da variável instrumental no caso PIMOESP corresponde à multiplicação na equação (B 9.5) por W^T , isto é

$$\frac{1}{\sqrt{N}}R_{22}Q_2W^T = \frac{1}{\sqrt{N}}\Gamma_i R_{x2}Q_xW^T - O(e)\left(\frac{1}{\sqrt{N}}R_{11}^T\right)^{-1}Q_1W^T + \frac{1}{\sqrt{N}}V_pW^T$$
(B 9.6)

Agora, Suponha que $\lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T$ exista.

Então tomando o limite para $N \rightarrow \infty$, na equação (B 9.6), resulta:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{22} Q_2 W^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i R_{x2} Q_x W^T - \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \lim_{N \to \infty} \frac{1}{\sqrt{N}} V_p W^T$$
$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{22} Q_2 W^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i R_{x2} Q_x W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T + \underbrace{\lim_{N \to \infty} O(e)}_{=0} \left(\frac{1}{\sqrt{N}} R_{11}^T \right)^{-1} Q_1 W^T$$

Desta última equação concluímos que a variável instrumento escolhida tem que satisfazer as seguintes condições

i)Tem que ser estatisticamente independente da sinal de ruído, isto é $\lim_{N \to \infty} \frac{1}{\sqrt{N}} V_p W^T = 0$

ii) E
$$posto\left(\frac{1}{\sqrt{N}}\Gamma_i R_{x2} Q_x W^T\right) = n$$
.

B.10 Prova do teorema 2

A equação (2.45) é dada por :

$$\begin{bmatrix} U_{i+1,i,N} \\ U_{1,i,N} \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \text{ ou } \begin{bmatrix} U_f \\ U_p \\ Y_f \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$
(B 10.1)

Provaremos primeiro a equação (2.46)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f(Q_2)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_f(Q_2)^T$$

Para provar a equação (2.46) é preciso provar:

i)
$$\lim_{N\to\infty}\frac{1}{N}V_fQ_2^T=0.$$

Em efeito, como os dados de entrada é linearmente independe em relação à sinal v(k), então

$$\frac{1}{N}V_{f}U_{p}^{T} = E[V_{k}U_{j}^{T}] = 0 \quad \text{quando} \quad N \to \infty$$
(B 10.2)

substituindo a equação (B 10.1) na equação (B 10.2), se obtém:

$$\frac{1}{N}V_f Q_1^T R_{21}^T + \frac{1}{N}V_f Q_2^T R_{22}^T = O(e)$$
(B 10.3)

ou

$$\lim_{N \to \infty} \frac{1}{N} V_f Q_1^T R_{21}^T + \lim_{N \to \infty} \frac{1}{N} V_f Q_2^T R_{22}^T = \lim_{N \to \infty} O(e) = 0$$
(B 10.4)

substituindo-se a equação (B 10.2) na equação (B 10.4), resulta:

$$\lim_{N \to \infty} \frac{1}{N} V_f Q_2^T R_{22}^T = 0.$$
 (B 10.5)

Por ser u(k) sinal de entrada persistentemente excitante, então a matriz R_{22} é não singular, então da equação (B 10.5) temos:

$$\lim_{N\to\infty}\frac{1}{N}V_f Q_2^T = 0.$$

De (2.37) e (B 10.1) temos que :

$$Y_{f} = \Gamma_{i}X_{f} + H_{i}U_{f} + V_{f}$$

$$Y_{f} = \Gamma_{i}X_{f} + H_{i}R_{11}Q_{1} + V_{f}$$
(B 10.6)

Multiplica-se a equação (B 10.6) por Q_2^T de ambos lados, resulta

$$Y_{f}Q_{2}^{T} = \Gamma_{i}X_{f}Q_{2}^{T} + H_{i}R_{11}\underbrace{Q_{1}Q_{2}^{T}}_{=0} + V_{f}Q_{2}^{T}$$

$$Y_{f}Q_{2}^{T} = \Gamma_{i}X_{f}Q_{2}^{T} + V_{f}Q_{2}^{T}$$
(B 10.7)

Da equação (B 10.1) pode-se expressar Y_f como:

$$Y_f = R_{31}Q_1 + R_{32}Q_2 + R_{33}Q_3 \tag{B 10.8}$$

Multiplica-se a equação (B 10.8) por Q_2^T de ambos lados, resulta:

$$Y_{f}Q_{2}^{T} = R_{31}\underbrace{Q_{1}Q_{2}^{T}}_{=0} + R_{32}\underbrace{Q_{2}Q_{2}^{T}}_{=I} + R_{33}\underbrace{Q_{3}Q_{2}^{T}}_{=0}$$

$$Y_{f}Q_{2}^{T} = R_{32}$$
(B 10.9)

Igualando as equações (B 10.9) e (B 10.7) se obtém:

$$R_{32} = \Gamma_i X_f Q_2^T + V_f Q_2^T$$
(B 10.10)

tomando limite para $N \rightarrow \infty$ na equação (B 10.10)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f Q_2^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{32} = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_f Q_2^T + \lim_{N \to \infty} \frac{1}{\sqrt{N}} V_f Q_2^T$$
(B 10.11)

o último termo do lado direito na equação (B 10.11) é zero, portanto

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}Y_fQ_2^T = \lim_{N\to\infty}\frac{1}{\sqrt{N}}\Gamma_iX_fQ_2^T.$$

Para encontrar as matrizes $A \in C$, aplica-se SVD em R_{32} (ver algoritmo MOESP ordinario).

$$R_{32} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T$$

onde $n = \dim(\Sigma_1)$, $[U_1]_{lixn}$ e $[U_1]_{lix(li-n)}$. Então

$$\Gamma_i X_f Q_2^T = R_{22} = U_1 \Sigma_1 V_1^T$$

defina-se

$$\Gamma_i = U_1 \Sigma_1^{1/2}$$

a matriz C é facilmente obtida de

$$C = \Gamma_i(1:l,1:n)$$

e a matriz A é obtida de

$$\Gamma_i(1:l(i-1),1:n)A = \Gamma_i(l+1:il,1:n)$$

Agora se provara (2.47)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_{i+1,i,N}(Q_1)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \left(\Gamma_i X_{i+1,N}(Q_1)^T + H_i R_{11} \right)$$

Da equação (B 10.10)

$$Y_f = \Gamma_i X_f + H_i R_{11} Q_1 + V_f$$
 (B 10.12)

multiplica-se a equação (B 10.12) por Q_1^T ambos lados, resulta:

$$Y_{f}Q_{1}^{T} = \Gamma_{i}X_{f}Q_{1}^{T} + H_{i}R_{11}\underbrace{Q_{1}Q_{1}^{T}}_{=I} + \underbrace{V_{f}Q_{1}^{T}}_{=0}$$

$$Y_{f}Q_{1}^{T} = \Gamma_{i}X_{f}Q_{1}^{T} + H_{i}R_{11}$$
(B 10.13)

Da equação (B 10.1) Y_f pode ser expressa como:

$$Y_f = R_{31}Q_1 + R_{32}Q_2 + R_{33}Q_3$$
 (B 10.14)

multiplica-se a equação (B 10.14) por Q_1^T de ambos lados, resulta

$$Y_{f}Q_{1}^{T} = R_{31}\underbrace{Q_{1}Q_{1}^{T}}_{=I} + R_{32}\underbrace{Q_{2}Q_{1}^{T}}_{=0} + R_{33}\underbrace{Q_{3}Q_{1}^{T}}_{=0}$$

$$Y_{f}Q_{1}^{T} = R_{31}$$
(B 10.15)

Igualando as equações (B 10.15) e (B 10.13)

$$Y_f Q_1^T = R_{31} = \Gamma_i X_f Q_1^T + H_i R_{11}$$
(B 10.16)

tomando limite para $N \! \rightarrow \! \infty$ na equação (B 10.16)

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f Q_1^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{31} = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_f Q_1^T + \frac{1}{\sqrt{N}} H_i R_{11} \right).$$
(B 10.17)

Encontrando as matrizes B e D

Como U_1 é uma estimativa consistente do espaço coluna da matriz observabilidade estendida Γ_i e $(U_2^{\perp})^T$ complemento ortogonal de U_1 , então a equação (B 10.17) pode ser reduzida:

$$(U_2^{\perp})^T \frac{1}{\sqrt{N}} R_{31} = (U_2^{\perp})^T H_i \frac{1}{\sqrt{N}} R_{11} + O(e)$$
(B 10.18)

Por ser u(k) sinal de entrada persistentemente excitante, então a matriz R_{11} é não singular, portanto da equação (B 10.18), resulta:

$$(U_2^{\perp})^T \frac{1}{\sqrt{N}} R_{31} (R_{11})^{-1} = (U_2^{\perp})^T H_i \frac{1}{\sqrt{N}} + O(e)$$
(B 10.19)

denotando $\Xi = (U_2^{\perp})^T R_{31} R_{11}^{-1}$ e $N \to \infty$, a equação (B 10.19) é expressada por

$$\Xi = (U_2^{\perp})^T H_i + O(e)$$

Logo as matrizes $B \in D$ são encontradas da mesma forma que no algoritmo MOESP ordinário.

B.11 Notações matriciais usadas no algoritmo POMOESP

Alem das notações citadas no algoritmo MOESP ordinário y algoritmo PIMOESP define-se a seguinte equação matricial das saídas:

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N} + \Phi_i W_{k,i,N} + V_{k,i,N}$$
(B 11.1)

Não é difícil obter esta equação recursiva, pois é somente agregar as duas últimas matrizes na equação (B 3.10), por exemplo a equação (B 3.7) pode ser expressada para o caso POMOESP como:

$$\begin{bmatrix} y_{s} \\ y_{s+1} \\ \vdots \\ y_{s+i-2} \\ y_{s+i-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-2} \\ CA^{i-2} \\ CA^{i-1} \end{bmatrix} x_{s} + \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ CA^{i-3}B & CA^{i-4}B & \cdots & D & 0 \\ CA^{i-2}B & CA^{i-3}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u_{s} \\ u_{s+1} \\ \vdots \\ u_{s+i-2} \\ u_{s+i-1} \end{bmatrix} + \\ + \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ CB & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3} & CA^{i-4} & \cdots & 0 & 0 \\ CA^{i-2} & CA^{i-3} & \cdots & C & 0 \end{bmatrix} \begin{bmatrix} w_{s} \\ w_{s+1} \\ \vdots \\ w_{s+i-2} \\ w_{s+i-1} \end{bmatrix} + \begin{bmatrix} v_{s} \\ v_{s+1} \\ \vdots \\ v_{s+i-2} \\ v_{s+i-1} \end{bmatrix}$$
(B 11.2)

A equação (B 11.2) pode-se expressar em forma mais general como:

$$\begin{bmatrix} y_{k} & y_{k+1} & \cdots & y_{k+s} & \cdots & y_{k+N-1} \\ y_{k} & y_{k+2} & \cdots & y_{k+s+1} & \cdots & y_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{k+i-1} & y_{k+i} & \cdots & y_{k+s+i} & \cdots & y_{k+N+i-2} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-2} \\ CA^{i-1} \end{bmatrix} \begin{bmatrix} x_{k} & x_{k+1} & \cdots & x_{k+s} & \cdots & x_{k+N-1} \end{bmatrix} +$$

$$+ \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3}B & CA^{i-4}B & \cdots & D & 0 \\ CA^{i-2}B & CA^{i-3}B & \cdots & CB & D \end{bmatrix} \begin{bmatrix} u_k & u_{k+1} & \cdots & u_{k+s} & \cdots & u_{k+N-1} \\ u_{k+1} & u_{k+2} & \cdots & u_{k+s+1} & \cdots & u_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{k+i-1} & u_{k+i} & \cdots & u_{k+s+i} & \cdots & u_{k+N+i-2} \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ CB & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3} & CA^{i-4} & \cdots & 0 & 0 \\ CA^{i-2} & CA^{i-3} & \cdots & C & 0 \end{bmatrix} \begin{bmatrix} w_{k} & w_{k+1} & \cdots & w_{k+s} & \cdots & w_{k+N-1} \\ w_{k} & w_{k+2} & \cdots & w_{k+s+1} & \cdots & w_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{k+i-1} & w_{k+i} & \cdots & w_{k+s+i} & \cdots & w_{k+N+i-2} \end{bmatrix} + \\ + \begin{bmatrix} v_{k} & v_{k+1} & \cdots & v_{k+s} & \cdots & v_{k+N-1} \\ v_{k} & v_{k+2} & \cdots & v_{k+s+1} & \cdots & v_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{k+i-1} & v_{k+i} & \cdots & v_{k+s+i} & \cdots & v_{k+N+i-2} \end{bmatrix}$$
(B 11.3)

A equação (B 11.3) pode ser expressa a traves da forma compacta como:

$$Y_{k,i,N} = \Gamma_i X_{k,N} + H_i U_{k,i,N} + \Phi_i W_{k,i,N} + V_{k,i,N}$$
(B 11.4)

Onde Φ_i é a matriz em blocos de Toeplitz, definida como:

$$\Phi_{i} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ CB & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ CA^{i-3} & CA^{i-4} & \cdots & 0 & 0 \\ CA^{i-2} & CA^{i-3} & \cdots & C & 0 \end{bmatrix}$$
(B 11.5)

E as matrizes em blocos de ruídos dos estados e ruído de saída do processo são expressas por:

$$W_{k,i,N} = \begin{bmatrix} w_k & w_{k+1} & \cdots & w_{k+s} & \cdots & w_{k+N-1} \\ w_k & w_{k+2} & \cdots & w_{k+s+1} & \cdots & w_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{k+i-1} & w_{k+i} & \cdots & w_{k+s+i} & \cdots & w_{k+N+i-2} \end{bmatrix}$$
(B 11.6)

$$V_{k,i,N} = \begin{bmatrix} v_k & v_{k+1} & \cdots & v_{k+s} & \cdots & v_{k+N-1} \\ v_k & v_{k+2} & \cdots & v_{k+s+1} & \cdots & v_{k+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{k+i-1} & v_{k+i} & \cdots & v_{k+s+i} & \cdots & v_{k+N+i-2} \end{bmatrix}$$
(B 11.7)

B.12 Prova do teorema 3

A equação (2.63) é dada por:

$$\begin{bmatrix} U_{f} \\ U_{p} \\ Y_{p} \\ Y_{f} \end{bmatrix} = \begin{bmatrix} U_{i+1,i,N} \\ U_{1,i,N} \\ Y_{1,i,N} \\ Y_{i+1,i,N} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q_{1} \\ Q_{2} \\ Q_{3} \\ Q_{4} \end{bmatrix}$$
(B 12.1)

Provaremos primeiro a equação (2.64) expressada por:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_f(Q_2)^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} \Gamma_i X_f(Q_2)^T$$
(B 12.3)

Considerando
$$k = i + 1$$
 na equação (B 11.4) temos

$$Y_{i+1,i,N} = \Gamma_i X_{i+1,N} + H_i U_{i+1,i,N} + \Phi_i W_{i+1,i,N} + V_{i+1,i,N}$$
(B 12.4)

Multiplica-se a equação (B 12.4) por $\frac{1}{\sqrt{N}} Q_2^T$ de ambos lados, resulta

$$\frac{1}{\sqrt{N}}Y_{i+1,i,N}Q_{2}^{T} = \frac{1}{\sqrt{N}}\Gamma_{i}X_{i+1,N}Q_{2}^{T} + \frac{1}{\sqrt{N}}H_{i}\underbrace{U_{i+1,i,N}Q_{2}^{T}}_{I_{1}} + \frac{1}{\sqrt{N}}\Phi_{i}\underbrace{W_{i+1,i,N}Q_{2}^{T}}_{I_{2}} + \frac{1}{\sqrt{N}}\underbrace{V_{i+1,i,N}Q_{2}^{T}}_{I_{3}} + \frac{1}{\sqrt{N}}\underbrace{V_{i+1,i,N}Q_{2}}$$

Onde os últimos ter termos, isto é $I_1,I_2\,$ e $I_3\,$ tendem a zero quando $\,N \,{\to}\,\infty$

Em efeito

No caso I_1

Da equação (B 12.1) o termo $U_f = U_{i+1,i,N}$ é expresso como:

$$U_f = U_{i+1,i,N} = R_{11}Q_1$$
 (B 12.6)

Multiplica-se a equação (B 12.6) por Q_2^T de ambos lados, resulta:

$$U_{i+1,i,N}Q_2^T = R_{11}Q_1Q_2^T = 0$$
.

Para os casos I_2 e I_3

Da equação (B 12.1) o termo $U_p = U_{1,i,N}$ é expresso como:

$$U_p = U_{1,i,N} = R_{21}Q_1 + R_{22}Q_2 \tag{B 12.7}$$

Em efeito, como os dados de entrada é linearmente independe em relação à sinal v(k), então

$$\frac{1}{N}V_{f}U_{p}^{T} = \frac{1}{N}V_{i+1,i,N}U_{1,i,N}^{T} = E[V_{k}U_{j}^{T}] = 0 \quad \text{quando } N \to \infty$$
(B 12.8)

substituindo a equação (B 12.7) na equação (B 12.8), se obtém:

$$\frac{1}{N}V_{f}Q_{1}^{T}R_{21}^{T} + \frac{1}{N}V_{f}Q_{2}^{T}R_{22}^{T} = O(e)$$
(B 12.9)

ou

$$\lim_{N \to \infty} \frac{1}{N} V_f Q_1^T R_{21}^T + \lim_{N \to \infty} \frac{1}{N} V_f Q_2^T R_{22}^T = \lim_{N \to \infty} O(e) = 0$$
(B 12.10)

Portanto da equação (B 12.10) temos que:

$$\lim_{N \to \infty} \frac{1}{N} V_f Q_2^T R_{22}^T = 0$$
 (B 12.11)

Ou

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} V_f Q_2^T \frac{1}{\sqrt{N}} R_{22}^T = 0$$
(B 12.12)

Por ser o sinal de entrada u(k) persistentemente excitante, então a matriz $\frac{1}{\sqrt{N}}R_{22}$ é

invertivel, então multiplica-se a equação (B 12.11) por $\left(\frac{1}{\sqrt{N}}R_{22}^{T}\right)^{-1}$ de ambos lados,

resulta:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} V_f Q_2^T \left(\frac{1}{\sqrt{N}} R_{22}^T \right) \left(\frac{1}{\sqrt{N}} R_{22}^T \right)^{-1} = 0 \left(\frac{1}{\sqrt{N}} R_{22}^T \right)^{-1}$$

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} V_f Q_2^T = 0.$$
 (B 12.13)

Baixo o mesmo argumento se demonstra:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} W_f Q_2^T = \lim_{N \to \infty} \frac{1}{\sqrt{N}} W_{i+1,i,N} Q_2^T = 0.$$
 (B 12.14)

Para $N \rightarrow \infty$, a equação (B 12.5) é dada por:

$$\frac{1}{\sqrt{N}}Y_{i+1,i,N}Q_2^T = \frac{1}{\sqrt{N}}\Gamma_i X_{i+1,N}Q_2^T + O(\varsigma)$$

Ou em forma equivalente

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}Y_f(Q_2)^T = \lim_{N\to\infty}\frac{1}{\sqrt{N}}\Gamma_i X_f(Q_2)^T.$$

Agora se provará (2.65)

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}Y_f(Q_3)^T = \lim_{N\to\infty}\frac{1}{\sqrt{N}}\Gamma_iX_f(Q_3)^T$$

Para provar (2.65) precisamos primeiro provar que:

i)
$$\frac{1}{N}Y_pW_f^T = O(\varsigma)$$
 (B 12.15)

ii)
$$\frac{1}{\sqrt{N}}Q_{1}W_{f}^{T} = O(\varsigma)$$
 (B 12.16)

iii)
$$\frac{1}{\sqrt{N}}Q_2W_f^T = O(\varsigma)$$
 (B 12.17)

Em efeito provaremos primeiro a parte i)

Seja k = 1 na equação (B 11.4) e multiplica-se esta por $\frac{1}{N}W_f^T = \frac{1}{N}W_{i+1,i,N}^T$ de ambos lados, resulta:

$$\frac{1}{N}Y_{1,i,N}W_{i+1,i,N}^{T} = \frac{1}{N}\Gamma_{i}X_{1,N}W_{i+1,i,N}^{T} + \frac{1}{N}H_{i}U_{1,i,N}W_{i+1,i,N}^{T} + \frac{1}{N}\Phi_{i}W_{1,i,N}W_{i+1,i,N}^{T} + \frac{1}{N}V_{1,i,N}W_{i+1,i,N}^{T}$$
ou

$$\frac{1}{N}Y_{p}W_{f}^{T} = \frac{1}{N}\Gamma_{i}\underbrace{X_{p}W_{f}^{T}}_{I1} + \frac{1}{N}H_{i}\underbrace{U_{p}W_{f}^{T}}_{I2} + \frac{1}{N}\Phi_{i}\underbrace{W_{p}W_{f}^{T}}_{I3} + \frac{1}{N}\underbrace{V_{p}W_{f}^{T}}_{I4} + \frac{1}{N}\underbrace{W_{p}W_{f}^{T}}_{I4} + \frac{1}{N}\underbrace{W_{p}W_{p}}_{I4} + \frac{1}{N}\underbrace{W_{p$$

Por ser u_k independentemente de w_j para todo k, j (ver equação (B 12.8)), o termo *I*2 é $O(\varsigma)$. O mesmo acontece para os termos *I*3 e *I*4, por propriedade de ruído branco de w_j e v_j [ver definição B 7.2], portanto a equação (B 12.8) é expressa por:

$$\frac{1}{N}Y_pW_f^T = \frac{1}{N}\Gamma_i \underbrace{X_pW_f^T}_{I1} + O(\varsigma)$$
(B 12.19)

Afirmamos que o termo $I1 = O(\varsigma)$ para $N \to \infty$, em efeito,

Como a sequência de estados passados é dada por $X_p = \begin{bmatrix} x_1 & x_2 & \dots & x_{N-1} & x_N \end{bmatrix}$, então da equação (2.48) se obtém a equação dos estados em função do estado inicial x_0 , isto é:

$$x_{k} = A^{k} x_{0} + \sum_{i=1}^{k} A^{i-1} B u_{s-1} + \sum_{i=1}^{k} A^{i-1} w_{s-1}$$
(B 12.19)

Desta forma a equação (B 12.19) caracteriza os elementos dos estados passados X_p .

Seja a sequência do processo de ruído é $W_j = \begin{bmatrix} w_j & w_{j+1} & \dots & w_{N+j-2} & w_{N+j-1} \end{bmatrix}$ onde cada elemento desta sequência representa o vetor coluna da matriz $W_{i+1,i,N}$ definida por:

$$W_{f} = W_{i+1,i,N} = \begin{bmatrix} w_{i+1} & w_{i+2} & \cdots & w_{i+s} & \cdots & w_{i+N} \\ w_{i+2} & w_{i+3} & \cdots & w_{i+s+2} & \cdots & w_{i+1+N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{2i} & w_{2i+1} & \cdots & w_{2i+1+s} & \cdots & w_{2i+N-1} \end{bmatrix}_{ixN}$$

Agora computando:

$$E[Xp W_f^T] = E[[x_1 \quad x_2 \quad \dots \quad x_{N-1} \quad x_N] \cdot W_f^T]$$
$$E[Xp W_j^T] = E[\left(A^k x_0 + \sum_{i=1}^k A^{i-1} B u_{k-1} + \sum_{i=1}^k A^{i-1} w_{k-1}\right) w_l^T]$$

Onde k = 1,.., N e l = i + 1,..,2i + N - 1

$$E[Xp W_j^T] = A^k E[x_0 w_l^T] + \sum_{i=1}^k A^{i-1} BE[u_{k-1} w_l^T] + \sum_{i=1}^k A^{i-1} E[w_{k-1} w_l^T]$$
(B 12.20)

Pela hipótese do problema, a equação (B 12.20) é expressada como:

$$\frac{1}{N} E[Xp W_j^T] = O(\xi)$$

Pelo exposto acima, então a equação (B 12.19) resulta:

$$\frac{1}{N}Y_pW_f^T = O(\varsigma)$$

Prova-se agora a parte ii) $\frac{1}{\sqrt{N}}Q_{I}W_{f}^{T} = O(\varsigma)$

Da equação (B 12.1) $U_f = U_{i+1,i,N}$ é expresso como:

$$U_f = U_{i+1,i,N} = R_{11}Q_1$$

e por ser linearmente independe do ruído, resulta:

$$\frac{1}{N}R_{11}Q_1W_f^T = \frac{1}{N}U_fW_f^T = O(\varsigma)$$
(B 12.21)

Por ser a entrada u_k persistentemente excitante, então a matriz R_{11} é invertivel, portanto a equação (B 12.21) é expressa como:

$$\frac{1}{\sqrt{N}}Q_{\mathrm{I}}W_{f}^{\mathrm{T}}=O(\varsigma)\,.$$

Por último, prova-se agora a parte iii) $\frac{1}{\sqrt{N}}Q_2W_f^T = O(\varsigma)$

Isto foi demonstrado na equação (B 12.14).

Agora prova-se a equação (2.65)

Da equação (B 12.1) $Y_p = Y_{1,i,N}$ é expresso como:

$$Y_p = Y_{1,i,N} = R_{31}Q_1 + R_{32}Q_2 + R_{33}Q_3$$
(B 12.22)

Multiplica-se a equação (B 12.22) por $\frac{1}{N}W_f^T = \frac{1}{N}W_{i+1,i,N}^T$ de ambos lados, resulta:

$$Y_{p} \frac{1}{N} W_{f}^{T} = R_{31} Q_{1} \frac{1}{N} W_{f}^{T} + R_{32} Q_{2} \frac{1}{N} W_{f}^{T} + R_{33} Q_{3} \frac{1}{N} W_{f}^{T}$$

n	11
v	u

$$\frac{1}{N}Y_{p}W_{f}^{T} = \frac{1}{\sqrt{N}}R_{31}\frac{1}{\sqrt{N}}Q_{1}W_{f}^{T} + \frac{1}{\sqrt{N}}R_{32}\frac{1}{\sqrt{N}}Q_{2}W_{f}^{T} + \frac{1}{\sqrt{N}}R_{33}\frac{1}{\sqrt{N}}Q_{3}W_{f}^{T}$$
(B 12.23)

Aplicando i), ii) e iii) na equação (B 12.23) se obtém:

$$O(\varsigma) = \frac{1}{\sqrt{N}} R_{33} \frac{1}{\sqrt{N}} Q_3 W_f^T$$
(B 12.23)

Por ser a entrada u_k persistentemente excitante, então a matriz R_{33} é invertivel, portanto a equação (B 12.23) é expressa como:

$$\frac{1}{\sqrt{N}}Q_3W_f^T = O(\varsigma). \tag{B 12.24}$$

Da mesma forma obtemos:

$$\frac{1}{\sqrt{N}}Q_3V_f^T = O(\varsigma). \tag{B 12.25}$$

Multiplica-se a equação (B 12.4) por $\frac{1}{\sqrt{N}} Q_3^T$ de ambos lados, resulta

$$\frac{1}{\sqrt{N}}Y_{f}Q_{3}^{T} = \frac{1}{\sqrt{N}}\Gamma_{i}X_{f}Q_{3}^{T} + \frac{1}{\sqrt{N}}H_{i}\underbrace{U_{f}Q_{3}^{T}}_{I_{1}} + \frac{1}{\sqrt{N}}\Phi_{i}\underbrace{W_{f}Q_{3}^{T}}_{I_{2}} + \frac{1}{\sqrt{N}}\underbrace{V_{f}Q_{3}^{T}}_{I_{3}}$$
(B 12.26)

Onde os últimos ter termos, isto é I_1, I_2 e I_3 tendem a zero quando $N \rightarrow \infty$

Em efeito

Para I_1

.

Multiplica-se a equação (B 12.6) por Q_3^T de ambos lados, resulta:

$$U_{i+1,i,N}Q_3^T = R_{11}Q_1Q_3^T = 0 (B 12.27)$$

Para os casos I_2 e I_3 é demonstrado pelas equações (B 12.24) e (B 12.25) respectivamente, portanto a equação (B 12.26) é expressa por:

$$\frac{1}{\sqrt{N}}Y_f Q_3^T = \frac{1}{\sqrt{N}}\Gamma_i X_f Q_3^T + O(\varsigma)$$

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}Y_f(Q_3)^T = \lim_{N\to\infty}\frac{1}{\sqrt{N}}\Gamma_i X_f(Q_3)^T.$$

Agora prova-se a equação (2.66):

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_p(Q_1^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_p(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{21}) \right)$$

Seja k = 1 na equação (B 11.4) e multiplica-se esta por $\frac{1}{\sqrt{N}}Q_1^T$ de ambos lados, resulta:

$$\frac{1}{\sqrt{N}}Y_{1,i,N}Q_1^T = \frac{1}{\sqrt{N}}\Gamma_i X_{1,N}Q_1^T + \frac{1}{\sqrt{N}}H_i U_{1,i,N}Q_1^T + \frac{1}{\sqrt{N}}\Phi_i W_{1,i,N}Q_1^T + \frac{1}{\sqrt{N}}V_{1,i,N}Q_1^T$$
(B 12.28)

substituindo-se a equação (B 12.7) na equação (B 12.28) resulta

$$\frac{1}{\sqrt{N}}Y_{1,i,N}Q_1^T = \frac{1}{\sqrt{N}}\Gamma_i X_{1,N}Q_1^T + \frac{1}{\sqrt{N}}H_i R_{21} + \frac{1}{\sqrt{N}}\Phi_i W_{1,i,N}Q_1^T + \frac{1}{\sqrt{N}}V_{1,i,N}Q_1^T$$

ou

$$\frac{1}{\sqrt{N}}Y_pQ_1^T = \frac{1}{\sqrt{N}}\Gamma_iX_pQ_1^T + \frac{1}{\sqrt{N}}H_iR_{21} + \frac{1}{\sqrt{N}}\Phi_i\underbrace{W_pQ_1^T}_{I_1} + \frac{1}{\sqrt{N}}\underbrace{V_pQ_1^T}_{I_2}$$
(B 12.29)

Por ser u_k independentemente de w_j , v_j para todo k, j (ver equação (B 12.8)), o termo Il e I2 é $O(\varsigma)$.

Portanto a equação (B 12.18) é expressa como:

$$\frac{1}{\sqrt{N}}Y_{p}Q_{1}^{T} = \frac{1}{\sqrt{N}}\Gamma_{i}X_{p}Q_{1}^{T} + \frac{1}{\sqrt{N}}H_{i}R_{21} + O(\varsigma).$$

Ou em forma equivalente

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} Y_p(Q_1^T) = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_p(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{21}) \right)$$

Agora prova-se (2.67)

Seja k = 1 na equação (B 11.4) e multiplica-se esta por $\frac{1}{\sqrt{N}}Q_2^T$ de ambos lados, resulta:

$$\frac{1}{\sqrt{N}}Y_{1,i,N}Q_2^T = \frac{1}{\sqrt{N}}\Gamma_i X_{1,N}Q_2^T + \frac{1}{\sqrt{N}}H_i U_{1,i,N}Q_2^T + \frac{1}{\sqrt{N}}\Phi_i W_{1,i,N}Q_2^T + \frac{1}{\sqrt{N}}V_{1,i,N}Q_2^T$$
(B 12.28)

substituindo-se a equação (B 12.7) na equação (B 12.28) resulta

$$\frac{1}{\sqrt{N}}Y_{1,i,N}Q_2^T = \frac{1}{\sqrt{N}}\Gamma_i X_{1,N}Q_2^T + \frac{1}{\sqrt{N}}H_i R_{22} + \frac{1}{\sqrt{N}}\Phi_i W_{1,i,N}Q_2^T + \frac{1}{\sqrt{N}}V_{1,i,N}Q_2^T$$

ou

$$\frac{1}{\sqrt{N}}Y_pQ_2^T = \frac{1}{\sqrt{N}}\Gamma_i X_p Q_2^T + \frac{1}{\sqrt{N}}H_i R_{22} + \frac{1}{\sqrt{N}}\Phi_i \underbrace{W_p Q_2^T}_{J_1} + \frac{1}{\sqrt{N}}\underbrace{V_p Q_2^T}_{J_2}$$
(B 12.29)

Por ser u_k independentemente de w_j , v_j para todo k, j (ver equação (B 12.8)), o termo J1e J2 é $O(\varsigma)$.

Portanto a equação (B 12.18) é expressa como:

$$\frac{1}{\sqrt{N}}Y_{p}Q_{2}^{T} = \frac{1}{\sqrt{N}}\Gamma_{i}X_{p}Q_{2}^{T} + \frac{1}{\sqrt{N}}H_{i}R_{22} + O(\varsigma).$$

ou em forma equivalente

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}Y_p(Q_2^T) = \lim_{N\to\infty}\left(\frac{1}{\sqrt{N}}\Gamma_iX_p(Q_2^T) + \frac{1}{\sqrt{N}}H_i(R_{22})\right).$$

Agora se prova a equação (2.68)

Considerando k = i + 1 na equação (B 11.4) temos

$$Y_{i+1,i,N} = \Gamma_i X_{i+1,N} + H_i U_{i+1,i,N} + \Phi_i W_{i+1,i,N} + V_{i+1,i,N}$$
(B 12.30)

Multiplica-se a equação (B 12.30) por $\frac{1}{\sqrt{N}} Q_1^T$ de ambos lados, resulta

$$\frac{1}{\sqrt{N}}Y_{i+1,i,N}Q_{1}^{T} = \frac{1}{\sqrt{N}}\Gamma_{i}X_{i+1,N}Q_{1}^{T} + \frac{1}{\sqrt{N}}H_{i}U_{i+1,i,N}Q_{1}^{T} + \frac{1}{\sqrt{N}}\Phi_{i}W_{i+1,i,N}Q_{1}^{T} + \frac{1}{\sqrt{N}}V_{i+1,i,N}Q_{1}^{T}$$

$$+ \frac{1}{\sqrt{N}}V_{i+1,i,N}Q_{1}^{T} \qquad (B\ 12.31)$$

Por ser u_k independentemente de w_j , v_j para todo k, j (ver equação (B 12.8)), os dois últimos termos do lado direito na equação (B 12.31) é $O(\varsigma)$, portanto, temos:

$$\frac{1}{\sqrt{N}}Y_{i+1,i,N}Q_{1}^{T} = \frac{1}{\sqrt{N}}\Gamma_{i}X_{i+1,N}Q_{1}^{T} + \frac{1}{\sqrt{N}}H_{i}U_{i+1,i,N}Q_{1}^{T} + O(\varsigma)$$
(B 12.32)

Substituindo-se a equação (B 12.6) na equação (B 12.32), resulta:

$$\frac{1}{\sqrt{N}}Y_{i+1,i,N}Q_1^T = \frac{1}{\sqrt{N}}\Gamma_i X_{i+1,N}Q_1^T + \frac{1}{\sqrt{N}}H_i R_{11} + O(\varsigma)$$
(B 12.33)

ou em forma equivalente

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}Y_f(Q_1^T) = \lim_{N\to\infty}\left(\frac{1}{\sqrt{N}}\Gamma_iX_f(Q_1^T) + \frac{1}{\sqrt{N}}H_i(R_{11})\right).$$

Portanto o teorema 3 esta demonstrado.

B.13 Prova da equação (2.59)

Da equação (B 12.1) Y_f é expresso por:

$$Y_f = R_{41}Q_1 + R_{42}Q_2 + R_{43}Q_3 + R_{44}Q_4$$
(B 13.1)

Multiplica-se a equação (B 13.1) por Q_2^T de ambos os lados, resulta:

$$Y_f Q_2^T = R_{42} (B \ 13.2)$$

Substituindo-se a equação (B 13.2) na equação (B 2.64), se obtém:

$$\frac{1}{\sqrt{N}}R_{42} = \frac{1}{\sqrt{N}}\Gamma_i X_f (Q_2)^T + O(\varsigma)$$
(B 13.3)

Multiplica-se a equação (B 13.1) por Q_3^T de ambos lados, resulta:

$$Y_f Q_3^T = R_{43}$$
 (B 13.4)

Substituindo-se a equação (B 13.2) na equação (B 2.65), se obtém:

$$\frac{1}{\sqrt{N}}R_{43} = \frac{1}{\sqrt{N}}\Gamma_i X_f (Q_3)^T + O(\varsigma)$$
(B 13.5)

Expressando em forma matricial as equações (B 13.3) e (B 13.5), resulta:

$$\frac{1}{\sqrt{N}} \begin{bmatrix} R_{42} & R_{43} \end{bmatrix} = \frac{1}{\sqrt{N}} \Gamma_i X_f \begin{bmatrix} Q_2^T & Q_3^T \end{bmatrix} + O(\varsigma)$$

Ou em forma equivalente

$$\lim_{N\to\infty}\frac{1}{\sqrt{N}}\begin{bmatrix}R_{42} & R_{43}\end{bmatrix} = \lim_{N\to\infty}\frac{1}{\sqrt{N}}\Gamma_i X_{i,N}\begin{bmatrix}Q_2^T & Q_3^T\end{bmatrix}.$$

B.14 Prova da equação (2.61)

Da equação (B 12.1) Y_f é expresso por:

$$Y_p = R_{31}Q_1 + R_{32}Q_2 + R_{33}Q_3 \tag{B 14.1}$$

Multiplica-se a equação (B 14.1) por Q_1^T de ambos lados, resulta:

$$Y_p Q_1^T = R_{31}$$
 (B 14.2)

Substituindo-se a equação (B 14.2) na equação (B 2.66), se obtém:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{31} = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_p(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{21}) \right)$$
(B 14.3)

Multiplica-se a equação (B 14.1) por Q_2^T de ambos lados, resulta:

$$Y_p Q_2^T = R_{32} (B 14.4)$$

Substituindo-se a equação (B 14.4) na equação (B 2.67), se obtém:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{32} = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_p(Q_2^T) + \frac{1}{\sqrt{N}} H_i(R_{22}) \right)$$
(B 14.5)

Multiplica-se a equação (B 13.1) por Q_1^T de ambos lados, resulta:

$$Y_f Q_1^T = R_{41} (B \ 14.6)$$

Substituindo-se a equação (B 14.6) na equação (B 2.68), se obtém:

$$\lim_{N \to \infty} \frac{1}{\sqrt{N}} R_{41} = \lim_{N \to \infty} \left(\frac{1}{\sqrt{N}} \Gamma_i X_f(Q_1^T) + \frac{1}{\sqrt{N}} H_i(R_{11}) \right)$$
(B 14.7)

Expressando em forma matricial as equações (B 14.3), (B 14.5) e (B 14.7), resulta:

$$\frac{1}{\sqrt{N}} \begin{bmatrix} R_{31} & R_{32} & R_{41} \end{bmatrix} = \Gamma_i \frac{1}{\sqrt{N}} \begin{bmatrix} X_p(Q_1^T) & X_p(Q_2^T) & X_f(Q_1^T) \end{bmatrix} + H_i \frac{1}{\sqrt{N}} \begin{bmatrix} R_{21} & R_{22} & R_{11} \end{bmatrix} + O(\varsigma) \,.$$

APÊNDICE C

C.1 Notação usada no capítulo 3

Matrizes em blocos de Hankel

Entradas passadas	i	$ \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} $	u_1 u_2	<i>u</i> ₂ <i>u</i> ₃	····	u_{N-1}	m			
L	linhas		•••	•••	•••	•••		$m \cdot i$		
II.	def	<i>u</i> _{<i>i</i>-1}	<i>u</i> _i	u_{i+1}		u_{i+N-2}	m	filas	$2 \cdot m \cdot m$	i
$U_{0 2i-1}$	=	<i>u</i> _{<i>i</i>}	u_{i+1}	u_{i+2}		u_{i+N-1}	m		filas	
Entradas futuras	i	u_{i+1}	u_{i+2}	u_{i+3}		u_{i+N}	m	m∙i		
	linhas		•••	•••	•••			filas		
		u_{2i-1}	u_{2i}	u_{2i+1}	•••	u_{2i+N-2}) m			

Os elementos da matriz $U_{0|2i-1}$ são vetores, isto é, $u_0 \in \Re^m$, $u_1 \in \Re^m$,....., $u_{2i+j-2} \in \Re^m$. Isto quer dizer que cada linha da matriz $U_{0|2i-1}$ tem *m* filas e, como ela contém 2.*i* linhas, a matriz $U_{0|2i-1}$ possui 2.*m.i* filas.

- O número de linhas "i" é um índice definido pelo usuário e deve ser um valor alto, isto é, ele deve ser pelo menos tão grande quanto a ordem máxima do sistema que se deseje identificar (i > n).
- O número de colunas N é tipicamente dado por: N = s 2i + 1, onde "s" é o número de amostras coletadas dos dados de entrada.
- A matriz $U_{0|2i-1}$ pode ser escrita como:

$$U_{0|2i-1} \stackrel{def}{=} \begin{pmatrix} u_0 & u_1 & u_2 & \dots & u_{j-1} \\ u_1 & u_2 & u_3 & \dots & u_j \\ \dots & \dots & \dots & \dots & \dots \\ u_{i-1} & u_i & u_{i+1} & \dots & u_{i+j-2} \\ \hline u_i & u_{i+1} & u_{i+2} & \dots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \dots & u_{i+j} \\ \dots & \dots & \dots & \dots & \dots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \dots & u_{2i+j-2} \end{pmatrix}^{def} \left(\begin{matrix} U_{0|i-1} \\ U_{i|2i-1} \end{matrix} \right)^{def} \left(\begin{matrix} U_p \\ U_f \end{matrix} \right)$$

ou então como:

As matrizes em blocos de Hankel das saídas $Y_{0|2i-1}, Y_p, Y_f, Y_p^+, Y_f^-$ são definidas de forma similar.

Tempo	U	Y	W	Χ
Passado	$U_{0 i-1} = U_{p}$	$Y_{0 i-1} = Y_p$	$W_p = \begin{pmatrix} U_p \\ Y_p \end{pmatrix}$	$X_p^d = X_0^d$
	$U_{0 i} = U_p^+$	$Y_{0 i} = Y_p^+$	$W_p^+ = \begin{pmatrix} U_p^+ \\ Y_p^+ \end{pmatrix}$	
Futuro	$U_{i 2i-1} = U_f$	$Y_{i 2i-1} = Y_f$	$W_f = \begin{pmatrix} U_f \\ Y_f \end{pmatrix}$	$X_f^d = X_i^d$
	$U_{i+1 2i-1} = U_f^-$	$Y_{i+1 2i-1} = Y_f^-$	$W_f^+ = \begin{pmatrix} U_f^+ \\ Y_f^+ \end{pmatrix}$	

Notação usada:

A matriz estendida de controlabilidade Δ_i^d , onde *i* (*i* > *n*) denota o número de colunas de blocos, é definida como:

$$\Delta_{i}^{d} = \underbrace{\begin{bmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{bmatrix}}_{mxi \quad colunas} |n \quad linhas \ , \in \Re^{nxmi}, A \in \Re^{nxm}, B \in \Re^{nxm}$$

C. 2 Idéia do algoritmo N4SID caso determinístico

i) A sequência de estados X_f^d está contida no espaço linha de W_f

A seqüência de estados X_f^d esta contida no espaço linha de W_p iv)

v) Aplicando-se o teorema da dimensão de Grassmann

$$\dim(espaço linha[W_p] \cap espaço linha[W_f]) = n$$

Prova

Primeiro provaremos i):

Da equação (B 3.10), se obtém:

$$Y_f = \Gamma_i X_f + H_i U_f \tag{C 2.1}$$

Isolando a seqüência de estado na equação (C 2.1) :

$$X_{f} = \Gamma_{i}^{*} Y_{f} - \Gamma_{i}^{*} H_{i} U_{f}$$
 (C 2.2)

A equação (C 2.2) demonstra que o vetor de estados é uma combinação linear do espaço linha da matriz Y_f e U_f , portanto X_f está contido no espaço linha W_f :

$$X_f = \Gamma_i^* Y_f - \Gamma_i^* H_i U_f \in span(W_f).$$
(C 2.3)

Agora se prova ii)

*

Da equação (B 3.10), se obtém:

$$Y_p = \Gamma_i X_p + H_i U_p \tag{C 2.4}$$

Isolando a seqüência de estados na equação (C 2.4) :

$$X_{p} = \Gamma_{i}^{*} Y_{p} - \Gamma_{i}^{*} H_{i} U_{p}$$
(C 2.5)

Associado ao sistema (3.15), o vetor de estados para o instante de tempo k+i é expresso como:

$$x(k+i) = A^{k}x(i) + \begin{bmatrix} A^{k-1}B & A^{k-2}B & \cdots B \end{bmatrix} \begin{bmatrix} u(i) \\ u(i+1) \\ \vdots \\ u(i+k-1) \end{bmatrix}$$

Então variando o valor do índice k se obtém a seqüência de estado:

$$X_p = A^k X_p + \Delta_k U_p \tag{C 2.6}$$

Substituindo a equação (C 2.5) na equação (C 2.6), resulta:

$$X_{p} = A^{k} \left(\Gamma_{i}^{*} Y_{p} - \Gamma_{i}^{*} H_{i} U_{p} \right) + \Delta_{k} U_{p}$$

$$X_{p} = \left(\Delta_{k} - A^{k} \Gamma_{i}^{*} H_{i} \right) U_{p} + \left(A^{k} \Gamma_{i}^{*} \right) Y_{p}$$
(C 2.7)

A equação (C 2.7) demonstra que o vetor de estados é uma combinação linear do espaço linha da matriz Y_p e U_p , portanto X_f está contido no espaço linha W_p

$$X_{p} = \left[\Delta_{k} - A^{k} \Gamma_{i}^{*} H_{i} \quad A^{k} \Gamma_{i}^{*}\right] W_{p} \quad \in span(W_{p}).$$
(C 2.8)

Por último provaremos iii):

Então provemos primeiro $posto(X_f) = n$

Por definição da sequência de estado temos

$$posto(X_f) \le n$$
 (C 2.9)

Por hipótese, na identificação de sistemas: $posto(\Delta_k) = n = posto(\Gamma_k) \operatorname{com} k > n$ e

pela equação (C 2.6) temos:

$$posto(X_f) = posto(A^k X_p + \Delta_k U_p) \ge n$$
(C 2.10)

por ser o sinal de entrada u_k persistentemente excitante.

Portanto das equações (C 2.10) e (C 2.9), resulta:

$$n \le posto(X_f) \le n$$
 (C 2.11)

Da equação (C 2.11) concluímos: $posto(X_f) = n$.

Da equação (C 2.4) e da matriz U_p , obtem-se:

$$Y_{p} = \Gamma_{i}X_{p} + H_{i}U_{p}$$

$$U_{p} = U_{p}$$

$$\Longrightarrow \begin{bmatrix} Y_{p} \\ U_{p} \end{bmatrix} = \begin{bmatrix} \Gamma_{i} & H_{i} \\ 0 & I_{mi} \end{bmatrix} \begin{bmatrix} X_{p} \\ U_{p} \end{bmatrix}$$
(C 2.12)

Da equação (C 2.12), se obtém:

$$posto\begin{bmatrix} Y_p \\ U_p \end{bmatrix} = posto\begin{bmatrix} X_p \\ U_p \end{bmatrix} = mi + n$$
(C 2.13)

Da mesma forma obtemos:

$$posto\begin{bmatrix} Y_f\\ U_f \end{bmatrix} = mi + n \tag{C 2.14}$$

$$posto\begin{bmatrix} Y_p \\ Y_f \\ U_p \\ U_f \end{bmatrix} = 2mi + n$$
(C 2.15)

Agora aplicando o teorema de Grassmann:

$$\dim(espaço \ fila[W_p] \cap espaço \ fila[W_f]) = posto[W_p] + posto[W_f] - posto[W_f]$$
$$= (mi+n) + (mi+n) - (2mi+n) = n$$

 $\dim(espaço fila[W_p] \cap espaço fila[W_f]) = n.$

C.3 Prova da equação (3.17)

Na equação (C 2.8), denotando $L_p = \left[\Delta_k - A^k \Gamma_i^* H_i \quad A^k \Gamma_i^* \right]$, resulta: $X_p = L_p W_p$.

APÊNDICE D

D.1 Matrizes utilizadas no capítulo 6

Além da notação dos capítulos anteriores, apresentaremos as seguintes matrizes: Matriz de observabilidade e de controlabilidade do controlador, dadas por:

$$\Gamma_{i}^{c} = \begin{bmatrix} C_{c} \\ C_{c}A_{c} \\ \vdots \\ C_{c}A_{c}^{i-1} \end{bmatrix} \in \Re^{mixn_{c}} \qquad e \qquad \Delta_{i}^{c} = \begin{bmatrix} A_{c}^{i-1}B_{c} & \cdots & A_{c}B_{c} & B_{c} \end{bmatrix} \in \Re^{n_{c}xli} \qquad D.1.1$$

A matriz em blocos triangular inferior de Toeplitz do controlador é dada por:

$$H_{i}^{c} \stackrel{def}{=} \begin{bmatrix} D_{c} & 0 & 0 & \dots & 0 \\ C_{c}B_{c} & D_{c} & 0 & \dots & 0 \\ C_{c}A_{c}B_{c} & C_{c}B_{c} & D_{c} & \dots & 0 \\ \dots & \dots & \dots & \ddots & 0 \\ C_{c}A_{c}^{i-2}B_{c} & C_{c}A_{c}^{i-3}B_{c} & C_{c}A_{c}^{i-4}B_{c} & \dots & D_{c} \end{bmatrix} \in \Re^{mixli}$$
D.1.2

A matriz em blocos triangular inferior de Toeplitz T_i é definida como:

$$T_i \stackrel{def}{=} \left(I_{li} + H_i H_i^c \right)^{-1} \in \Re^{lixli}$$
D.1.3

Pela hipótese, na identificação em malha fechada, de que o sistema seja bem posto, temos que existe a inversa da matriz T_i .

Da mesma forma que foram definidas as matrizes em blocos de entrada $U_{0|2i-1}$ e saída $Y_{0|2i-1}$, podemos definir a matriz em blocos de referência $R_{0|2i-1}$.

Para simplificar as futuras demonstrações, definem-se matrizes auxiliares $N \in M$, estas matrizes não são matrizes em blocos de Hankel,

$$N_{p|q} \stackrel{def}{=} \left[U_{0|2i-1} + H_{2i}^{c} Y_{0|2i-1} \right]$$
D.1.4

A matriz $N_{p|q}$ é formada por todas as linhas compreendidas entre a linha p até a linha q.

$$M_{p|q} \stackrel{def}{=} U_{p|q} + H_{q-p+1}^{c} Y_{p|q}$$
D.1.5
onde $0 \le p \le q \le 2i - 1$.

Para gerar as matrizes *N* ou *M* precisamos dos primeiros parâmetros de Markov, pois os dados de entrada e saída são conhecidos.

Da equação (D.1.4) pode-se deduzir o seguinte:

$$N_{p|q} \stackrel{def}{=} U_{p|q} + H_{q-p+1}^{c} Y_{p|q} + \Gamma_{q-p+1}^{c} \Delta_{p}^{c} Y_{0|p-1}$$
D.1.6

$$= M_{p|q} + \Gamma_{q-p+1}^{c} \Delta_{p}^{c} Y_{0|p-1}$$
 D.1.7

A demonstração da equação (D.1.6) não é tão simples como menciona o artigo [VAN OVERSCHEE; DE MOOR, 1997]. Apresentamos uma idéia desta demonstração sem o rigor matemático.

Seja:

$$H = \begin{bmatrix} D & 0 & 0 & 0 & 0 \\ A^{0} & D & 0 & 0 & 0 \\ A^{1} & A^{0} & D & 0 & 0 \\ A^{2} & A^{1} & A^{0} & D & 0 \\ A^{3} & A^{2} & A^{1} & A^{0} & D \end{bmatrix}_{5x5} e Y = \begin{bmatrix} Y_{0} & Y_{1} & Y_{2} & Y_{3} & Y_{4} & Y_{5} & Y_{6} \\ Y_{1} & Y_{2} & Y_{3} & Y_{4} & Y_{5} & Y_{6} & Y_{7} \\ Y_{2} & Y_{3} & Y_{4} & Y_{5} & Y_{6} & Y_{7} \\ Y_{2} & Y_{3} & Y_{4} & Y_{5} & Y_{6} & Y_{7} \\ Y_{3} & Y_{4} & Y_{5} & Y_{6} & Y_{7} & Y_{8} \\ Y_{3} & Y_{4} & Y_{5} & Y_{6} & Y_{7} & Y_{8} \\ Y_{4} & Y_{5} & Y_{6} & Y_{7} & Y_{8} & Y_{9} \\ Y_{4} & Y_{5} & Y_{6} & Y_{7} & Y_{8} & Y_{9} \\ Y_{4} & Y_{5} & Y_{6} & Y_{7} & Y_{8} & Y_{9} \\ \end{bmatrix}_{5x7} = \begin{bmatrix} Y_{0|p-1} = Y_{0|2} \\ Y_{0|p-1} \\ Y_{0|p-1} \\ Y_{0|p-1} \\ Y_{0|p-1} \\ Y_{0|p-1} \\$$

Então o produto de HY é dado por:

$$HY = \begin{bmatrix} Dy_0 & Dy_1 \\ A^0 y_0 + Dy_1 & A^0 y_1 + Dy_2 \\ A^1 y_0 + A^0 y_1 + Dy_2 & A^1 y_1 + A^0 y_2 + Dy_3 \\ A^2 y_0 + A^1 y_1 + A^0 y_2 + Dy_3 & A^2 y_1 + A^1 y_2 + A^0 y_3 + Dy_4 \\ A^3 y_0 + A^2 y_1 + A^1 y_2 + A^0 y_3 + Dy_4 & A^3 y_1 + A^2 y_2 + A^1 y_3 + A^0 y_4 + Dy_5 \end{bmatrix}$$

$$\begin{array}{cccc} Dy_5 & Dy_6 \\ A^0y_5 + Dy_6 & A^0y_6 + Dy_7 \\ A^1y_5 + A^0y_6 + Dy_7 & A^1y_6 + A^0y_7 + Dy_8 \\ A^2y_5 + A^1y_6 + A^0y_7 + Dy_8 & A^2y_6 + A^1y_7 + A^0y_8 + Dy_9 \\ A^3y_5 + A^2y_6 + A^1y_7 + A^0y_8 + Dy_9 & A^3y_6 + A^2y_7 + A^1y_8 + A^0y_9 + Dy_{10} \end{bmatrix}_{5x7} D.1.8$$

Considerando a sub-matriz de *HY* formada por todas as linhas compreendidas entre a linha *p* até a linha *q* (incluindo os extremos), então para p = 3 e q = 5 o produto *HY* é expresso como:

$$\begin{bmatrix} HY \end{bmatrix}_{p|q} = \begin{bmatrix} A^{1}y_{0} + A^{0}y_{1} + Dy_{2} & A^{1}y_{1} + A^{0}y_{2} + Dy_{3} & A^{2}y_{1} + A^{1}y_{2} + A^{0}y_{3} + Dy_{4} & A^{2}y_{1} + A^{1}y_{2} + A^{0}y_{3} + Dy_{4} & A^{2}y_{1} + A^{1}y_{2} + A^{0}y_{3} + Dy_{4} & A^{3}y_{1} + A^{2}y_{2} + A^{1}y_{3} + A^{0}y_{4} + Dy_{5} & A^{1}y_{5} + A^{0}y_{6} + Dy_{7} & A^{1}y_{6} + A^{0}y_{7} + Dy_{8} & A^{2}y_{6} + A^{1}y_{7} + A^{0}y_{8} + Dy_{9} & A^{2}y_{6} + A^{1}y_{7} + A^{0}y_{8} + Dy_{9} & A^{3}y_{6} + A^{2}y_{7} + A^{1}y_{8} + A^{0}y_{9} + Dy_{10} \end{bmatrix}_{3x6} D.1.9$$

A equação (D.1.9) pode ser expressa como a soma de duas matrizes:

$$HY = \begin{bmatrix} A^{1}y_{0} + A^{0}y_{1} & A^{1}y_{1} + A^{0}y_{2} \cdots & A^{1}y_{5} + A^{0}y_{6} & A^{1}y_{6} + A^{0}y_{7} \\ A^{2}y_{0} + A^{1}y_{1} & A^{2}y_{1} + A^{1}y_{2} \cdots & A^{2}y_{5} + A^{1}y_{6} & A^{2}y_{6} + A^{1}y_{7} \\ A^{3}y_{0} + A^{2}y_{1} & A^{3}y_{1} + A^{2}y_{2} \cdots & A^{3}y_{5} + A^{2}y_{6} & A^{3}y_{6} + A^{2}y_{7} \end{bmatrix} +$$

$$\begin{bmatrix} Dy_2 & Dy_3 & \cdots \\ A^0y_2 + Dy_3 & A^0y_3 + Dy_4 & \cdots \\ A^1y_2 + A^0y_3 + Dy_4 & A^1y_3 + A^0y_4 + Dy_5 & \cdots \end{bmatrix}$$

$$\begin{array}{ccc} Dy_{7} & Dy_{8} \\ A^{0}y_{7} + Dy_{8} & A^{0}y_{8} + Dy_{9} \\ A^{1}y_{7} + A^{0}y_{8} + Dy_{9} & A^{1}y_{8} + A^{0}y_{9} + Dy_{10} \end{array} \right] \text{D.1.10}$$

A equação (D.1.10) pode ser expressa novamente como:

$$\begin{bmatrix} HY \end{bmatrix}_{p|q} = \begin{bmatrix} A^{1} & A^{0} \\ A^{2} & A^{1} \\ A^{3} & A^{2} \end{bmatrix} \begin{bmatrix} y_{0} & y_{1} & y_{2} & \cdots & y_{6} \\ y_{1} & y_{2} & y_{3} & \cdots & y_{7} \end{bmatrix} + \begin{bmatrix} D & 0 & 0 \\ A^{0} & D & 0 \\ A^{1} & A^{0} & D \end{bmatrix} \begin{bmatrix} y_{2} & y_{3} & \cdots & y_{7} & y_{8} \\ y_{3} & y_{4} & \cdots & y_{8} & y_{9} \\ y_{4} & y_{5} & \cdots & y_{9} & y_{10} \end{bmatrix}$$

Esta última equação pode ser expressa por:

$$\left[H^{c}Y\right]_{p|q} = \Gamma^{c}{}_{q-p+1}\Delta^{c}{}_{p}Y_{0|p-1} + H^{c}{}_{q-p+1}Y_{p|q}$$
D.1.11

Para facilitar as contas e sem perda de generalidade, foram adotadas as seguintes convenções:

Para os elementos da matriz H^c levar em consideração que: $A^r = C_c A_c^r B_c$ para todo r = 0,1,2,...

Para os elementos da matriz Γ^c levar em consideração que: $A^r = C_c A_c^r$ para todo r = 0,1,2,...

Para os elementos da matriz Δ^c levar em consideração que: $A^r = A_c^r B_c$ para todo r = 0,1,2,...

Desta forma, pode-se obter a matriz:

1.0

$$N_{p|q} = U_{p|q} + H_{q-p+1}^{c} Y_{p|q} + \Gamma_{q-p+1}^{c} \Delta_{p}^{c} Y_{0|p-1}.$$

A matriz $M_{i+1|2i-1}$ não é igual à matriz $M_{i|2i-1}$, a diferença é que na primeira matriz faltam as primeiras *i* linhas em blocos (denotada por $\overline{M}_{i|2i-1}$).

A matriz $M_{i+1|2i-1}$ pode ser expressa como:

$$M_{i+1|2i-1} = \overline{M}_{i|2i-1} - \Gamma_{i-1}^{c} B_{c} Y_{i|i}$$
D.1.12

A sequência de estados é definida como:

$$X_{i} = \begin{bmatrix} x_{i} & x_{i+1} & \cdots & x_{i+j-1} \end{bmatrix}, \quad X_{i}^{c} = \begin{bmatrix} x_{i}^{c} & x_{i+1}^{c} & \cdots & x_{i+j-1}^{c} \end{bmatrix}, \quad X_{i}^{d} = \begin{bmatrix} x_{i}^{d} & x_{i+1}^{d} & \cdots & x_{i+j-1}^{d} \end{bmatrix}$$
e

 $X_i^s = \begin{bmatrix} x_i^s & x_{i+1}^s & \cdots & x_{i+j-1}^s \end{bmatrix}$, onde os índices *c*, *d* e *s* são usados para indicar os estados do controlador, estados determinísticos e estados estocásticos, respectivamente.

Para a análise teórica se supõe que $j \rightarrow \infty$. Assume-se que sejam são ergódigos, então se pode substituir as médias pela esperança matemática [VAN OVERSCHEE; DE MOOR, 1996].

$$Ej[\bullet] \stackrel{def}{=} \lim_{j \to \infty} \frac{1}{j} [\bullet]$$

Com esta definição de *Ej*, pode-se definir as matrizes de correlação como:

$$Ej\left[\binom{N_{0|2i-1}}{X_{0}}\left(N_{0|2i-1}^{T} \quad X_{0}^{T}\right)\right]^{def}\left(\binom{R_{nn}}{S_{nn}} \quad S_{nn}^{T}\right)$$
D.1.13

$$Ej\left[\begin{pmatrix}Y_{0|2i-1}\\Y_{i|2i-1}^{s}\end{pmatrix}\left((Y_{0|2i-1}^{s})^{T} \quad (Y_{i|2i-1}^{s})^{T}\right)\right] \stackrel{def}{=} \begin{pmatrix}L_{i} \quad (\Delta_{i}^{G})^{T}\Gamma_{i}^{T}\\\Gamma_{i}\Delta_{i}^{G} \quad L_{i}\end{pmatrix}$$
D.1.14

$$Ej\left[\begin{pmatrix} X_0^d \\ X_0^s \end{pmatrix} ((X_0^d)^T \quad (X_0^s)^T)\right] \stackrel{def}{=} \begin{pmatrix} \Sigma_{dd} \quad \Sigma_{sd}^T \\ \Sigma_{sd} \quad \Sigma_{ss} \end{pmatrix}$$
D.1.15

onde

$$\Sigma_{ss} \stackrel{\text{def}}{=} E\left[x_k^s(x_k^s)^T\right], \ G \stackrel{\text{def}}{=} E\left[x_{k+1}^s(y_k^s)^T\right], \ \Lambda_0 \stackrel{\text{def}}{=} E\left[y_k^s(y_k^s)^T\right] e \ \Delta_i^G \stackrel{\text{def}}{=} \left[A^{i-1}G \cdots AG \quad G\right]$$

Para maiores informações ver [VAN OVERSCHEE; BART DE MOOR, 1996].

D.2 Prova da equação 6.6

Para provar a equação (6.6):

$$Y_{i|2i-1} = T \Big[\Gamma_i A^i X_0^d + Y_{i|2i-1}^s + \Gamma_i \Delta_i^d U_{0|i-1} + H_i^d M_{i|2i-1} \Big],$$
 D.2.1

primeiro considere a equação (6.5), a qual é dada por:

$$Y_{i|2i-1} = \Gamma_i \left[A^i X_0^d + \Delta_i^d U_{0|i-1} \right] + H_i^d U_{i|2i-1} + Y_{i|2i-1}^s.$$
 D.2.2

Nesta equação o termo que contém $U_{i|2i-1}$ pode ser expresso como uma combinação linear da matriz $M_{i|2i-1}$ e da matriz $Y_{i|2i-1}$, para isto considera-se a equação (D.1.5), para o caso particular de p = i e q = 2i - 1, isto é:

$$M_{i|2i-1} = U_{i|2i-1} + H_i^c Y_{i|2i-1} \implies U_{i|2i-1} = M_{i|2i-1} - H_i^c Y_{i|2i-1}$$
D.2.3

$$\begin{split} Y_{i|2i-1} &= \Gamma_i \Big[A^i X_0^d + \Delta_i^d \ U_{0|i-1} \Big] + H_i^d \Big(M_{i|2i-1} - H_i^c Y_{i|2i-1} \Big) + Y_{i|2i-1}^s \\ Y_{i|2i-1} &= \Gamma_i \Big[A^i X_0^d + \Delta_i^d \ U_{0|i-1} \Big] + H_i^d \ M_{i|2i-1} - H_i^d \ H_i^c Y_{i|2i-1} + Y_{i|2i-1}^s \\ \Big(I + H_i^d \ H_i^c \Big) Y_{i|2i-1} &= \Gamma_i A^i X_0^d + \Gamma_i \Delta_i^d \ U_{0|i-1} + H_i^d \ M_{i|2i-1} + Y_{i|2i-1}^s \end{split}$$
 D.2.4

Fazendo:

$$T^{-1} = \left[I + H_i^d H_i^c\right], \text{ a equação (D.2.4) pode ser expressa como:}$$
$$Y_{i|2i-1} = T \left[\Gamma_i A^i X_0^d + Y_{i|2i-1}^s + \Gamma_i \Delta_i^d U_{0|i-1} + H_i^d M_{i|2i-1}\right].$$

D.3 Prova do teorema 1

Primeiro provaremos a equação (6.11). Para isto, considerar:

$$\begin{bmatrix} N_{0|2i-1} \\ Y_{0|i-1} \end{bmatrix} = \begin{bmatrix} N_{0|i-1} \\ N_{i|2i-1} \\ Y_{0|i-1} \end{bmatrix} = \begin{bmatrix} I_{mi} & 0 & H_i^c \\ 0 & I_{mi} & \Gamma_i^c \Delta_i^c \\ 0 & 0 & I_{li} \end{bmatrix} \begin{bmatrix} U_{0|i-1} \\ M_{i|2i-1} \\ Y_{0|i-1} \end{bmatrix}.$$
 D.3.1

Para obter as matrizes do segundo membro em (D.3.1) se usaram as equações (D.1.4) e (D.1.7), isto é , de (D.1.4):

$$\begin{split} N_{p|q} \stackrel{def}{=} & \left[U_{p|q} + H_{q-p+1}^{c} Y_{p|q} \right], \text{ fazendo } p = 0 \text{ e } q = i-1, \text{ obtém-se:} \\ N_{0|i-1} &= & \left[U_{0|i-1} + H_{i}^{c} Y_{0|i-1} \right] \\ \text{de (D.1.7)} \\ N_{p|q} &= M_{p|q} + \Gamma_{q-p+1}^{c} \Delta_{p}^{c} Y_{0|p-1}, \text{ fazendo } p = i \text{ e } q = 2i-1, \text{ obtém-se:} \end{split}$$

$$N_{i|2i-1} = M_{i|2i-1} + \Gamma_i^c \Delta_i^c Y_{0|i-1}$$

No segundo membro em (D.3.1) a matriz do lado esquerdo é uma matriz triangular superior e sua diagonal está inteiramente formada por uns, isto implica que tem posto completo. Então, por uma propriedade de posto de uma matriz:

$$posto\left(\begin{bmatrix}I_{mi} & 0 & H_{i}^{c}\\0 & I_{mi} & \Gamma_{i}^{c}\Delta_{i}^{c}\\0 & 0 & I_{li}\end{bmatrix}_{(2m+l)ix(2m+l)i} \begin{bmatrix}U_{0|i-1}\\M_{i|2i-1}\\Y_{0|i-1}\end{bmatrix}_{(2m+l)ixj}\right) = posto\left(\begin{bmatrix}U_{0|i-1}\\M_{i|2i-1}\\Y_{0|i-1}\end{bmatrix}\right)$$
$$posto\left(\begin{bmatrix}N_{0|2i-1}\\Y_{0|i-1}\end{bmatrix}\right) = posto\left(\begin{bmatrix}U_{0|i-1}\\M_{i|2i-1}\\Y_{0|i-1}\end{bmatrix}\right)$$

O posto de uma matriz é invariante na troca de linhas, isto prova a equação (6.11).

Agora provaremos a equação (6.12)

Em primeiro lugar provaremos duas propriedades que serão utilizadas na prova desta equação:

$$Ej\left[(\Gamma_{2i}X_0^d + Y_{0|2i-1}^s)N_{0|2i-1}^T\right] = \Gamma_{2i}S_{xn}$$
 D.3.2

$$Ej\left[Y_{0|2i-1}^{s}\left(X_{0}^{d}\right)^{T}\right]=\Gamma_{2i}\Sigma_{sd}$$
 D.3.3

Para provar a equação (D.3.2) precisa-se do sistema puramente estocástico, o qual é dado por:

$$\boldsymbol{x}^{\boldsymbol{s}_{k+1}} = A\boldsymbol{x}^{\boldsymbol{s}_k} + \boldsymbol{w}_k$$

$$\mathbf{y}^{\mathbf{s}_{k+1}} = C\mathbf{x}^{\mathbf{s}_k} + \mathbf{v}_k$$

Variando-se k se obtém a seguinte equação recursiva:

$$Y_{0|2i-1}^{s} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{2i-1} \end{bmatrix} \begin{bmatrix} x_{0}^{s} & \cdots & x_{j-1}^{s} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C & 0 & \cdots & 0 \\ \vdots \\ CA^{i-2} & CA^{i-3} & \cdots & 0 \end{bmatrix} W_{0|2i-1} + V_{0|2i-1}$$

$$Y_{0|2i-1}^{s} = \Gamma_{2i} X_{0}^{s} + H_{2i}^{w} W_{0|2i-1} + V_{0|2i-1}.$$
 D.3.4

Substituindo-se (D.3.4) em (D.3.2) resulta:

$$Ej\left[(\Gamma_{2i}X_{0}^{d} + Y_{0|2i-1}^{s})N_{0|2i-1}^{T}\right] = Ej\left[(\Gamma_{2i}X_{0}^{d} + \Gamma_{2i}X_{0}^{s} + H_{2i}^{w}W_{0|2i-1} + V_{0|2i-1})N_{0|2i-1}^{T}\right]$$
$$= Ej\left[(\Gamma_{2i}X_{0} + H_{2i}^{w}W_{0|2i-1} + V_{0|2i-1})N_{0|2i-1}^{T}\right]$$
D.3.5

É fácil mostrar que:

$$N_{0|2i-1} = R_{0|2i-1} - \Gamma_{2i}^c X_0^c .$$
 D.3.6

Faremos explicitamente os casos k=0,1; o resultado geral segue sem muita dificuldade. Lembremos que as equações do controlador (6.2a,b) são:

$$x_{k+1}^c = A_c x_k^c - B_c y_k$$
 D.3.7

$$u_k + D_c y_k = r_k - C_c x_k^c$$
. D.3.8

Considerando k = 0 nas equações (D.3.7) e (D.3.8):

$$x_{1}^{c} = A_{c}x_{0}^{c} - B_{c}y_{0}$$

$$u_{0} + D_{c}y_{0} = r_{0} - C_{c}x_{0}^{c} \Longrightarrow u_{0} + D_{c}y_{0} = r_{0} - C_{c}x_{0}^{c}.$$

Para $k = 1$ nas equações (D.3.7) e (D.3.8):

$$x_{2}^{c} = A_{c}x_{1}^{c} - B_{c}y_{1}$$

$$u_{1} + D_{c}y_{1} = r_{k} - C_{c}x_{1}^{c} \Longrightarrow u_{1} + D_{c}y_{1} = r_{k} - C_{c}Ax_{0}^{c} - B_{c}y_{0} \Longrightarrow$$

$$u_{1} + C_{c}B_{c}y_{0} + D_{c}y_{1} = r_{k} - C_{c}A_{c}x_{0}^{c}$$

ou em forma matricial:

$$\begin{bmatrix} u_0 \\ u_1 \end{bmatrix} + \begin{bmatrix} D_c & 0 \\ C_c B_c & D_c \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} - \begin{bmatrix} C_c \\ C_c A_c \end{bmatrix} X_0^c.$$

Em forma recursiva se obtém:

$$N_{p|q} = U_{p|q} + H_{q-p+1}^{c} Y_{p|q} = R_{0|2i-1} - \Gamma_{2i}^{c} X_{0}^{c}$$

$$N_{p|q} = R_{0|2i-1} - \Gamma_{2i}^{c} X_{0}^{c}.$$
D.3.9

Usando a linearidade em Ej, de (D.3.5) se obtém:

$$Ej\left[(\Gamma_{2i}X_0^d + Y_{0|2i-1}^s)N_{0|2i-1}^T\right] = Ej\left[(\Gamma_{2i}X_0 + H_{2i}^w W_{0|2i-1} + V_{0|2i-1})N_{0|2i-1}^T\right]$$
$$= Ej \left[(\Gamma_{2i} X_0) N_{0|2i-1}^T \right] + Ej \left[(H_{2i}^w W_{0|2i-1}) N_{0|2i-1}^T \right] + Ej \left[(V_{0|2i-1}) N_{0|2i-1}^T \right]$$
$$= \Gamma_{2i} Ej \left[(X_0) N_{0|2i-1}^T \right] + H_{2i}^w Ej \left[(W_{0|2i-1}) N_{0|2i-1}^T \right] + Ej \left[(V_{0|2i-1}) N_{0|2i-1}^T \right]$$
D.3.10

De (D.1.13) se tem que:

 $\Gamma_{2i} Ej \left[(X_0) N_{0|2i-1}^T \right] = \Gamma_{2i} S_{xn}$

e pela hipótese do teorema 1 para malha fechada se tem que:

$$H_{2i}^{w} Ej[(W_{0|2i-1})N_{0|2i-1}^{T}] = H_{2i}^{w} Ej[(W_{0|2i-1})R_{0|2i-1}^{T}] - H_{2i}^{w} Ej[(W_{0|2i-1})X_{0}^{T}](\Gamma_{2i}^{c})^{T} = 0$$
$$Ej[(V_{0|2i-1})N_{0|2i-1}^{T}] = 0$$

Então a equação (D.3.10) fica expressa como:

$$Ej\left[(\Gamma_{2i}X_0^d + Y_{0|2i-1}^s)N_{0|2i-1}^T\right] = \Gamma_{2i}S_{xn}.$$

Para provar (D.3.3)

$$Ej\left[Y_{0|2i-1}^{s}(X_{0}^{d})^{T}\right]=\Gamma_{2i}\Sigma_{sd}$$

se procede da mesma forma que na prova anterior, substituindo-se esta última equação em (D.3.4) e pela hipóteses do teorema 1 para malha fechada se tem que:

$$\begin{split} Ej \Big[Y_{0|2i-1}^{s} (X_{0}^{d})^{T} \Big] &= Ej \Big[\Big(\Gamma_{2i} X_{0}^{s} + H_{2i}^{w} W_{0|2i-1} + V_{0|2i-1} \Big) (X_{0}^{d})^{T} \Big] \\ &= \Gamma_{2i} Ej \Big[\Big(X_{0}^{s} \Big) (X_{0}^{d})^{T} \Big] + H_{2i}^{w} Ej \Big[\Big(W_{0|2i-1} \Big) (X_{0}^{d})^{T} \Big] + Ej \Big[\Big(V_{0|2i-1} \Big) (X_{0}^{d})^{T} \Big] \\ Ej \Big[Y_{0|2i-1}^{s} (X_{0}^{d})^{T} \Big] &= \Gamma_{2i} \Sigma_{sd} \,. \end{split}$$

Agora provaremos a equação (6.12), que é dada pela projeção ortogonal:

$$Z_{i} = Y_{i|2i-1} / \binom{N_{0|i-1}}{Y_{0|i-1}} = T_{i} \Big[\Gamma_{i} \hat{X}_{i} + H_{i}^{d} M_{i|2i-1} \Big]$$

A projeção ortogonal é dada por [VAN OVERSCHEE; DE MOOR, 1996]: $A/B = \phi[A, B](\phi[B, B])^{+}B$, onde $\phi[A, B] = Ej[AB^{T}]$

Projetando
$$Y_{i|2i-1}$$
 sobre $\begin{bmatrix} N_{0|2i-1} \\ Y_{0|i-1} \end{bmatrix} = \begin{bmatrix} U_{0|i-1} \\ M_{i|2i-1} \\ Y_{0|i-1} \end{bmatrix}$ se obtém:
 $Y_{i|2i-1} \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} = \phi \begin{bmatrix} Y_{i|2i-1}, \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} \end{bmatrix} \left(\phi \begin{bmatrix} \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}, \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} \end{bmatrix} \right)^{+} \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$
 $Y_{i|2i-1} \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} = E_{j} \begin{bmatrix} (Y_{i|2i-1}) (N^{T}_{0|i-1} | Y^{T}_{0|i-1}) \end{bmatrix} \left(E_{j} \begin{bmatrix} \begin{pmatrix} N_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} (N^{T}_{0|i-1} | Y^{T}_{0|i-1}) \end{bmatrix} \right)^{-1} \left(\frac{N_{0|i-1}}{Y_{0|i-1}} \right) D.3.11$

Esta projeção ortogonal é uma combinação linear dos elementos $U_{0|i-1}$, $M_{i|2i-1}$ e $Y_{0|i-1}$. Do outro lado, da equação (6.6) se obtém que:

$$Y_{i|2i-1} = T \left[\underbrace{\prod_{i} A^{i} X_{0}^{d} + Y_{i|2i-1}^{s}}_{\text{Termo 1}} + \underbrace{\prod_{i} \Delta_{i}^{d} U_{0|i-1} + H_{i}^{d} M_{i|2i-1}}_{\text{Termo 2}} \right].$$
 D.3.12

Quando se projeta $Y_{i|2i-1}$ sobre o subespaço $\begin{pmatrix} U_{0|i-1} \\ M_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$ como na equação (6.12) observa-se

que o termo 2 na equação (D.3.12) não muda, pois ele se encontra no espaço linha onde se está projetando. Então precisamos encontrar a projeção do termo 1. Da mesma forma que foi definida em [apêndice A], a equação (D.3.11) ficaria expressa como:

$$Y_{i|2i-1} / \binom{N_{0|i-1}}{Y_{0|i-1}} = Ej \left[\left(\Gamma_i A^i X_0^d + Y_{i|2i-1}^s \right) \left(N^T_{0|i-1} \middle| Y^T_{0|i-1} \right) \right] \left(Ej \left[\left(\frac{N_{0|i-1}}{Y_{0|i-1}} \right) \left(N^T_{0|i-1} \middle| Y^T_{0|i-1} \right) \right] \right)^{-1} \left(\frac{N_{0|i-1}}{Y_{0|i-1}} \right) + T_i \Gamma_i \Delta_i^d U_{0|i-1} + T_i H_i^d M_{i|2i-1}.$$

Para encontrar o primeiro e o segundo termos do segundo membro desta equação, introduzimos a seguinte notação:

$$F = Ej \left[\left(\Gamma_i A^i X_0^d + Y_{i|2i-1}^s \right) \left(N_{0|2i-1}^T - Y_{0|i-1}^T \right) \right] = \begin{bmatrix} F_1 & F_2 \end{bmatrix}$$
D.3.13

$$N = E_{j} \left[\left(\frac{N_{0|2i-1}}{Y_{0|i-1}} \right) \left(N_{0|2i-1}^{T} \quad Y_{0|i-1}^{T} \right) \right] = \left[\begin{array}{cc} N_{11} & N_{21}^{T} \\ N_{21} & N_{22} \end{array} \right]$$
D.3.14

Observa-se que todos os termos são conhecidos, exceto o termo $Y_{0|i-1}$

É fácil verificar que, ver [SANTOS MIRANDA, 2004]:

$$Y_{0|i-1} = \Gamma_i X_{0|i-1}^d + H_i^d U_{0|i-1} + Y_{0|i-1}^s$$
D.3.15

de (D.1.4) se obtém que:

$$N_{0|i-1} = U_{0|i-1} + H_i^c Y_{0|i-1}$$
D.3.16

Substituindo-se (D.3.16) em (D.3.15) se obtém que:

$$Y_{0|i-1} = \Gamma_i X_{0|i-1}^d + H_i^d \left(N_{0|i-1} - H_i^c Y_{0|i-1} \right) + Y_{0|i-1}^s$$

$$Y_{0|i-1} = \Gamma_i X_{0|i-1}^d + H_i^d N_{0|i-1} - H_i^d H_i^c Y_{0|i-1} + Y_{0|i-1}^s$$

$$\left(I_{li} - H_i^d H_i^c \right) Y_{0|i-1} = \Gamma_i X_{0|i-1}^d + H_i^d N_{0|i-1} + Y_{0|i-1}^s$$
Como $T_i^{-1} = \left(I_{li} - H_i^d H_i^c \right)$ então:

$$Y_{0|i-1} = T_i \Big[\Gamma_i X_{0|i-1}^d + H_i^d N_{0|i-1} + Y_{0|i-1}^s \Big].$$
 D.3.17

Agora efetuando as operações em (D.3.13) e (D.3.14) e usando-se as equações auxiliares (D.3.2) e (D.3.3), se obtém:

$$F_{1} = \Gamma_{i}A^{i}S_{xn}$$

$$F_{2} = \left[\Gamma_{i}A^{i}\left(\Sigma_{dd} + \Sigma_{sd} + \Sigma_{sd}^{T}\right)\Gamma_{i}^{T} + \Gamma_{i}A^{i}S_{xn} + \Gamma_{i}\Delta_{i}^{G}\right]T_{i}^{T}$$

$$N_{11} = R_{nn}$$

$$N_{21} = T_{i}\left[\Gamma_{i}A^{i}S_{xn} + H_{i}^{d}\left(I - 0\right)R_{nn}\right]$$

$$N_{22} = T_{i}\left[\Gamma_{i}\left(\Sigma_{dd} + \Sigma_{sd} + \Sigma_{sd}^{T}\right)\Gamma_{i}^{T} + L_{i} + \Gamma_{i}S_{xn}\left(\frac{I}{0}\right)\left(H_{i}^{d}\right)^{T} + H_{i}^{d}\left(I - 0\right)S_{xn}^{T}\Gamma_{i}^{T} + H_{i}^{d}\left(I - 0\right)R_{nn}\left(\frac{I}{0}\right)\left(H_{i}^{d}\right)^{T}\right]T_{i}^{T}$$

Então a projeção ortogonal é expressa por:

$$Z_{i} = Y_{i|2i-1} / \binom{N_{0|i-1}}{Y_{0|i-1}} = T_{i} F N^{-1} \binom{N_{0|i-1}}{Y_{0|i-1}} + T_{i} \Gamma_{i} \Delta_{i}^{d} U_{0|i-1} + T_{i} H_{i}^{d} M_{i|2i-1}$$
D.3.18

Usando a matriz inversa do Lema para a matriz N e com Ω definido em (6.10), P_0 e \hat{X}_0

como em (6.14)-(6.15) finalmente se obtém:

$$Z_{i} = T_{i}\Gamma_{i}\left[\left(A^{i} - \Omega_{i}\Gamma_{i}\right)S_{xn}R_{nn}^{-1}N_{0|2i-1} + \Omega_{i}\left(T^{-1}Y_{0|i-1} - H_{i}^{d}N_{0|i-1}\right) + \Delta_{i}^{-1}U_{0|i-1}\right] + T_{i}H_{i}^{d}M_{i|2i-1} \text{ D.3.19}$$

Na equação (D.3.19) se têm os seguintes termos:

$$S_{xn}R_{nn}^{-1}N_{0|2i-1} = Ej \Big[X_0 \ N_{0|2i-1}^T \Big] Ej \Big[N_{0|2i-1} \ N_{0|2i-1}^T \Big]^{-1}N_{0|2i-1} = X_0 / N_{0|2i-1} = \hat{X}_0$$
 D.3.20
$$(T^{-1}Y_{0|i-1} - H_i^d N_{0|i-1}) = \Gamma_i X_0^d + Y_{0|i-1}^s \text{ isto \acute{e} obtido de (D.3.17)}$$
$$= \Gamma_i X_0^d + Y_{0|i-1}^s = Y_{0|i-1} - H_i^d U_{0|i-1} \text{ isto \acute{e} obtido de (D.3.15)}$$

$$(T^{-1}Y_{0|i-1} - H_i^d N_{0|i-1}) = Y_{0|i-1} - H_i^d U_{0|i-1}$$
 D.3.21

Substituindo-se (D.3.20) e (D.3.21) em (D.3.19) se obtém:

$$Z_{i} = T_{i}\Gamma_{i}\left[\left(A^{i} - \Omega_{i}\Gamma_{i}\right)\hat{X}_{0} + \Omega_{i}\left(Y_{0|i-1} - H^{d}_{i}U_{0|i-1}\right) + \Delta^{-1}_{i}U_{0|i-1}\right] + T_{i}H^{d}_{i}M_{i|2i-1}$$
$$Z_{i} = T_{i}\Gamma_{i}\left[\left(A^{i} - \Omega_{i}\Gamma_{i}\right)\hat{X}_{0} + \left(\Delta^{-1}_{i} - \Omega_{i}H^{d}_{i}\right)U_{0|i-1} + \Omega_{i}Y_{0|i-1}\right] + T_{i}H^{d}_{i}M_{i|2i-1}$$

Por último fazendo

$$\hat{X}_{i} = (A^{i} - \Omega_{i}\Gamma_{i})\hat{X}_{0} + (\Delta_{i}^{-1} - \Omega_{i}H_{i}^{d})U_{0|i-1} + \Omega_{i}Y_{0|i-1}$$

se obtém que:

$$Z_{i} = Y_{i|2i-1} / \binom{N_{0|i-1}}{Y_{0|i-1}} = T_{i} \Big[\Gamma_{i} \hat{X}_{i} + H_{i}^{d} M_{i|2i-1} \Big].$$

D.4. Prova do teorema 2 Principal

Idéias para a demonstração: Observa-se em (6.17) que a matriz ϑ_i é a projeção oblíqua das saídas futuras $Y_{i|2i-1}$ em $M_{i|2i-1}$ sobre $W_{0|i-1}$ onde:

$$W_{0|i-1} = \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix}$$

A idéia desta demonstração é obter a projeção oblíqua em termos da projeção ortogonal, o seguinte gráfico mostra isto:



Projeção ortogonal

Figura D.4.1 Projeção oblíqua das saídas futuras $Y_{i|2i-1}$ em $M_{i|2i-1}$ sobre $W_{0|i-1}$.

Da projeção ortogonal se obtém que:

$$Y_{i|2i-1} / \begin{bmatrix} W_{0|i-1} \\ M_{i|2i-1} \end{bmatrix} = Y_{i|2i-1} / M_{i|2i-1} W_{0|i-1} + Y_{i|2i-1} / M_{0|i-1} M_{i|2i-1}$$
D.4.1

Se em (D.4.1) eliminarmos o segundo termo do segundo membro, então se obtém a projeção oblíqua, isto é:

$$Y_{i|2i-1} / \begin{bmatrix} W_{0|i-1} \\ M_{i|2i-1} \end{bmatrix} = Y_{i|2i-1} / M_{i|2i-1} W_{0|i-1} = \vartheta_i$$
D.4.2

Agora demonstraremos o teorema 2:

Para fazer esta demonstração, considerar primeiro a projeção ortogonal obtida no teorema 1 para malha fechada:

$$Z_{i} \stackrel{def}{=} Y_{i|2i-1} / \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \\ M_{i|2i-1} \end{pmatrix} = T_{i} \Big[\Gamma_{i} \hat{X}_{i} + H_{i}^{d} M_{i|2i-1} \Big]$$
D.4.3

Usando a expressão de \hat{X}_i definida no teorema 1:

$$\hat{X}_{i} = \left(A^{i} - \Omega_{i}\Gamma_{i}\right)\hat{X}_{0} + \left(\Delta_{i}^{-1} - \Omega_{i}H_{i}^{d}\right)U_{0|i-1} + \Omega_{i}Y_{0|i-1}$$
D.4.4

e \hat{X}_0 definida por:

$$\hat{X}_{0} = X_{0} / N_{0|2i-1} = X_{0} / \binom{N_{0|i-1}}{N_{i|2i-1}} = X_{0} / \binom{N_{0|i-1}}{N_{0|i-1}} N_{0|i-1} + X_{0} / \binom{N_{0|i-1}}{N_{0|i-1}} N_{i|2i-1}$$
D.4.5

Da definição das matrizes auxiliares, equação (D.1.15):

$$M_{p|q} \stackrel{def}{=} U_{p|q} + H_{q-p+1}^{c} Y_{p|q},$$

se obtém:
$$N_{i|2i-1} = M_{i|2i-1},$$
 D.4.6

Substituindo-se (D.4.6) em (D.4.5):

$$\hat{X}_{0} = X_{0} /_{M_{i|2i-1}} N_{0|i-1} + X_{0} /_{N_{0|i-1}} M_{i|2i-1}, \qquad D.4.7$$

Substituindo-se (D.4.4) em (D.4.3):

$$Z_{i} = T_{i} \Big[\Gamma_{i} \Big(\Big(A^{i} - \Omega_{i} \Gamma_{i} \Big) \hat{X}_{0} + \Big(\Delta_{i}^{-1} - \Omega_{i} H_{i}^{d} \Big) U_{0|i-1} + \Omega_{i} Y_{0|i-1} \Big) + H_{i}^{d} M_{i|2i-1} \Big],$$
D.4.8

Substituindo-se (D.4.7) em (D.4.8):

$$Z_{i} = T_{i} \Big[\Gamma_{i} \Big(\Big(A^{i} - \Omega_{i} \Gamma_{i} \Big) \Big] X_{0} /_{M_{i}|2i-1} N_{0|i-1} + X_{0} /_{N_{0}|i-1} M_{i|2i-1} \Big] + \Big(\Delta_{i}^{-1} - \Omega_{i} H_{i}^{d} \Big) U_{0|i-1} + \Omega_{i} Y_{0|i-1} \Big) + H_{i}^{d} M_{i|2i-1} \Big], \quad D.4.9$$

Se em (D.4.9) se eliminam os termos que contêm a matriz $M_{i|2i-1}$, então se obtém a projeção oblíqua:

$$Z_{i} = T_{i} \Gamma_{i} \left[\left(A^{i} - \Omega_{i} \Gamma_{i} \right) X_{0} / _{M_{i|2i-1}} N_{0|i-1} \right] + \left(\Delta_{i}^{-1} - \Omega_{i} H_{i}^{d} \right) U_{0|i-1} + \Omega_{i} Y_{0|i-1} \right]$$

ou em forma matricial:

$$Z_{i} = T_{i} \Gamma_{i} \left[\left(A^{i} - \Omega_{i} \Gamma_{i} \right) \left(\Delta_{i}^{-1} - \Omega_{i} H_{i}^{d} \right) \quad \Omega_{i} \right] \left[\frac{X_{0} /_{M_{i}|2i-1}}{U_{0|i-1}} \right]$$
D.4.10

A equação (4.10) é a projeção oblíqua denotada por ϑ_i :

$$\vartheta_i = T_i \Gamma_i(\tilde{X}_i)$$

onde:

$$\widetilde{X}_{i} = \left[\left(A^{i} - \Omega_{i} \Gamma_{i} \right) \left(\Delta_{i}^{-1} - \Omega_{i} H_{i}^{d} \right) \quad \Omega_{i} \right] \left[\frac{X_{0} / M_{i|2i-1} N_{0|i-1}}{U_{0|i-1}} \right]$$

Comparando-se esta última expressão com o estado do filtro de Kalman em (D.4.4), se mostra que \tilde{X}_i é efetivamente uma seqüência do estado de filtro de Kalman, com estado inicial \tilde{X}_0 e matriz de covariância P_0 definida por:

$$\widetilde{X}_{i} \stackrel{def}{=} X_{i}^{k} [\widetilde{X}_{0}, P_{0}]; \quad \widetilde{X}_{0} = \widehat{X}_{0} /_{M_{i}|2i-1} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} e \quad P_{0} = -\left[\Sigma_{xx} - \Sigma_{ss} - S_{xn} R_{nn}^{-1} S_{xn}^{T} \right]$$

Isto prova a parte 1 do teorema 2 para malha fechada.

Agora provaremos a parte 2 do teorema.

Para provar que a ordem do sistema é n, considerar a projeção oblíqua:

$$\vartheta_i = T_i \Gamma_i (\tilde{X}_i) = U_1 S_1 V_1^T$$

Observa-se que Γ_i tem somente *n* colunas e $\widetilde{X}_i = \hat{X}_i$ tem somente *n* filas e a matriz T_i é inversível de posto completo *lixli*, então por propriedade de posto de uma matriz se tem que:

$$posto(T_i\Gamma_i(\widetilde{X}_i)) = posto(\Gamma_i(\widetilde{X}_i))$$

É o mesmo que posto de ϑ_i o qual é a ordem do sistema, isto é *n*.

Prova das partes 3 e 4 do teorema

A decomposição da projeção oblíqua ϑ_i na equação (6.18) pode ser desdobrada em duas partes:

 $\vartheta_i = U_1 S_1^{1/2} S_1^{1/2} V_1^T$. Usando a seguinte notação,

 $T_i \Gamma_i = U_1 S_1^{1/2}$ e $\widetilde{X}_i = S_1^{1/2} V_1^T$,

e de $\vartheta_i = T_i \Gamma_i(\widetilde{X}_i)$ obtém-se:

$$\widetilde{X}_i = (T_i \Gamma_i)^+ \vartheta_i.$$

APÊNDICE E

E.1 Prova do teorema 7.3.1

Sem perda de generalidade e sem criar confusão, considere f=f(k) onde f é qualquer função no instante k.

Da figura 7.1 obtêm-se:

$$e_2 = r_2 - y \tag{E.1.2}$$

A equação (7.19) é dada por:

Substituindo-se a equação (E.1.1) na equação (E.1.3) resulta:

$$y = C_p s + D_p (e_1 + r_1)$$

 $y = C_p s + D_p e_1 + D_p r_1$ E.1.4

$$e_1 = C_c v + D_c e_2 \tag{E.1.5}$$

Substituindo-se a equação (E.1.5) na equação (E.1.4) resulta:

$$y = C_p s + D_p (C_c v + D_c e_2) + D_p r_1$$
 E.1.6

Substituindo a equação (E.1.2) na equação (E.1.6) obtém-se:

$$y = C_{p}s + D_{p}C_{c}v + D_{p}D_{c}(r_{2} - y) + D_{p}r_{1}$$

$$y = C_{p}s + D_{p}C_{c}v + D_{p}D_{c}r_{2} - D_{p}D_{c}y + D_{p}r_{1}$$

$$y + D_{p}D_{c}y = C_{p}s + D_{p}C_{c}v + D_{p}D_{c}r_{2} + D_{p}r_{1}$$

$$(I + D_{p}D_{c})y = C_{p}s + D_{p}C_{c}v + D_{p}D_{c}r_{2} + D_{p}r_{1}$$
E.1.7

Pela hipótese do teorema, a matriz $(I + D_p D_c)$ tem inversa. Seja:

$$\Phi = (I + D_p D_c)^{-1}$$
E.1.8

Multiplicando-se ambos dos lados da equação (E.1.7) pela equação (E.1.8), resulta:

$$y = \Phi C_p s + \Phi D_p C_c v + \Phi D_p D_c r_2 + \Phi D_p r_1$$
 E.1.9

O objetivo é expressar os estados da planta e do controlador em função das entradas r_1 e r_2 .

A equação (7.18) é dada por:

$$s(k+1) = A_p s + B_p u$$
 E.1.10

Substituindo-se a equação (E.1.1) na equação (E.1.10) resulta:

$$s(k+1) = A_p s + B_p (e_1 + r_1)$$

$$s(k+1) = A_p s + B_p e_1 + B_p r_1$$

E.1.11

Substituindo-se a equação (E.1.5) na equação (E.1.11) resulta:

$$s(k+1) = A_p s + B_p (C_c v + D_c e_2) + B_p r_1$$

$$s(k+1) = A_p s + B_p C_c v + B_p D_c e_2 + B_p r_1$$

E.1.12

Substituindo-se a equação (E.1.2) na equação (E.1.12) resulta:

$$s(k+1) = A_p s + B_p C_c v + B_p D_c (r_2 - y) + B_p r_1$$

$$s(k+1) = A_p s + B_p C_c v + B_p D_c r_2 - B_p D_c y + B_p r_1$$

E.1.13

Substituindo-se a equação (E.1.9) na equação (E.1.13) resulta: $s(k+1) = A_{p}s + B_{p}C_{c}v + B_{p}D_{c}r_{2} - B_{p}D_{c}(\Phi C_{p}s + \Phi D_{p}C_{c}v + \Phi D_{p}D_{c}r_{2} + \Phi D_{p}r_{1}) + B_{p}r_{1}$ $s(k+1) = A_{p}s + B_{p}C_{c}v + B_{p}D_{c}r_{2} - B_{p}D_{c}\Phi C_{p}s - B_{p}D_{c}\Phi D_{p}C_{c}v - B_{p}D_{c}\Phi D_{p}D_{c}r_{2}$ $-B_{p}D_{c}\Phi D_{p}r_{1} + B_{p}r_{1}$ $s(k+1) = (A_{p} - B_{p}D_{c}\Phi C_{p})s + (B_{p}C_{c} - B_{p}D_{c}\Phi D_{p}C_{c})v + (B_{p}D_{c} - B_{p}D_{c}\Phi D_{p}D_{c})r_{2} + (B_{p} - B_{p}D_{c}\Phi D_{p})r_{1}$ $s(k+1) = (A_{p} - B_{p}D_{c}\Phi C_{p})s + B_{p}(I - D_{c}\Phi D_{p})C_{c}v + B_{p}(I - D_{c}\Phi D_{p})r_{1} + B_{p}D_{c}(I - \Phi D_{p}D_{c})r_{2}$ E.1.14 Da mesma forma para os estados do controlador, tem-se que a equação (7.20) é dada por:

$$v(k+1) = A_c v(k) + B_c e_2(k)$$
 E.1.15

Substituindo-se a equação (E.1.2) na equação (E.1.15) resulta:

$$v(k+1) = A_c v + B_c (r_2 - y)$$

$$v(k+1) = A_c v + B_c r_2 - B_c y$$

E.1.16

Substituindo-se a equação (E.1.2) na equação (E.1.16) resulta:

$$v(k+1) = A_{c}v + B_{c}r_{2} - B_{c}(\Phi C_{p}s + \Phi D_{p}C_{c}v + \Phi D_{p}D_{c}r_{2} + \Phi D_{p}r_{1})$$

$$v(k+1) = A_{c}v + B_{c}r_{2} - B_{c}\Phi C_{p}s - B_{c}\Phi D_{p}C_{c}v - B_{c}\Phi D_{p}D_{c}r_{2} - B_{c}\Phi D_{p}r_{1}$$

$$v(k+1) = -B_{c}\Phi C_{p}s + (A_{c} - B_{c}\Phi D_{p}C_{c})v - B_{c}\Phi D_{p}r_{1} + (B_{c} - B_{c}\Phi D_{p}D_{c})r_{2}$$

$$v(k+1) = -B_{c}\Phi C_{p}s + (A_{c} - B_{c}\Phi D_{p}C_{c})v - B_{c}\Phi D_{p}r_{1} + B_{c}(I - \Phi D_{p}D_{c})r_{2}$$
E.1.17

Logo, as equações (E.14) e (E.17) podem ser expressas na forma matricial:

$$\begin{bmatrix} s(k+1)\\v(k+1)\end{bmatrix} = A \begin{bmatrix} s(k)\\v(k)\end{bmatrix} + B \begin{bmatrix} r_1(k)\\r_2(k)\end{bmatrix}$$
E.1.18

onde

$$A = \begin{bmatrix} A_p - B_p D_c \Phi C_p & B_p (I - D_c \Phi D_p) C_c \\ - B_c \Phi C_p & A_c - B_c \Phi D_p C_c \end{bmatrix}$$
E.1.19

e

$$B = \begin{bmatrix} B_p (I - D_c \Phi D_p) & B_p D_c (I - \Phi D_p C_c) \\ - B_c \Phi D_p & B_p (I - \Phi D_p C_c) \end{bmatrix}$$
E.1.20

Da mesma forma:

Substituindo-se a equação (E.1.2) na equação (E.1.5) resulta:

$$e_{1} = C_{c}v + D_{c}(r_{2} - y)$$

$$e_{1} = C_{c}v + D_{c}r_{2} - D_{c}y$$
E.1.21

Substituindo-se a equação (E.1.9) na equação (E.1.21) resulta:

$$e_{1} = C_{c}v + D_{c}r_{2} - D_{c}(\Phi C_{p}s + \Phi D_{p}C_{c}v + \Phi D_{p}D_{c}r_{2} + \Phi D_{p}r_{1})$$

$$e_{1} = C_{c}v + D_{c}r_{2} - D_{c}\Phi C_{p}s - D_{c}\Phi D_{p}C_{c}v - D_{c}\Phi D_{p}D_{c}r_{2} - D_{c}\Phi D_{p}r_{1}$$

$$e_{1} = -D_{c}\Phi C_{p}s + (C_{c} - D_{c}\Phi D_{p}C_{c})v + (D_{c} - D_{c}\Phi D_{p}D_{c})r_{2} - D_{c}\Phi D_{p}r_{1}$$

$$e_{1} = -D_{c}\Phi C_{p}s + (I - D_{c}\Phi D_{p})C_{c}v - D_{c}\Phi D_{p}r_{1} + D_{c}(I - \Phi D_{p}D_{c})r_{2}$$
E.1.22

Substituindo-se a equação (E.1.2) na equação (E.1.9) resulta:

$$e_2 = r_2 - (\Phi C_p s + \Phi D_p C_c v + \Phi D_p D_c r_2 + \Phi D_p r_1)$$

 $e_2 = -\Phi C_p s - \Phi D_p C_c v - \Phi D_p D_c r_2 - \Phi D_p r_1 + r_2$
 $e_2 = -\Phi C_p s - \Phi D_p C_c v - \Phi D_p r_1 + (I - \Phi D_p D_c) r_2$
E.1.23

Substituindo-se a equação (E.1.1) na equação (E.1.22) resulta:

$$u = [-D_c \Phi C_p s + (I - D_c \Phi D_p)C_c v - D_c \Phi D_p r_1 + D_c (I - \Phi D_p D_c) r_2] + r_1$$

$$u = -D_c \Phi C_p s + (I - D_c \Phi D_p)C_c v + (I - D_c \Phi D_p) r_1 + D_c (I - \Phi D_p D_c) r_2$$
E.1.24

Logo, as equações (E.1.22), (E.1.23), (E.1.24) e (E.1.9) podem ser expressas na forma matricial:

$$\begin{bmatrix} e_1(k) \\ e_2(k) \\ u(k) \\ y(k) \end{bmatrix} = C \begin{bmatrix} s(k) \\ v(k) \end{bmatrix} + D \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$
E.1.25

onde

$$C = \begin{bmatrix} -D_{c}\Phi C_{p} & (I - D_{c}\Phi D_{p})C_{c} \\ -\Phi C_{p} & -\Phi D_{p}C_{c} \\ -D_{c}\Phi C_{p} & (I - D_{c}\Phi D_{p})C_{c} \\ \Phi C_{p} & \Phi D_{p}C_{c} \end{bmatrix}$$

$$E.1.26$$

$$D = \begin{bmatrix} -D_{c}\Phi D_{p} & D_{c}(I - \Phi D_{p}D_{c}) \\ -\Phi D_{p} & I - \Phi D_{p}D_{c} \\ I - D_{c}\Phi D_{p} & D_{c}(I - \Phi D_{p}D_{c}) \\ \Phi D_{p} & \Phi D_{p}D_{c} \end{bmatrix}$$

$$E.1.27$$

O modelo global é dado pelas equações (E.1.18) e (E.1.25), isto é:

$$\begin{bmatrix} s(k+1)\\ v(k+1) \end{bmatrix} = A \begin{bmatrix} s(k)\\ v(k) \end{bmatrix} + B \begin{bmatrix} r_1(k)\\ r_2(k) \end{bmatrix}$$
E.1.28

$$\begin{bmatrix} e_1(k) \\ e_2(k) \\ u(k) \\ y(k) \end{bmatrix} = C \begin{bmatrix} s(k) \\ v(k) \end{bmatrix} + D \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$
E.1.29

E.2 Prova da equação (7.26)

Igualando as equações (7.15) e (E.1.29) se obtém que:

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \\ D_{31} & D_{32} \\ D_{41} & D_{42} \end{bmatrix} = \begin{bmatrix} -D_c \Phi D_p & D_c (I - \Phi D_p D_c) \\ -\Phi D_p & I - \Phi D_p D_c \\ I - D_c \Phi D_p & D_c (I - \Phi D_p D_c) \\ \Phi D_p & \Phi D_p D_c \end{bmatrix}$$
E.1.30

Da equação (E.1.30) obtém-se:

$$D_{31} = I - D_c \Phi D_p$$

ou
$$D_{31} = I - D_c (I - D_p D_c)^{-1} D_p$$

E.1.31
a inversa da matriz, dada em (E 1 31) é necessária para computar as matrizes da plante

a inversa da matriz dada em (E.1.31) é necessária para computar as matrizes da planta através da equação (7.16), aplicando matriz inversa de Leman, obtém-se:

$$D_{31}^{-1} = I + D_c D_p$$
.

E.2 Prova da equação (7.28)

Igualando as equações (7.15) e (E.1.29) se obtém que:

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \\ D_{31} & D_{32} \\ D_{41} & D_{42} \end{bmatrix} = \begin{bmatrix} -D_c \Phi D_p & D_c (I - \Phi D_p D_c) \\ -\Phi D_p & I - \Phi D_p D_c \\ I - D_c \Phi D_p & D_c (I - \Phi D_p D_c) \\ \Phi D_p & \Phi D_p D_c \end{bmatrix}$$
E.1.32

Da equação (E.1.32) obtém-se:

$$D_{22} = I - \Phi D_p D_c$$

$$D_{22} = I - (I - D_p D_c)^{-1} D_p D_c$$
 E.1.33

a inversa da matriz dada em (E.1.33) é necessária para computar as matrizes do controlador através da equação (7.17).

Para encontrar a inversa da matriz dada pela equação (E.1.33) basta demonstrar:

$$(I - D_p D_c)^{-1} D_p D_c = D_p (I - D_p D_c)^{-1} D_c$$
 E.1.34

Pela hipótese do teorema 7.3.1 existem $(I + D_p D_c)^{-1}$ e $(I + D_c D_p)^{-1}$ então existem $(D_p D_c)^{-1}$ e $(D_c D_p)^{-1}$, isto quer dizer:

$$I = (D_c D_p) \cdot (D_c D_p)^{-1}$$
 E.1.35

Considerar:

$$D_c = D_c$$
$$D_c = I.D_c$$
E.1.36

Substituindo-se a equação (E.1.35) na equação (E.1.36) resulta:

$$D_c = (D_c D_p) \cdot (D_c D_p)^{-1} \cdot D_c$$
 E.1.37

Multiplicando-se ambos os lados da equação (E.1.37) pela matriz D_p , resulta:

$$D_{p}D_{c} = D_{p}(D_{c}D_{p}).(D_{c}D_{p})^{-1}.D_{c}$$

$$D_{p}D_{c} = (D_{p}D_{c})D_{p}(D_{c}D_{p})^{-1}.D_{c}$$
E.1.38

Multiplicando-se ambos os lados da equação (E.1.38) pela matriz $(D_p D_c)^{-1}$, resulta:

$$(D_p D_c)^{-1} D_p D_c = (D_p D_c)^{-1} (D_p D_c) D_p (D_c D_p)^{-1} D_c$$

$$I = D_p (D_c D_p)^{-1} D_c$$

E.1.39

Somando-se ambos os lados da equação (E.139) pela matriz $D_p D_c$, obtém-se:

$$D_p D_c + I = D_p D_c + D_p (D_c D_p)^{-1} D_c$$
 E.1.40

considerar:

$$I = (D_p D_c)^{-1} (D_p D_c)$$
 E.141

Substituindo-se a equação (E.1.41) na equação (E.1.40) resulta:

$$D_p D_c + (D_p D_c)^{-1} (D_p D_c) = D_p D_c + D_p (D_c D_p)^{-1} D_c$$
$$[I + (D_p D_c)^{-1}] D_p D_c = D_p [I + (D_c D_p)^{-1}] D_c.$$

Observar que (aplicando, matriz inversa de Leman):

$$(I + D_p D_c)^{-1} = I^{-1} - I^{-1} D_p (I + D_c I D_p)^{-1} D_c I^{-1}$$

$$(I + D_p D_c)^{-1} = I - D_p (I + D_c D_p)^{-1} D_c$$

E.1.42

Substituindo-se a equação (E.1.34) na equação (E.1.42) resulta:

$$(I + D_p D_c)^{-1} = I - (I + D_p D_c)^{-1} D_p D_c$$
 E.1.43

O lado direito da equação E.1.43 é dado pela equação (E.1.33), isto é:

$$(I + D_p D_c)^{-1} = D_{22}$$
 E.1.44

Da equação (E.1.44) obtém-se:

 $(D_{22})^{-1} = (I + D_p D_c).$

E.4 Prova da equação (7.29)

Considerar a transformação linear T, dada por:

$$T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ então } T^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

Aplicando esta transformação ao sistema dado pela equação (7.16), obtém-se:

$$\widetilde{A} = T^{-1}AT$$

$$\widetilde{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} A & -B_1 D_{31}^{-1} C_3 \\ 0 & A - B_1 D_{31}^{-1} C_3 \end{pmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$\widetilde{A} = \begin{pmatrix} A & 0 \\ 0 & A - B_1 D_{31}^{-1} C_3 \end{pmatrix}$$

 $\widetilde{B} = T^{-1}B$

$$\widetilde{B} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} B_1 D_{31}^{-1} \\ B_1 D_{31}^{-1} \end{pmatrix}$$
$$\widetilde{B} = \begin{pmatrix} 0 \\ B_1 D_{31}^{-1} \end{pmatrix}$$

$$\widetilde{C} = CT$$

$$\widetilde{C} = \begin{pmatrix} C_4 & -D_{41}D_{31}^{-1}C_3 \end{pmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\widetilde{C} = \begin{pmatrix} C_4 & C_4 - D_{41}D_{31}^{-1}C_3 \end{pmatrix}$$

$$\widetilde{D} = D_{41} D_{31}^{-1}$$

$$P = \left[\widetilde{A}; \widetilde{B}; \widetilde{C}; \widetilde{D}\right].$$

APÊNDICE F

Algoritmo MOESP (caso determinístico)

function [A,B,C,D]=Mo_Ord(u,y,i);

$$\label{eq:mN} \begin{split} &[m,N] = size(u); \mbox{ if } m > N; \mbox{ } u=u'; \mbox{ end}; \mbox{ } [m,N] = size(u); \\ &[L,N] = size(y); \mbox{ if } L > N; \mbox{ } y=y'; \mbox{ end}; \mbox{ } [L,N] = size(y); \end{split}$$

% Matriz de Hankel

UT=hank(u,i); YT=hank(y,i); U=UT(m*i+1:2*m*i,:); Y=YT(L*i+1:2*L*i,:);

% Entradas futuras % Saídas futuras

% Matriz LQ

im=size(U,1); iL=size(Y,1); r=triu(qr([U;Y]'))'; L11=r(1:im,1:im); L21=r(im+1:im+iL,1:im); L22=r(im+1:im+iL,im+1:im+iL);

% Calculando a SVD da matriz L_{22}

[Un,ss,Vn]=svd(L22); SS=diag(ss); t=1:length(SS); figure(2);hold off; plot(t,SS); hold on; plot(t,SS,'k*'); title('valores singulares'); n = input(' Ingresse ordem do sistema ');

% Determinando as matrizes A e C

un1=Un(1:(i-1)*L,1:n); un2=Un(L+1:i*L,1:n); A=un1\un2; C=un1(1:L,:);

% Determinando as matrizes B e D

u2=Un(:,n+1:i*L); K=u2'*L21*inv(L11); qL=L*i-n; sig=zeros(i*qL,m);

```
for k=1:i
sig((k-1)*qL+1:k*qL,:)=K(:,(k-1)*m+1:k*m);
end
Q=u2';
```

```
\begin{array}{l} QQ = zeros(qL^{*i}, i^{*}L); \\ for \ k = 1:i \\ QQ((k-1)^{*}qL + 1:k^{*}qL, 1:(i-(k-1))^{*}L) = Q(:, (k-1)^{*}L + 1:L^{*}i); \\ end \\ IU = [eye(L) \ zeros(L,n); zeros(L^{*}(i-1), L) \ un1]; \end{array}
```

DB=(QQ*IU)\sig;

D=DB(1:L,:); B=DB(L+1:L+n,:);

% FIN DO ALGORITMO

Algoritmo PIMOESP

function [A,B,C,D]=PI_MOESP(u,y,i);

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

U = hank(u,i); Y = hank(y,i);

% Para algoritmo PIMOESP precisamos de h=[Uf Up Yf]

% Matriz LQ

RR = triu(qr(h'))'; R = RR(1:(2*m+L)*i,1:(2*m+L)*i); clear U Y; R32=R(2*m*i+1:(2*m+L)*i,m*i+1:2*m*i);

% Truncamento da matriz R

% Calculando a SVD da matriz R_{32}

[Un,ss,Vn]=svd(R32); SS=diag(ss); t=1:length(SS); figure(2);hold off; plot(t,SS); hold on; plot(t,SS,'k*'); title('valores singulares'); n = input(' Ingresse ordem do sistema ');

% Calculando a matriz A e C

un1=Un(1:(i-1)*L,1:n); un2=Un(L+1:i*L,1:n); A=un1\un2; C=un1(1:L,:);

% Determinando as matrizes B e D

u2=Un(:,n+1:i*L); R11 = R(1:m*i,1:m*i); R31 = R(2*m*i+1:(2*m+L)*i,1:m*i); K=u2'*R31*inv(R11);

qL=L*i-n; sig=zeros(i*qL,m); for k=1:i sig((k-1)*qL+1:k*qL,:)=K(:,(k-1)*m+1:k*m); end Q=u2';

```
\begin{array}{l} QQ = zeros(qL^{*i},i^{*}L); \\ for \ k=1:i \\ QQ((k-1)^{*}qL+1:k^{*}qL,1:(i-(k-1))^{*}L) = Q(:,(k-1)^{*}L+1:L^{*i}); \\ end \\ IU = [eye(L) \ zeros(L,n); zeros(L^{*}(i-1),L) \ un1]; \end{array}
```

DB=(QQ*IU)\sig;

D=DB(1:L,:); B=DB(L+1:L+n,:);

% FIN DO ALGORITMO

Algoritmo POMOESP

function [A,B,C,D]=P0_MOESP(u,y,i);

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

% PARA ALGOTIMO MOESP PRECISAMOS DE h=[Uf Up Yp Yf]

 $\begin{array}{ll} U &= hank(u,i); & Y = hank(y,i); \\ UP &= U(1:m^{*}i,:); \\ UF &= U(m^{*}i+1:2^{*}m^{*}i,:); \\ h &= [UF;UP;Y]; \end{array}$

% MATRIZ LQ

 $\begin{array}{ll} R &= triu(qr(h'))'; \\ R &= R(1:2^*(m+L)^*i, 1:2^*(m+L)^*i); \\ clear U Y; \end{array} \\ \end{array} \\ \begin{array}{ll} \% & Truncamento \ da \ matriz \ R \\ \end{array}$

 $\begin{array}{l} \mathsf{R11} = \mathsf{R}(1:\mathsf{m}^*\mathsf{i},1:\mathsf{m}^*\mathsf{i});\\ \mathsf{R21} = \mathsf{R}(\mathsf{m}^*\mathsf{i}+1:2^*\mathsf{m}^*\mathsf{i},1:\mathsf{m}^*\mathsf{i});\\ \mathsf{R22} = \mathsf{R}(\mathsf{m}^*\mathsf{i}+1:2^*\mathsf{m}^*\mathsf{i},\mathsf{m}^*\mathsf{i}+1:2^*\mathsf{m}^*\mathsf{i});\\ \mathsf{R31} = \mathsf{R}(2^*\mathsf{m}^*\mathsf{i}+1:(2^*\mathsf{m}+\mathsf{L})^*\mathsf{i},1:\mathsf{m}^*\mathsf{i});\\ \mathsf{R32} = \mathsf{R}(2^*\mathsf{m}^*\mathsf{i}+1:(2^*\mathsf{m}+\mathsf{L})^*\mathsf{i},\mathsf{m}^*\mathsf{i}+1:2^*\mathsf{m}^*\mathsf{i});\\ \mathsf{R41} = \mathsf{R}((2^*\mathsf{m}+\mathsf{L})^*\mathsf{i}+1:2^*(\mathsf{m}+\mathsf{L})^*\mathsf{i},1:\mathsf{m}^*\mathsf{i});\\ \mathsf{R42} = \mathsf{R}((2^*\mathsf{m}+\mathsf{L})^*\mathsf{i}+1:2^*(\mathsf{m}+\mathsf{L})^*\mathsf{i},\mathsf{m}^*\mathsf{i}+1:2^*\mathsf{m}^*\mathsf{i});\\ \mathsf{R43} = \mathsf{R}((2^*\mathsf{m}+\mathsf{L})^*\mathsf{i}+1:2^*(\mathsf{m}+\mathsf{L})^*\mathsf{i},2^*\mathsf{m}^*\mathsf{i}+1:(2^*\mathsf{m}+\mathsf{L})^*\mathsf{i});\\ \end{array}$

% CALCULANDO A SVD DA MATRIZ [R42 R43]

[Un,ss,Vn]=svd([R42 R43]); SS=diag(ss); t=1:length(SS); figure(2);hold off; plot(t,SS); hold on; plot(t,SS,'k*'); title('valores singulares'); n = input(' Ingresse ordem do sistema ');

% CALCULANDO A MATRIZ A E C

un1=Un(1:(i-1)*L,1:n); un2=Un(L+1:i*L,1:n); A=un1\un2; C=un1(1:L,:);

% CALCULANDO AS MATRIZES B E D

u2=Un(:,n+1:i*L); R1C= [R21 R22 R11]; R2C= [R31 R32 R41]; K=u2'*R2C/R1C;

```
qL=L*i-n;
sig=zeros(i*qL,m);
for k=1:i
sig((k-1)*qL+1:k*qL,:)=K(:,(k-1)*m+1:k*m);
end
Q=u2';
```

 $\begin{array}{l} QQ = zeros(qL^{*i},i^{*}L);\\ for \ k=1:i\\ QQ((k-1)^{*}qL+1:k^{*}qL,1:(i-(k-1))^{*}L) = Q(:,(k-1)^{*}L+1:L^{*}i);\\ end\\ IU = [eye(L) \ zeros(L,n); zeros(L^{*}(i-1),L) \ un1]; \end{array}$

DB=(QQ*IU)\sig; D=DB(1:L,:); B=DB(L+1:L+n,:); % FIN DO ALGORITMO

Algoritmo determinístico 1 (N4SID) (empregando estados)

function [A,B,C,D]=Det1_N4sid(u,y,i); [m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

U=hank(u,i); Y=hank(y,i); h=[U;Y]; R = triu(qr(h'))';R=R(1:2*(m+L)*i,1:2*(m+L)*i); % Trucanmento de R; clear U;Y;

% Computando Projeçao obliqua

 $Up = R(1:m^{*}i,:);$ % Entradas Passadas $Uf = R(m^{i+1}:2^{m^{i}});$ % Entradas futuras Yp = R(2*m*i+1:2*m*i+L*i,:); % Saídas Passadas Yf = R(2*m*i+L*i+1:(2*m*i+2*L*i),:); % Saídas futuras Wp = [Up; Yp];

AB=Yf -Yf*(Uf')*pinv(Uf*(Uf'))*Uf; Oi= AB*pinv(CB)*Wp;

CB=Wp -Wp*(Uf')*pinv(Uf*(Uf'))*Uf;

clear AB CB

% Computando a outra projecao obliqua

 $Uf_ = R(m^{*}(i+1)+1:2^{*}m^{*}i,:); Yf_ = R(i^{*}(2^{*}m+L)+L+1:2^{*}i^{*}(m+L),:);$ Upmas = $R(1:m^{*}(i+1),:);$ Ypmas = R(2*m*i+1:i*(2*m+L)+L,:);Wpmas = [Upmas;Ypmas];

AB=Yf -Yf *(Uf ')*pinv(Uf *(Uf '))*Uf ; CB=Wpmas -Wpmas*(Uf ')*pinv(Uf *(Uf '))*Uf ; Oi = AB*pinv(CB)*Wpmas;clear AB CB

[U,S,V] = svd(Oi);ss = diag(S); plot(ss);

Ordem do sistema ? '); n = input('

% Encontrando U1

U1 = U(:,1:n);

% Determine ri e Ri_

ri = U1*diag(sqrt(ss(1:n))); ri_= ri(1:L*(i-1),:); Xi = pinv(ri)*Oi; Xi_mas = pinv(ri_)*Oi_;

% 6 determine as matrizes A,B,C,D

K = [Xi;Uii];

Uii = $R(m^{i+1}:m^{i+m}:);$ Yii = $R((2^{m+L})^{i+1}:(2^{m+L})^{i+L}:);$ J = [Xi_mas;Yii];

X=J/K;A=X(1:n,1:n);B=X(1:n,n+1:n+m); C=X(n+1:n+L,1:n); D=X(n+1:n+L,n+1:n+m);

% FIN DO ALGORITMO

Algoritmo determinístico 2 (N4SID) (empregando a matriz de observabilidade estendida)

function [A,B,C,D]=Det_N4sid2(u,y,i);

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

% Computando Projeção obligua $Up = R(1:m^{*}i,:);$ % Entradas Pasadas $Uf = R(m^{i+1}:2^{m^{i}});$ % Entradas futuras Yp = R(2*m*i+1:2*m*i+L*i,:); % Saidas Pasadas Yf = R(2*m*i+L*i+1:(2*m*i+2*L*i),:); % Saidas futuras Wp = [Up; Yp];AB=Yf -Yf*(Uf')*pinv(Uf*(Uf'))*Uf; % projecao ortogonal CB=Wp -Wp*(Uf')*pinv(Uf*(Uf'))*Uf; Oi= AB*pinv(CB)*Wp; clear AB CB; [U,S,V] = svd(Oi); S1 = diag(S); plot(S1); n = input(' Ordem do sistema ? '); U1 = U(:,1:n);% Matriz para obter Mat Obsrva estendida = [ri] Lx n S2 = sqrt(S1(1:n));U2 = U(:,n+1:L*i);% Matriz complemento rotogonal de ri de [1:L,n+1:Li] % Considerando W=I, % Matriz de Observabilidade estendida = ri ri = U1*diag(sqrt(S2(1:n)));

ri_ortg = U2'; % Matriz complemento ortogonal de ri ri_ = ri(1:L*(i-1),:); % matriz de onservabildade extendida sem as ultimas linhas

 $r_{i} = r_{i}(1:L^{-(i-1),:});$ % matriz de onservabildade extendida sem as ultimas linnas $r_{i} = r_{i}(L+1:L^{*}_{i},:);$ % ri sem as primeiras L linhas. $A = pinv(r_{i})^{*}rr_{i};$ $C = r_{i}(1:L,:);$

% Calculando as Mtraizes B e D % Primeiro encontramos MM e LLL

% Computando LLL

for k=1:i LLL((k-1)*T+1:k*T,1:(i-k+1)*L) = LL(:,(k-1)*L+1:L*i); end

%eMM

$$\begin{split} M &= zeros(i^{*}T,m); \\ for \ k=1:i \\ M(T^{*}(k-1)+1:k^{*}T,:) = Mm(:,m^{*}(k-1)+1:k^{*}m); \\ end \end{split}$$

% Aux, donde M = LLL*Aux*[D;B] Aux = [eye(L,L), zeros(L,n); zeros(L*(i-1),L), ri_]; LLLAux = LLL*Aux;

%soluçao do [D;B]=LLLAux\M e Recuperando as matrizes DB = LLLAux\M; D = DB(1:L,:); B = DB(L+1:L+n,:); % FIN DO ALGORITMO

Algoritmo 1 N4SID

(Empregando seqüência de estados \widetilde{X}_i)

function [A,B,C,D]=N4sid1(u,y,i);

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

U = hank(u,i);Y = hank(y,i);h=[U;Y]; R = triu(qr(h'))'; $R = R(1:2^{*}(m+L)^{*}i, 1:2^{*}(m+L)^{*}i);$ % Truncamento da matriz R clear U Y; Up = R(1:m*i,:); % Entradas passadas Uf = R(m*i+1:2*m*i,:); % Entradas futuras Yp = R(2*m*i+1:(2*m+L)*i,:);% Saidas passadas Yf = R((2*m+L)*i+1:2*(m+L)*i,:);% Saidas futuras Wp = [Up; Yp]; $G = pinv(Uf^*(Uf'))^*Uf;$ % auxiliar para computar A/B $AB = Yf - Yf^*(Uf')^*G;$ % Projeçao ortogonal de Yf/Uf % Projeçao ortogonal de Wp/Uf $CB = Wp - Wp^*(Uf')^*G;$ Obli= AB*pinv(CB)*Wp; % Projecao Obligua clear AB CB G; [Un,ss,Vn] = svd(Obli);% Determine a ordem dos valores singulares, %U1 e S1 SS = diag(ss); plot(SS);% Determine n n = input(' Ordem do sistema ? '); U1 = Un(:,1:n);% Matriz para obter Mat Obsrva estendida = [ri] Lixn S1 = sqrt(SS(1:n));% Considerando W=I, % Matriz de Observabilidade estendida = ri ri = U1*diag(sqrt(SS(1:n))); % matriz de onservabildade extendida sem as as ultimas linhas $ri_ = ri(1:L^{*}(i-1),:);$ % Determine as seguencia de estados % precisamos computar a proj Oi+1=Yf_/Uf_ Wpmas Uf = $R(m^{(i+1)+1}:2^{m^{i}}:);$ Yf = $R(i^{*}(2^{*}m+L)+L+1:2^{*}i^{*}(m+L),:);$ Upmas = $R(1:m^{*}(i+1),:);$ Ypmas = R(2*m*i+1:i*(2*m+L)+L,:);Wpmas = [Upmas;Ypmas];

GG = (Uf_')*pinv(Uf_*(Uf_'))*Uf_; AB = Yf_-Yf_*GG; CB = Wpmas -Wpmas*GG; Obli_mas= AB*pinv(CB)*Wpmas;

% e suas seudos inversas

Xi = pinv(ri)*Obli; Xi_mas = pinv(ri_)*Obli_mas;

% Determine as matrizes A,B,C,D

Algoritmo 2 N4SID

(Empregando seqüência de estados \hat{X}_i)

function [A,B,C,D]=N4sid2(u,y,i);

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

U = hank(u,i);Y = hank(y,i); h=[U;Y];

 $\begin{array}{l} {\sf R} &= triu(qr(h'))'; \\ {\sf R} &= {\sf R}(1:2^*(m+L)^*i,1:2^*(m+L)^*i); \\ {\sf clear} \; {\sf U} \; {\sf Y}; \end{array}$

Up = R(1:m*i,:); Uf = R(m*i+1:2*m*i,:); Yp = R(2*m*i+1:(2*m+L)*i,:); Yf = R((2*m+L)*i+1:2*(m+L)*i,:); Wp = [Up;Yp];

 $G = pinv(Uf^*(Uf'))^*Uf;$ $AB = Yf - Yf^*(Uf')^*G;$ $CB = Wp - Wp^*(Uf')^*G;$ $Obli= AB^*pinv(CB)^*Wp;$ % auxiliar para computar A/B % Projeção ortogonal de Yf/Uf % Projeção ortogonal de Wp/Uf % Projeção Oblígua

% Truncamento da matriz R

% Entradas passadas

% Entradas futuras

% Saídas passadas

% Saídas futuras

clear AB CB G;

[Un,ss,Vn] = svd(Obli); SS = diag(ss); plot(SS);

% Determine a ordem dos valores singulares, %U1 e S1

n = input(' Ordem do sistema ? '); U1 = Un(:,1:n); % Matriz para obter Mat Obsrva estendida = [ri] Lixn S1 = sqrt(SS(1:n));

% Considerando W=I,

ri = U1*diag(sqrt(SS(1:n))); % Matriz de Observabilidade estendida = ri ri_ = ri(1:L*(i-1),:); % matriz de onservabildade extendida sem as ultimas linhas

% CALCULANDO AS PROJECOES ORTOGONAIS

$$\begin{split} &Zi=[R((2^*m+L)^*i+1:2^*(m+L)^*i,1:(2^*m+L)^*i)]; &\% \text{ formula}^{***} \text{ da Teses} \\ &Zimas=[R((2^*m+L)^*i+L+1:2^*(m+L)^*i,1:(2^*m+L)^*i+L)]; &\% \text{ Formula }^{****} \text{ da teses} \end{split}$$

Yii=R(2*m*i+L*i+1:2*m*i+L*i+L,1:(2*m+L)*i+L); % formula**** da teses Uif=Uf(:,1:(2*m+L)*i+L); % se escolhe 5 colunas para dimensionar con Yii

XY=[pinv(ri_)*Zimas;Yii]; XU=[pinv(ri)*Zi,zeros(n,L);Uif];

ACK=XY/XU;

% recuperar A C e K a traves de minimos quadrados

% Recuperando as matriz A e C e K

A = ACK(1:n,1:n); C = ACK(n+1:n+L,1:n); K = ACK(:,n+1:end);

% Obtendo as matriz B e D LL=[A;C]*pinv(ri);

% Observar que LL=[L1;L2] onde [L1]nxLi e [L2] LxLi L1=LL(1:n,:); L2=LL(n+1:n+L,:); M=pinv(ri_);

% observar que K=[km1;km2]

km1=K(1:n,:); km2=K(n+1:n+L,:); K1=zeros(i*n,m); K2=zeros(i*L,m);

for k=1:i $K1(n^{*}(k-1)+1:k^{*}n,:)=km1(:,m^{*}(k-1)+1:k^{*}m);$ $K2(L^{*}(k-1)+1:k^{*}L,:)=km2(:,m^{*}(k-1)+1:k^{*}m);$ end

% Agora formando a matriz N

MM=[zeros(n,L) M]; ML=MM-L1;

L22=-L2(:,L+1:L*i); L2_=[eye(L)-L2(:,1:L), L22]; N2=zeros(L*i,L*i); N1=zeros(i*n,L*i);

for k=1:i N1(n*(k-1)+1:k*n,1:(i-k+1)*L)=ML(:,(k-1)*L+1:L*i); N2(L*(k-1)+1:k*L,1:(i-k+1)*L)=L2_(:,(k-1)*L+1:L*i); end

N=[N1;N2]* [eye(L,L) , zeros(L,n);zeros(L*(i-1),L), ri_];

% por ultimo extraindo as matrizes B e D % [K1;K2]=N[D;B] DB=N\[K1;K2];

% recuperando as matrizes D=DB(1:L,:); B=DB(L+1:n+L,:);

% FIN DO ALGORITMO

Algoritmo determinístico MON4SID 1

(empregando estados)

% y=[y(1), y(2) ,...y(Ndat)]; pxNdat matriz % u=[u(1), u(2) ,...u(Ndat)]; mxNdat matriz % n = dim(x); % k=numeros de blocos fila

function [A,B,C,D]=MON4SID1(u,y,k);

% ADIMENSIONANDO OS DADOS

$$\label{eq:mN} \begin{split} &[m,N] = size(u); \mbox{ if } m > N; \mbox{ } u=u'; \mbox{ end}; \mbox{ } [m,N] = size(u); \\ &[L,N] = size(y); \mbox{ if } L > N; \mbox{ } y=y'; \mbox{ end}; \mbox{ } [L,N] = size(y); \\ &clear \mbox{ } m,N,L \end{split}$$

% 1. Construir as matrizes de Hankel U_f, U_p e Y_f, Y_p .

```
[p,Ndat]=size(y); [m,Ndat]=size(u);
N=Ndat-2*k
                     % Numero de Colunas da matriz em blocos se Hankel
ii=0;
for i=1:m:2*k*m-m+1
  ii=ii+1;
  U(i:i+m-1,:)=u(:,ii:ii+N-1);
                              % Dados da matriz U
end
ii=0;
for i=1:p:2*k*p-p+1
  ii=ii+1;
  Y(i:i+p-1,:)=y(:,ii:ii+N-1);
                              % Dados da matriz Y
end
UUf=U(k*m+1:2*k*m,:);
                              % Entradas futuras
YYf=Y(k*p+1:2*k*p,:);
                              % Saídas futuras
UUp=U(1:k*m,:);
                              % Entradas passadas
YYp=Y(1:k*p,:);
                              % Saídas passadas
Wp=[UUp;YYp];
Wf=[UUf;YYf];
H=(1/N)*[UUf;Wp;YYf];
                              % Matriz de Hankel
% 2. Decomposição LQ ou QR
[Q,L]=qr(H',0);
R=L';
```

% COMPUTANDO GAMA METODO MOESP %

% 3. Calcular a SVD da matriz L_{32} . L32 = R(k*(2*m+p)+1:2*k*(m+p),k*m+1:k*(2*m+p)); i=k; L=p [Un,ss,Vn]=svd(L32);

% 4. Determinar a ordem do sistema por inspeção dos valores singulares em S e particionar a SVD para obter U_1 . SS=diag(ss); plot(SS) n = input(' Ingrese ordem do sistema ');

% 5. Determinar $\Gamma_i e \Gamma_{i-1}$ como ri=Un(:,1:n); % Matriz observabilidade estendida ri_= ri(1:L*(i-1),:);

% COMPUTANDO A PROJECAO OBLIQUA

L21=R(k*m+1:k*(2*m+p),1:k*m); L22=R(k*m+1:k*(2*m+p),k*m+1:k*(2*m+p)); Oi=L32*inv(L22)*[L21 L22]; % **PORJEÇAO OBLIQUA**

% Projecao obliqua para O1+1

 $\begin{array}{lll} Uf_=R(m+1:m^*k,:); & Yf_=R(k^*(2^*m+L)+L+1:2^{*i*}(m+L),:); \\ Up=R(m^*i+1:2^*m^*i,:); & Yp=R(2^*m^*i+1:2^*m^*i+L^*i,:); \\ Ui=R(1:m,:); & Yi=R(k^*(2^*m+L)+1:k^*(2^*m+L)+L,:); \end{array}$

Upmas = [Up;Ui] ; Ypmas = [Yp;Yi]; Wpmas = [Upmas;Ypmas];

AB=Yf_-Yf_*(Uf_')*pinv(Uf_*(Uf_'))*Uf_; CB=Wpmas -Wpmas*(Uf_')*pinv(Uf_*(Uf_'))*Uf_; Oi_= AB*pinv(CB)*Wpmas; clear AB CB

% 6. Determine X_i^d e X_{i+1}^d pelas equações (4.13) e (4.14). Xi= [pinv(ri)*Oi zeros(n,L*k)];

Xi_mas = pinv(ri_)*Oi_;

% 7. Determine as matrizes A,B,C,D

KK = [Xi;Ui]; J = [Xi_mas;Yi]; X=J/KK;

A=X(1:n,1:n); B=X(1:n,n+1:n+m); C=X(n+1:n+L,1:n); D=X(n+1:n+L,n+1:n+m);

% FIN DO ALGORITMO

Algoritmo determinístico MON4SID 2

(empregando a matriz de observabilidade estendida)

% y=[y(1), y(2) ,...y(Ndat)]; pxNdat matriz % u=[u(1), u(2) ,...u(Ndat)]; mxNdat matriz % n = dim(x); k=numeros de blocos fila

function [A,B,C,D]=MON4SID2(u,y,k);

```
% ADIMENSIONANDO OS DADOS
[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u);
[L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);
clear m.N.L
```

% 1. Construir as matrizes de Hankel $U_f, U_p \in Y_f, Y_p$.

```
[p,Ndat]=size(y); [m,Ndat]=size(u);
N=Ndat-2*k
                  % Numero de Colunas da matriz em blocos se Hankel
ii=0;
for i=1:m:2*k*m-m+1
 ii=ii+1:
 U(i:i+m-1,:)=u(:,ii:ii+N-1); % Dados da matriz U
end
ii=0;
for i=1:p:2*k*p-p+1
 ii=ii+1:
 Y(i:i+p-1,:)=y(:,ii:ii+N-1);
                         % Dados da matriz Y
end
UUf=U(k*m+1:2*k*m,:);
                         % Entradas futuras
                         % Saídas futuras
YYf=Y(k*p+1:2*k*p,:);
UUp=U(1:k*m,:);
                         % Entradas passadas
                         % Saídas passadas
YYp=Y(1:k*p,:);
Wp=[UUp;YYp];
Wf=[UUf;YYf];
H=(1/N)*[UUf;Wp;YYf];
                         % Matriz de Hankel
% Decomposicao LQ ou QR
[Q,L]=qr(H',0);
R=L';
%
      COMPUTANDO GAMA metodo MOESP
%
% 3. Calcular a SVD da matriz L_{32}.
L32 = R(k^{*}(2^{*}m+p)+1:2^{*}k^{*}(m+p),k^{*}m+1:k^{*}(2^{*}m+p));
i=k; L=p
[Un,ss,Vn]=svd(L32);
```

% 4. Determinar a ordem do sistema por inspeção dos valores singulares em S e particionar a SVD para obter U_1 . SS=diag(ss); plot(SS) n = input(' Ingresse ordem do sistema '); % 5. Determinar Γ_i e Γ_{i-1} como: ri=Un(:,1:n); % Matriz observabilidade estendida un1=Un(1:(i-1)*L,1:n); un2=Un(L+1:i*L,1:n);

% COMPUTAMO as matrizes A C

A=un1\un2; C=un1(1:L,:);

% COMPUTANDO MATRIZES D e B "METODO N4SID"

ri ortg = U2';

******* U2 = Un(:,n+1:L*i); % Matriz complemento rotogonal de ri % Matriz complemento ortogonal de ri ri_ = ri(1:L*(i-1),:); % matriz de onservabildade extendida sem as ultimas linhas

% Calculando as Mtraizes B e D % Primeiro encontramos MM e LLL

Uf = R(1:m*i,:); % Entradas futuras COMPUTADAS DE R Yf = R(2*m*i+L*i+1:(2*m*i+2*L*i),:); % Saídas futuras COMPUTADAS DE R

Mm = ri ortg*Yf*pinv(Uf);LL = ri_ortg;

T=(L*i-n); % T = Numero de linhas de LLL LLL = zeros(i*T,L*i);

% Computando LLL

for k=1:i LLL((k-1)*T+1:k*T,1:(i-k+1)*L) = LL(:,(k-1)*L+1:L*i);end

%eMM

 $M = zeros(i^{T},m);$ for k=1:i $M(T^{*}(k-1)+1:k^{*}T,:)=Mm(:,m^{*}(k-1)+1:k^{*}m);$ end

% Aux, donde M = LLL*Aux*[D;B] Aux = $[eye(L,L), zeros(L,n); zeros(L^{*}(i-1),L), ri_];$ LLLAux = LLL*Aux;

%soluçao do [D;B]=LLLAux\M DB = LLLAuxM;

% Recuperando as matrizes D = DB(1:L,:);B = DB(L+1:L+n,:);

% FIM DO ALGORIMO

Algoritmo determinístico MON4SID 3

(empregando estados totais)

% y=[y(1), y(2) ,...y(Ndat)]; pxNdat matriz % u=[u(1), u(2) ,...u(Ndat)]; mxNdat matriz % n = dim(x); % k=numeros de blocos fila

function [Kn,A,B,C,D]=MON4SID3(u,y,k);

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y); clear m,N,L

% 1. Construir as matrizes de Hankel $U_f, U_p \in Y_f, Y_p$.

```
[p,Ndat]=size(y);
[m,Ndat]=size(u);
N=Ndat-2*k
ii=0;
for i=1:m:2*k*m-m+1
    ii=ii+1;
    U(i:i+m-1,:)=u(:,ii:ii+N-1); % Dados da matriz
end
ii=0;
for i=1:p:2*k*p-p+1
    ii=ii+1;
    Y(i:i+p-1,:)=y(:,ii:ii+N-1); % Dados da matriz
end
```

```
UUf=U(k*m+1:2*k*m,:); YYf=Y(k*p+1:2*k*p,:);
UUp=U(1:k*m,:); YYp=Y(1:k*p,:); Wp=[UUp;YYp];
```

```
H=(1/N)*[UUf;Wp;YYf];
```

```
% 2. Calcular a fatoração LQ, equação (4.10).
[Q,L]=qr(H',0);
L=L';
```

```
L22= L(k*m+1:k*(2*m+p),k*m+1:k*(2*m+p));
L32 = L( k*(2*m+p)+1:2*k*(m+p),k*m+1:k*(2*m+p)); i=k;
```

% 3. Calcular a SVD da matriz L_{32} . [Un,ss,Vn]=svd(L32);

% 4. Determinar a ordem do sistema por inspeção dos valores singulares em *S* e particionar a SVD para obter U_1 SS=diag(ss); t=1:length(SS); figure(2);hold off; plot(t,SS); hold on; plot(t,SS,'k*'); title('valores singulares');

n = input(' Inprese ordem do sistema ');

U1=Un(:,1:n);

% 5. Determinar Γ_i como gama=U1;

% 6. Determinar X pela equações (4.18) e as seqüências de estados X_i^d e X_{i+1}^d X=pinv(gama)*L32*inv(L22)*Wp;

XX=X(:,2:N); X=X(:,1:N-1);

% 7. Solucionar o seguinte conjunto de equações lineares para obter *A*, *B*, *C* e *D*: U=UUf(1:m,1:N-1); Y=YYf(1:p,1:N-1);

ABCD=[XX;Y]/[X;U]; A=ABCD(1:n,1:n); B=ABCD(1:n,n+1:n+m); C=ABCD(n+1:n+p,1:n); D=ABCD(n+1:n+p,n+1:n+m);

% FIM DO ALGORIMO

Algoritmo determinístico estocástico MON4SID 4

% 1. Compute os passos de 1 até 6 do algoritmo MON4SID 1.

% 2. Solucione o conjunto de equações lineares para encontrar A, B, C e D:

% 3. Determine *Q*, *S* e *R* dos resíduos, como: % do algoritmo MON4SID1 obtemos KK e J e n res=J-[A B;C D]*KK; cov=res*res';

% Determine Q, R S de cov Qs=cov(1:n,1:n); Ss=cov(1:n,n+1:n+L); Sts=cov(n+1:n+L,1:n); Rs=cov(n+1:n+L,n+1:n+L);

n % ordem do sistema LqL=dlqe(A,eye(n),C,Qs,Rs);

K=A*LqL; % ganho do filtro de Kalman

% FIM DO ALGORIMO

Algoritmo determinístico estocástico MON4SID 5

% 1. Compute os passos de 1 até 6 do algoritmo MON4SID 3.

% 2. Solucione o conjunto de equações lineares para encontrar A, B, C e D:

% 3. Determine *Q*, *S* e *R* dos resíduos, como: % do algoritmo MON4SID3 obtemos XX, Y, X, U e n % Encontrando os residuais res=[XX;Y]-ABCD*[X;U];; cov=res*res';

% Determine Q, R S de cov
L % numero de saídas do processo;
Qs=cov(1:n,1:n);
Ss=cov(1:n,n+1:n+L);
Sts=cov(n+1:n+L,1:n);
Rs=cov(n+1:n+L,n+1:n+L);

n % ordem do sistema LqL=dlqe(A,eye(n),C,Qs,Rs);

K=A*LqL;

% ganho do filtro de Kalman

% FIM DO ALGORIMO

Algoritmo 1 para malha fechada N4SIDC

% Se precisa conhecer as matrizes do controlador (Ac,Bc,Cc,Dc) % ou as entradas impulso do controlador

% Dados de entrada:

% R : Sinal de referencia

% U : Sinal de entrada da planta

% Y : Sinal de saída da planta

% (A,B,C,D): matrizes do controlador % i= 10

% Dados de saida

% As matrizes da planta

function [AA,BB,CC,DD,K]=N4SIDmf1(u,y,A,B,C,D,i);

% CONTROLADOR

A = [2.65 -3.11 1.75 -0.39; 1 0 0 0; 0 1 0 0; 0 0 1 0]; B = [1 0 0 0]'; C = [-0.4135 0.8629 -0.7625 0.2521]; D = 0.61;

[m,N]=size(u); if m > N; u=u'; end; [m,N]=size(u); [L,N]=size(y); if L > N; y=y'; end; [L,N]=size(y);

 $\begin{array}{l} U &= hank(u,i); \\ Y &= hank(y,i); \\ h = [U;Y]; \end{array}$

R = triu(qr(h'))'; RR = R(1:2*(m+L)*i,1:2*(m+L)*i); % Truncamento da matriz R clear U Y; % CONSTRUENCDO A MATRIZ Hc controlador [L,d]=size(C); ri=C; for k=2:i ri((k-1)*L+1:k*L,:)=ri((k-2)*L+1:(k-1)*L,:)*A; end rr=[D;ri*B]; [LL,m]=size(D); Hc=zeros(LL*i,m*i); for k=1:i; Hc((k-1)*LL+1:LL*i,(k-1)*m+1:k*m)=rr(1:LL*(i-k+1),:); end

% 1. Construa as matrizes $M_{i|2i-1} = U_{i|2i-1} + H_i^c Y_{i|2i-1}$ e $M_{i+1|2i-1} = U_{i+1|2i-1} + H_{i-1}^c Y_{i+1|2i-1}$, onde H_i^c e H_{i-1}^c contêm os parâmetros de Markov do controlador.

% CONSTRUENDO A MATRIZ M

 $\label{eq:uf=RR} \begin{array}{l} Uf=RR(m^{*}i+1:2^{*}m^{*}i,:); \\ Yf=RR((2^{*}m+L)^{*}i+1:2^{*}i^{*}(m+L),:); \end{array}$

Mf=Uf+Hc*Yf;

% Saidas Futuros

% construendo MF_ Hc_=Hc(1:m*(i-1),1:L*(i-1)); Uf_=RR(m*(i+1)+1:2*m*i,:); Yf_=RR(i*(2*m+L)+L+1:2*i*(m+L),:);

Mf_=Uf_+Hc_*Yf_;

% Saidas pasadas

% 2. CALCULANDO PROJECAO OBLIQUA E ORTOGONAL Up=RR(1:m*i,:); Yp=RR(2*m*i+1:(2*m+L)*i,:); Wp=[Up;Yp];

% Projeçao oblíqua. MfT=Mf'; Yf_Mf= Yf-Yf *MfT * pinv(Mf *MfT)*Mf ; Wp_Mf =Wp-Wp*MfT * pinv(Mf *MfT)*Mf ; Obli=Yf_Mf * pinv(Wp_Mf)*Wp ;

% Analogamente para a outra projeção

Upmas=RR(1:m*(i+1),:); Ypmas=RR(2*m*i+1:i*(2*m+L)+L,:); Wpmas=[Upmas;Ypmas];

Mft=Mf_';

Yf_MMf= Yf_-Yf_*Mft*pinv(Mf_*Mft)*Mf_; Wp_MMf =Wpmas-Wpmas*Mft * pinv(Mf_ *Mft)*Mf_; Obli_mas=Yf_MMf * pinv(Wp_MMf)*Wpmas ;

% 3. Calculando os SVD da projeção oblíqua ponderada Uu,Ss,Vv]=svd(Obli); % Determine a ordem dos valores singulares, %U1 e S1 % Plotando os valores singulares significativos SS = diag(Ss); H = length(SS); plot(H);

% Determine n por inspeção dos valores %singulares da gráfica n = input(' Ordem do sistema ');

% 4. DERTERMINE Gg

U1=Uu(:,1:n); S1=sqrt(SS(1:n)); U2=Uu(:,n+1:L*i); % colunas Li pois ri e uma matriz de (nxLi)

Gg=U1*diag(sqrt(S1(1:n)));

% 5. DETERMINE OS ESTADOS

xi=pinv(Gg)*Obli; Gg_1=Gg(1:L*(i-1),:); ximas=pinv(Gg_1)*Obli_mas;

%6. Compute S e os Residuias

D = 0.61 % Matriz do controlador

Uii=RR(m*i+1:m*i+m,:); Yii=RR((2*m+L)*i+1:(2*m+L)*i+L,:); Mii=Uii+D*Yii;

K=[xi;Mii]; J=[ximas;Yii]; Xs=J/K;

% 7. residuais T res=J-Xs*K;

%8. Determine A,B,C,D, (T1) com Co dado

%[m,nc]=size(C); S11=Xs(1:n,1:n); S21=Xs(n+1:n+m,1:n); S12=Xs(1:n,n+1:n+m); S22=Xs(n+1:n+m,n+1:n+m);

BB=S12*inv(eye(m)-D*S22) DD=S22*inv(eye(m)-D*S22); T1=inv(eye(L)+DD*D); CC=inv(T1)*S21; AA=S11+BB*D*S21; % matrizes da planta (AA, BB, CC, DD)

%9. DETERMINE Q S R DOS RESIDUAIS

clear Ss; cov = res*res'; % covariancia Qs = cov(1:n,1:n); Ss = cov(1:n,n+1:n+L); Rs = cov(n+1:n+L,n+1:n+L);

n % ordem do sistema LqL=dlqe(AA,eye(n),CC,Qs,Rs);

K=AA*LqL;

% ganho do filtro de Kalman da planta

% FIM DO ALGORIMO

Algoritmo 2 para caso em malha fechada N4SIDC

% Se precisa conhecer as matrizes do controlador (Ac,Bc,Cc,Dc) % ou as entradas impulso do controlador

% Dados de entrada:

% R : Sinal de referencia % U : Sinal de entrada da planta % Y : Sinal de saída da planta

% (A,B,C,D): matrizes do controlador

% Dados de saida

% As matrizes da planta

function [Ap,Bp,Cp,Dp,K]=N4SIDmf2(u,y,i);

% CONTROLADOR

A = [2.65 -3.11 1.75 -0.39; 1 0 0 0; 0 1 0 0; 0 0 1 0]; B = [1 0 0 0]'; C = [-0.4135 0.8629 -0.7625 0.2521]; D = 0.61;

$$\label{eq:mn} \begin{split} & [m,N] = size(u); \mbox{ if } m > N; \mbox{ } u = u'; \mbox{ end}; \mbox{ } [m,N] = size(u); \\ & [L,N] = size(y); \mbox{ if } L > N; \mbox{ } y = y'; \mbox{ end}; \mbox{ } [L,N] = size(y); \end{split}$$

% CONSTRUENCDO A MATRIZ Hc controlador

[L,d]=size(C); ri=C;

for k=2:i
 ri((k-1)*L+1:k*L,:)=ri((k-2)*L+1:(k-1)*L,:)*A;
end
rr=[D;ri*B];
[LL,m]=size(D);

Hc=zeros(LL*i,m*i); for k=1:i; Hc((k-1)*LL+1:LL*i,(k-1)*m+1:k*m)=rr(1:LL*(i-k+1),:); end

% CONSTRUENDO A MATRIZ M

Uf=RR(m*i+1:2*m*i,:); Yf=RR((2*m+L)*i+1:2*i*(m+L),:); Up=RR(1:m*i,:); Yp=RR(2*m*i+1:(2*m+L)*i,:); Wp=[Up;Yp];

Mf=Uf+Hc*Yf;

% Saidas Futuros

% Saidas perpendiculares futuras

Rfp=Yf - (Yf/Mf)*Mf;

% saidas perpendiculares pasadas Rpp=Wp-(Wp/Mf)*Mf;

% Projeçao obliqua

Obli=(Rfp/Rpp)*Wp;

% Calculando os SVD da projeção oblíqua

[Uu,Ss,Vv]=svd(Obli); SS = diag(Ss); H = length(SS); plot(H);

% Determine n por inspeção dos valores singulares da gráfica

n = input(' Ordem do sistema ') S1=sqrt(SS(1:n)); U1=Uu(:,1:n); % deten U2=Uu(:,n+1:L*i)'; % dete

% determine U1; % determine U2 = complemento ortogonal de U1

% Determine gam e gamm

gam=U1*diag(sqrt(S1(1:n))); gamm=gam(1:L*(i-1),:);

% As pseudo inversas gam_inv = pinv(gam); gamm_inv = pinv(gamm);

% Determine Zi e Zip

Upmas=RR(1:m*(i+1),:); Ypmas=RR(2*m*i+1:i*(2*m+L)+L,:); Wpmas=[Upmas;Ypmas];

% construendo MF_

 $\begin{array}{l} \mbox{Hc}_{=}\mbox{Hc}(1:m^{*}(i-1),1:\overline{L^{*}}(i-1)); \\ \mbox{Uf}_{=}\mbox{RR}(m^{*}(i+1)+1:2^{*}m^{*}i,:); \mbox{Yf}_{=}\mbox{RR}(i^{*}(2^{*}m+L)+L+1:2^{*}i^{*}(m+L),:); \end{array}$

Mf_=Uf_+Hc_*Yf_;

% Saidas pasadas

spi=[Up;Yp;Mf]; spip=[Upmas;Ypmas;Mf_];

Zi=Yf*pinv(spi)*spi; Zip=Yf_*pinv(spip)*spip;

```
% Determine K como
Lhs = U2*Zi;
Lhs = Lhs(:);
Rhs = zeros(2*i*(L*i-n)*(L+m),i*L*m);
for k=1:i
  Mk = Mf((k-1)*m+1:k*m,:);
  Uk = U2(:,(k-1)*L+1:i*L);
  ax = [1:(i-k+1)*L]; offs = [0:L*i:L*i*m-1];
  L1 = length(ax);
  L2 = length(offs);
  axt = kron(ones(1,L2),ax) + kron(offs,ones(1,L1));
  Rhs(:,axt) = Rhs(:,axt) + kron(Mk',Uk);
end
sol = Rhs\Lhs;
  % Computando a matriz K
  K = zeros(L*i,m*i);
  for k=1:i
     ax = [1:L];
   offs = [(k-1)*L:L*i:L*i*m-1];
     L1=length(ax);
     L2=length(offs);
     axt = kron(ones(1,L2),ax) + kron(offs,ones(1,L1));
     Kk = reshape(sol(axt),L,m);
     for ki = k:i
     K((ki-1)*L+1:ki*L,(ki-k)*m+1:(ki-k+1)*m) = Kk;
     end
   end
```

% Determine os estados

Xi = gam_inv * (Zi -K*Mf); Xip = gamm_inv * (Zip - K(1:L*(i-1),1:m*(i-1))*Mf_);

% Determine as matrizes da planta

Rhs = [Xi;Mf(1:m,:)]; Lhs = [Xip;Yf(1:L,:)];

XXX = Lhs/Rhs;

```
S11 = XXX(1:n,1:n);

S12 = XXX(1:n,n+1:n+m);

S21 = XXX(n+1:n+L,1:n);

S22 = XXX(n+1:n+L,n+1:n+m);

clear A; B; C;

C0= D;

clear D;
```

 $\begin{array}{l} tt = inv(eye(m) - C0^*S22);\\ Bp = S12^*tt;\\ Dp = S22^*tt;\\ Ap = S11 + B^*C0^*S21;\\ TLi = eye(L) + D^*C0;\\ Cp = TLi^*S21; \end{array}$

% Determine Q S R desde os residuais

```
res = Lhs - XXX*Rhs; % Residuais
```
cov = res*res'; Qs = cov(1:n,1:n); Ss = cov(1:n,n+1:n+L); Rs = cov(n+1:n+L,n+1:n+L);

% FIM DO ALGORIMO

Algoritmos N4sid Matlab (sistema determinístico)

T = input('Sample time ')n = input('Ordem do sistema ')ny = input('Numero de saída ')z=[y u];% ingrese os dados de entrada e saidaTR=30;% Numero maior que a ordem do sistemath=n4sid(z,n,ny,TR,[1,0,0],324480,T,'trace')[A,B,C,D]=th2ss(th);% recuperando as matrizes do sistema

Algoritmos PEM Matlab (sistema determinístico)

n = 4; % orden do sistema ny = 2; % numero de saidas z=[y u]; teta=canstart(z,n,ny,[1,0,0]) TH= pem(z,teta); [A,B,C,D]=th2ss(TH); % recuperando as matrizes do sistema

Algoritmos MOESP (Verhaegen) (sistema determinístico)

s=30; % Número maior que ordem do sistema % u = dados de entrada; % y = dados de saída [Sn, R]=dordpo(u,y,s); n=4; % orden do sistema [A,C]=dmodpo(R,n); [B,D]=dac2bd(A,C,u,y);

Algoritmos N4sid Matlab (sistema determinístico-estocástico)

T= 1;	% Sample time	
TR=20;	% Numero maior que a ordem do sistema	
n = 7;	% Ordem do sistema	
ny = 2;	% Numero de saidas	
z=[y u];	% Dados de saída e entrada	
th=n4sid(z,n,ny,TR,[1,1,1],324480,T,'trace');		

[A,B,C,D,Kn]=th2ss(th); %recuperando as matrizes do sistema

Algoritmos PEM Matlab (sistema determinístico estocástico)

n = 6; % orden do sistema ny = 2; % numero de saidas z=[y u]; % Dados de saída e entrada teta=canstart(z,n,ny,[1,1,1]) TH= pem(z,teta); [Aa,Ba,Ca,Da,Ka,x0]=th2ss(TH); %recuperando as matrizes do sistema

Algoritmos MOESP (Verhaegen) (sistema determinístico – estocástico)

s=20; % Número maior que ordem do sistema % u = dados de entrada; % y = dados de saída [Sn, R]=dordpo(u,y,s); n=7; % orden do sistema [A,C,K]=dmodpo(R,n); [B,D]=dac2bd(A,C,u,y);

ALGORITMO ARXS

% y=[y(1), y(2) ,...y(Ndat)]; pxNdat matriz % u=[u(1), u(2) ,...u(Ndat)]; mxNdat matriz

function [A,B,C,D]=arxs(ys,us)
[m,N]=size(us); if m > N; us=us'; end; [m,N]=size(us);
[L,N]=size(ys); if L > N; ys=ys'; end; [L,N]=size(ys);
clear m,N,L

z=[ys us];	
maxo=10;	% oderm maximal
na= maxo + 3;	% numero de deley das saidas
nb= maxo + 3;	% numero de deley nas entradas
[Sy,ny]=size(ys);	% ny numero de saidas

[N,nz]=size(z); y=z(:,1:ny); u=z(:,ny+1:nz); nu=nz-ny; % nu numero de entradas nk=1; m=arx(z,[na*ones(ny,ny) nb*ones(ny,nu) nk*ones(ny,nu)]); [a,b]=th2arx(m); % recupera os coeficientes [dum,nca]=size(a); [dum,ncb]=size(b); thy=-a(:,ny+1:nca)';

```
thu=b(:,nu+1:ncb)'
%nk=1
```

FIY=[]; FIU=[];

```
for kk=1:na
  FIY(:,(kk-1)*ny+1:kk*ny)=y(na-kk+1:N-kk,:);
end
for kk=1:nb
  FIU(:,(kk-1)*nu+1:kk*nu)=u(nb-kk+1:N-kk,:);
end
dalen=N-max([maxo,na,nb]);
Y1=zeros(dalen,maxo);
%Y1 e o vetor de k-pasos preditores
for kk=1:maxo;
  Ydum=FIY*thy + FIU*thu;
  Y1(:,(kk-1)*ny+1:kk*ny)=Ydum;
  FIY=[Ydum FIY(:,1:(na-1)*ny)];
  FIU=[zeros(dalen,nu) FIU(:,1:(nb-1)*nu)];
end
[U,S,V]=svd(Y1);
diag(S)
Dg=diag(S);
t=1:length(Dg);
figure(2);hold off;
plot(t,Dg);
hold on
plot(t,Dg,'k*');
%determine o valor de n por inspeção dos valores %singulares da gráfica
             Ordem do sistema ? ');
n = input('
X=Y1*V(:,1:n);
start=max([maxo,nb,na])+1;
yr=y(start:N,:);
C=X\yr
                               % estima minimos quadrados de C
e1=yr-X*C;
C=C':
Xnew=X(2:dalen,:);
Xr=X(1:dalen-1,:);
abk=[Xr u(start:N-1,:) e1(1:dalen-1,:)]\Xnew;
% Estimativa dos minimos quadraos de a , b e k
abk=abk';
A=abk(1:n,1:n);
B=abk(1:n,n+1:n+nu);
K=abk(1:n,n+nu+1:n+nu+ny);
D=zeros(ny,nu);
% FIM DO ALGORITMO
```

MATRIK DE HANKEL

function yh=hank(y,i);

[m,my]=size(y); if m > my; y=y'; end; [m,my]=size(y);

%i=10 %numero ingressado pelo usuário

M=length(y); % Ndat=M-2*i+1; % o numero de colunas da matriz de hankel yh=zeros(2*i*m,Ndat);

for s=1:2*Ndat; yh((s-1)*m+1:s*m,:)=y(:,s:s+Ndat-1); end yh=(1/sqrt(Ndat))*yh;