



Universidade Estadual de Campinas

LCSI

Faculdade de Engenharia Elétrica e de Computação

Departamento de Máquinas, Componentes e Sistemas Inteligentes

Laboratório de Controle e Sistemas Inteligentes

Metodos de Subespaços para Identificação de Sistemas: Propostas de Alterações, Implementações e Avaliações

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do título de

Mestre em Engenharia Elétrica
por

David Giraldo Clavijo

Prof. Dr. Gilmar Barreto

Orientador - FEEC/UNICAMP/SP

14 de novembro de 2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Giraldo, David

G441m Métodos de Subespaços para Identificação de Sistemas:
Propostas de Alterações, Implementações e Avaliações
David Giraldo. – Campinas, SP:
[s.n.], 2008.

Orientador: Gilmar Barreto.
Dissertação de Mestrado - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Identificação de Sistemas. 2. Sistemas MIMO.
3. Espaço de Estado. 4. Sistemas lineares – Identificação.
5. Modelagem de dados.
I. Barreto, Gilmar. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título

Titulo em Inglês: Subspace methods for systems identification: proposals of alterations, implementations and evaluations

Palavras-chave em Inglês: Systems Identification, Multivariable Systems, State Space models, Subspace Methods, Algorithms

Área de concentração: Automação

Titulação: Mestre em Engenharia Elétrica e de Computação

Banca examinadora: Celso Pascoli Bottura, Annabell Del Real Tamariz

Data da defesa: 14/11/2008

Programa de Pós Graduação: Engenharia Elétrica

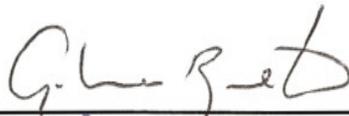
COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: David Giraldo Clavijo

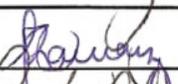
Data da Defesa: 14 de novembro de 2008

Título da Tese: "Métodos de Subespaços para Identificação de Sistemas: Propostas de Alterações, Implementações e Avaliações"

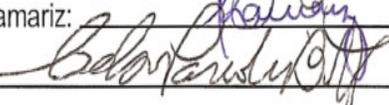
Prof. Dr. Gilmar Barreto (Presidente):



Profa. Dra. Annabell Del Real Tamariz:



Prof. Dr. Celso Pascoli Bottura:





Universidade Estadual de Campinas

LCSI

Faculdade de Engenharia de Elétrica e Computação
Departamento de Máquinas, Componentes e Sistemas Inteligentes
Laboratório de Controle e Sistemas Inteligentes

Tese : **Métodos de Subespaços para Identificação de Sistemas:
Propostas de Alterações, Implementações e Avaliações**

Autor : **David Giraldo Clavijo**

Orientador : **Prof. Dr. Gilmar Barreto**

Aprovada em 14 de novembro de 2008 pela banca examinadora:

- Prof. Dr. Gilmar Barreto (Presidente)
- Profa. Dra. Annabell Del Real Tamariz - CCT/UNEF
- Prof. Dr. Celso Pascoli Bottura - FEEC/UNICAMP

Agradecimentos

Ao meu orientador, Prof. Gilmar Barreto, sou grato pela dedicação na orientação, pela amizade e oportunidade.

Ao Prof. Celso Pascoli Bottura pela ajuda neste trabalho, conselhos e apoio.

Aos colegas César, Mateus, Neliton, André, Fernando e Marcus pela amizade, sugestões e auxílios na realização desta tese.

Ao meu quase irmão Mauricio e toda sua família, pela grande amizade e apoio sentimental.

Aos meus amigos os Memos, Leo, Luchis e xNata pelo apoio e companhia durante esta jornada.

À CAPES, pelo apoio financeiro.

*Cuando te sientas con una hermosa chica por dos horas,
parece como si hubieran pasado dos minutos.
Cuando te sientas en una estufa caliente por dos minutos,
parece como si hubieran pasado dos horas.
¡Eso es la Relatividad!*

Albert Einstein

*Aos meus pais, María José e Francisco
irmãs, María José e Nathalia
meu sobrinho, e minha afilhada,
avós, tios e amigos*

Resumo

Este estudo apresenta os fundamentos teóricos para modelagem de dados multivariáveis no espaço de estado através de Métodos de Subespaços para Identificação de sistemas lineares invariantes no tempo, discretos no tempo. O trabalho contém alguns conceitos básicos de sistemas dinâmicos, um pouco da história e os elementos de identificação de sistemas, modelos no espaço de estado e modelos estendidos no espaço de estado. Dois Métodos de Subespaços são analisados e tratados, o Multivariable Output-Error State-sPace (MOESP) e o Numerical algorithm for Subspace State-Space System IDentification (N4SID). Modificações nos seus algoritmos são propostas e implementadas. Experimentos com *benchmarks* são realizados para exemplificar o procedimento de identificação por subespaços e para avaliar os algoritmos modificados.

Palavras-chave: Identificação de Sistemas Multivariáveis, Modelos no Espaço de Estado, Métodos de Subespaços, Algoritmos MOESP e N4SID.

Abstract

This study presents the theoretical foundations of multivariable data modeling in state space by Subspace Methods for Systems Identification of linear time invariant, discrete time, systems. The work contains some basic concepts of dynamic systems, a little of history and the elements of systems identification, state space models and extended state space models. Two Subspace Methods are analyzed and applied, Multivariable Output-Error State-sPace (MOESP) and Numerical algorithm for Subspace State-Space System IDentification (N4SID). Some modifications then are proposed and implemented. Experiments with benchmarks are realized to show the procedure of Identification by subspace and to evaluate the modified algorithms.

Keywords: Multivariable System Identification, State Space models, Subspace Methods, MOESP and N4SID algorithms.

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xv
Lista de Símbolos	xvii
Trabalhos Publicados Pelo Autor	xix
PREFÁCIO	xxi
PREFÁCIO	xxi
1 MODELAGEM DE DADOS	1
1.1 Sistemas Dinâmicos	1
1.2 Identificação de Sistemas	4
1.2.1 Elementos da Identificação de Sistemas	4
1.2.2 História	6
1.2.3 Períodos no desenvolvimento da Identificação de Sistemas	9
1.2.4 Novos métodos	11
1.3 Modelo no Espaço de estado	12
1.3.1 Resposta ao impulso de sistema monovariável	15
1.3.2 Resposta ao impulso de sistema multivariável	16
1.3.3 Operador de Hankel	17
1.4 Modelo Estendido no Espaço de Estado	21
2 MÉTODOS DE SUBESPAÇOS	25
2.1 Introdução	25
2.2 Seqüência de Estados \mathbf{X}_f	34
2.2.1 Algoritmos de intersecção	34
2.2.2 Algoritmos de projecção	35
2.3 Matrizes do Sistema	37
2.3.1 Algoritmos que usam os estados	37
2.3.2 Algoritmos que usam a matriz de observabilidade estendida	39
2.4 Método MOESP	40
2.5 Método N4SID	48

2.6	Conclusões	54
3	MÉTODOS DE SUBESPAÇOS: EXPERIMENTAÇÕES	57
3.1	Introdução	57
3.2	Benchmark 1	58
3.2.1	Introdução	58
3.2.2	Algoritmos	59
3.2.3	Parâmetros de identificação	61
3.2.4	Resultados	64
3.2.5	Validação do modelo	68
3.2.6	Conclusões	72
3.3	Benchmark 1 com ruído	78
3.3.1	Introdução	78
3.3.2	Algoritmos	78
3.3.3	Parâmetros de identificação	78
3.3.4	Resultados	80
3.3.5	Validação do modelo	83
3.3.6	Conclusões	85
3.4	Benchmark 2	86
3.4.1	Introdução	86
3.4.2	Algoritmos	86
3.4.3	Parâmetros de identificação	88
3.4.4	Resultados	88
3.4.5	Validação do modelo	88
3.4.6	Conclusões	88
4	COMENTÁRIOS E CONCLUSÕES	89
A	PROGRAMAS COMPUTACIONAIS	91
A.1	Algoritmo MOESP baseado em Katayama:	91
A.2	Algoritmo MOESP modificado:	93
A.3	Algoritmo N4SID modificado:	94
B	ÁLGEBRA LINEAR	97
B.1	Projeções ortogonais:	97
B.2	Projeções oblíquas:	98
B.3	Prova equação 2.14:	98
B.4	Prova equação 2.31:	101
C	EXPERIMENTOS	103
C.1	Benchmark 1 com ruído	103
C.1.1	Validação do modelo	103
C.2	Benchmark 2	106
C.2.1	Parâmetros de identificação	106
C.2.2	Resultados	106

C.2.3 Validação do modelo	109
Referências bibliográficas	115

Lista de Figuras

1.1	Modelo de um Sistema Dinâmico determinístico	2
1.2	Um diagrama de fluxo da identificação de sistemas.	7
1.3	Resposta à entrada zero de um sistema LTI	19
2.1	Diferentes caminhos que seguem os Métodos Clássicos e os Métodos de Subespaços.	28
2.2	Método MOESP referência.	42
2.3	Método MOESP baseado em Katayama.	45
2.4	Método MOESP modificado.	47
2.5	Método N4SID referência.	49
2.6	Método N4SID baseado em Katayama.	53
2.7	Método N4SID modificado.	55
3.1	Seqüência de dados de entrada.	60
3.2	Seqüência de dados de saída.	60
3.3	Valores singulares para o MOESP de Katayama.	62
3.4	Gráfico de barras dos valores singulares para o MOESP de Katayama.	62
3.5	Gráfico de barras dos valores singulares para o MOESP modificado.	63
3.6	Valores singulares para o N4SID referência.	63
3.7	Gráfico de barras dos valores singulares para o N4SID modificado.	64
3.8	Gráfico de encaixe do modelo para o algoritmo MOESP 1.	72
3.9	Zoom no gráfico de encaixe do modelo para o algoritmo MOESP 1.	73
3.10	Gráfico de encaixe do modelo para o algoritmo MOESP 2.	73
3.11	Zoom no gráfico de encaixe do modelo para o algoritmo MOESP 2.	74
3.12	Gráfico de encaixe do modelo para o algoritmo MOESP 3.	74
3.13	Zoom no gráfico de encaixe do modelo para o algoritmo MOESP 3.	75
3.14	Gráfico de encaixe do modelo para o algoritmo N4SID 1.	75
3.15	Zoom no gráfico de encaixe do modelo para o algoritmo N4SID 1.	76
3.16	Gráfico de encaixe do modelo para o algoritmo N4SID 2.	76
3.17	Zoom no gráfico de encaixe do modelo para o algoritmo N4SID 2.	77
3.18	Seqüência de dados de entrada.	79
3.19	Seqüência de dados de saída.	79
3.20	Gráfico de barras dos valores singulares para o MOESP de Katayama.	80
3.21	Seqüência de dados de entrada.	87
3.22	Seqüência de dados de saída.	87

B.1	Projeção ortogonal.	98
B.2	Projeção oblíqua.	99
C.1	Gráfico de encaixe do modelo para o algoritmo MOESP 1.	103
C.2	Gráfico de encaixe do modelo para o algoritmo MOESP 2.	104
C.3	Gráfico de encaixe do modelo para o algoritmo MOESP 3.	104
C.4	Gráfico de encaixe do modelo para o algoritmo N4SID 1.	105
C.5	Gráfico de encaixe do modelo para o algoritmo N4SID 2.	105
C.6	Gráfico de barras dos valores singulares para o MOESP de Katayama.	106
C.7	Gráfico de encaixe do modelo para o algoritmo MOESP 1.	112
C.8	Gráfico de encaixe do modelo para o algoritmo MOESP 2.	113
C.9	Gráfico de encaixe do modelo para o algoritmo MOESP 3.	113
C.10	Gráfico de encaixe do modelo para o algoritmo N4SID 1.	114
C.11	Gráfico de encaixe do modelo para o algoritmo N4SID 2.	114

Lista de Tabelas

1.1	Conferencias IFAC, com os pesquisadores mais destacados e temas principais de cada uma.	9
3.1	Tempos (em segundos) para os algoritmos MOESP com dados sem ruído.	67
3.2	Tempos (em segundos) para os algoritmos N4SID com dados sem ruído.	68
3.3	Erro de validação (em porcentagem) para os algoritmos MOESP com dados sem ruído.	71
3.4	Erro de validação (em porcentagem) para os algoritmos N4SID com dados sem ruído.	71
3.5	Erro de validação (em porcentagem) para os algoritmos MOESP com dados com ruído.	85
3.6	Erro de validação (em porcentagem) para os algoritmos N4SID com dados com ruído.	85
C.1	Erro de validação (em porcentagem) para os algoritmos MOESP.	112
C.2	Erro de validação (em porcentagem) para os algoritmos N4SID.	112

Lista de Símbolos

δ	-	Delta de Kronecker
H	-	Matriz de Hankel
\mathcal{T}	-	Matriz de Toeplitz
\mathcal{O}	-	Matriz de Observabilidade
\mathcal{C}	-	Matriz de Atingibilidade
\mathcal{C}^{\leftarrow}	-	Matriz de Atingibilidade reordenada
\dagger	-	Pseudoinversa
MOESP 1	-	algoritmo MOESP referência
MOESP 2	-	algoritmo MOESP baseado em Katayama
MOESP 3	-	algoritmo MOESP modificado
N4SID 1	-	algoritmo N4SID referência
N4SID 2	-	algoritmo N4SID modificado

Trabalhos Publicados Pelo Autor

1. D. Giraldo, G. Barreto. "Subspace Methods: Performance Analysis with benchmarks." 7th Brazilian Conference on Dynamics, Control and Applications, DINCON 2008.

PREFÁCIO

Modelos matemáticos que representem a realidade o mais fielmente possível são necessários na indústria moderna. A teoria de identificação de sistemas provê métodos para aproximar sistemas reais usando conjuntos de modelos baseados em dados de entrada - saída experimentais, isto com a finalidade de entender o comportamento de sistemas, modelar matematicamente sistemas e/ou sinais, prever, controlar, otimizar, diagnosticar, etc. A estimação de parâmetros para determinar modelos mediante métodos clássicos, por exemplo o método de erro de predição, tem como objetivo minimizar a diferença entre a saída real e a saída estimada do processo; mas estes métodos requerem um conhecimento do sistema a ser identificado, além de um elevado esforço computacional para sistemas com múltiplas entradas - múltiplas saídas (MIMO).

O problema principal tratado na identificação de sistemas no espaço de estado pode ser definido como: Dado um grande número de medidas de entradas, u_1, u_2, \dots, u_k , e de saídas, y_1, y_2, \dots, y_k , geradas por um sistema desconhecido, determinar a ordem do sistema, n , e as matrizes A, B, C e D , a menos de uma transformação de similaridade.

Os Métodos de Subespaço têm provado ser uma excelente alternativa aos métodos de erro de predição clássicos para a identificação de sistemas lineares multivariáveis de ordem elevada; estes métodos permitem a determinação direta dos estados do sistema através de técnicas de álgebra linear numérica.

Podemos agora fazer a seguinte pergunta: por que a Identificação por Subespaços é requerida se temos uma teoria de identificação de sistemas clássica bem estabelecida, como os métodos de erro de predição? Uma das principais vantagens para Identificação por Subespaços está na facilidade que estes métodos apresentam, em relação aos métodos tradicionais, de trabalhar com sistemas de múltiplas entradas e múltiplas saídas (MIMO). Além disso, os métodos de erro de predição requerem informação a priori do sistema a ser identificado, e maior esforço computacional nas suas implementações. Métodos de Subespaços não são métodos iterativos onde tem-se que otimizar uma função de custo. São métodos baseados em álgebra linear numérica, onde não precisamos tratar com problemas de otimização não lineares.

Este estudo apresenta os fundamentos teóricos para modelagem de dados multivariáveis no es-

paço de estado através de Métodos de Subespaços para Identificação de sistemas lineares invariantes no tempo, discretos no tempo. Contém alguns conceitos básicos de sistemas dinâmicos, um pouco da história e os elementos da Identificação de Sistemas, modelos no espaço de estado e modelos estendidos no espaço de estado. Dois Métodos de Subespaços são analisados e tratados, o MOESP (Multivariable Output-Error State-space) e o N4SID (Numerical algorithm for Subspace State-Space System Identification). Modificações nos seus algoritmos são propostas e implementadas. Experimentos com *benchmarks* são realizados para avaliar tanto os algoritmos convencionais como os propostos. Neste trabalho apresentamos os códigos dos algoritmos modificados aqui explicados. Espera-se que esta tese seja útil para pesquisadores em modelagem, controle e processamento de sinais de sistemas complexos.

No Capítulo 1 introduzimos alguns conceitos básicos de Modelagem de Dados, como sistemas dinâmicos, elementos e um pouco da história dos períodos do desenvolvimento da identificação de sistemas, e alguns dos novos métodos; aqui, nestes novos métodos aparecem os métodos de subespaços no espaço de estado, tema principal desta tese. Antes de tratar em detalhe os Métodos de Subespaços no Capítulo 2, ainda no Capítulo 1 também temos informação sobre o modelo no espaço de estado, e finalmente sobre o modelo estendido no espaço de estado, utilizados pelos Métodos de Subespaços que são tratados em detalhes ao longo desta tese.

Depois de ter apresentado toda uma base teórica dos Métodos de Subespaços no Capítulo 1, podemos analisar o teorema principal dos Métodos de Subespaços, no Capítulo 2. Neste Capítulo, a importância do estado nos métodos de subespaços é explicada e depois apresentada uma forma para obter este estado, para em seguida mostrar uma maneira de obter as matrizes do sistema. Finalmente os métodos MOESP, N4SID, são tratados; modificações para implementações destes algoritmos são apresentadas. Assim concluímos o Capítulo 2 que trata dos Métodos de Subespaços.

O Capítulo 3 tem como objetivo mostrar, com base no Capítulo 2, a parte de experimentos realizados com os diversos algoritmos MOESP (referência, baseado no Katayama e Modificado) e N4SID (referência e modificado). Vários exemplos foram experimentados, testados e os resultados dos experimentos realizados são apresentados e analisados. Os modelos utilizados para realização dos experimentos, *benchmarks*, são também apresentados.

Esta tese é finalizada com nossas conclusões e considerações sobre os algoritmos tratados e propostas para continuação deste estudo. As referências bibliográficas apresentadas dão um panorama do estado da arte em Métodos de Subespaços.

Capítulo 1

MODELAGEM DE DADOS

1.1 Sistemas Dinâmicos

Vamos começar com uma definição da palavra *sistema*; o dicionário *Merriam-Webster* define um sistema como:

- Um sistema é um grupo de elementos interdependentes ou que interatuam regularmente, e que formam um todo.

Assim, um sistema é uma combinação de componentes que atuam juntos e realizam um objetivo determinado; um sistema não necessariamente tem que ser físico. Temos diferentes tipos de sistemas, e, de fato, quase tudo com o que temos contato em nossa vida é um sistema ou forma parte de um sistema. Um sistema pode ser qualquer objeto no qual variáveis de diferentes tipos interagem e produzem sinais observáveis. Estes sinais observáveis são conhecidos como *saídas*; os estímulos externos que afetam o sistema, e que podem ser controlados, são chamados de *entradas*. Outros estímulos que afetam o sistema são as *perturbações*, que se apresentam de forma aleatória e incontrolada. Agora, adicionando a palavra *dinâmico*, temos que um sistema dinâmico é aquele cujo modelo matemático é constituído por equações diferenciais ou a diferenças, onde o tempo é uma variável independente; ou seja, é aquele em que alguns dos seus aspectos variam com o tempo; [4], [31], [34].

A Figura 1.1 mostra um diagrama esquemático de um sistema dinâmico com variáveis u_k , y_k e v_k , que representam a entrada, saída e perturbação para um instante k , respectivamente. O sistema é acionado por variáveis de entrada e perturbações. O usuário pode observar as entradas e saídas, u_k e y_k , respectivamente, mas não as perturbações, v_k ; e pode manipular diretamente as entradas, mas não as saídas nem as perturbações. Os dados de entrada e de saída são variáveis que fornecem uma informação útil sob o comportamento do sistema tratado, mesmo não se conhecendo a estrutura interna

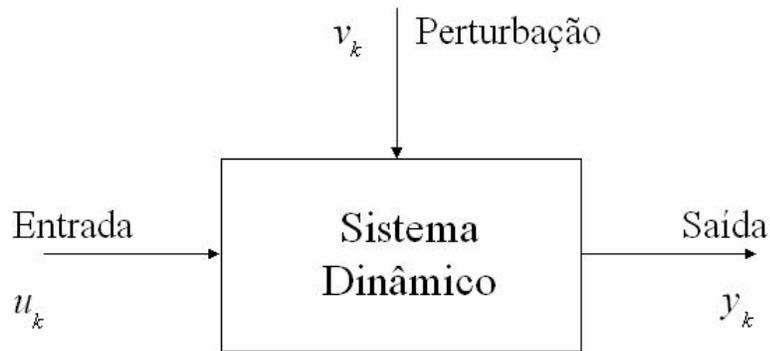


Fig. 1.1: Modelo de um Sistema Dinâmico determinístico

do sistema. Assim, podem-se construir modelos que descrevem as dinâmicas do sistema de interesse com ajuda dos dados de entrada - saída observados; [24].

Os modelos dos sistemas dinâmicos podem ser de diversos tipos, incluindo os seguintes:

- Modelos mentais, intuitivos ou verbais
- Gráficos ou tabelas
- Modelos matemáticos

Podemos ter assim, modelos mentais, intuitivos, verbais, ou aproximações graficamente orientadas como gráficos e tabelas, mas neste trabalho estamos interessados nos últimos, nos modelos matemáticos. Um modelo matemático pode ser visto como um algoritmo ou conjunto de equações que juntamente com um grupo de valores dados, representa o comportamento de um sistema, processo ou fenômeno. Tais modelos são dados por equações que descrevem o comportamento dinâmico de um sistema em função do tempo; estas equações podem ser diferenciais (para o caso de tempo contínuo) ou em diferenças (para tempo discreto). Modelos matemáticos de sistemas dinâmicos existem em todas as disciplinas científicas, e são usados para simulações, treinamento de operador, análise, monitoração, detecção de falhas, previsão, otimização, projeto de sistemas de controle, controle de qualidade, etc. O desenvolvimento de um modelo matemático para um sistema do mundo real pode ser uma tarefa muito complexa; em casos em que a dinâmica do sistema não é bem entendida, uma série de experimentos tem que ser realizados para coletar dados que serão processados usando várias técnicas. Podemos falar, em geral, de sistemas com entradas e saídas, mas têm-se sistemas onde os sinais de entrada podem não existir; este conjunto de dados de saída ordenado no tempo é conhecido

como série temporal, e podem ser, dependendo do tipo de amostragem, contínuas ou discretas. [10]; [3]. Para nosso trabalho apenas consideraremos os sistemas com conjuntos de dados de entradas e saídas discretas.

Modelos dinâmicos para predição e controle incluem funções de transferência, modelos no espaço de estado, modelos de séries temporais, etc, que são parametrizados em termos de um número finito de parâmetros. Assim, estes modelos dinâmicos são conhecidos como modelos *paramétricos*, e requerem a escolha de uma possível estrutura de modelo, de um critério de ajuste de parâmetros, e da estimação dos parâmetros que melhor ajustem o modelo aos dados experimentais. Também são usados modelos *não paramétricos* como curvas de respostas ao impulso, respostas em frequência, análise de correlação, função densidade espectral, etc. Neste tipo de modelos não existe um número finito de parâmetros; [17]. Nosso trabalho limita-se a modelos paramétricos.

Existem diversas aproximações para gerar um modelo de um sistema. Uma delas, por exemplo, pode ser começando pelos princípios básicos (como leis físicas) que expliquem detalhadamente os mecanismos essenciais subjacentes do fenômeno observado; isto é, dividir o sistema em subsistemas, os quais têm propriedades bem conhecidas de experiência previa. Esta é uma aproximação analítica, conhecida como modelos "*caixa branca*"; neste tipo de modelo tem-se conhecimento necessário *a priori* do sistema, baseado em leis físicas que representam o comportamento do sistema dinâmico. Por exemplo, um processo que é projetado por um operador, onde ele sabe como se efetua a transformação de entradas em saídas; [33]. Para os engenheiros, no entanto, este tipo de tratamento não é realmente útil. A razão é que os engenheiros não estão realmente interessados no modelo exato como tal, mas sim nas aplicações potenciais dele na engenharia. Nesta perspectiva, um modelo matemático é só um passo no projeto global de um sistema. A qualidade de um modelo é ditada pelo objetivo final para o qual ele serve. A incerteza de um modelo é permitida sempre que a robustez total do sistema seja assegurada. Assim, utilizando identificação de sistemas, pode-se obter uma aproximação experimental. Para isto, experimentos são executados no sistema; medidas e observações do sistema são coletadas, e depois o sistema é modelado usando um modelo conhecido como modelo "*caixa preta*"; [5],[20]. Neste tipo de modelos nenhum conhecimento prévio (*a priori*) do sistema dinâmico é conhecido. Ou seja, não sabemos os elementos que compõem o sistema ou processo, só os sinais de entrada e saída. O problema consiste em escolher uma estrutura adequada para o modelo, e com isto obter a maior informação possível do sistema. Depois de escolher a estrutura, o passo a seguir é a estimação dos parâmetros do modelo a partir dos dados medidos do sistema dinâmico; e assim, obter um modelo paramétrico; [24]; [10]; [9]. Nos vamos trabalhar com este tipo de modelos caixa preta, onde não precisamos conhecer o comportamento interno do sistema, só um conjunto de dados de entrada e saída. Tem-se casos nos quais as equações do sistema são conhecidas, mas alguns parâmetros não, e tem-se que estimar usando algum método de estimação de parâmetros; é uma

mistura entre os modelos de caixa branca e preta, chamados modelos "*caixa cinza*".

Um modelo complexo vai levar a um projeto sofisticado, no entanto um modelo simples vai deteriorar o desempenho (*performance*) e robustez global da implementação final. Assim, engenheiros tipicamente usam técnicas de identificação de sistemas para construir seus modelos. Este é o campo de modelagem de sistemas dinâmicos de dados experimentais: experimentos são executados em um sistema, certa classe de modelos parametrizados é predefinida pelo usuário e valores numéricos apropriados são associados aos parâmetros para representar o mais próximo possível os dados gravados. Finalmente tem-se um passo de validação, no qual o modelo é testado com dados experimentais que não foram usados no experimento de identificação do sistema, para analisar se a representação obtida é apropriada. Se não é o caso, pode-se considerar uma estrutura de modelo mais complexa, ou estimar seus parâmetros novamente, etc.

1.2 Identificação de Sistemas

O termo *Identificação de Sistemas* foi inventado por Lotfi Zadeh em 1962. Ele definiu a Identificação de Sistemas como; [23]:

"Identificação é a determinação, baseado nas entradas e nas saídas, de um sistema dentro de uma classe específica de sistemas, o qual é equivalente ao sistema sob o teste"

Isto vai ser, para nosso caso, construir modelos matemáticos de sistemas dinâmicos baseados nos dados de entrada e saída observados. Esta definição é altamente orientada aos sistemas, e não reflete as fortes características estatísticas das técnicas clássicas de identificação de sistemas. Não obstante, o termo foi adotado e tornou-se terminologia padrão no âmbito de controle. A identificação de sistemas representa a interface entre o mundo dos modelos matemáticos e o mundo real. Qualquer aplicação de teoria de controle ao mundo real tem que, de uma ou outra forma, tratar com problemas de identificação de sistemas.

1.2.1 Elementos da Identificação de Sistemas

Identificação de sistemas é uma metodologia desenvolvida principalmente na área do controle automático, pela qual podemos escolher o melhor modelo (ou melhores modelos) de um conjunto de modelos candidatos, baseados nos dados observados de entrada - saída do sistema. Daqui, e da definição do Zadeh, temos que o problema de identificação de sistemas pode ser especificado por três elementos principais; [17]; [25]; [24]; [1]:

- Um conjunto de dados \mathcal{D} obtido com medidas das entradas - saídas.

- Um conjunto, classe, ou estrutura de modelos M , que contém os modelos candidatos.
- Uma equivalência que é frequentemente definida em termos de um critério, ou função objetivo, L para a seleção do melhor modelo (ou melhores modelos), ou uma regra para avaliar os modelos candidatos, baseados nos dados. Este critério é função da saída do processo e da saída do modelo.

Quando a equivalência é definida em termos da função objetivo, o problema de identificação fica como um simples problema de otimização: encontrar um modelo M_o pertencente ao conjunto de modelos M , tal que a função objetivo seja minimizada. Mas na identificação ficam perguntas como:

- Tem uma única solução?
- Como evitar parâmetros redundantes?
- Que tipo de análise deve ser feita aos resultados do experimento para obter o desejado?
- A solução obtida é devido à influência dos sinais de entrada escolhidos?
- Qual é complexidade apropriada para descrever o comportamento do sistema observado?

Para o primeiro elemento da identificação de sistemas, os dados de entrada - saída \mathcal{D} , temos que estes são coletados mediante experimentos. Neste caso, precisa-se projetar o experimento decidindo quais sinais de entrada (ou sinais de prova), e quais sinais de saída devem ser medidos, o intervalo de amostragem, etc, para que desse modo as características do sistema sejam refletidas nos dados observados. É sabido que simplificações significativas nos cálculos podem ser conseguidas escolhendo sinais de entrada de um tipo especial, por exemplo, funções impulso, degrau, ruído branco, etc. Além disso, para obter dados úteis para a identificação do sistema, precisa-se de alguma informação *a priori*, ou algum conhecimento físico, sobre o sistema. Por exemplo, têm casos onde não se podem realizar experimentos em malha aberta (*open loop*) devido a diversas razões, como segurança, algumas causas técnicas e / ou econômicas, etc, então só podem ser utilizados dados medidos sob condições de operação normal; [1].

Modelos e sistemas podem ser divididos em muitas classes, como lineares ou não lineares, variantes ou invariantes no tempo, de tempo contínuo ou discreto, etc. A escolha da classe de modelos M a utilizar é um problema difícil em identificação de sistemas, mas usualmente uma variedade de classes de sistemas discretos no tempo lineares invariantes no tempo (LTI) são utilizados. Um dos problemas mais difíceis é encontrar uma boa estrutura de modelos, ou ajustar a ordem dos modelos, baseado nos dados de entrada - saída disponíveis; [24]; [9]. Neste documento vamos trabalhar com modelos LTI.

É bem conhecido que sistemas reais de interesse são não lineares, variantes no tempo, e podem conter atraso, e algumas variáveis ou sinais de grande importância podem não ser mensuráveis. Também é certo que os sistemas LTI são os mais simples e importantes na classe de sistemas dinâmicos usados na prática e na literatura. Com eles têm-se modelos ideais, e as experiências mostram que eles podem aproximar muito bem os processos industriais. Adicionalmente, os projetos de controle baseados em modelos LTI geralmente levam a bons resultados. Além, deve-se enfatizar que a identificação de sistemas é uma técnica de aproximação de sistemas reais por meio de modelos, pois não existe um sistema "ideal" nas aplicações práticas [17]; [1].

O próximo passo é obter um modelo no conjunto de modelos M , pelo qual os dados experimentais sejam explicados da melhor maneira. Para isto, precisamos um critério para medir a diferença entre o modelo e o sistema real, assim o critério deve ter significado físico e ser simples para manejar matematicamente.

Para obter o melhor modelo $M_o \in M$, precisamos de:

- Uma condição para a existência de um modelo que minimize o critério.
- Um algoritmo para computar o modelo.
- Um método para validar o modelo.

O modelo estimado deve ser avaliado a partir de um novo conjunto de dados e analisado se consegue atingir o objetivo desejado; em síntese, a validação do modelo é para determinar se o modelo identificado deve, ou não, ser aceito como descrição apropriada que expresse as dinâmicas do sistema. Assim, a validação de modelo depende da forma como o modelo vai ser usado, em informação *a priori* do sistema, aptidão do modelo aos dados reais, etc. Têm-se diversos testes de validação de modelos como análise estatística de resíduos, simples inspeção, por meio de simulações, entre outros; [1]. Além disso, quando se projeta um sistema de controle, deve-se ter em consideração o desempenho do sistema obtido pela identificação; se este não é satisfatório, temos que voltar alguns passos na identificação, e procurar outro modelo, outra classe de modelos, ou fazer um reprojeto do experimento, etc. Na figura 1.2 temos um fluxograma do procedimento de identificação.

1.2.2 História

Uma notável quantidade de técnicas e teorias de identificação de sistemas têm bases em fundamentos estatísticos. Pessoas como K. F. Gauss (1809) e R. A. Fisher (1912) e muitos outros famosos matemáticos e estatísticos têm implantado os fundamentos para a identificação de sistemas. Uma das primeiras perspectivas estatísticas no campo da identificação de sistemas é possivelmente o artigo de Åström e Bohlin, que desenvolveram um método de máxima verossimilhança para modelos

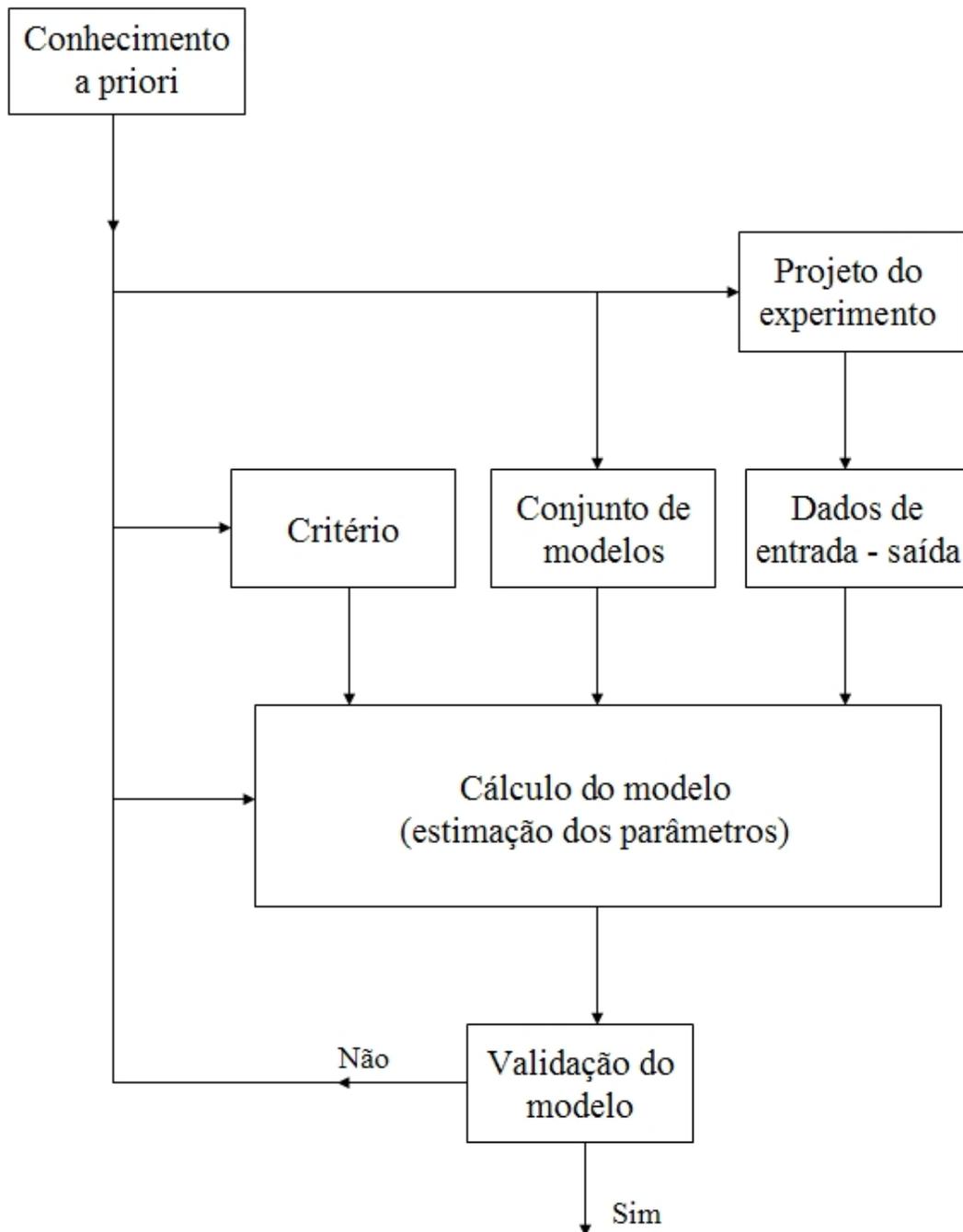


Fig. 1.2: Um diagrama de fluxo da identificação de sistemas.

ARMAX (no ano 1965). A Identificação de Sistemas é uma técnica bem conhecida para desenvolver modelos matemáticos baseados nas seqüências de dados de entrada e saída. Estas técnicas têm sido aplicadas desde que tem-se controle por realimentação. Antes, existiam métodos como a análise de resposta transitória para ajustar controladores PID, análise de freqüência junto com sínteses clássicas de Bode-Nichols, etc; mas foi só com a teoria de controle moderno, em torno de 1960, com o uso explícito de modelos paramétricos, que começaram atividades mais substanciais em estimação de sistemas. Uma variedade de aproximações para estimação de parâmetros em modelos de sistemas dinâmicos e sistemas lineares em geral, teve lugar no começo dos anos 60. Pode-se dizer que a identificação de sistemas foi estabelecida como um campo de pesquisa certificado dentro da área de controle automático em meados dos anos 60: com um artigo de Eykhoff, em 1966, no terceiro congresso da IFAC (International Federation of Automatic Control), em Londres. Um ano depois, 1967, o primeiro simpósio IFAC em identificação de sistemas foi organizado em Praga. Este é na atualidade o simpósio mais antigo das séries de simpósios IFAC. Desde então, a identificação de sistemas tem sido estabelecida como um campo do controle automático com sessões regulares nas reuniões gerais de controle como os congressos da IFAC e CDC (Conference on Decision and Control), com edições especiais em jornais de controle, etc. O número de artigos em identificação de sistemas relacionado a problemas de controle em jornais, conferências e palestras, no ano 1996 era da ordem de 10^5 [23].

Por outro lado, raramente a identificação de sistemas tem apresentado algo bem definido, um problema aberto, que tenha reunido muitos pesquisadores com um interesse comum como o chamado problema de erro na variável que foi tratado e estudado por Kalman (1983) (o problema de erro na variável é um problema onde tanto os sinais de entrada como de saída são medidos com erros de caráter desconhecido). A primeira década de identificação de sistemas como campo estabelecido, que pode-se tomar desde meados dos 1960:s até meados dos 1970:s, consistiu em um período de intenso desenvolvimento e de grande interesse geral na área. Não obstante, incluso no simpósio em identificação de 1970, também em Praga, Åström e Eykhoff acabaram seu artigo com a frase: "*Após o IFAC em Praga, 1970, ainda há muito trabalho por ser feito*", [23], o que incitava a continuar pesquisas na área.

No entanto, durante os anos 1980:s, teve-se pouco interesse em identificação de sistemas; isto pode-se notar no número de sessões de simpósios internacionais e no número de artigos (*papers*) publicados nos jornais. Pode-se exemplificar mais claramente pelo fato que a IFAC's *Technical Board*, queria mudar os simpósios, tanto criando novos, como interrompendo alguns. E neste ponto, o simpósio de identificação de sistemas foi um dos indicados para ser suprimido, pois o interesse tinha-se movido para o controle adaptativo; [23]. Depois, no fim dos anos 1980 e no começo dos 1990, ocorreu um renascimento apreciável no desenvolvimento de técnicas para identificação de sistemas. Tem-se na tabela 1.1 algumas das conferências IFAC, pesquisadores destacados e temas principais; observa-

Lugar	Ano	Autor destacado	Tema principal
Prague	1967	Strejc e Peterka	Novos métodos
Prague	1970	Strejc e Peterka	Novos métodos e necessidade de unificação
The Hague	1973	Eykhoff	Comparações de métodos
Tiblisi	1976	Rajbmann	Identificação em malha fechada (<i>closed loop</i>)
Darmstadt	1979	Isermann	Uso pretendido do modelo (<i>intended use of the model</i>)
Washington	1982	Bekey e Saridis	Aplicações não técnicas
York	1985	Barker e Young	Adaptação, identificabilidade (<i>identifiability</i>)
Beijing	1988	Chen e Liu	Adaptação, aplicações em processamento de sinais
Budapest	1991	Bányász e Keviczky	Identificação para controle, novos modelos de ruído
Copenhagen	1994	Blanke e Söderström	Técnicas de subespaços, novos modelos tipo não linear

Tab. 1.1: Conferencias IFAC, com os pesquisadores mais destacados e temas principais de cada uma.

se que a Europa tem uma dominância no tema da identificação; isto pelo fato que os pesquisadores europeus têm sido mais ativos na área de identificação de sistemas.

1.2.3 Períodos no desenvolvimento da Identificação de Sistemas

Assim, fazendo uma grande simplificação podemos ter os seguintes períodos no desenvolvimento da identificação de sistemas:

- ... - 1960: desenvolvimento das raízes estatísticas
- 1960 - 1970: propagação dos métodos de identificação
- 1970 - 1985: consolidação da teoria e prática da identificação de sistemas
- 1985 - ...: surgimento de novas idéias sem fundamentos (raízes) estatísticos

O primeiro período pode-se tomar desde Gauss (1809) e vai até 1960, quando os modelos paramétricos explícitos tornaram-se de principal interesse na comunidade de controle; durante este tempo a identificação de sistemas recebeu alguns conceitos essenciais de estatística. Em particular, a aplicação de regressões lineares e do método dos mínimos quadrados nos modelos AR. Além disso, também formam base da identificação de sistemas os conceitos, ferramentas e análise para regressões não lineares e o método da máxima verossimilhança (Fisher 1912, Wald 1949, Cramér 1946). A aproximação estocástica foi desenvolvida no começo dos anos 1950 (Robbins e Monro 1951) e depois tornou-se uma importante fonte de inspiração para as técnicas de identificação *on-line* (recursivas).

Vários aspectos da identificação de sistemas são na realidade variações da análise de séries temporais; os desenvolvimentos tanto de métodos espectrais como de métodos paramétricos para séries

temporais começaram com Yule (1927), e tiveram um impacto profundo no campo da identificação de sistemas. No fim deste período foram publicados livros como os de Grenander e Rosenblatt (1957) e Blackman e Tukey (1959), mais tarde vieram notáveis pesquisadores na área de identificação, como Whittle (1963), Hannan (1970), Box e Jenkins (1970) e Jenkins e Watts (1968); [23].

Para o segundo período, o *status* no campo da identificação no ano 1970 foi descrito por Åström e Eykhoff (1971) com a seguinte frase:

"O campo da identificação neste momento é algo desconcertante, inclusive para os chamados especialistas. Muitos métodos têm sido analisados e tratados. Novos métodos são propostos em grande quantidade, e o campo vê-se mais como uma caixa com vários truques do que como uma questão unificada."

Neste período surgiram uma grande quantidade de métodos de identificação, devido a aspectos tais como:

- A equação linear a diferenças com relações entrada - saída podia ser escrita como uma regressão linear e assim o método dos mínimos quadrados podia ser aplicado; mas isto levava a estimações com desvio, exceto sob situações de ruído específicas. Isso abriu tanto uma área para aproximações sistemáticas, como uma quantidade de truques para tratar este desvio (*bias*). Assim surgiram métodos como "o método da matriz estendida", "mínimos quadrados generalizados", "método da variável instrumental", "mínimos quadrados recursivos", "mínimos quadrados estendidos", "o método da máxima verossimilhança"(o qual por muito tempo foi reservado essencialmente para modelos ARMAX), etc.
- Técnicas de espectro e correlação para séries temporais foram bem desenvolvidas e adaptadas para estimação de sistemas de controle.
- Em adição a estas aproximações orientadas à estatística, foi natural pegar um ponto de vista orientado a sistemas, e começar com a relação básica de convolução entre a entrada e a saída. Muitas técnicas de deconvolução, realização, e expansão da função para a resposta ao impulso foram desenvolvidas.

Existem alguns livros como Lee (1964); Mendel (1973), e Eykhoff (1974), que têm informação muito boa do desenvolvimento da identificação de sistemas neste período.

Para o seguinte período, o artigo de Åström e Eykhoff (1971) mostra um desejo de ter um trabalho na unificação e em comparações entre as diferentes aproximações. Duas linhas principais podem ser distinguidas:

- tentativas de encontrar conexões entre os diferentes métodos

- trabalho em programas computacionais

Assim, no começo deste período (final dos anos 1960 e começo dos 1970), falou-se muito da necessidade de comparações entre os diferentes métodos, mas quase nada foi feito nesse aspecto. Como conclusão pode-se dizer que uma razão importante para ter acontecido isto foi a dificuldade para trocar programas e dados; como um exemplo, Ljung passou 4 meses em 1971, tentando converter um conjunto de cartões perfurados - de dados do alternador da turbina de Stanton - em uma fita entendível; [23]. O desenvolvimento do computador e do software permitiu e levou a pacotes de software de identificação com ambientes amigáveis para o usuário. Um dos primeiros e mais conhecidos pacotes interativos deste período foi o IDPAC, de Åström (1983), desenvolvido em Lund, mas muitos similares e relacionados foram desenvolvidos em universidades com área de pesquisa em identificação. Deve-se ressaltar que este desenvolvimento teve o efeito de que pesquisadores começaram a entender os métodos dos outros. Uma das conclusões neste período foi que a grande quantidade de métodos existentes correspondia a suposições diferentes de ruídos e variações das estruturas dos modelos, mais do que ser “modelos diferentes”. Isto permitiu uma simplificação substancial e contração na descrição do campo. Também deu luz para as comparações: quando é possível, e quando não tem sentido perguntar "qual é o melhor método"?

Outro tópico importante neste período foi o projeto experimental e a influência das condições experimentais no resultado da identificação (incluindo identificação em malha fechada). Ambos podiam ser tratados com estrutura estatística: é uma questão de computar e analisar a matriz de informação de Fisher. Um livro influente neste período, com ênfase particular no projeto experimental é o de Goodwin e Payne (1977).

Finalmente, em meados dos 1980:s, a visão estatística da identificação de sistemas tinha amadurecido e se estabelecido. As estruturas tradicional e clássica para estimação de parâmetros tinham sido acertadas e coerentemente escolhidas para o mundo dos sistemas dinâmicos. Os métodos correspondentes têm se mostrado como poderosas e práticas ferramentas, e pacotes de *software* comerciais começaram a aparecer.

1.2.4 Novos métodos

Com todos os métodos e teorias desenvolvidas em meados dos 1980:s, algumas pessoas pensaram que tinha-se acabado o interesse de pesquisa na área. Mas que surgiram um número de diferentes tópicos que pouco ou nada tinham de fundamentos estatísticos. Alguns dos mais importantes são:

- Métodos de subespaços para modelagem de dados no espaço de estado
- Identificação para controle

- Propriedades do ruído com rejeição de média
- Modelos *caixa preta* não lineares

No começo dos anos 1990:s, os métodos baseados na teoria de realização para estimação de modelos no espaço de estado (por exemplo, primeiro encontrar os estados a partir dos dados, e depois estimar o sistema), conhecidos geralmente como método de subespaços, tornaram-se os métodos mais efetivos e usados, em particular para sistemas multivariáveis. Estes, os Métodos de Subespaços para modelagem no espaço de estado, são o tema principal de nosso trabalho, e, segundo Ljung: "Este é na minha opinião, o desenvolvimento mais interessante na última década"; [23].

1.3 Modelo no Espaço de estado

Modelos matemáticos existem de todas as formas e tamanhos. Em nosso trabalho, vamos nós restringir aos modelos no *espaço de estado*, discretos no tempo, lineares, invariantes no tempo. Esta "limitação" pode-se ver como uma restrição ampla nas classes de modelos, especialmente a parte de que são lineares, mas, como falamos anteriormente: "*os modelos LTI são os mais simples e importantes na classe de sistemas dinâmicos usados na prática e na literatura. Com eles têm-se modelos ideais, e as experiências mostram que eles podem aproximar bem muitos dos processos industriais*". Além disso, o número de ferramentas para o projeto de sistemas de controle disponíveis para este tipo de modelos é vasto. Especialmente por esta razão, esta classe de modelos é de muito interesse. Um dos objetivos da teoria de sistemas é a determinação da dinâmica de um sistema através dos dados de entrada - saída medidos; isto pode ser feito por meio de modelos no espaço de estado. Se pelo menos n variáveis são necessárias para descrever completamente o comportamento do sistema dinâmico, então tais n variáveis são o conjunto chamado de variáveis de estado. [3], [10]. Quando ordenamos esta coleção de variáveis em um vetor, este é chamado de vetor de estado. Assim, o vetor de estado de um sistema determinístico constitui a mínima coleção de conhecimento necessário para obter o comportamento de um sistema dinâmico. Os modelos no espaço de estado oferecem vantagens como a baixa sensibilidade dos parâmetros com respeito a perturbações para sistemas de ordem elevada, e tem mostrado sua capacidade para identificar sistemas de múltiplas entradas e múltiplas saídas com dimensões de estado mínimas; [28]. O espaço de estado é n -dimensional, e qualquer estado do sistema pode ser representado pelos vetores de estado. Quando um espaço de estado de dimensão finita não for suficiente para capturar o comportamento do sistema, então a dinâmica deste terá dimensão infinita. Nesta tese apenas a dimensão finita e modelos discretos no espaço de estado são tratados.

Matematicamente, os modelos multivariáveis discretos no tempo, no espaço de estado são de-

scritos pelo seguinte conjunto de equações, [3], [10], [21], [19]:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k \end{cases} \quad (1.1)$$

Neste modelo temos os seguintes elementos:

- Os vetores $\mathbf{u}_k \in \mathfrak{R}^m$ e $\mathbf{y}_k \in \mathfrak{R}^l$ são medidas no instante k das m entradas e l saídas, respectivamente, do processo. O vetor $\mathbf{x}_k \in \mathfrak{R}^n$ é o vetor de estado do processo no tempo discreto k e contém os valores numéricos das n variáveis de estado. Este estado pode não ter significado físico, correspondendo a variáveis internas do modelo, e é muito importante nos Métodos de Subespaços como veremos no capítulo seguinte.
- $\mathbf{A} \in \mathfrak{R}^{n \times n}$ é chamada a matriz do sistema dinâmico, ou a matriz de estado; ela, como seu nome diz, descreve a dinâmica do sistema. $\mathbf{B} \in \mathfrak{R}^{n \times m}$ é a matriz de entrada que representa uma transformação linear pela qual as entradas influenciam no próximo estado. $\mathbf{C} \in \mathfrak{R}^{l \times n}$ é a matriz de saída que descreve como o estado interno é transferido ao mundo exterior nas medidas de \mathbf{y}_k . $\mathbf{D} \in \mathfrak{R}^{l \times m}$ é a matriz de realimentação. O par de matrizes \mathbf{A}, \mathbf{C} é suposto ser observável, o que implica que uma matriz chamada de observabilidade tem posto n . O par de matrizes \mathbf{A}, \mathbf{B} é suposto ser atingível, o que implica que uma matriz chamada de atingibilidade tem posto n . As matrizes de atingibilidade e observabilidade são de muita importância nos Métodos de subespaços como observaremos mais para a frente.

Com as equações que definem o modelo no espaço de estado, equação (1.1), em resposta à seqüência de entrada $\{\mathbf{u}_k\}$, obtemos o seguinte conjunto de vetores de estado:

$$\begin{cases} \mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0 \\ \mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1 = \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_0 + \mathbf{B}\mathbf{u}_1 \\ \vdots \\ \mathbf{x}_k = \mathbf{A}^k\mathbf{x}_0 + \sum_{i=0}^{k-1} \mathbf{A}^{k-i-1}\mathbf{B}\mathbf{u}_i \end{cases} \quad (1.2)$$

E a resposta do sistema à seqüência de entrada $\{\mathbf{u}_k\}$, fazendo uso dos vetores de estado, equações (1.2), é dada por:

$$\begin{cases} \mathbf{y}_0 = \mathbf{C}\mathbf{x}_0 + \mathbf{D}\mathbf{u}_0 \\ \mathbf{y}_1 = \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{u}_1 = \mathbf{C}\mathbf{A}\mathbf{x}_0 + \mathbf{C}\mathbf{B}\mathbf{u}_0 + \mathbf{D}\mathbf{u}_1 \\ \vdots \\ \mathbf{y}_k = \mathbf{C}\mathbf{A}^k\mathbf{x}_0 + \sum_{i=0}^{k-1} \mathbf{C}\mathbf{A}^{k-i-1}\mathbf{B}\mathbf{u}_i + \mathbf{D}\mathbf{u}_k \end{cases} \quad (1.3)$$

Supondo o sistema com entrada nula, $\mathbf{u}_k = \mathbf{0}$, chamado sistema livre, temos que as equações do modelo no espaço de estado podem ser reescritas como:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \\ \mathbf{y}_k = \mathbf{C}\mathbf{x}_k \end{cases} \quad (1.4)$$

Por iterações sucessivas a partir do estado inicial dado \mathbf{x}_0 para o sistema livre, obtemos:

$$\begin{cases} \mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 \\ \mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 = \mathbf{A}^2\mathbf{x}_0 \\ \mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 = \mathbf{A}^3\mathbf{x}_0 \\ \vdots \\ \mathbf{x}_k = \mathbf{A}^k\mathbf{x}_0 \end{cases} \quad (1.5)$$

E das equações (1.4) e (1.5), a resposta do sistema livre ou resposta à entrada zero é dada por:

$$\mathbf{y}_k = \mathbf{C}\mathbf{A}^k\mathbf{x}_0 \quad (1.6)$$

Se definimos uma matriz de transição de estados $\Phi_{(k,k_0)}$, desde o instante k_0 até o instante k como:

$$\mathbf{x}_k := \Phi_{(k,k_0)}\mathbf{x}_{k_0} \quad (1.7)$$

do conjunto de equações (1.5), temos que a matriz de transição do estado inicial 0, até o estado k pode ser definida como:

$$\Phi_{(k,0)} = \Phi_k = \mathbf{A}^k \quad (1.8)$$

Analogamente, tomando as equações do modelo no espaço de estado, equações (1.1), mas supondo agora o estado inicial zero, $\mathbf{x}_0 = \mathbf{0}$, temos o seguinte conjunto de vetores de estado:

$$\begin{cases} \mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0 = \mathbf{B}\mathbf{u}_0 \\ \mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1 = \mathbf{A}\mathbf{B}\mathbf{u}_0 + \mathbf{B}\mathbf{u}_1 \\ \vdots \\ \mathbf{x}_k = \sum_{i=0}^{k-1} \mathbf{A}^{k-i-1}\mathbf{B}\mathbf{u}_i \end{cases} \quad (1.9)$$

Assim, temos que a resposta do sistema à seqüência de entrada $\{\mathbf{u}_k\}$ e estado inicial zero, é dada por:

$$\begin{cases} \mathbf{y}_0 = \mathbf{C}\mathbf{x}_0 + \mathbf{D}\mathbf{u}_0 = \mathbf{D}\mathbf{u}_0 \\ \mathbf{y}_1 = \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{u}_1 = \mathbf{C}\mathbf{B}\mathbf{u}_0 + \mathbf{D}\mathbf{u}_1 \\ \vdots \\ \mathbf{y}_k = \sum_{i=0}^{k-1} \mathbf{C}\mathbf{A}^{k-i-1}\mathbf{B}\mathbf{u}_i + \mathbf{D}\mathbf{u}_k \end{cases} \quad (1.10)$$

que é chamada de resposta ao estado zero.

Analisando a equação (1.3) temos que a resposta de um sistema linear é a soma da resposta à entrada zero (equação (1.6)) com a resposta ao estado inicial zero (equação (1.10)). Estes resultados são muito importantes para o operador de Hankel, e um pouco para frente veremos mais detalhadamente a obtenção deste.

1.3.1 Resposta ao impulso de sistema monovariável

Se temos um sistema discreto monovariável representado no espaço de estado, e na sua entrada aplicamos um impulso discreto unitário; [3]; [17]; [11]:

$$\mathbf{u}_k = \delta_k := \begin{cases} 1 & \text{se } k = 0 \\ 0 & \text{se } k \neq 0 \end{cases}$$

e com o estado inicial sendo nulo, $\mathbf{x}_0 = 0$, a partir do instante em que ocorre o impulso temos:

- Instante 0:

$$\begin{cases} \mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0 = \mathbf{B} \\ \mathbf{y}_0 = \mathbf{C}\mathbf{x}_0 + \mathbf{D}\mathbf{u}_0 = \mathbf{D}\mathbf{u}_0 = \mathbf{D} = \mathbf{G}_0 \end{cases}$$

pois $\mathbf{x}_0 = 0$ e $\mathbf{u}_0 = 1$

- Instante 1:

$$\begin{cases} \mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1 = \mathbf{A}\mathbf{B} \\ \mathbf{y}_1 = \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{u}_1 = \mathbf{C}\mathbf{B}\mathbf{u}_0 = \mathbf{C}\mathbf{B} = \mathbf{G}_1 \end{cases}$$

pois $\mathbf{x}_1 = \mathbf{B}$ e $\mathbf{u}_1 = 0$

- Instante 2:

$$\begin{cases} \mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u}_2 = \mathbf{A}^2\mathbf{B} \\ \mathbf{y}_2 = \mathbf{C}\mathbf{x}_2 + \mathbf{D}\mathbf{u}_2 = \mathbf{C}\mathbf{A}\mathbf{B}\mathbf{u}_0 = \mathbf{C}\mathbf{A}\mathbf{B} = \mathbf{G}_2 \end{cases}$$

pois $\mathbf{x}_2 = \mathbf{A}\mathbf{B}$ e $\mathbf{u}_2 = 0$

E assim sucessivamente; por tanto a resposta ao impulso do sistema monovariável é dada por:

$$\mathbf{G}_k = \begin{cases} \mathbf{D} & \text{se } k = 0 \\ \mathbf{C}\mathbf{A}^{k-1}\mathbf{B} & \text{se } k \geq 1 \end{cases} \quad (1.11)$$

Tais termos são chamados de parâmetros de Markov do sistema LTI definido pela quádrupla $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, e expressam os efeitos das entradas passadas na saída atual \mathbf{y}_k .

1.3.2 Resposta ao impulso de sistema multivariável

Um sistema discreto multivariável representado no espaço de estado, com m entradas, pode ser representado como a entrada \mathbf{u}_k constituída de uma combinação linear de impulsos $\delta_i = [\dots 0 0 1 0 0 \dots]^T$ com apenas 1 na i -ésima componente, um para cada i -ésima componente da entrada, ponderado por:

$$\mathbf{u}_{ki} = \begin{cases} 1 & \text{se } k = i \\ 0 & \text{se } k \neq i \end{cases}$$

$$\mathbf{u}_k = \sum_{i=1}^m \mathbf{u}_{ki} \delta_i = \mathbf{u}_{k1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \mathbf{u}_{k2} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \dots + \mathbf{u}_{km} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (1.12)$$

A resposta ao impulso do sistema multivariável é então a combinação linear das respostas aos impulsos δ_i ponderadas por \mathbf{u}_{ki} , onde i é o contador dos experimentos, e a resposta final será a sequência ordenada de todas as respostas, obtidas da seguinte maneira:

- O impulso para a primeira entrada é aplicado e observada a resposta deste impulso para o experimento, \mathbf{G}_{k1} .
- O impulso é aplicado na segunda entrada e observada a resposta deste impulso para o experimento, \mathbf{G}_{k2} .
- Repetimos este procedimento para as m entradas do experimento.

Este procedimento é idealizado considerando que todas as respostas são obtidas no mesmo instante k . Assim, a resposta ao impulso unitário nesse instante k é dada por:

$$\mathbf{G}_k \equiv \left[\mathbf{G}_{k1} \quad \mathbf{G}_{k2} \quad \mathbf{G}_{k3} \quad \dots \quad \mathbf{G}_{km} \right] \quad (1.13)$$

onde \mathbf{G}_{ki} corresponde à resposta ao impulso no k -ésimo instante de tempo da entrada i .

Com a equação (1.12) como entrada, e supondo $\mathbf{x}_0 = 0$, a resposta ao impulso vai ter uma estrutura da forma; [3]; [11]:

$$\mathbf{G}_k = \begin{cases} 0 & \text{se } k < 0 \\ \mathbf{D} & \text{se } k = 0 \\ \mathbf{CA}^{k-1}\mathbf{B} & \text{se } k \geq 1 \end{cases} \quad (1.14)$$

1.3.3 Operador de Hankel

As equações que definem o modelo em espaço de estado, equação (1.1), podem ser reescritas em forma matricial como:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \quad (1.15)$$

Podemos dizer que o problema de realização é obter uma realização no espaço de estado, tal que para um dado operador de transferência \mathbf{T} tenhamos uma equivalência da equação (1.15), com a relação:

$$\mathbf{y} = \mathbf{T}\mathbf{u}$$

onde \mathbf{u} e \mathbf{y} representam respectivamente as seqüências de vetores de entrada e saída.

Na análise de sistemas tem-se um operador muito importante e útil, o operador de Hankel; este pode ser obtido desenvolvendo a equação em espaço de estado do sistema desde o instante de tempo inicial $k = 0$; [3]; [17]:

$$\begin{cases} \mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0 \\ \mathbf{y}_0 = \mathbf{C}\mathbf{x}_0 + \mathbf{D}\mathbf{u}_0 \end{cases} \quad (1.16)$$

$$\begin{cases} \mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1 \\ \mathbf{y}_1 = \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{u}_1 \end{cases} \quad (1.17)$$

Substituindo o estado \mathbf{x}_1 da equação (1.16) na equação (1.17), temos:

$$\begin{cases} \mathbf{x}_2 = \mathbf{A}(\mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0) + \mathbf{B}\mathbf{u}_1 = \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_0 + \mathbf{B}\mathbf{u}_1 \\ \mathbf{y}_1 = \mathbf{C}(\mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_0) + \mathbf{D}\mathbf{u}_1 = \mathbf{C}\mathbf{A}\mathbf{x}_0 + \mathbf{C}\mathbf{B}\mathbf{u}_0 + \mathbf{D}\mathbf{u}_1 \end{cases} \quad (1.18)$$

Para o instante seguinte temos:

$$\begin{cases} \mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u}_2 \\ \mathbf{y}_2 = \mathbf{C}\mathbf{x}_2 + \mathbf{D}\mathbf{u}_2 \end{cases} \quad (1.19)$$

E substituindo \mathbf{x}_2 , obtemos:

$$\begin{cases} \mathbf{x}_3 = \mathbf{A}^3 \mathbf{x}_0 + \mathbf{A}^2 \mathbf{B} \mathbf{u}_0 + \mathbf{A} \mathbf{B} \mathbf{u}_1 + \mathbf{B} \mathbf{u}_2 \\ \mathbf{y}_2 = \mathbf{C} \mathbf{A}^2 \mathbf{x}_0 + \mathbf{C} \mathbf{A} \mathbf{B} \mathbf{u}_0 + \mathbf{C} \mathbf{B} \mathbf{u}_1 + \mathbf{D} \mathbf{u}_2 \end{cases} \quad (1.20)$$

E assim sucessivamente, até obter uma expressão geral dada por:

$$\mathbf{y}_k = \begin{cases} \mathbf{C} \mathbf{x}_0 + \mathbf{D} \mathbf{u}_0 & \text{para } k = 0 \\ \mathbf{C} \mathbf{A}^k \mathbf{x}_0 + \sum_{j=0}^{k-1} \mathbf{C} \mathbf{A}^{k-j-1} \mathbf{B} \mathbf{u}_j + \mathbf{D} \mathbf{u}_k & \text{para } k \geq 1 \end{cases} \quad (1.21)$$

Daqui, temos que, uma coleção de saídas \mathbf{y} , de uma coleção de entradas \mathbf{u} , desde um instante 0 até um instante k , pode ser escrita como:

$$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C} \mathbf{A} \\ \mathbf{C} \mathbf{A}^2 \\ \vdots \\ \mathbf{C} \mathbf{A}^k \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} \mathbf{D} & 0 & 0 & \cdots & 0 \\ \mathbf{C} \mathbf{B} & \mathbf{D} & 0 & \cdots & 0 \\ \mathbf{C} \mathbf{A} \mathbf{B} & \mathbf{C} \mathbf{B} & \mathbf{D} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C} \mathbf{A}^{k-1} \mathbf{B} & \mathbf{C} \mathbf{A}^{k-2} \mathbf{B} & \mathbf{C} \mathbf{A}^{k-3} \mathbf{B} & \cdots & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_k \end{bmatrix} \quad (1.22)$$

Supondo um estado inicial $\mathbf{x}_0 = 0$, e com a equação (1.11) podemos reescrever esta equação como:

$$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{G}_0 & 0 & 0 & 0 & \cdots & 0 \\ \mathbf{G}_1 & \mathbf{G}_0 & 0 & 0 & \cdots & 0 \\ \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{G}_0 & 0 & \cdots & 0 \\ \mathbf{G}_3 & \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{G}_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \end{bmatrix} \quad (1.23)$$

pois os termos da matriz correspondem às respostas ao impulso do sistema.

Agora, supondo que aplicamos entradas não nulas somente até o instante $k = -1$, que vamos chamar de entradas *passadas* (com respeito ao instante presente, $k = 0$), ou seja, entradas nulas para $k = 0, 1, 2, \dots$; e que medimos as saídas para os instantes $k \geq 0$, que chamamos de saídas *futuras*,

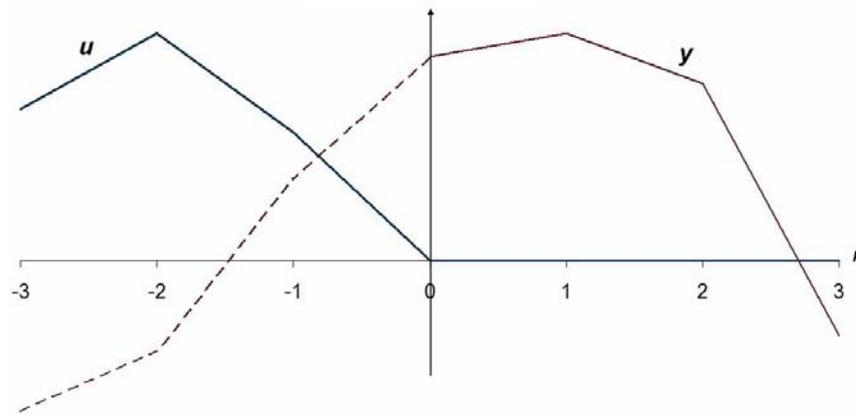


Fig. 1.3: Resposta à entrada zero de um sistema LTI

ver figura 1.3, podemos escrever a representação no espaço de estado como:

$$\begin{bmatrix} \vdots \\ \times \\ \times \\ \mathbf{y}_0 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \times & \times & 0 & 0 & \dots \\ \dots & \times & \times & \times & 0 & \dots \\ \dots & \mathbf{G}_3 & \mathbf{G}_2 & \mathbf{G}_1 & \times & \dots \\ \dots & \mathbf{G}_4 & \mathbf{G}_3 & \mathbf{G}_2 & \mathbf{G}_1 & \dots \\ \dots & \mathbf{G}_5 & \mathbf{G}_4 & \mathbf{G}_3 & \mathbf{G}_2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ u_{-2} \\ u_{-1} \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (1.24)$$

Ou seja, a saída para a seqüência de entrada aplicada ao sistema LTI discreto no tempo pode ser expressa como:

$$\mathbf{y}_k = \sum_{i=0}^{\infty} \mathbf{G}_i \mathbf{u}_{k-i} \quad \text{para } k = 0, 1, 2, \dots \quad (1.25)$$

e definindo as seqüências de entradas passadas:

$$\mathbf{u}_- := \begin{bmatrix} \mathbf{u}_{-1} \\ \mathbf{u}_{-2} \\ \mathbf{u}_{-3} \\ \vdots \end{bmatrix} \quad (1.26)$$

e saídas futuras:

$$\mathbf{y}_+ := \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \end{bmatrix} \quad (1.27)$$

respectivamente, podemos reescrever a relação das seqüências de entrada e saída como:

$$\mathbf{y}_+ = \mathbf{H}\mathbf{u}_- \quad (1.28)$$

com o operador \mathbf{H} , que recebe o nome de operador de Hankel, e é definido como:

$$\mathbf{H} := \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \mathbf{G}_3 & \mathbf{G}_4 & \cdots \\ \mathbf{G}_2 & \mathbf{G}_3 & \mathbf{G}_4 & \mathbf{G}_5 & \cdots \\ \mathbf{G}_3 & \mathbf{G}_4 & \mathbf{G}_5 & \mathbf{G}_6 & \cdots \\ \mathbf{G}_4 & \mathbf{G}_5 & \mathbf{G}_6 & \mathbf{G}_7 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \mathcal{OC} \quad (1.29)$$

onde os operadores de observabilidade e atingibilidade são definidos como:

$$\mathcal{O} := \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix} \quad (1.30)$$

e:

$$\mathcal{C} := \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots \end{bmatrix} \quad (1.31)$$

respectivamente.

A matriz de Hankel apresenta várias propriedades interessantes e que são requeridas para o desenvolvimento dos métodos de identificação por subespaços, e que vamos realizar nos próximos capítulos; [3]; [11]; [17], [13], [34].

Os Métodos de Subespaços, tema principal deste trabalho, consideram os modelos no espaço de estado. Mas por que é uma boa idéia tentar encaixar dados de processos (industriais) experimentais ao modelo no espaço de estado descrito? A razão principal é que para sistemas de múltiplas entradas - múltiplas saídas (MIMO), a representação no espaço de estado é o modelo mais conveniente para trabalhar com por exemplo, o projeto de sistemas de controle assistido por computador (*computer aided control system design*) (CACSD). A maioria dos controladores pode ser computada efetivamente em

termos do modelo no espaço de estado, quando em outras representações (como por exemplo formas fracionais matriciais) os cálculos, a análise e o projeto são mais complicados.

1.4 Modelo Estendido no Espaço de Estado

Definamos as matrizes bloco de Hankel, de dimensões $km \times N$ e $kl \times N$ como, [3], [19]:

$$\mathbf{U}_{0|k-1} := \begin{pmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{N-1} \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \cdots & \mathbf{u}_N \\ \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 & \cdots & \mathbf{u}_{N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_{k-1} & \mathbf{u}_k & \mathbf{u}_{k+1} & \cdots & \mathbf{u}_{k+N-2} \end{pmatrix} \quad (1.32)$$

que vamos chamar de entradas passadas, $\mathbf{U}_{0|k-1} = \mathbf{U}_p$; e:

$$\mathbf{Y}_{0|k-1} := \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{N-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_N \\ \mathbf{y}_2 & \mathbf{y}_3 & \mathbf{y}_4 & \cdots & \mathbf{y}_{N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{y}_{k-1} & \mathbf{y}_k & \mathbf{y}_{k+1} & \cdots & \mathbf{y}_{k+N-2} \end{pmatrix} \quad (1.33)$$

de saídas passadas, $\mathbf{Y}_{0|k-1} = \mathbf{Y}_p$, respectivamente.

Os Métodos de Subespaços fazem uso da estrutura presente nas matrizes de observabilidade, equação (1.30), e atingibilidade, equação (1.31); a matriz de observabilidade estendida ($k \geq n$) é definida como:

$$\mathcal{O}_k := \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{k-1} \end{pmatrix} \in \mathfrak{R}^{kl \times n} \quad (1.34)$$

tal que $\text{posto}(\mathcal{O}_k) = n$, e a matriz de atingibilidade estendida ($k \geq n$) por:

$$\mathcal{C}_k := \left(\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{k-1}\mathbf{B} \right) \in \mathfrak{R}^{n \times km} \quad (1.35)$$

Além das matrizes de observabilidade e atingibilidade, precisamos de uma matriz bloco, triangular

inferior de Toeplitz, definida por:

$$\mathcal{T}_k := \begin{pmatrix} \mathbf{D} & 0 & 0 & \dots & 0 \\ \mathbf{CB} & \mathbf{D} & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{k-1}\mathbf{B} & \mathbf{CA}^{k-2} & \mathbf{CA}^{k-3} & \dots & \mathbf{D} \end{pmatrix} \in \mathfrak{R}^{kl \times km} \quad (1.36)$$

e o vetor de estado inicial definido como:

$$\mathbf{X}_0 := \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{N-1} \end{pmatrix} \in \mathfrak{R}^{n \times N} \quad (1.37)$$

que podemos chamar de vetor de estados passados, $\mathbf{X}_0 = \mathbf{X}_p$.

Das equações (1.1) que representam o sistema, e com as definições (1.32), (1.33), (1.34), (1.36), e (1.37), podemos reescrever a seguinte equação matricial de entradas-saídas com estado inicial \mathbf{X}_0 dado por (1.37), como o modelo estendido no espaço de estado; [3], [17], [26], [7]:

$$\mathbf{Y}_p = \mathcal{O}_k \mathbf{X}_p + \mathcal{T}_k \mathbf{U}_p \quad (1.38)$$

que é similar a equação (1.22), mas com as entradas, saídas e estados em forma matricial. De forma análoga para instantes posteriores, definimos:

$$\mathbf{U}_{k|2k-1} := \begin{pmatrix} \mathbf{u}_k & \mathbf{u}_{k+1} & \mathbf{u}_{k+2} & \dots & \mathbf{u}_{k+N-1} \\ \mathbf{u}_{k+1} & \mathbf{u}_{k+2} & \mathbf{u}_{k+3} & \dots & \mathbf{u}_{k+N} \\ \mathbf{u}_{k+2} & \mathbf{u}_{k+3} & \mathbf{u}_{k+4} & \dots & \mathbf{u}_{k+N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_{2k-1} & \mathbf{u}_{2k} & \mathbf{u}_{2k+1} & \dots & \mathbf{u}_{2k+N-2} \end{pmatrix} \in \mathfrak{R}^{km \times N} \quad (1.39)$$

que vamos chamar de entradas futuras, $\mathbf{U}_{k|2k-1} = \mathbf{U}_f$; e:

$$\mathbf{Y}_{k|2k-1} := \begin{pmatrix} \mathbf{y}_k & \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \dots & \mathbf{y}_{k+N-1} \\ \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \mathbf{y}_{k+3} & \dots & \mathbf{y}_{k+N} \\ \mathbf{y}_{k+2} & \mathbf{y}_{k+3} & \mathbf{y}_{k+4} & \dots & \mathbf{y}_{k+N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{y}_{2k-1} & \mathbf{y}_{2k} & \mathbf{y}_{2k+1} & \dots & \mathbf{y}_{2k+N-2} \end{pmatrix} \in \mathfrak{R}^{kl \times N} \quad (1.40)$$

de saídas futuras, $\mathbf{Y}_{k|2k-1} = \mathbf{Y}_f$; e definimos:

$$\mathbf{X}_k := \begin{pmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} & \mathbf{x}_{k+2} & \dots & \mathbf{x}_{k+N-1} \end{pmatrix} \in \mathfrak{R}^{n \times N} \quad (1.41)$$

como o vetor de estados futuros, $\mathbf{X}_k = \mathbf{X}_f$.

De forma análoga como para o passado, mas com as definições (1.39), (1.40), e (1.41), temos a equação matricial de entradas-saídas com estado inicial \mathbf{X}_k dado por (1.41), como:

$$\mathbf{Y}_f = \mathcal{O}_k \mathbf{X}_f + \mathcal{T}_k \mathbf{U}_f \quad (1.42)$$

Definindo a matriz de atingibilidade reordenada como:

$$\mathcal{C}_k^{\leftarrow} := \begin{pmatrix} \mathbf{A}^{k-1} \mathbf{B} & \mathbf{A}^{k-2} \mathbf{B} & \dots & \mathbf{A} \mathbf{B} & \mathbf{B} \end{pmatrix} \quad (1.43)$$

para os estados, podemos escrever a equação matricial:

$$\mathbf{X}_f = \mathbf{A}^k \mathbf{X}_p + \mathcal{C}_k^{\leftarrow} \mathbf{U}_p \quad (1.44)$$

Também precisamos referir-nos as seguintes matrizes:

$$\mathbf{W}_{0|k-1} := \begin{pmatrix} \mathbf{U}_{0|k-1} \\ \mathbf{Y}_{0|k-1} \end{pmatrix} \in \mathfrak{R}^{(km+kl) \times N} \quad (1.45)$$

como matriz de dados passados, $\mathbf{W}_{0|k-1} = \mathbf{W}_p$, e:

$$\mathbf{W}_{k|2k-1} := \begin{pmatrix} \mathbf{U}_{k|2k-1} \\ \mathbf{Y}_{k|2k-1} \end{pmatrix} \in \mathfrak{R}^{(km+kl) \times N} \quad (1.46)$$

como matriz de dados futuros, $\mathbf{W}_{k|2k-1} = \mathbf{W}_f$.

Para todas as definições anteriores, devem ser satisfeitos os critérios de $k > n$ e o valor de N suficientemente grande.

O problema da identificação de sistemas por Métodos de Subespaços pode-se escrever finalmente como:

Dados um grande número de medidas (N) de entradas, $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$, e saídas, $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k$, de um sistema desconhecido, determinar a ordem n do sistema, e as matrizes $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$, à menos de uma transformação de similaridade.

Capítulo 2

MÉTODOS DE SUBESPAÇOS

2.1 Introdução

A automação dos processos industriais nas últimas décadas levou a um aumento de produtividade em alguns setores, obrigando a indústria a adaptar-se às demandas do mercado e a aumentar sua competitividade. Para aumentar dita competitividade é necessário maximizar a eficiência dos processos e, conseqüentemente, implantar controladores mais eficientes e confiáveis. Para isso é imprescindível conhecer o comportamento dinâmico do processo. Na atualidade, cada vez mais, o trabalho de um engenheiro consiste na obtenção de modelos matemáticos destes processos; [24]. Os modelos são utilizados em áreas tão distintas como: economia, bioengenharia, meteorologia, engenharia, etc. O campo de utilização desses modelos é muito amplo, destacando-se aplicações como: controle, supervisão, predição, simulação, otimização, etc. As técnicas de identificação de sistemas têm evoluído bastante, procurando atender essa demanda com modelos cada vez mais rápidos e precisos. Enfatiza-se ainda que os processos industriais estão sujeitos a severos requisitos de eficiência, disponibilidade e segurança. A complexidade dos mesmos aumenta constantemente e isso torna necessário o desenvolvimento de ferramentas automáticas de auxílio ao operador humano: os sistemas de supervisão. Em termos de disponibilidade e segurança, é necessário destacar as áreas destinadas à detecção e diagnóstico de falhas. Com uma rápida detecção das falhas pode-se evitar desde uma perda de produção até uma deterioração do sistema com conseqüências que podem ser catastróficas para o próprio sistema e, pior ainda, para o pessoal da planta. Um dos métodos utilizados para isso é a comparação de algumas variáveis do processo com as saídas equivalentes de um modelo da planta. Tais métodos são denominados de diagnóstico de falhas baseado em modelo. E para isto, é necessário dispor de um modelo matemático que ajuste-se bem ao comportamento do sistema estudado. Nesse sentido, grande parte da literatura referente à identificação de sistemas trata como achar modelos polinomiais, como os métodos de erro de predição (PEM) e os métodos de variáveis instrumentais (IVM). No entanto,

o uso destes métodos acarreta vários problemas que precisam ser resolvidos. Os métodos PEM, por exemplo, têm excelentes propriedades estatísticas desde que o modelo seja bem encontrado, o que pode ser às vezes difícil. Por outro lado, os métodos IVM tentam dar uma estimação dos parâmetros solucionando um sistema de equações lineares. De qualquer forma, a obtenção dos modelos PEM e IVM é muito difícil para o caso de sistemas multivariáveis, e a confiabilidade numérica pode ser inaceitável em problemas complexos, incluindo sistemas de grandes dimensões, com grande número de entradas - saídas. No caso de sistemas complexos, os Métodos de Subespaços surgem como uma alternativa aos métodos PEM e IVM. Para sistemas multivariáveis estes métodos provêm modelos confiáveis no espaço de estado diretamente a partir dos dados de entrada e saída. Os algoritmos envolvidos são numericamente robustos e não usam técnicas de otimização não lineares, o que significa que os algoritmos são rápidos (não iterativos) e precisos (não existem problemas de mínimos locais). A complexidade computacional destes métodos é menor quando comparada ao método PEM, especialmente quando o número de entradas e saídas é grande; [39]. Eles não requerem uma representação canônica do modelo no espaço de estado e não precisam de muitos parâmetros de inicialização, como é o caso do método PEM. Dado que sistemas de grandes dimensões são comumente encontrados na indústria, a aplicação dos algoritmos de identificação por subespaços neste campo é bastante promissora.

Os Métodos de Subespaços para modelagem no espaço de estado têm suas origens na teoria de realização no espaço de estado devido a Ho e Kalman (1966). Estas técnicas determinam um modelo no espaço de estado para uma resposta ao impulso dada, o qual gerou muita atenção nos anos 1970. Na área de identificação, geralmente tem-se dados de entrada - saída, e não medidas da resposta ao impulso. Assim, os Métodos de Subespaços foram desenvolvidos a finais dos anos 1980. O termo "Método de Subespaço para Identificação" foi introduzido por Verhaegen e Deprettere (1991). Na atualidade existem muitas versões de Métodos de Subespaços: Canonical Variate Analysis (CVA) devido a Larimore (1990), Multivariable Output-Error State-Space (MOESP) devido a Verhaegen e Dewilde (1992), Numerical algorithm for Subspace State-Space System Identification (N4SID) devido a Van Overschee e De Moor (1994), entre outros.

Os Métodos de identificação por Subespaços em tempo discreto referem-se a uma classe de algoritmos cuja principal característica é trabalhar com os subespaços gerados pelas linhas ou colunas das matrizes bloco de Hankel formadas pelos dados de entrada - saída, para calcular um modelo confiável no espaço de estado. Os modelos no espaço de estado são descritos matematicamente pelas equações:

$$\begin{cases} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k \end{cases} \quad (2.1)$$

Os modelos no espaço de estado, como já citamos no capítulo 1, apresentam vantagens sobre tudo na parte de aplicação a sistemas de múltiplas entradas - múltiplas saídas (MIMO), ficando tão simples manejar-los como se fosse um sistema de uma entrada - uma saída (SISO), [24], [10], [11], [26]. Os sistemas no espaço de estado têm os seguintes elementos:

- $\mathbf{A} \in \mathfrak{R}^{n \times n}$ é chamada a matriz do sistema.
- $\mathbf{B} \in \mathfrak{R}^{n \times m}$ é chamada a matriz de entrada.
- $\mathbf{C} \in \mathfrak{R}^{l \times n}$ é chamada a matriz de saída.
- $\mathbf{D} \in \mathfrak{R}^{l \times m}$ é chamada a matriz de realimentação direta.
- $\mathbf{u}_k \in \mathfrak{R}^m$ que são as entradas do sistema; sinais controláveis.
- $\mathbf{y}_k \in \mathfrak{R}^l$ que são as saídas do sistema; sinais observáveis.
- $\mathbf{x}_k \in \mathfrak{R}^n$ que é o vetor de estados do sistema.

Os estados do sistema são muito importantes nos métodos de subespaços, e sua importância é uma das maiores diferenças com os métodos clássicos de erro de predição (PEM).

Qual a diferença entre estes dois métodos?

Ao contrário dos métodos clássicos onde primeiro as matrizes do sistema são obtidas, e depois, se necessários, os estados, nos métodos de subespaços computam-se primeiro os estados (implícita ou explicitamente), e depois obtém-se o modelo do sistema; ver figura 2.1.

Qual é a vantagem deste fato?

É que com o modelo no espaço de estado, equação (2.1), se tem-se as entradas \mathbf{u}_k e as saídas \mathbf{y}_k , e se conhecermos os estados \mathbf{x}_k , as matrizes \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} podem ser obtidas por meio do método dos mínimos quadrados, pois o sistema vai ficar como um problema linear, com estas matrizes como incógnitas. Isto, em contraste com os métodos clássicos de erro de predição (PEM), onde tem-se que lidar com problemas de otimização não lineares; esta é a vantagem dos métodos de subespaços; [10], [17], [26], [45], [37].

Então, o problema central vai ser determinar a ordem n e as matrizes \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} do sistema, à menos de uma transformação de similaridade, utilizando Métodos de Subespaços para identificação. Com as definições apresentadas no capítulo 1, todas as condições necessárias para obter o modelo estendido no espaço de estado foram tratadas e estamos prontos para analisar o seguinte teorema, que constitui a base dos Métodos de Subespaços.

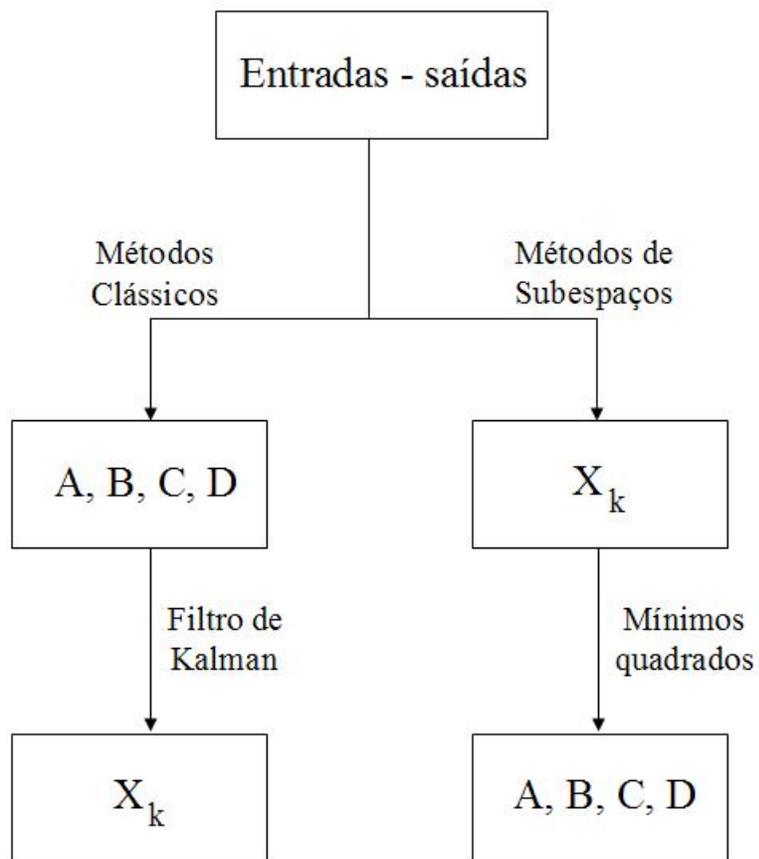


Fig. 2.1: Diferentes caminhos que seguem os Métodos Clássicos e os Métodos de Subespaços.

Teorema 2.1 (Teorema Principal dos Métodos de Subespaços) *Considerando que; [10], [32], [17]:*

i) *A entrada é persistentemente excitante de ordem k , isto é que:*

$$\text{posto}(\mathbf{U}_p) = km$$

onde m corresponde ao número de entradas, k é a dimensão das matrizes bloco de Hankel, e se satisfaz o critério de $k > n$.

ii) *A interseção do espaço linha das entradas futuras, \mathbf{U}_f , e do espaço linha dos estados passados, \mathbf{X}_p , é vazia:*

$$\mathcal{R}(\mathbf{U}_f^T) \cap \mathcal{R}(\mathbf{X}_p^T) = \emptyset$$

iii) *As matrizes $\mathbf{W}_1 \in \mathfrak{R}^{kl \times kl}$ e $\mathbf{W}_2 \in \mathfrak{R}^{N \times N}$ são tais que: o posto da matriz \mathbf{W}_1 , é completo; e a matriz \mathbf{W}_2 é tal que:*

$$\text{posto}(\mathbf{W}_p) = \text{posto}(\mathbf{W}_p \mathbf{W}_2)$$

onde $\mathbf{W}_p = \begin{pmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{pmatrix}$,

e definindo $\bar{\mathbf{P}}_k$ como a projeção oblíqua (para algumas noções de projeções ortogonal e oblíqua, ver apêndice B):

$$\bar{\mathbf{P}}_k := \mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p \quad (2.2)$$

e a decomposição em valores singulares (SVD: Singular Value Decomposition):

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \quad (2.3)$$

$$= \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T \quad (2.4)$$

temos o seguinte:

a) *A projeção oblíqua das saídas futuras, \mathbf{Y}_f , no subespaço das entradas e saídas passadas, \mathbf{W}_p , na direção das entradas futuras, \mathbf{U}_f , é igual ao produto da matriz de observabilidade estendida pela matriz de estados futuros:*

$$\bar{\mathbf{P}}_k = \mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p = \mathcal{O}_k \mathbf{X}_f \quad (2.5)$$

b) *A ordem do sistema (2.1), é igual ao número de valores singulares na equação (2.3).*

c) A matriz de observabilidade estendida é dada pela expressão:

$$\mathcal{O}_k = \mathbf{W}_1^{-1} \mathbf{U}_1 \Sigma_1^{1/2} \mathbf{T} \quad (2.6)$$

e os estados contidos no subespaço coluna de \mathbf{W}_2 podem ser obtidos de:

$$\mathbf{X}_f \mathbf{W}_2 = \mathbf{T}^{-1} \Sigma_1^{1/2} \mathbf{V}_1^T \quad (2.7)$$

onde $\mathbf{T} \in \mathbb{R}^{n \times n}$ é uma transformação de similaridade, não singular.

d) A seqüência de estados é dada pela equação:

$$\mathbf{X}_f = \mathcal{O}_k^\dagger \bar{\mathbf{P}}_k \quad (2.8)$$

Prova:

a) Podemos reescrever a equação do modelo estendido no espaço de estado, (1.38):

$$\mathbf{Y}_p = \mathcal{O}_k \mathbf{X}_p + \mathcal{T}_k \mathbf{U}_p$$

como:

$$\mathbf{X}_p = \mathcal{O}_k^\dagger \mathbf{Y}_p - \mathcal{O}_k^\dagger \mathcal{T}_k \mathbf{U}_p$$

Substituindo na equação (1.44):

$$\mathbf{X}_f = \mathbf{A}^k \mathbf{X}_p + \mathcal{C}_k^{\leftarrow} \mathbf{U}_p$$

o valor de \mathbf{X}_p , obtemos:

$$\begin{aligned} \mathbf{X}_f &= \mathbf{A}^k \left(\mathcal{O}_k^\dagger \mathbf{Y}_p - \mathcal{O}_k^\dagger \mathcal{T}_k \mathbf{U}_p \right) + \mathcal{C}_k^{\leftarrow} \mathbf{U}_p \\ &= \left(\mathcal{C}_k^{\leftarrow} - \mathbf{A}^k \mathcal{O}_k^\dagger \mathcal{T}_k \right) \mathbf{U}_p + \left(\mathbf{A}^k \mathcal{O}_k^\dagger \right) \mathbf{Y}_p \end{aligned}$$

e fazendo:

$$\mathbf{L}_p = \left(\mathcal{C}_k^{\leftarrow} - \mathbf{A}^k \mathcal{O}_k^\dagger \mathcal{T}_k \quad \mathbf{A}^k \mathcal{O}_k^\dagger \right)$$

e:

$$\mathbf{W}_p = \begin{pmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{pmatrix}$$

podemos reescrever os estados futuros, \mathbf{X}_f , como uma combinação linear das entradas e saídas passadas, \mathbf{W}_p :

$$\mathbf{X}_f = \mathbf{L}_p \mathbf{W}_p \quad (2.9)$$

Substituindo na equação (1.42):

$$\mathbf{Y}_f = \mathcal{O}_k \mathbf{X}_f + \mathcal{T}_k \mathbf{U}_f$$

a equação de estados futuros, (2.9), temos:

$$\mathbf{Y}_f = \mathcal{O}_k \mathbf{L}_p \mathbf{W}_p + \mathcal{T}_k \mathbf{U}_f$$

Agora, trabalhando nesta equação, fazendo uma projeção no subespaço linha do complemento ortogonal das entradas futuras, \mathbf{U}_f^\perp :

$$\mathbf{Y}_f | \mathbf{U}_f^\perp = \mathcal{O}_k \mathbf{L}_p \mathbf{W}_p | \mathbf{U}_f^\perp + \mathcal{T}_k \mathbf{U}_f | \mathbf{U}_f^\perp$$

como $\mathbf{U}_f | \mathbf{U}_f^\perp = 0$, obtemos a equação:

$$\mathbf{Y}_f | \mathbf{U}_f^\perp = \mathcal{O}_k \mathbf{L}_p \mathbf{W}_p | \mathbf{U}_f^\perp \quad (2.10)$$

e aceitando que (prova no apêndice da referência [10]):

$$[\mathbf{W}_p | \mathbf{U}_f^\perp] [\mathbf{W}_p | \mathbf{U}_f^\perp]^\dagger \mathbf{W}_p = \mathbf{W}_p \quad (2.11)$$

então, multiplicando a equação (2.10) à direita por $[\mathbf{W}_p | \mathbf{U}_f^\perp]^\dagger \mathbf{W}_p$:

$$[\mathbf{Y}_f | \mathbf{U}_f^\perp] [\mathbf{W}_p | \mathbf{U}_f^\perp]^\dagger \mathbf{W}_p = \mathcal{O}_k \mathbf{L}_p [\mathbf{W}_p | \mathbf{U}_f^\perp] [\mathbf{W}_p | \mathbf{U}_f^\perp]^\dagger \mathbf{W}_p$$

Lembrando uma das equações de projeção oblíqua:

$$\mathbf{A}/_F \mathbf{G} = [\mathbf{A} | \mathbf{F}^\perp] [\mathbf{G} | \mathbf{F}^\perp]^\dagger \mathbf{G} \quad (2.12)$$

e com as equações (2.5), (2.9) e (2.11) vamos finalmente obter:

$$\mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p = \mathcal{O}_k \mathbf{L}_p \mathbf{W}_p$$

$$\bar{\mathbf{P}}_k = \mathcal{O}_k \mathbf{X}_f$$

b) Da equação (2.5):

$$\begin{aligned}\bar{\mathbf{P}}_k &= \mathcal{O}_k \mathbf{X}_f \\ \mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 &= \mathbf{W}_1 \mathcal{O}_k \mathbf{X}_f \mathbf{W}_2\end{aligned}$$

Lembrando as dimensões das matrizes:

$$\begin{aligned}\mathbf{W}_1 &\in \mathfrak{R}^{kl \times kl}, \\ \mathbf{W}_2 &\in \mathfrak{R}^{N \times N}, \\ \mathcal{O}_k &\in \mathfrak{R}^{kl \times n}, \\ \mathbf{X}_f &\in \mathfrak{R}^{n \times N}\end{aligned}$$

temos que o produto $\mathbf{W}_1 \mathcal{O}_k \in \mathfrak{R}^{kl \times n}$ e o produto $\mathbf{X}_f \mathbf{W}_2 \in \mathfrak{R}^{n \times N}$ têm *posto* = n , pela consideração iii) no enunciado do teorema ($\mathbf{W}_1 \mathcal{O}_k$ tem n colunas, e $\mathbf{X}_f \mathbf{W}_2$ tem n linhas); e o produto de duas matrizes de *posto* = n é uma matriz de *posto* = n .

c) Da decomposição em valores singulares (SVD), equação (2.4), temos:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \mathbf{W}_1 \mathcal{O}_k \mathbf{X}_f \mathbf{W}_2 = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T$$

que, por propriedades da SVD pode ser decomposta em:

$$\begin{aligned}\mathbf{W}_1 \mathcal{O}_k &= \mathbf{U}_1 \Sigma_1^{1/2} \mathbf{T} \\ \mathbf{X}_f \mathbf{W}_2 &= \mathbf{T}^{-1} \Sigma_1^{1/2} \mathbf{V}_1^T\end{aligned}$$

assim obtemos:

$$\mathcal{O}_k = \mathbf{W}_1^{-1} \mathbf{U}_1 \Sigma_1^{1/2} \mathbf{T}$$

onde \mathbf{T} é uma matriz arbitrária não singular de dimensão $n \times n$.

d) Da equação (2.5), temos:

$$\bar{\mathbf{P}}_k = \mathcal{O}_k \mathbf{X}_f$$

cuja solução pode ser obtida como:

$$\mathbf{X}_f = \mathcal{O}_k^\dagger \bar{\mathbf{P}}_k$$

C.Q.D.

Os Métodos de Subespaços em essência são constituídos por três passos principais: o primeiro, onde fazemos uma projeção (ortogonal ou oblíqua) das matrizes de dados, o segundo onde obtemos os

estados (implícita ou explicitamente) a partir do subespaço gerado pela projeção anterior, e finalmente no terceiro obtém-se as matrizes do sistema para o modelo no espaço de estado. Cada Método de Subespaços parece completamente diferente dos outros em conceito, ferramentas computacionais e interpretação, mas no fundo, todos correspondem a estes três passos principais; [14], [38].

Do anterior teorema temos que a equação (2.9) é muito importante, pois ela expressa-nos que os estados futuros estão no subespaço linha das entradas e saídas passadas, ou seja, podemos obter os estados futuros como uma combinação linear dos dados passados:

$$\mathbf{X}_f = \mathbf{L}_p \mathbf{W}_p$$

Além deste resultado, podemos resumir o teorema anterior nos seguintes itens:

- $\text{posto}(\bar{\mathbf{P}}_k) = n$
- $\mathcal{R}(\bar{\mathbf{P}}_k^T) = \mathcal{R}(\mathbf{X}_f^T)$
Isto é, o espaço linha gerado pela projeção oblíqua das saídas futuras no subespaço das entradas e saídas passadas, na direção das entradas futuras é igual ao espaço linha dos estados futuros.
- $\mathcal{R}(\bar{\mathbf{P}}_k) = \mathcal{R}(\mathcal{O}_k)$
O que significa que o espaço coluna gerado pela projeção oblíqua das saídas futuras no subespaço das entradas e saídas passadas, na direção das entradas futuras é igual ao espaço coluna da matriz de observabilidade.

Isto explica o nome de subespaços dado a estes algoritmos, pois eles obtêm matrizes relativas ao sistema (matriz de observabilidade, matriz de atingibilidade, matriz de Toeplitz) dos subespaços de matrizes de dados projetadas.

Agora, no caso especial quando temos uma entrada do tipo impulso unitário deslocado:

$$\mathbf{u}_{k-i} = \begin{cases} 1 & \text{se } i = 1 \\ 0 & \text{se } i \neq 1 \end{cases}$$

e o estado $\mathbf{x}_i = 0$ para $i = 0, 1, \dots, k-1$, vamos obter uma saída resposta ao impulso. Nesse caso, o vetor de estados será dado por (prova das equações (1.9), (1.41)):

$$\mathbf{X}_f = \left(\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{N-1}\mathbf{B} \right)$$

e da equação (2.5):

$$\bar{\mathbf{P}}_k = \mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p = \mathcal{O}_k \mathbf{X}_f$$

$$= \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{N-1}\mathbf{B} \end{pmatrix} \\ = \mathbf{H}$$

O que indica que neste caso o problema de identificação determinística é equivalente ao problema de realização determinística.

2.2 Seqüência de Estados \mathbf{X}_f

Como falamos anteriormente, os estados são muito importantes, pois já temos as seqüências de entrada e saída, e se conhecermos a seqüência de estados, podemos obter as matrizes do sistema por meio do método dos mínimos quadrados; a obtenção destes estados pode ser dividida em duas classes, os chamados algoritmos de intersecção e os chamados algoritmos de projeção. Estes nomes vêm dos métodos utilizados para conseguir as matrizes do sistema desde os dados de entrada - saída arrumados em matrizes bloco de Hankel.

2.2.1 Algoritmos de intersecção

A seqüência de estados futuros está no espaço linha das entradas - saídas passadas e no espaço linha das entradas - saídas futuras, ou seja:

$$\mathcal{R}(\mathbf{X}_f^T) = \mathcal{R}(\mathbf{W}_p^T) \cap \mathcal{R}(\mathbf{W}_f^T)$$

Para provar isto, por um lado, temos da equação (2.9), que foi uma das conclusões importantes do teorema, que:

$$\mathbf{X}_f = \mathbf{L}_p \mathbf{W}_p$$

Esta equação indica que a seqüência de estados é formada como uma combinação linear das linhas dos dados (entradas - saídas) passados, ou seja:

$$\mathcal{R}(\mathbf{X}_f^T) \subset \mathcal{R}(\mathbf{W}_p^T)$$

Por outro lado, temos que a equação do modelo no espaço de estado, (1.42):

$$\mathbf{Y}_f = \mathcal{O}_k \mathbf{X}_f + \mathcal{T}_k \mathbf{U}_f$$

pode ser reescrita como:

$$\begin{aligned} \mathbf{X}_f &= \mathcal{O}_k^\dagger \mathbf{Y}_f - \mathcal{O}_k^\dagger \mathcal{T}_k \mathbf{U}_f \\ &= \left(-\mathcal{O}_k^\dagger \mathcal{T}_k \right) \mathbf{U}_f + \left(\mathcal{O}_k^\dagger \right) \mathbf{Y}_f \\ &= \mathbf{L}_f \mathbf{W}_f \end{aligned}$$

onde:

$$\begin{aligned} \mathbf{L}_f &= \left(\begin{array}{cc} -\mathcal{O}_k^\dagger \mathcal{T}_k & \mathcal{O}_k^\dagger \end{array} \right) \\ \mathbf{W}_f &= \left(\begin{array}{c} \mathbf{U}_f \\ \mathbf{Y}_f \end{array} \right) \end{aligned}$$

indicando que a seqüência de estados futuros, \mathbf{X}_f é formado como uma combinação linear das linhas dos dados (entradas - saídas) futuros, ou seja:

$$\mathcal{R}(\mathbf{X}_f^T) \subset \mathcal{R}(\mathbf{W}_f^T)$$

Logo o espaço linha da seqüência de estados futuros, \mathbf{X}_f , está contido na intersecção dos espaços linha dos dados passados, e linha dos dados futuros. Esta intersecção pode ser obtida por meio da decomposição em valores singulares (SVD) da matriz concatenada de dados de entrada - saída passados e futuros:

$$\mathbf{W} := \left(\begin{array}{c} \mathbf{U}_{0|2k-1} \\ \mathbf{Y}_{0|2k-1} \end{array} \right)$$

2.2.2 Algoritmos de projeção

No fundo, a idéia destes algoritmos de projeção é simples, a partir da equação do modelo no espaço de estado, (1.42):

$$\mathbf{Y}_f = \mathcal{O}_k \mathbf{X}_f + \mathcal{T}_k \mathbf{U}_f$$

fazendo a projeção ortogonal no subespaço linha do complemento ortogonal das entradas futuras, \mathbf{U}_f^\perp , obtemos:

$$\begin{aligned} \mathbf{Y}_f | \mathbf{U}_f^\perp &= \mathcal{O}_k \mathbf{X}_f | \mathbf{U}_f^\perp + \mathcal{T}_k \mathbf{U}_f | \mathbf{U}_f^\perp \\ &= \mathcal{O}_k \mathbf{X}_f | \mathbf{U}_f^\perp \end{aligned} \tag{2.13}$$

Esta equação (2.13) constitui o primeiro passo do algoritmo MOESP como veremos mais para frente; daqui temos que pode-se obter a matriz de observabilidade estendida, \mathcal{O}_k , com técnicas de decomposição matricial, por exemplo, com decomposição em valores singulares (SVD); o posto da matriz de observabilidade vai dar a ordem do sistema, n . Conhecendo a matriz de observabilidade e a ordem, podemos obter as matrizes do sistema.

Agora vamos analisar o seguinte, utilizando o teorema, e com a matriz \mathbf{W}_1 igual à matriz identidade, e \mathbf{W}_2 igual à projeção ortogonal no subespaço linha do complemento ortogonal das entradas futuras:

$$\mathbf{W}_1 = \mathbf{I} \in \mathfrak{R}^{l \times k}$$

$$\mathbf{W}_2 = \mathbf{P}_{\mathbf{U}_f^\perp} \in \mathfrak{R}^{l \times k}$$

e com as equações (2.3) e (2.12):

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \mathbf{I} \left[\mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p \right] \mathbf{P}_{\mathbf{U}_f^\perp}$$

Podemos mostrar que (ver apêndice B):

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \left[\mathbf{Y}_f | \mathbf{U}_f^\perp \right] \mathbf{P} \left[\mathbf{W}_p | \mathbf{U}_f^\perp \right] \quad (2.14)$$

das equações (2.9) e (2.13) tem-se:

$$\begin{aligned} \mathbf{Y}_f | \mathbf{U}_f^\perp &= \mathcal{O}_k \mathbf{X}_f | \mathbf{U}_f^\perp \\ &= \mathcal{O}_k \mathbf{L}_p \mathbf{W}_p | \mathbf{U}_f^\perp \end{aligned}$$

então, na equação (2.14) tem-se:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \mathbf{Y}_f | \mathbf{U}_f^\perp$$

pois vamos ter que a projeção do subespaço linha de $\left[\mathbf{W}_p | \mathbf{U}_f^\perp \right]$ ocorre nele mesmo.

Esta dedução nos indica que o teorema teo:subespacos no fundo é baseado nestes algoritmos de projeção.

Aqui apresentamos o caso para:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \mathbf{Y}_f | \mathbf{U}_f^\perp$$

devido as matrizes \mathbf{W}_1 e \mathbf{W}_2 definidas acima. Este caso corresponde ao algoritmo MOESP. Quando \mathbf{W}_1 e \mathbf{W}_2 tomam o valor da matriz identidade:

$$\mathbf{W}_1 = \mathbf{W}_2 = \mathbf{I} \in \mathfrak{R}^{l \times k}$$

temos o caso do algoritmo N4SID, onde:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = \bar{\mathbf{P}}_k = \mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p$$

como veremos mais para frente.

2.3 Matrizes do Sistema

Até agora temos, do teorema anterior, a matriz de observabilidade estendida e/ou os estados; podemos assim obter as matrizes do sistema por dois métodos: com algoritmos que utilizam os estados, ou com algoritmos que utilizam a matriz de observabilidade estendida.

2.3.1 Algoritmos que usam os estados

Do teorema (2.1), temos a ordem do sistema, equação (2.3), a matriz de observabilidade estendida, equação (2.6), e a seqüência de estados \mathbf{X}_k , equação (2.8).

Agora, definimos as seguintes matrizes de dados:

$$\mathbf{U}_{0|k} := \begin{pmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{N-1} \\ \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \dots & \mathbf{u}_N \\ \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 & \dots & \mathbf{u}_{N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_k & \mathbf{u}_{k+1} & \mathbf{u}_{k+2} & \dots & \mathbf{u}_{k+N-1} \end{pmatrix} \in \mathfrak{R}^{(k+1)mxN} \quad (2.15)$$

que podemos escrever como $\mathbf{U}_{0|k} = \mathbf{U}_p^+$,

e para instantes seguintes:

$$\mathbf{U}_{k+1|2k-1} := \begin{pmatrix} \mathbf{u}_{k+1} & \mathbf{u}_{k+2} & \mathbf{u}_{k+3} & \dots & \mathbf{u}_{k+N} \\ \mathbf{u}_{k+2} & \mathbf{u}_{k+3} & \mathbf{u}_{k+4} & \dots & \mathbf{u}_{k+N+1} \\ \mathbf{u}_{k+3} & \mathbf{u}_{k+4} & \mathbf{u}_{k+5} & \dots & \mathbf{u}_{k+N+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_{2k-1} & \mathbf{u}_{2k} & \mathbf{u}_{2k+1} & \dots & \mathbf{u}_{2k+N-2} \end{pmatrix} \in \mathfrak{R}^{(k-1)mxN} \quad (2.16)$$

que podemos escrever como $\mathbf{U}_{k+1|2k-1} = \mathbf{U}_f^-$.

Analogamente para as saídas, definimos:

$$\mathbf{Y}_{0|k} := \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{N-1} \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_N \\ \mathbf{y}_2 & \mathbf{y}_3 & \mathbf{y}_4 & \cdots & \mathbf{y}_{N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{y}_k & \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \cdots & \mathbf{y}_{k+N-1} \end{pmatrix} \in \mathfrak{R}^{(k+1)lxN} \quad (2.17)$$

que podemos escrever como $\mathbf{Y}_{0|k} = \mathbf{Y}_p^+$,

e para instantes seguintes:

$$\mathbf{Y}_{k+1|2k-1} := \begin{pmatrix} \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \mathbf{y}_{k+3} & \cdots & \mathbf{y}_{k+N} \\ \mathbf{y}_{k+2} & \mathbf{y}_{k+3} & \mathbf{y}_{k+4} & \cdots & \mathbf{y}_{k+N} \\ \mathbf{y}_{k+3} & \mathbf{y}_{k+4} & \mathbf{y}_{k+5} & \cdots & \mathbf{y}_{k+N+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{y}_{2k-1} & \mathbf{y}_{2k} & \mathbf{y}_{2k+1} & \cdots & \mathbf{y}_{2k+N-2} \end{pmatrix} \in \mathfrak{R}^{(k-1)lxN} \quad (2.18)$$

que podemos escrever como $\mathbf{Y}_{k+1|2k-1} = \mathbf{Y}_f^-$.

Referimo-nos à seguinte matriz:

$$\mathbf{W}_p^+ := \begin{pmatrix} \mathbf{U}_p^+ \\ \mathbf{Y}_p^+ \end{pmatrix} \in \mathfrak{R}^{(km+kl)lxN} \quad (2.19)$$

como matriz de dados \mathbf{W}_p^+ .

Como fizemos no teorema, podemos mostrar que:

$$\bar{\mathbf{P}}_{k-1} := \mathbf{Y}_f^- / \mathbf{U}_f^- \mathbf{W}_p^+ \quad (2.20)$$

$$= \mathcal{O}_{k-1} \mathbf{X}_{k+1} \quad (2.21)$$

onde:

$$\mathcal{O}_{k-1} := \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{k-2} \end{pmatrix} \quad (2.22)$$

Podemos obter os estados \mathbf{X}_{k+1} da equação (2.21), como:

$$\mathbf{X}_{k+1} = \mathcal{O}_{k-1}^\dagger \bar{\mathbf{P}}_{k-1} \quad (2.23)$$

Do sistema (2.1):

$$\begin{pmatrix} \mathbf{X}_{k+1} \\ \mathbf{Y}_{0|k} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{X}_k \\ \mathbf{U}_{0|k} \end{pmatrix} \quad (2.24)$$

conhecemos as entradas, as saídas e os estados (\mathbf{X}_k da equação (2.8) e \mathbf{X}_{k+1} da equação (2.23)), e podemos resolver este problema pelo método dos mínimos quadrados e assim obter as matrizes do sistema, \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} .

2.3.2 Algoritmos que usam a matriz de observabilidade estendida

Do teorema teo:subespacos, para começar este tipo de algoritmos conhecemos a ordem do sistema, equação (2.3), e a matriz de observabilidade estendida, equação (2.6). A diferença com os algoritmos que usam os estados para resolver o problema de identificação, onde obtemos os estados \mathbf{X}_k e \mathbf{X}_{k+1} para construir um problema linear que resolvemos aplicando mínimos quadrados para assim obter as matrizes \mathbf{A} , \mathbf{B} , \mathbf{C} e \mathbf{D} , é que aqui primeiro obtemos as matrizes \mathbf{A} e \mathbf{C} da matriz de observabilidade estendida, e depois \mathbf{B} e \mathbf{D} de uma equação linear.

Para atingir as duas primeiras matrizes do sistema, temos que a matriz de observabilidade estendida satisfaz a seguinte identidade; [36]:

$$\begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{k-2} \end{pmatrix} \mathbf{A} = \begin{pmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \vdots \\ \mathbf{CA}^{k-1} \end{pmatrix}$$

$$\mathcal{O}_{k-1} \mathbf{A} = \mathcal{O}_k^\dagger \quad (2.25)$$

onde \mathcal{O}_k^\dagger é a matriz de observabilidade estendida deslocada para acima, ou seja, sem o primeiro bloco $n \times l$.

A matriz \mathbf{A} pode ser obtida da equação (2.25) pelo método dos mínimos quadrados como:

$$\mathbf{A} = \mathcal{O}_{k-1}^\dagger \mathcal{O}_k^\dagger \quad (2.26)$$

e a matriz \mathbf{C} diretamente do primeiro bloco ($n \times l$) da matriz de observabilidade estendida \mathcal{O}_k :

$$\mathbf{C} = \mathcal{O}_k(1 : l; 1 : n) \quad (2.27)$$

Agora, para as duas matrizes restantes, \mathbf{B} e \mathbf{D} , podemos trabalhar a equação (1.42) assim:

$$\begin{aligned} \mathbf{Y}_f &= \mathcal{O}_k \mathbf{X}_f + \mathcal{T}_k \mathbf{U}_f \\ \mathbf{Y}_f - \mathcal{O}_k \mathbf{X}_f &= \mathcal{T}_k \mathbf{U}_f \\ \mathbf{Y}_f \mathbf{U}_f^\dagger - \mathcal{O}_k \mathbf{X}_f \mathbf{U}_f^\dagger &= \mathcal{T}_k \end{aligned} \quad (2.28)$$

Analisando as dimensões destas matrizes:

$$\mathbf{U}_f \in \mathfrak{R}^{k m \times N},$$

$$\mathbf{Y}_f \in \mathfrak{R}^{k l \times N},$$

$$\mathcal{O}_k \in \mathfrak{R}^{k l \times n},$$

$$\mathbf{X}_f \in \mathfrak{R}^{n \times N},$$

os produtos da equação anterior têm dimensão:

$$\mathbf{Y}_f \mathbf{U}_f^\dagger \in \mathfrak{R}^{k l \times k m}$$

$$\mathcal{O}_k \mathbf{X}_f \mathbf{U}_f^\dagger \in \mathfrak{R}^{k l \times k m}$$

que é igual a dimensão da matriz de Toeplitz, definida na equação (1.36) como:

$$\mathcal{T}_k := \begin{pmatrix} \mathbf{D} & 0 & 0 & \dots & 0 \\ \mathbf{CB} & \mathbf{D} & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{k-1} \mathbf{B} & \mathbf{CA}^{k-2} \mathbf{B} & \mathbf{CA}^{k-3} \mathbf{B} & \dots & \mathbf{D} \end{pmatrix} \in \mathfrak{R}^{k l \times k m}$$

Assim, da equação (2.28) podemos obter as matrizes \mathbf{B} e \mathbf{D} do sistema.

2.4 Método MOESP

Um dos principais Métodos de Subespaços para identificação de sistemas é o MOESP (**M**ultivariable **O**utput **E**rror **S**tate **sP**ace), desenvolvido por Verhaegen e Dewilde [45], [43], [44]; graficamente, este algoritmo segue o procedimento mostrado na figura 2.2. Este método começa com a projeção ortogonal das saídas passadas, \mathbf{Y}_p no subespaço do complemento ortogonal das entradas passadas, \mathbf{U}_p^\perp ; [32],

[18], [16], [7]:

$$\mathbf{Y}_p | \mathbf{U}_p^\perp \quad (2.29)$$

Mas, neste documento vamos apresentar o algoritmo MOESP baseado no livro de Katayama "*Subspace Methods for System Identification*"; [17], onde supomos a seguinte decomposição LQ:

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \quad (2.30)$$

temos que esta projeção ortogonal, equação (2.29), em termos da decomposição LQ, é dada por (ver apêndice B):

$$\mathbf{Y}_p | \mathbf{U}_p^\perp = \mathbf{L}_{22} \mathbf{Q}_2^T \quad (2.31)$$

Da equação do modelo estendido no espaço de estado, equação (1.38), temos que, [10]:

$$\mathbf{Y}_p = \mathcal{O}_k \mathbf{X}_p + \mathcal{T}_k \mathbf{U}_p$$

e da equação (2.30):

$$\mathbf{Y}_p = \mathbf{L}_{21} \mathbf{Q}_1^T + \mathbf{L}_{22} \mathbf{Q}_2^T$$

a seguir podemos escrever:

$$\mathbf{L}_{21} \mathbf{Q}_1^T + \mathbf{L}_{22} \mathbf{Q}_2^T = \mathcal{O}_k \mathbf{X}_p + \mathcal{T}_k \mathbf{U}_p \quad (2.32)$$

Aqui podemos observar que os termos do lado esquerdo da equação são ortogonais, e do lado direito temos uma soma direta dos subespaços linha dos estados e das entradas, isto pela suposição ii) do teorema (2.1).

Pós-multiplicando a equação (2.32) por \mathbf{Q}_2 obtemos:

$$\begin{aligned} \mathbf{L}_{21} \mathbf{Q}_1^T \mathbf{Q}_2 + \mathbf{L}_{22} \mathbf{Q}_2^T \mathbf{Q}_2 &= \mathcal{O}_k \mathbf{X}_p \mathbf{Q}_2 + \mathcal{T}_k \mathbf{U}_p \mathbf{Q}_2 \\ \mathbf{L}_{22} &= \mathcal{O}_k \mathbf{X}_p \mathbf{Q}_2 \end{aligned} \quad (2.33)$$

pois \mathbf{Q}_1 e \mathbf{Q}_2 são ortogonais, e de (2.30), temos $\mathbf{U}_p = \mathbf{L}_{11} \mathbf{Q}_1^T$.

Agora, usando a decomposição em valores singulares (SVD) desta equação (2.33):

$$\mathbf{L}_{22} = \mathcal{O}_k \mathbf{X}_p \mathbf{Q}_2 = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T \quad (2.34)$$

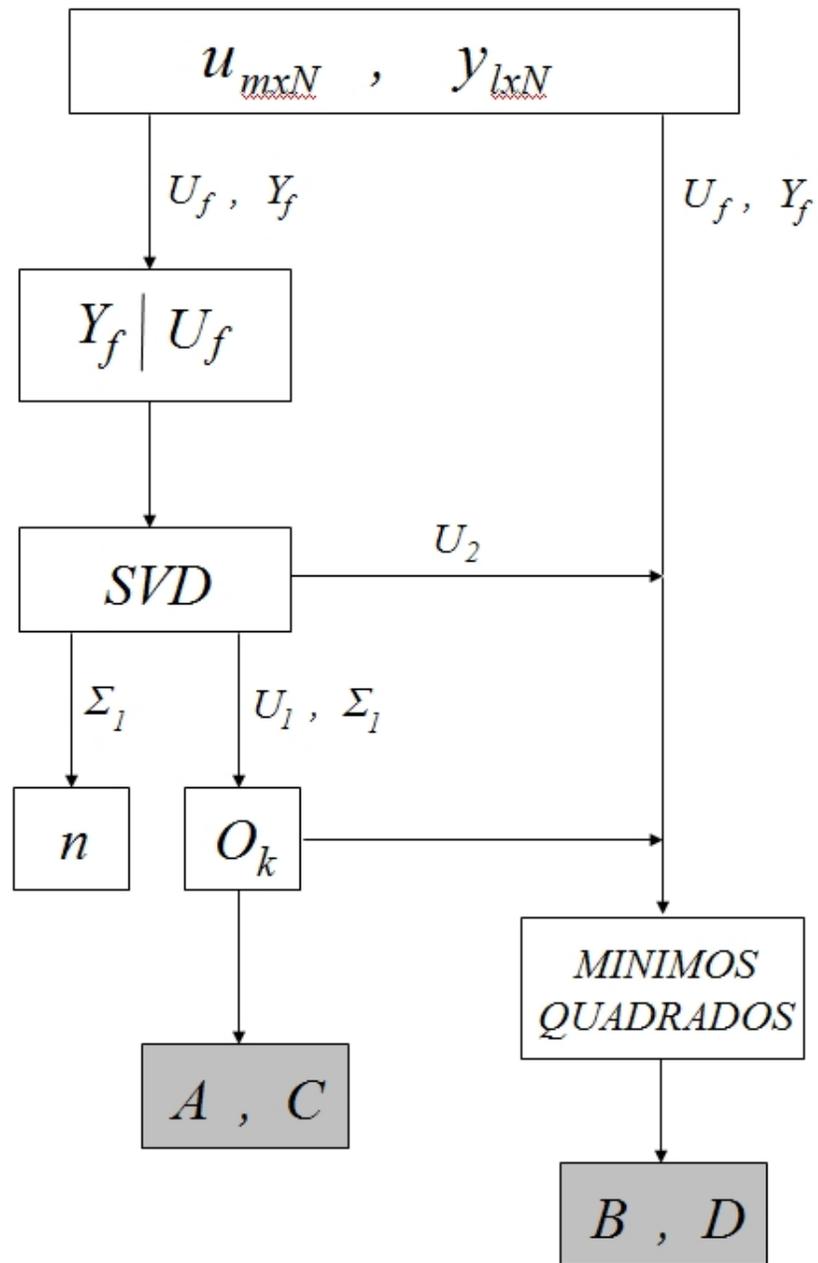


Fig. 2.2: Método MOESP referência.

onde, por meio da decomposição em valores singulares (SVD) obtemos:

$$\mathcal{O}_k = \mathbf{U}_1 \Sigma_1^{1/2} \quad (2.35)$$

e a dimensão do sistema é obtida fazendo uma inspeção aos valores singulares:

$$n = \dim \Sigma_1 \quad (2.36)$$

Com a matriz de observabilidade estendida, \mathcal{O}_k , as primeiras duas matrizes do sistema, \mathbf{A} e \mathbf{C} podem ser obtidas da equação (2.26):

$$\mathbf{A} = \mathcal{O}_{k-1}^\dagger \mathcal{O}_k^\dagger$$

e da equação (2.27):

$$\mathbf{C} = \mathcal{O}_k(1 : l; 1 : n)$$

Para computar as outras matrizes do sistema, \mathbf{B} e \mathbf{D} , temos, por outro lado, que pós-multiplicando a equação (2.32) por \mathbf{Q}_1 obtemos, [10]:

$$\mathbf{L}_{21} \mathbf{Q}_1^T \mathbf{Q}_1 + \mathbf{L}_{22} \mathbf{Q}_2^T \mathbf{Q}_1 = \mathcal{O}_k \mathbf{X}_p \mathbf{Q}_1 + \mathcal{T}_k \mathbf{U}_p \mathbf{Q}_1$$

$$\mathbf{L}_{21} = \mathcal{O}_k \mathbf{X}_p \mathbf{Q}_1 + \mathcal{T}_k \mathbf{U}_p \mathbf{Q}_1 \quad (2.37)$$

e pré-multiplicando por \mathbf{U}_2^T :

$$\mathbf{U}_2^T \mathbf{L}_{21} = \mathbf{U}_2^T \mathcal{O}_k \mathbf{X}_p \mathbf{Q}_1 + \mathbf{U}_2^T \mathcal{T}_k \mathbf{U}_p \mathbf{Q}_1$$

$$\mathbf{U}_2^T \mathbf{L}_{21} = \mathbf{U}_2^T \mathcal{T}_k \mathbf{L}_{11} \quad (2.38)$$

pois $\mathcal{O}_k = \mathbf{U}_1 \Sigma_1^{1/2}$, mas $\mathbf{U}_2^T \mathcal{O}_k = 0$ devido a que \mathbf{U}_1 e \mathbf{U}_2 são ortogonais, e de (2.30) temos que $\mathbf{U}_p = \mathbf{L}_{11} \mathbf{Q}_1^T$.

Reordenando esta equação, pois os termos \mathbf{L}_{11} , \mathbf{L}_{21} e \mathbf{U}_2 são conhecidos, e o termo \mathcal{T}_k , matriz de Toeplitz, é uma matriz linear em função dos termos faltantes, \mathbf{B} e \mathbf{D} , (pois \mathbf{A} e \mathbf{C} foram dados pela matriz de observabilidade estendida); portanto:

$$\mathbf{U}_2^T \mathcal{T}_k = \mathbf{U}_2^T \mathbf{L}_{21} \mathbf{L}_{11}^{-1} \quad (2.39)$$

Se definimos:

$$\mathbf{U}_2^T := \begin{bmatrix} \mathbf{L}_1 & \mathbf{L}_2 & \cdots & \mathbf{L}_k \end{bmatrix}$$

$$\mathbf{U}_2^T \mathbf{L}_{21} \mathbf{L}_{11}^{-1} := \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 & \cdots & \mathbf{M}_k \end{bmatrix}$$

onde:

$$\mathbf{L}_i \in \mathfrak{R}^{(kl-n) \times l}, i = 1, 2, \dots, k,$$

e:

$$\mathbf{M}_i \in \mathfrak{R}^{(kl-n) \times m}, i = 1, 2, \dots, k,$$

Podemos reescrever a equação linear com respeito a \mathbf{B} e \mathbf{D} como:

$$\begin{bmatrix} \mathbf{L}_1 & \bar{\mathbf{L}}_2 \mathcal{O}_{k-1} \\ \mathbf{L}_2 & \bar{\mathbf{L}}_3 \mathcal{O}_{k-2} \\ \vdots & \vdots \\ \mathbf{L}_{k-1} & \bar{\mathbf{L}}_k \mathcal{O}_1 \\ \mathbf{L}_k & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{D} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_{k-1} \\ \mathbf{M}_k \end{bmatrix} \quad (2.40)$$

onde:

$$\bar{\mathbf{L}}_i = \begin{bmatrix} \mathbf{L}_i & \mathbf{L}_{i+1} & \cdots & \mathbf{L}_k \end{bmatrix} \in \mathfrak{R}^{(kl-n) \times (k+1-i)l}, i = 2, 3, \dots, k,$$

Podemos estimar as matrizes \mathbf{B} e \mathbf{D} da equação (2.40) pelo método dos mínimos quadrados. Este Método de Subespaços MOESP, mostrado na figura 2.3, segue a lógica apresentada anteriormente como os *algoritmos que usam a matriz de observabilidade estendida*, na seção 2.3.2. O código em Matlab deste algoritmo está no apêndice A.

Uma alteração do algoritmo anterior (baseado em Katayama), que vamos chamar de *algoritmo MOESP modificado*, é apresentado a seguir. As matrizes \mathbf{A} e \mathbf{C} do sistema, como já dito, podem ser obtidas fazendo uso da matriz de observabilidade estendida, \mathcal{O}_k , com as equações (2.26) e (2.27). Agora, podemos ter um procedimento alternativo para calcular as matrizes do sistema \mathbf{B} e \mathbf{D} . A equação (2.39) pode ser expressa como:

$$\begin{aligned} \mathbf{U}_2^T \mathcal{T}_k &= \mathbf{U}_2^T \mathbf{L}_{21} \mathbf{L}_{11}^{-1} \\ \mathcal{T}_k &= \mathbf{L}_{21} \mathbf{L}_{11}^{-1} \end{aligned} \quad (2.41)$$

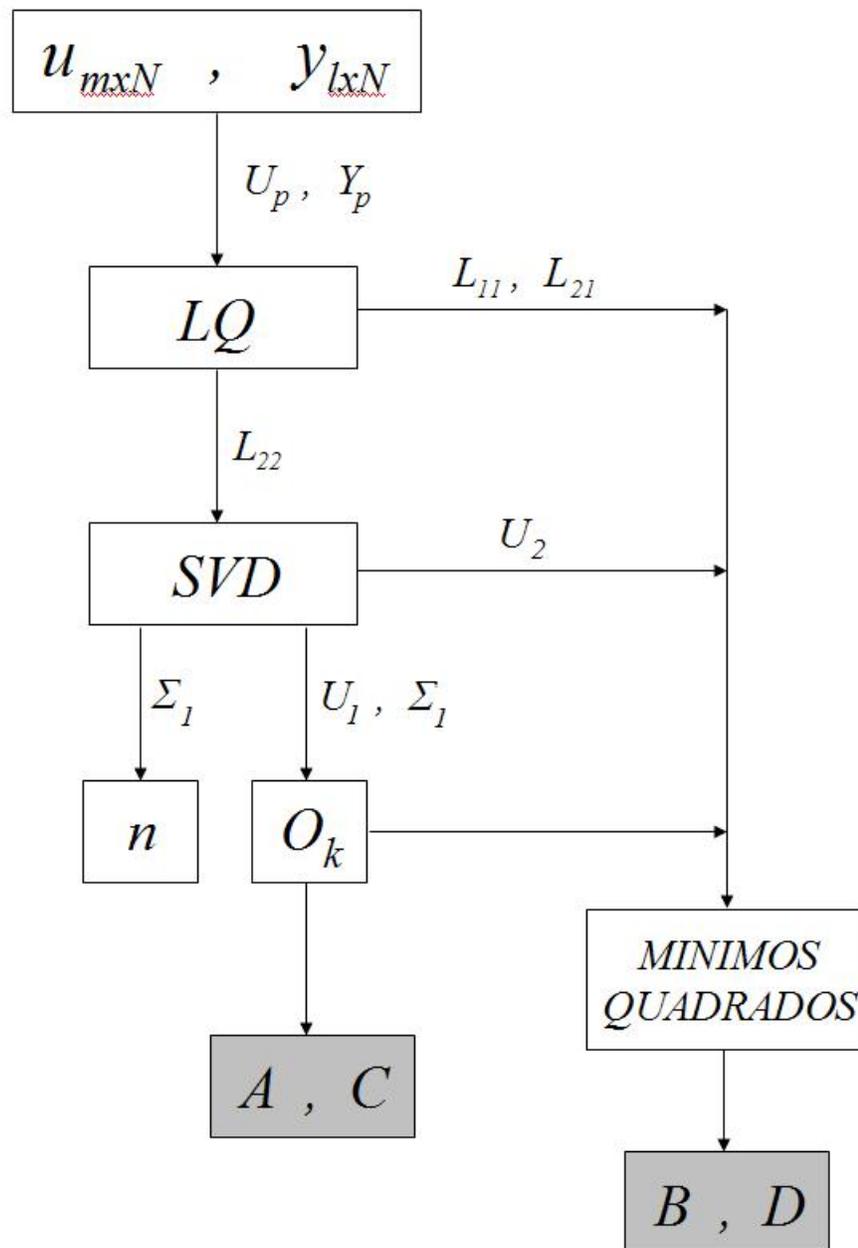


Fig. 2.3: Método MOESP baseado em Katayama.

O termo a esquerda corresponde à matriz de Toeplitz, dada pela equação (1.36):

$$\mathcal{T}_k := \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{k-1}\mathbf{B} & \mathbf{CA}^{k-2}\mathbf{B} & \mathbf{CA}^{k-3}\mathbf{B} & \cdots & \mathbf{D} \end{pmatrix} \in \mathfrak{R}^{kl \times km}$$

Fazendo um análise observamos que nesta matriz, o primeiro bloco coluna é composto pelas respostas ao impulso, equação (1.11):

$$\mathbf{G}_k := \begin{cases} \mathbf{D} & \text{se } k = 0 \\ \mathbf{CA}^{k-1}\mathbf{B} & \text{se } k \geq 1 \end{cases}$$

ou seja:

$$\text{O primeiro bloco coluna de } \mathbf{L}_{21}\mathbf{L}_{11}^{-1} = \begin{bmatrix} \mathbf{D} \\ \mathbf{CB} \\ \vdots \\ \mathbf{CA}^{k-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_k \end{bmatrix} \quad (2.42)$$

Podemos fazer uso das respostas ao impulso e, com elas, construir a matriz bloco de Hankel, equação (1.29), de dimensão $kl \times pm$:

$$\mathbf{H}_{k,p} := \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \mathbf{G}_3 & \cdots & \mathbf{G}_p \\ \mathbf{G}_2 & \mathbf{G}_3 & \mathbf{G}_4 & \cdots & \mathbf{G}_{p+1} \\ \mathbf{G}_3 & \mathbf{G}_4 & \mathbf{G}_5 & \cdots & \mathbf{G}_{p+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_k & \mathbf{G}_{k+1} & \mathbf{G}_{k+2} & \cdots & \mathbf{G}_{k+p-1} \end{bmatrix} \quad (2.43)$$

e como esta matriz satisfaz a equação:

$$\mathbf{H}_{k,p} := \mathcal{O}_k \mathcal{C}_p \quad (2.44)$$

onde a matriz bloco de Hankel e a matriz de observabilidade estendida são conhecidas; então podemos computar a matriz de atingibilidade estendida como:

$$\mathcal{C}_p = \mathcal{O}_k^\dagger \mathbf{H}_{k,p} \quad (2.45)$$

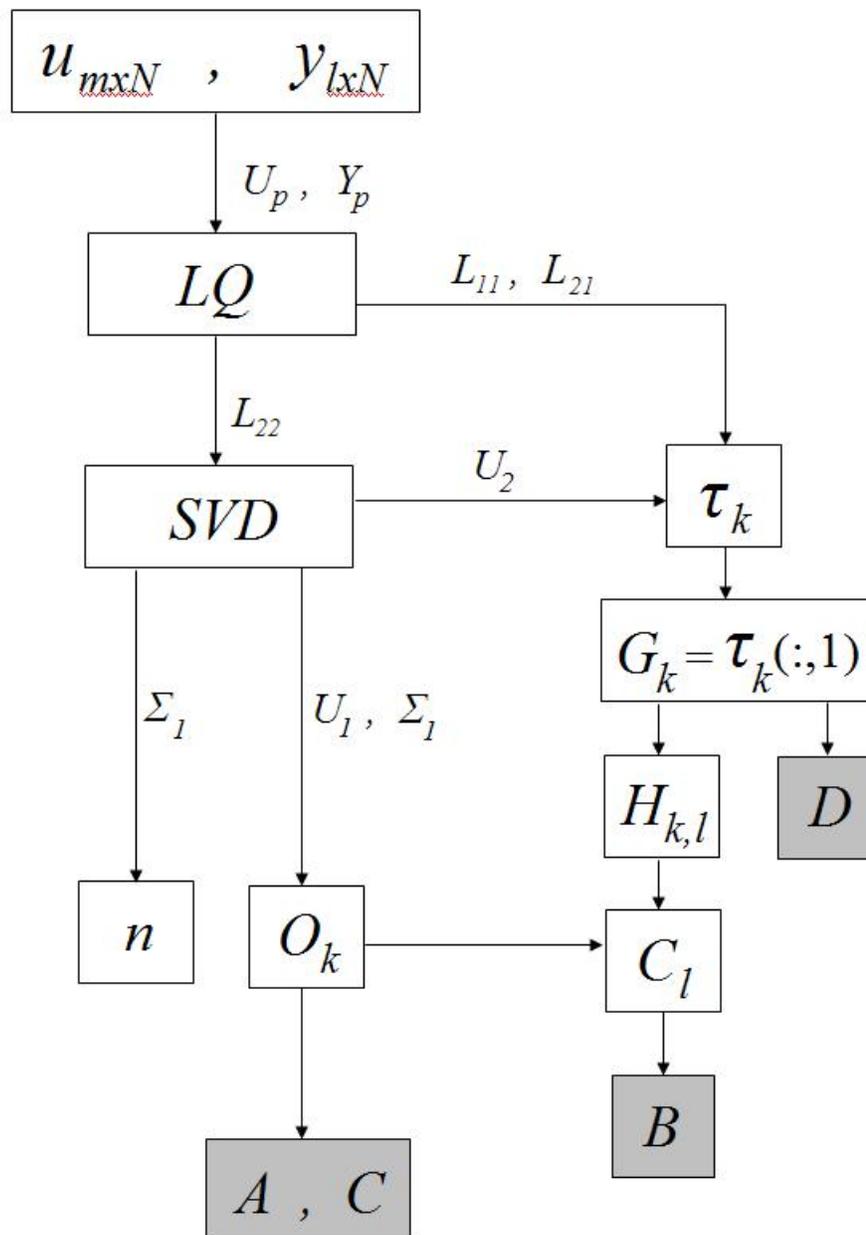


Fig. 2.4: Método MOESP modificado.

Assim, com esta matriz de atingibilidade, equação (1.35) dada como:

$$\mathcal{C}_k := \left(\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{k-1}\mathbf{B} \right) \in \mathfrak{R}^{n \times km}$$

a matriz faltante do sistema, \mathbf{B} , pode ser obtida do primeiro $n \times m$ bloco, já que a matriz \mathbf{D} foi dada pela resposta ao impulso no instante $k = 0$:

$$\begin{aligned} \mathbf{B} &= \mathcal{C}_p(1 : n, 1 : m) \\ \mathbf{D} &= \mathbf{G}_0 \end{aligned} \quad (2.46)$$

Este algoritmo modificado é mostrado na figura 2.4, e o seu código em Matlab está no apêndice A.

2.5 Método N4SID

Um segundo método de subespaços conhecido como N4SID (Numerical algorithm for Subspace State Space System Identification) foi desenvolvido por Van Overschee e De Moor, [37]; graficamente, este algoritmo segue o procedimento mostrado na figura 2.5. Este algoritmo é iniciado com a projeção oblíqua das saídas futuras, \mathbf{Y}_f , no subespaço das entradas e saídas passadas, \mathbf{W}_p , na direção das entradas futuras, \mathbf{U}_f ; [18], [16], [30],[7], [15]:

$$\mathbf{Y}_f /_{\mathbf{U}_f} \mathbf{W}_p \quad (2.47)$$

Mas, neste documento vamos apresentar o algoritmo N4SID também baseado no livro de Katayama "*Subspace Methods for System Identification*"; [17], onde supomos a seguinte decomposição LQ:

$$\begin{bmatrix} \mathbf{U}_f \\ \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{Y}_f \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} & \mathbf{0} \\ \mathbf{L}_{41} & \mathbf{L}_{42} & \mathbf{L}_{43} & \mathbf{L}_{44} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \\ \mathbf{Q}_3^T \\ \mathbf{Q}_4^T \end{bmatrix} \quad (2.48)$$

onde $\mathbf{L}_{44} = \mathbf{0}$, [18],[30], então podemos, por simplicidade, reescrever a equação (2.48) como, [10]:

$$\begin{bmatrix} \mathbf{U}_f \\ \mathbf{W}_p \\ \mathbf{Y}_f \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{0} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{Q}}_1^T \\ \overline{\mathbf{Q}}_2^T \\ \overline{\mathbf{Q}}_3^T \end{bmatrix} \quad (2.49)$$

onde:

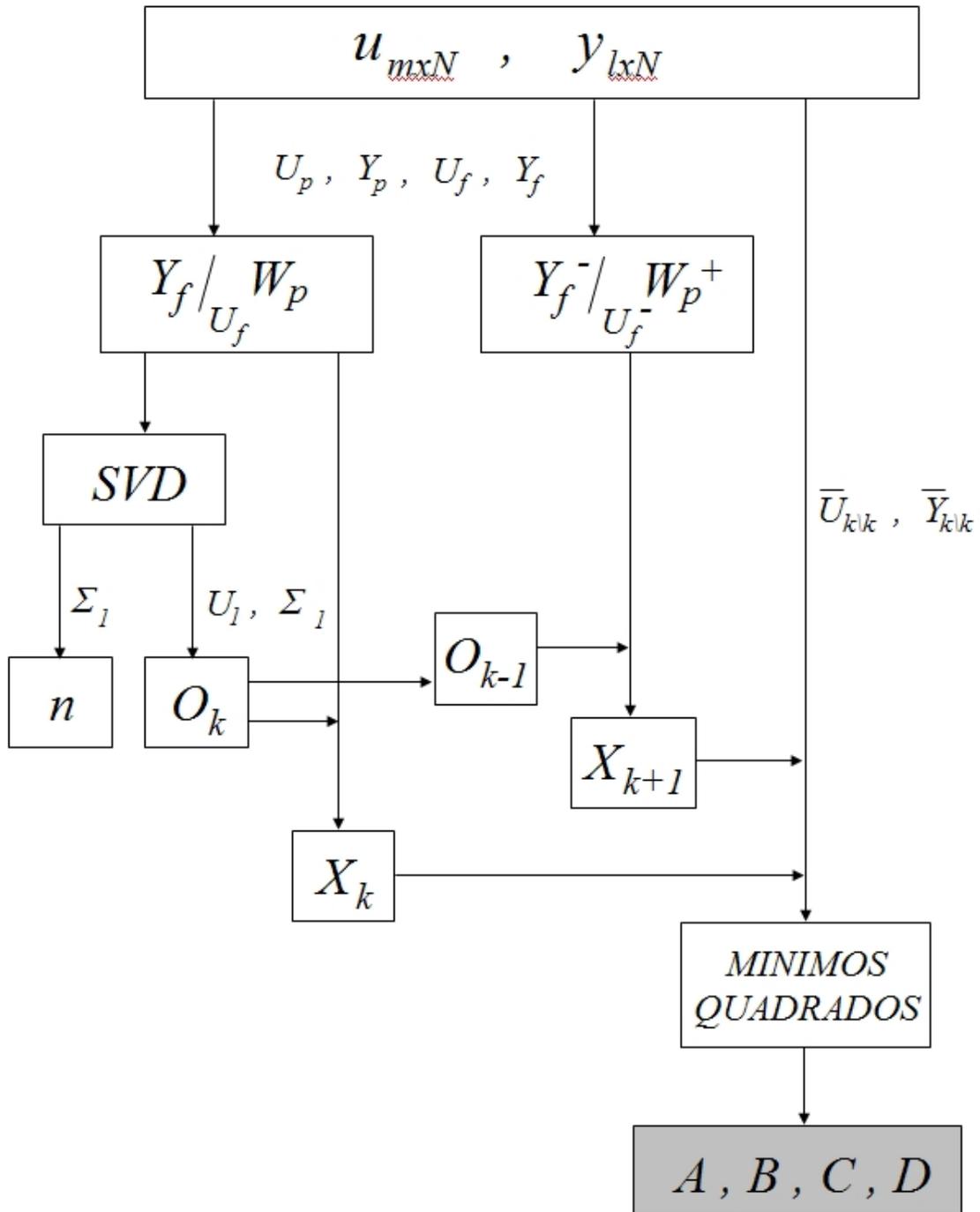


Fig. 2.5: Método N4SID referência.

$$\begin{aligned} \mathbf{R}_{11} &= [\mathbf{L}_{11}], & \mathbf{R}_{21} &= \begin{bmatrix} \mathbf{L}_{21} \\ \mathbf{L}_{31} \end{bmatrix}, & \mathbf{R}_{22} &= \begin{bmatrix} \mathbf{L}_{22} & \mathbf{0} \\ \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix}, & \mathbf{R}_{31} &= [\mathbf{L}_{41}], \\ \mathbf{R}_{32} &= [\mathbf{L}_{42} \quad \mathbf{L}_{43}], \\ \text{e:} \\ \bar{\mathbf{Q}}_1^T &= [\mathbf{Q}_1^T], & \bar{\mathbf{Q}}_2^T &= \begin{bmatrix} \mathbf{Q}_2^T \\ \mathbf{Q}_3^T \end{bmatrix}, & \bar{\mathbf{Q}}_3^T &= [\mathbf{Q}_4^T]. \end{aligned}$$

Da equação (2.49) temos que:

$$\begin{aligned} \mathbf{U}_f &= \mathbf{R}_{11} \bar{\mathbf{Q}}_1^T \\ \bar{\mathbf{Q}}_1^T &= \mathbf{R}_{11}^{-1} \mathbf{U}_f \end{aligned} \quad (2.50)$$

além disso, temos que:

$$\begin{aligned} \mathbf{W}_p &= \mathbf{R}_{21} \bar{\mathbf{Q}}_1^T + \mathbf{R}_{22} \bar{\mathbf{Q}}_2^T \\ \mathbf{R}_{22} \bar{\mathbf{Q}}_2^T &= \mathbf{W}_p - \mathbf{R}_{21} \bar{\mathbf{Q}}_1^T \\ \bar{\mathbf{Q}}_2^T &= \mathbf{R}_{22}^\dagger (\mathbf{W}_p - \mathbf{R}_{21} \bar{\mathbf{Q}}_1^T) \end{aligned} \quad (2.51)$$

e finalmente:

$$\mathbf{Y}_f = \mathbf{R}_{31} \bar{\mathbf{Q}}_1^T + \mathbf{R}_{32} \bar{\mathbf{Q}}_2^T \quad (2.52)$$

Substituindo $\bar{\mathbf{Q}}_1^T$ e $\bar{\mathbf{Q}}_2^T$, equações (2.50) e (2.51) respectivamente, em (2.52) obtemos:

$$\begin{aligned} \mathbf{Y}_f &= \mathbf{R}_{31} \mathbf{R}_{11}^{-1} \mathbf{U}_f + \mathbf{R}_{32} \mathbf{R}_{22}^\dagger (\mathbf{W}_p - \mathbf{R}_{21} \mathbf{R}_{11}^{-1} \mathbf{U}_f) \\ &= \mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{W}_p + (\mathbf{R}_{31} - \mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{R}_{21}) \mathbf{R}_{11}^{-1} \mathbf{U}_f \end{aligned} \quad (2.53)$$

Lembrando o modelo estendido no espaço de estado, equação (1.42):

$$\mathbf{Y}_f = \mathcal{O}_k \mathbf{X}_f + \mathcal{T}_k \mathbf{U}_f$$

e comparando-o com a última expressão temos que:

$$\mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{W}_p = \mathcal{O}_k \mathbf{X}_f \quad (2.54)$$

e:

$$\mathcal{T}_k = (\mathbf{R}_{31} - \mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{R}_{21}) \mathbf{R}_{11}^{-1} \quad (2.55)$$

Assim, da equação (2.5):

$$\bar{\mathbf{P}}_k = \mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p = \mathcal{O}_k \mathbf{X}_f$$

e com a equação (2.54), tem-se que esta projeção oblíqua, equação (2.47), em termos da decomposição LQ, é dada por:

$$\mathbf{Y}_f / \mathbf{U}_f \mathbf{W}_p = \mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{W}_p \quad (2.56)$$

Usando a seguinte decomposição em valores singulares (SVD):

$$\mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{W}_p = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T \quad (2.57)$$

podemos obter as equações:

$$\mathcal{O}_k = \mathbf{U}_1 \Sigma_1^{1/2} \quad (2.58)$$

e:

$$\mathbf{X}_f = \Sigma_1^{1/2} \mathbf{V}_1^T \quad (2.59)$$

e a dimensão do sistema dada por:

$$n = \dim \Sigma_1 \quad (2.60)$$

Este método segue a lógica apresentada na parte dos *algoritmos que usam os estados*; [19]; mas para evitar o procedimento mostrado anteriormente para obter os estados \mathbf{X}_{k+1} , ou seja, as equações (2.20) à (2.23), podemos manipular os dados já conhecidos da seguinte forma; com o vetor de estados, equação (2.59), dado como:

$$\mathbf{X}_k = \mathbf{X}_f = \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}(k+N-2) & \mathbf{x}(k+N-1) \end{bmatrix} \quad (2.61)$$

como definido anteriormente, equação (1.41), definimos as seguintes matrizes:

$$\bar{\mathbf{X}}_{k+1} := \begin{bmatrix} \mathbf{x}_{k+1} & \mathbf{x}_{k+2} & \cdots & \mathbf{x}(k+N-1) \end{bmatrix} \quad (2.62)$$

$$\bar{\mathbf{X}}_k := \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} & \cdots & \mathbf{x}(k+N-2) \end{bmatrix} \quad (2.63)$$

$$\bar{\mathbf{U}}_{k|k} := \begin{bmatrix} \mathbf{u}_k & \mathbf{u}_{k+1} & \cdots & \mathbf{u}(k+N-2) \end{bmatrix} \quad (2.64)$$

$$\bar{\mathbf{Y}}_{k|k} := \begin{bmatrix} \mathbf{y}_k & \mathbf{y}_{k+1} & \cdots & \mathbf{y}(k+N-2) \end{bmatrix} \quad (2.65)$$

Ou seja, com o vetor de estados obtido \mathbf{X}_k , equação (2.61), de dimensão $n \times N$, temos a possibilidade de rearranjar as matrizes para deixá-las com dimensões $n \times (N-1)$ e assim ter \mathbf{X}_{k+1} ; e com isto

o que queremos finalmente é obter o sistema no espaço de estado, equação (2.24), expresso como:

$$\begin{bmatrix} \bar{\mathbf{X}}_{k+1} \\ \bar{\mathbf{Y}}_{k|k} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}}_{k+1} \\ \bar{\mathbf{U}}_{k|k} \end{bmatrix} \quad (2.66)$$

Este é um sistema linear que tem como incógnitas as matrizes \mathbf{A} , \mathbf{B} , \mathbf{C} , e \mathbf{D} , as quais podem ser estimadas aplicando, como já dito, o método dos mínimos quadrados:

$$\begin{bmatrix} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hat{\mathbf{C}} & \hat{\mathbf{D}} \end{bmatrix} = \left(\begin{bmatrix} \bar{\mathbf{X}}_{k+1} \\ \bar{\mathbf{Y}}_{k|k} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}}_{k+1} \\ \bar{\mathbf{U}}_{k|k} \end{bmatrix}^T \right) \left(\begin{bmatrix} \bar{\mathbf{X}}_k \\ \bar{\mathbf{U}}_{k|k} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}}_{k+1} \\ \bar{\mathbf{U}}_{k|k} \end{bmatrix}^T \right)^{-1} \quad (2.67)$$

Na figura 2.6 é apresentado o algoritmo N4SID explicado.

Uma forma alternativa de obter as matrizes do sistema é apresentada a seguir:

Como no método MOESP, fazendo uso da matriz de observabilidade estendida, equação (2.58), podemos obter as matrizes \mathbf{A} e \mathbf{C} , equações (2.26) e (2.27), respectivamente. Agora, para computar as matrizes \mathbf{B} e \mathbf{D} , podemos ter o procedimento dado a seguir:

Na comparação dos termos, equação (2.55), temos que:

$$\mathcal{T}_k = \left(\mathbf{R}_{31} - \mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{R}_{21} \right) \mathbf{R}_{11}^{-1}$$

Esta é a matriz de Toeplitz, dada pela equação (1.36):

$$\mathcal{T}_k := \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{k-1} \mathbf{B} & \mathbf{CA}^{k-2} \mathbf{B} & \mathbf{CA}^{k-3} \mathbf{B} & \dots & \mathbf{D} \end{pmatrix} \in \mathfrak{R}^{kl \times km}$$

Analisando esta matriz, vemos que o primeiro bloco coluna é composto pelas respostas ao impulso, equação (1.11):

$$\mathbf{G}_k := \begin{cases} \mathbf{D} & \text{se } k = 0 \\ \mathbf{CA}^{k-1} \mathbf{B} & \text{se } k \geq 1 \end{cases}$$

ou seja:

$$\begin{aligned} \text{O primeiro bloco coluna de} \\ \left(\mathbf{R}_{31} - \mathbf{R}_{32} \mathbf{R}_{22}^\dagger \mathbf{R}_{21} \right) \mathbf{R}_{11}^{-1} \end{aligned} = \begin{bmatrix} \mathbf{D} \\ \mathbf{CB} \\ \vdots \\ \mathbf{CA}^{k-1} \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_k \end{bmatrix} \quad (2.68)$$

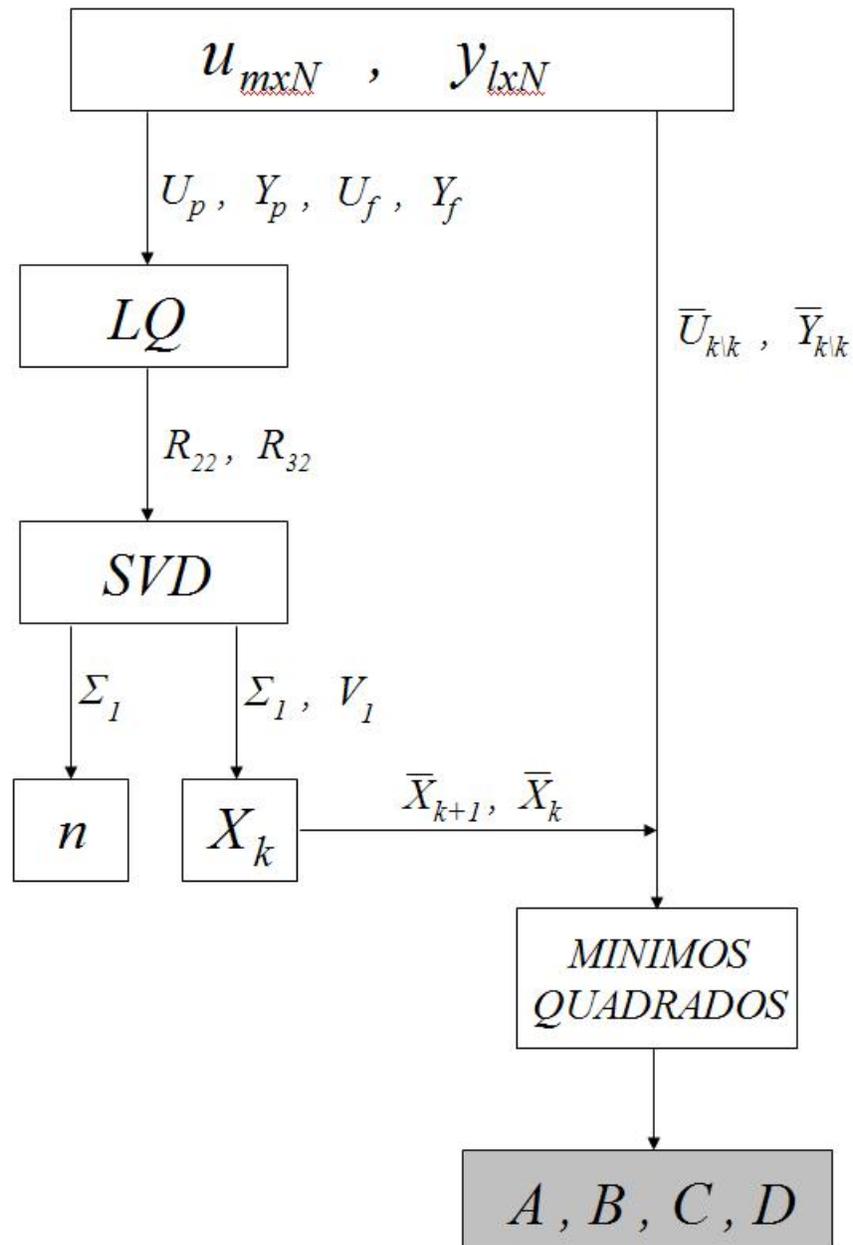


Fig. 2.6: Método N4SID baseado em Katayama.

Podemos fazer uso das respostas ao impulso e, com elas, construir a matriz bloco de Hankel, equação (1.29), de dimensão $kl \times pm$:

$$\mathbf{H}_{k,p} := \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \mathbf{G}_3 & \cdots & \mathbf{G}_p \\ \mathbf{G}_2 & \mathbf{G}_3 & \mathbf{G}_4 & \cdots & \mathbf{G}_{p+1} \\ \mathbf{G}_3 & \mathbf{G}_4 & \mathbf{G}_5 & \cdots & \mathbf{G}_{p+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_k & \mathbf{G}_{k+1} & \mathbf{G}_{k+2} & \cdots & \mathbf{G}_{k+p-1} \end{bmatrix} \quad (2.69)$$

e como esta matriz satisfaz a equação:

$$\mathbf{H}_{k,p} := \mathcal{O}_k \mathcal{C}_p \quad (2.70)$$

onde a matriz bloco de Hankel e a matriz de observabilidade estendida são conhecidas; então podemos computar a matriz de atingibilidade estendida como:

$$\mathcal{C}_p = \mathcal{O}_k^\dagger \mathbf{H}_{k,p} \quad (2.71)$$

Assim, com esta matriz de atingibilidade, dada como, equação (1.35):

$$\mathcal{C}_k := \left(\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{k-1}\mathbf{B} \right) \in \mathfrak{R}^{n \times km}$$

a última incógnita do sistema, a matriz \mathbf{B} , pode ser obtida do primeiro $n \times m$ bloco, pois a matriz \mathbf{D} é dada pela resposta ao impulso no instante $k = 0$:

$$\begin{aligned} \mathbf{B} &= \mathcal{C}_p(1 : n, 1 : m) \\ \mathbf{D} &= \mathbf{G}_0 \end{aligned} \quad (2.72)$$

Vamos chamar esta forma alternativa do algoritmo N4SID implementado por Katayama, apresentado na figura 2.7, de *algoritmo N4SID modificado*; o código em Matlab deste algoritmo pode ser encontrado no apêndice A.

2.6 Conclusões

Fazendo uma comparação dos Métodos de Subespaços com os Métodos Clássicos de Erro de Predição (PEM) temos como uma das principais diferenças a importância dos estados para o procedimento de identificação, além de outras diferenças ou vantagens como no chamado problema de parametrização, que é particularmente não trivial para sistemas de múltiplas saídas; o modelo nos

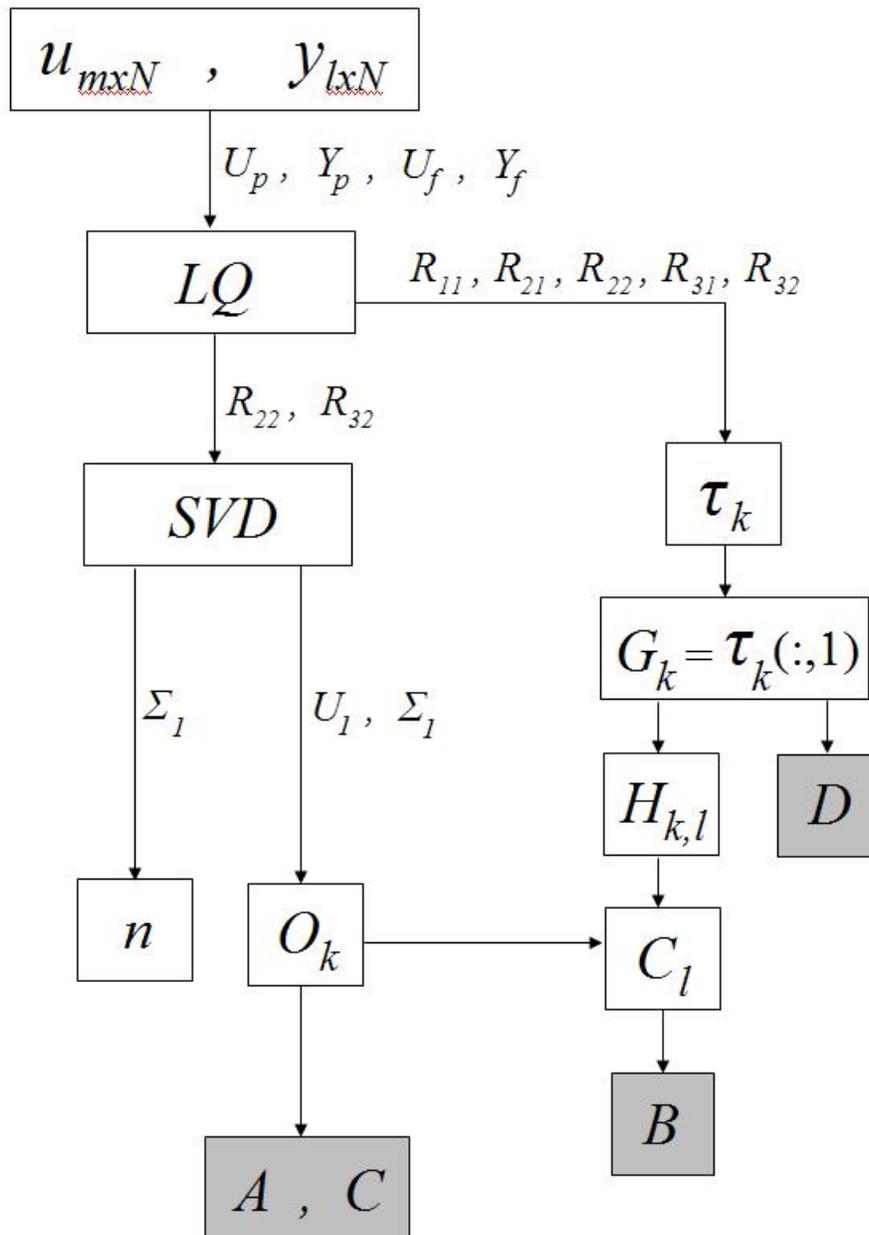


Fig. 2.7: Método N4SID modificado.

Métodos de Subespaços é parametrizado pelo modelo no espaço de estado, e a ordem é decidida no procedimento de identificação (comumente utilizando a decomposição em valores singulares). Está pode ser considerada como uma das grandes vantagens dos métodos MOESP e N4SID quando comparados com os PEM, pois o uso da decomposição em valores singulares (SVD) permite analisar e obter mais facilmente uma estimativa da ordem do sistema. Além disso, não há muita complicação nos algoritmos de Subespaços para passar de sistemas SISO a sistemas MIMO. O estado inicial com valor diferente de zero não apresenta problemas adicionais em termos de parametrização, típicos em alguns Métodos Clássicos (PEM). Os Métodos de Subespaços são rápidos e apresentam boa exatidão em diversas aplicações práticas, pois não contêm técnicas de otimização não linear, o que leva a melhoras nas propriedades numéricas de velocidade e exatidão; por outra parte, os Métodos Clássicos de Erro de Predição (PEM) são mais exatos do que os Métodos de Subespaços, mas se não tem-se uma boa estimativa inicial dos parâmetros do modelo, a solução obtida pode diferir da solução ótima. Assim, pode-se obter um bom modelo inicial com Métodos de Subespaços de maneira rápida, e se precisa-se de exatidão, utilizar a seguir métodos de erro de predição. Isto indica que os Métodos Clássicos e os Métodos de Subespaços são, na realidade, complementares. Deixando de lado os Métodos Clássicos (PEM), os Métodos de Subespaços têm a vantagem que eles só requerem dados de entrada - saída e nenhum conhecimento a priori. Adicionalmente, estes métodos são baseados em algoritmos de álgebra linear numérica, estáveis e não iterativos como por exemplo a decomposição QR (ou QL), e SVD. Um problema da identificação por subespaços é que se pode perder uma interpretação física do modelo; ou seja, ainda que o estado seja uma variável física, nem sempre é possível associá-lo diretamente como uma variável do processo. Além disso, precisa-se de grandes quantidades de dados para obter modelos exatos. As modificações feitas nos algoritmos MOESP e N4SID permitem obter as matrizes **B** e **D** por meio das respostas ao impulso, fazendo uso das matrizes de Toeplitz, de Hankel e de atingibilidade.

Capítulo 3

MÉTODOS DE SUBESPAÇOS: EXPERIMENTAÇÕES

3.1 Introdução

Existem diversas implementações de Métodos de Subespaços para Identificação de Sistemas; estas apresentam muitas diferenças, significativamente na implementação prática; por exemplo, nas referências [27], [8], [39], têm-se estudos comparativos entre os Métodos Clássicos de Erro de Predição (PEM) e os Métodos de Subespaços; nestas são apresentadas as vantagens de uns frente aos outros, comparações de complexidade computacional para cada algoritmo, informação a priori requerida do sistema do qual quer-se fazer a identificação, etc; um pouco mais enfocadas para nosso trabalho, têm-se as referências [14], [27], [19], [16], [6], aonde estudos comparativos entre os algoritmos de subespaços mais comuns, MOESP, N4SID, CVA, são feitos; estes algoritmos são comparados nos aspectos de complexidade computacional, erro de predição e erro de simulação. Uma das conclusões gerais destes estudos é que os três algoritmos diferem muito na complexidade computacional; por exemplo, a conclusão de um dos trabalhos é que o método CVA apresenta um número significativamente menor de operações em ponto flutuante (Floating point Operations Per Second, FLOPS) comparado com os outros dois métodos. Outros estudos foram propostos para procurar a causa desta diferença, e examinar as possíveis reduções computacionais necessárias para os algoritmos MOESP e N4SID.

Em nosso trabalho queremos analisar estes últimos dois algoritmos de Subespaços: o MOESP e o N4SID, cujas bases foram apresentadas no capítulo 2; mostrar algumas variações nas suas implementações, e fazer uma análise do seu comportamento com os conjuntos de dados que aqui vão ser tratados.

Na hora de procurar estes dados para fazer os experimentos apresentam-se diversos problemas;

por exemplo, no começo nós queríamos trabalhar com dados de algum tipo de aeronave: um avião, helicóptero ou nave espacial, mas não conseguimos um conjunto de dados de entrada - saída deste tipo, só algumas series temporais; [29], [12]. Encontramos uma livraria *on line* chamada DAISY (DAtabase for Identification of SYstems), da Katholieke Universiteit Leuven, implementada pelo Professor Bart De Moor, [2], que contém conjuntos de dados em diversas áreas: processos industriais, sistemas mecânicos, sistemas biomédicos, series temporais, etc. Mas a idéia de trabalhar com estes dados foi abandonada devido a que existem documentos que tratam com estes dados; [14], [16], [27].

Com o propósito de que as pessoas que peguem este trabalho como referência consigam refazer os experimentos e entender melhor o tema aqui tratado, escolhemos dois benchmarks, de igual ordem, de duas referências diferentes; [10], [3]. A idéia de trabalhar com estes conjuntos é permitir a reprodutibilidade, pois nas áreas de análise de dados, identificação de sistemas e processamento de sinais, este aspecto é geralmente ignorado, já que, freqüentemente, apresentam-se artigos ou documentos onde algum algoritmo é aplicado, e conclui-se que esse é melhor que tal outro algoritmo para determinado conjunto de dados. Mas, comumente, o conjunto de dados usado no documento não é acessível ou não está facilmente disponível. Assim, se nós queremos reproduzir um experimento de determinado artigo, vamos encontrar muitas dificuldades, ou simplesmente não vamos conseguir fazê-lo.

Os algoritmos apresentados neste trabalho foram implementados em Matlab, num computador Intel Core 2 de 1.86 GHz, e 1 GB de RAM, sempre tentando manter o uso da CPU e o uso do arquivo de paginação em valores constantes durante os experimentos. Os códigos em Matlab destes algoritmos estão disponíveis no Apêndice A.

3.2 Benchmark 1

3.2.1 Introdução

Para realizar esta análise dos algoritmos, primeiro tomamos um benchmark, de ordem 4, de 2 entradas e 2 saídas, utilizado em um trabalho de comparação de Métodos de Subespaços e de Métodos Clássicos de De Moor e Van Overschee, referência [10]. Um conjunto de 6000 dados de entradas - saídas foi gerado usando o modelo (benchmark); estes conjuntos foram divididos em dois subconjuntos: um primeiro usado para o procedimento de identificação (que corresponde aos primeiros 5000 dados, N), e o subconjunto restante para o procedimento de validação (os outros 1000 dados). O

benchmark é dado pelas seguintes matrizes:

$$A = \begin{bmatrix} 0.6030 & 0.6030 & 0 & 0 \\ -0.6030 & 0.6030 & 0 & 0 \\ 0 & 0 & -0.6030 & -0.6030 \\ 0 & 0 & 0.6030 & -0.6030 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.1650 & -0.6965 \\ 0.6268 & 1.6961 \\ 0.0751 & 0.0591 \\ 0.3516 & 1.7971 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.2641 & -1.4462 & 1.2460 & 0.5774 \\ 0.8717 & -0.7012 & -0.6390 & -0.3600 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

Nas figuras 3.1 e 3.2 temos o conjunto de dados de entrada - saída usados no procedimento de identificação (onde u_1 e u_2 correspondem as entradas, e y_1 e y_2 as saídas).

3.2.2 Algoritmos

No começo, a nossa idéia era comparar os Métodos Clássicos (de erro de predição, PEM) com os Métodos de Subespaços; das referências [39], [40], [41], temos que, na realidade, estes métodos não devem ser confrontados, pois eles, além de serem muito diferentes, são complementares. Como dito no final do capítulo 2, cada um dos métodos apresenta suas vantagens e desvantagens, e um bom caminho para identificar o modelo de um sistema, pode ser começar aplicando Métodos de Subespaços, para obter uma boa estimativa inicial do modelo de forma rápida, e utilizar a seguir Métodos Clássicos que são mais exatos. Assim, a nossa idéia mudou para a de comparar uns algoritmos de referência (algoritmos base) com os algoritmos implementados via decomposição LQ, apresentados por Katayama, [17]. Mas na hora de implementar a versão do algoritmo N4SID baseado no Katayama, os resultados não deram certo. Começamos a analisar detalhadamente o procedimento, fazendo passo a passo o algoritmo, comparamos resultados intermédios (como, por exemplo, as matrizes de observabilidade e de atingibilidade), e observamos que até certo ponto dava tudo correto. Desse ponto para frente, e fazendo uso dos resultados obtidos na análise detalhada, implementamos outra forma para acabar o algoritmo, e assim obter as matrizes do sistema. Deste modo, tentando implementar o algoritmo N4SID via decomposição LQ, conseguimos uma forma alternativa para obter as matrizes do

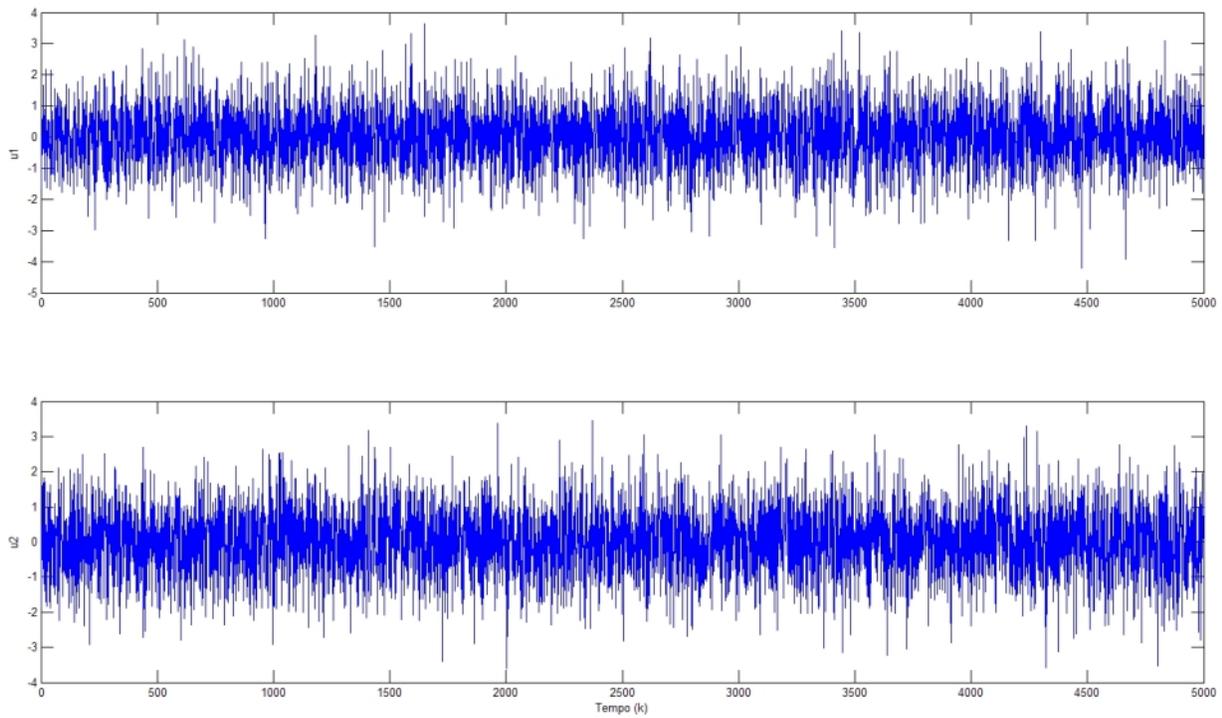


Fig. 3.1: Seqüência de dados de entrada.

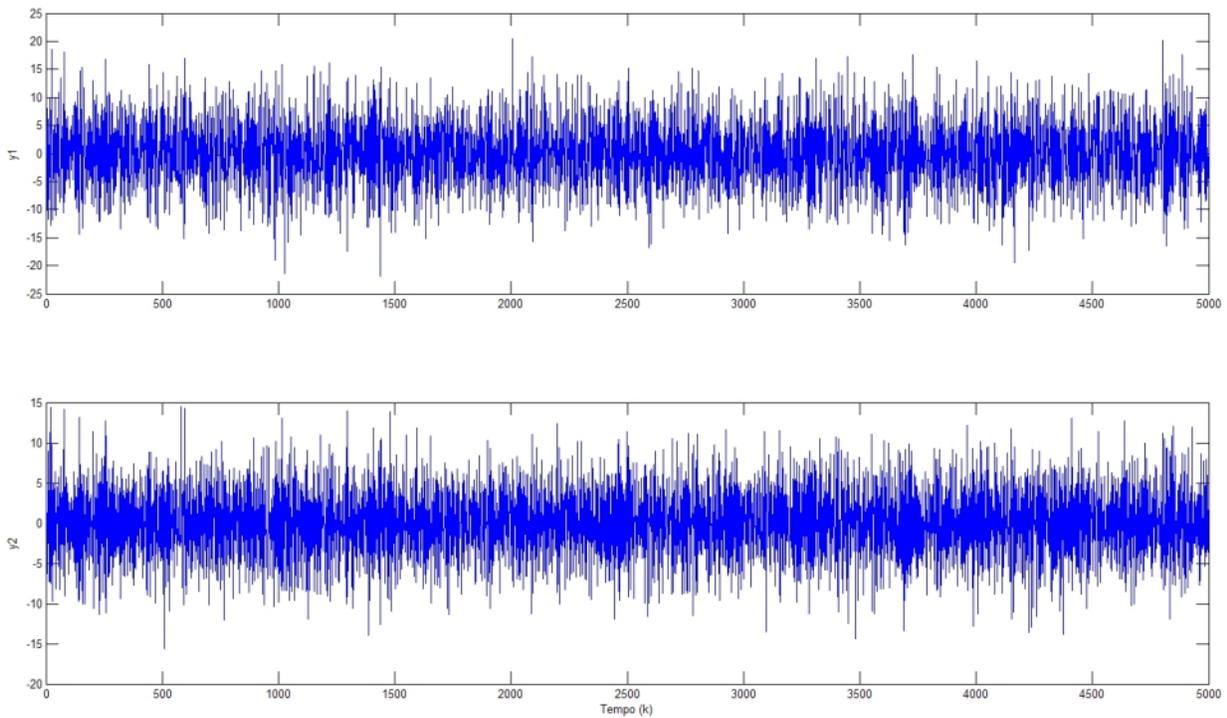


Fig. 3.2: Seqüência de dados de saída.

sistema, fazendo uso da matriz de Toeplitz, respostas ao impulso, matriz de Hankel e matrizes de observabilidade e atingibilidade, forma que foi apresentada no capítulo 2. Depois de implementar esta modificação no algoritmo N4SID, implementamo-la também para o outro algoritmo aqui tratado, o MOESP. Temos como algoritmos de referência para fazer as comparações, os algoritmos mostrados nos trabalhos [10], e [22], implementados nas funções *subid* e *n4sid* do Matlab versão 7.0 (esta última função com valor *MOESP* na opção *N4Weight*); a idéia é tomar uns algoritmos que têm sido usados em documentos anteriores, em nosso caso, os algoritmos que chamamos de referência, e compará-los com os apresentados neste trabalho, no capítulo 2, que têm diferenças como a implementação via decomposição LQ (baseados em Katayama) e exibem algumas modificações na obtenção das matrizes do sistema (algoritmos que chamamos de modificados).

3.2.3 Parâmetros de identificação

Os parâmetros necessários nos Métodos de Subespaços, k e n , que são a dimensão das matrizes bloco de Hankel, e a ordem do sistema a identificar, foram ajustados nos valores de $k = 10$ e $n = 4$ respectivamente; este ultimo valor é a ordem real do sistema original (*benchmark*). Esta ordem pode ser obtida no passo da decomposição em valores singulares, como consegue-se observar nas figuras 3.3 à 3.7. Estes gráficos apresentam os valores singulares para os algoritmos MOESP baseado em Katayama, MOESP modificado, N4SID referência, e N4SID modificado, respectivamente. O algoritmo MOESP referência não nos entrega um gráfico para escolher a ordem analisando os valores singulares, pois ele tem a opção de obter o modelo para cada uma das ordens de 0 até 10, e escolher o melhor deles, ou a opção de introduzirmos o valor do modelo desejado. Na figura 3.3 temos os valores singulares do algoritmo MOESP baseado em Katayama graficados com a função de Matlab *plot*; mas na figura 3.4 observa-se mais claramente a diferença entre os valores singulares, graficados com a função de Matlab *bar*, onde a ordem escolha para o sistema é $n = 4$. Devido a isto, os algoritmos implementados foram trabalhados com este último tipo de gráficos, chamados gráficos de barras. Por parte dos algoritmos N4SID, o algoritmo referência apresenta os valores singulares em uma escala logarítmica como pode-se observar na figura 3.6, e o algoritmo N4SID modificado que nós implementamos foi trabalhado nesta parte de obtenção da ordem do sistema, também com gráficos de barras, onde a ordem escolhida para o sistema também tem valor $n = 4$. Os valores selecionados de $k = 10$ e $n = 4$ satisfazem o critério dos Métodos de Subespaços, $kl > n$. Aqui observa-se uma das vantagens que apresentam os Métodos de Subespaços, pois eles requerem mínima informação *a priori*, k e n (n é obtido em um passo intermedio do algoritmo), e isto indica que o usuário não precisa conhecer muita informação sobre o sistema a ser identificado.

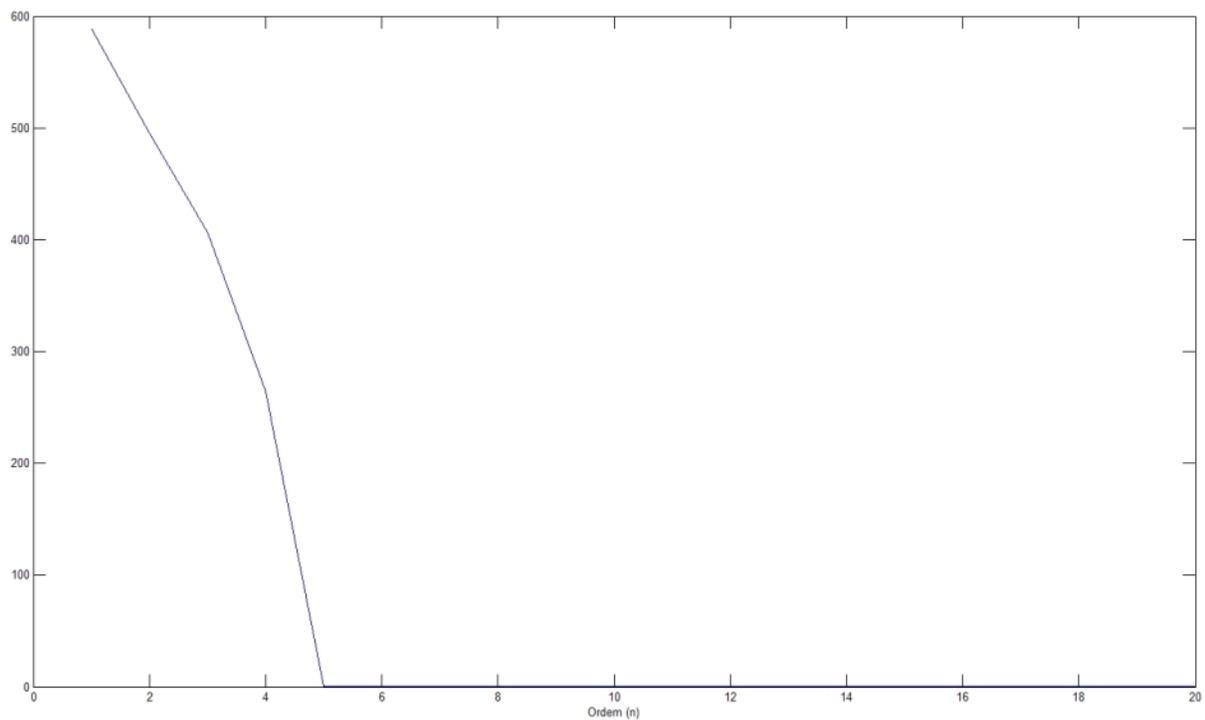


Fig. 3.3: Valores singulares para o MOESP de Katayama.

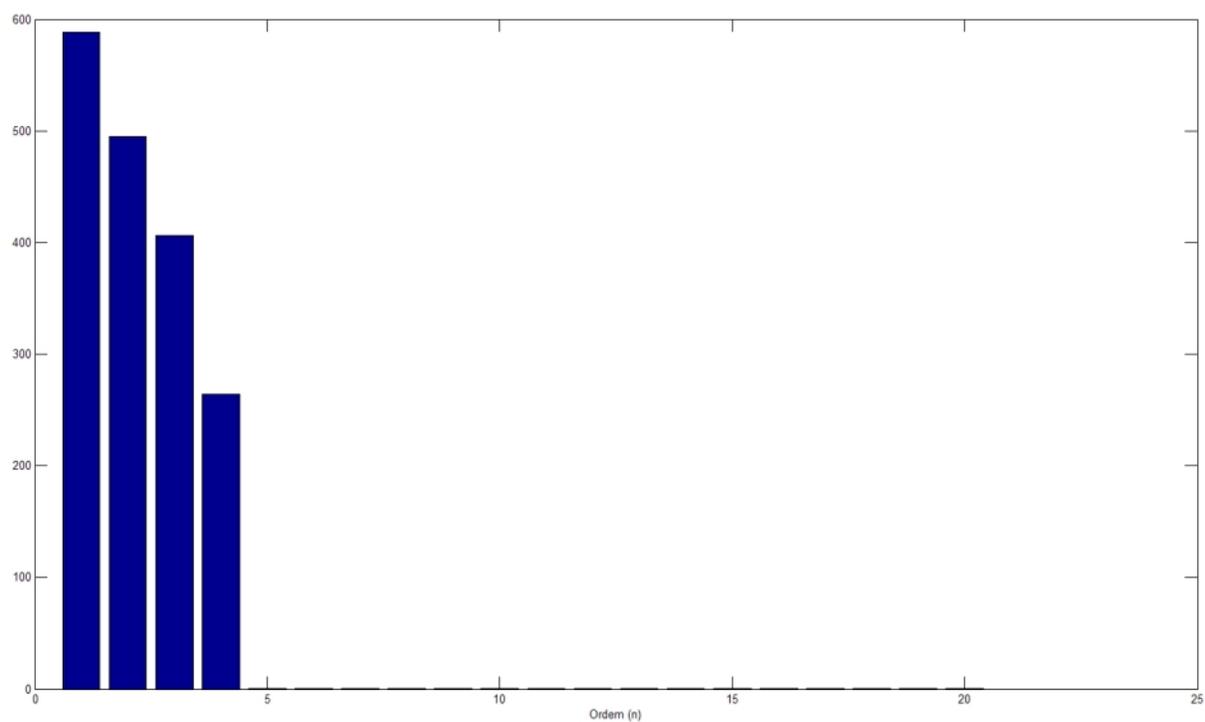


Fig. 3.4: Gráfico de barras dos valores singulares para o MOESP de Katayama.

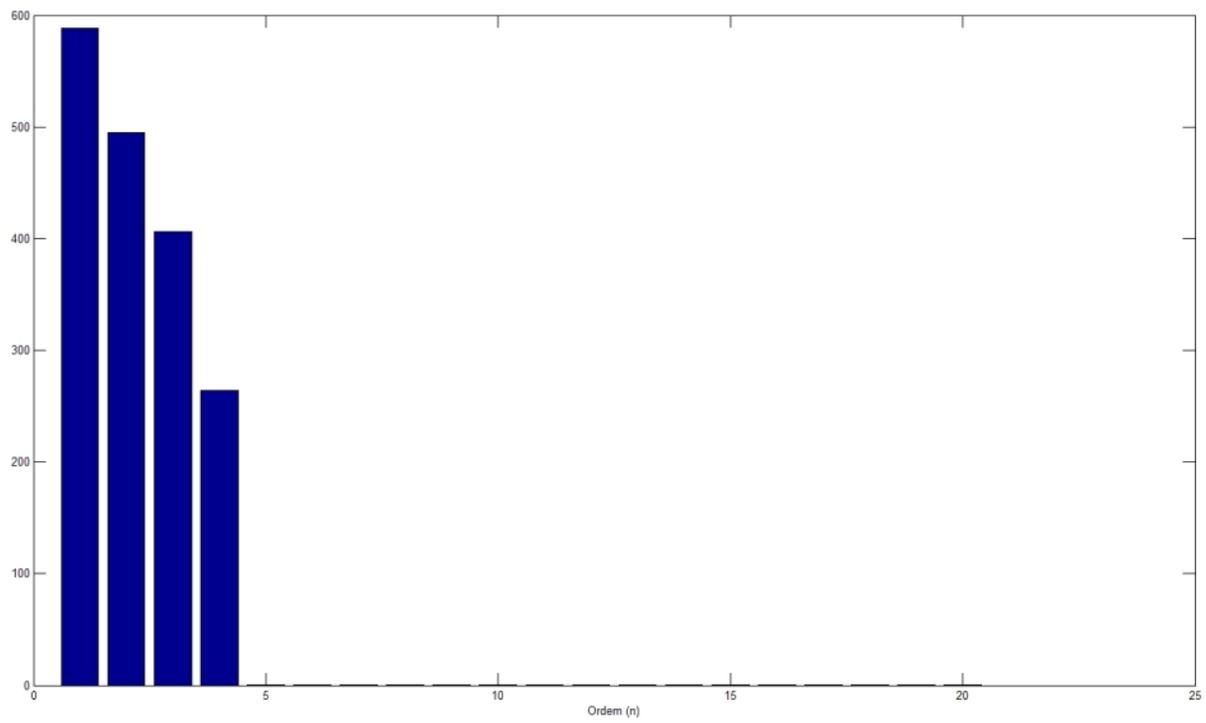


Fig. 3.5: Gráfico de barras dos valores singulares para o MOESP modificado.

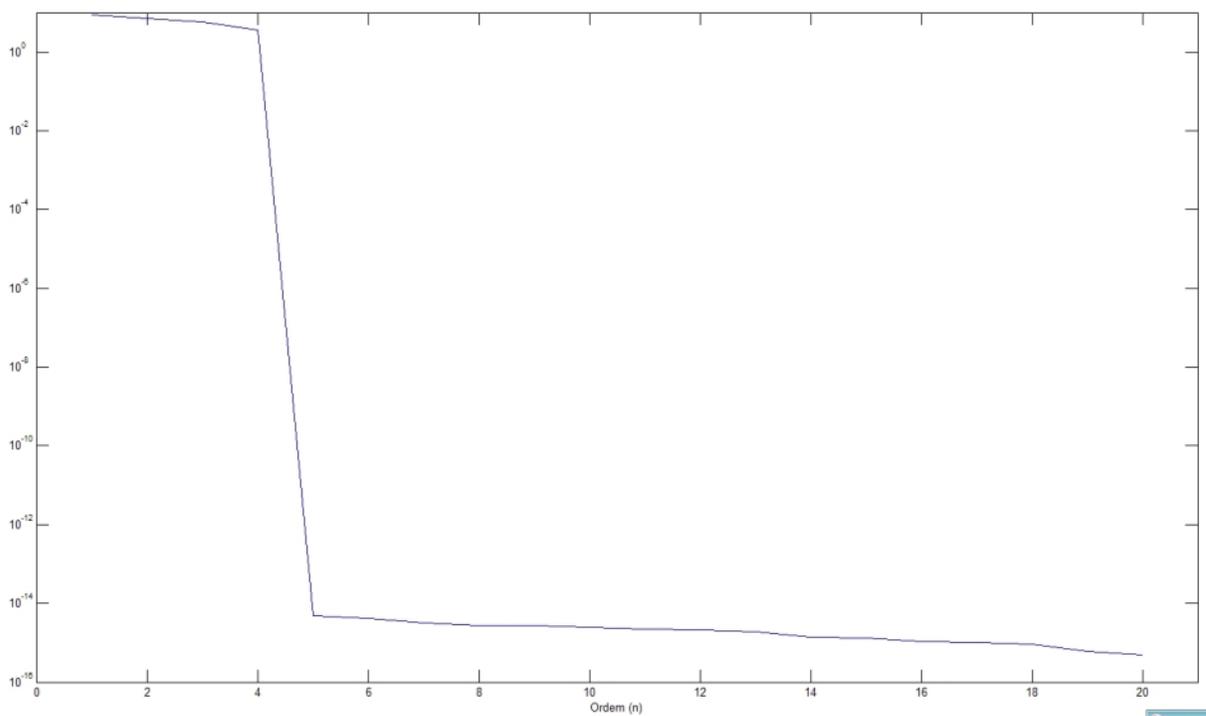


Fig. 3.6: Valores singulares para o N4SID referência.

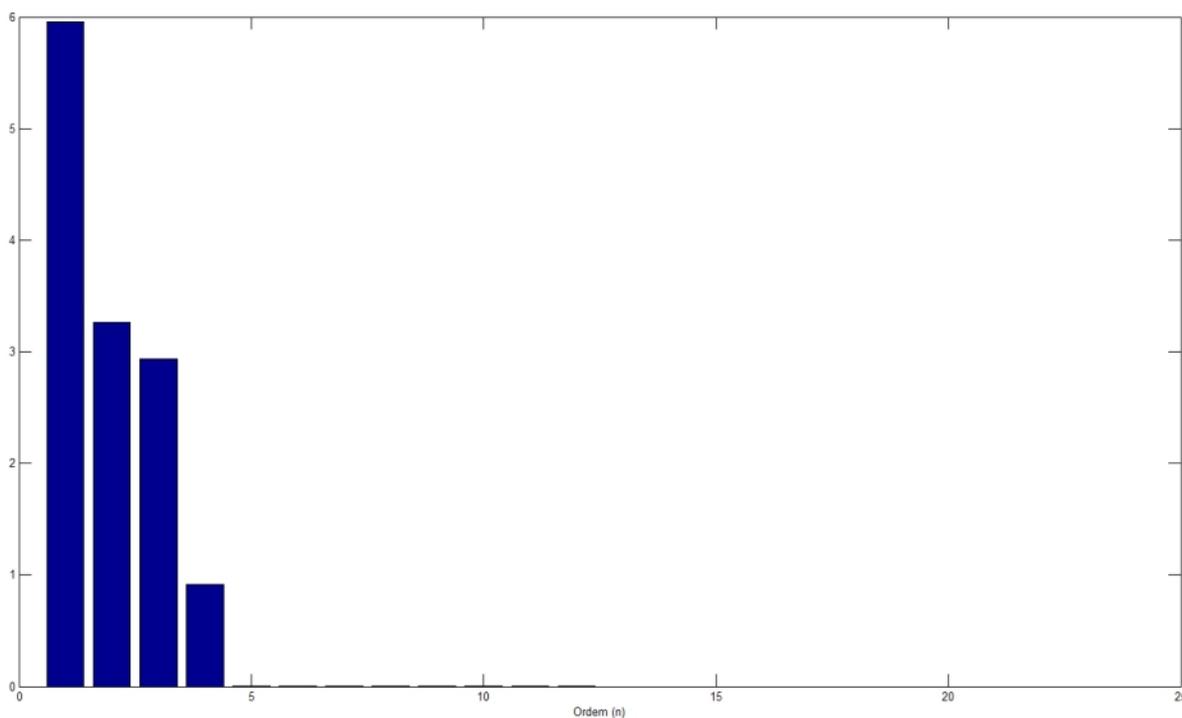


Fig. 3.7: Gráfico de barras dos valores singulares para o N4SID modificado.

3.2.4 Resultados

Os modelos obtidos aplicando cada um dos métodos são os seguintes:

MOESP referência

$$A = \begin{bmatrix} 0.5243 & -0.6445 & 0.3839 & -0.1818 \\ 0.5615 & 0.1800 & -0.7789 & 0.1056 \\ 0.0094 & -0.4853 & -0.2026 & 0.7998 \\ -0.1637 & -0.1862 & -0.4124 & -0.5018 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0423 & -4.9269 \\ -1.9177 & 1.0220 \\ 1.8091 & 2.0966 \\ -0.0354 & -0.7512 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.5430 & 0.4990 & 0.3518 & 0.1302 \\ 0.4163 & -0.3224 & -0.1343 & -0.2411 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

MOESP Katayama:

$$A = \begin{bmatrix} 0.4789 & 0.5542 & 0.3520 & 0.1717 \\ -0.6669 & 0.3538 & 0.5167 & -0.0316 \\ 0.1839 & 0.5034 & -0.3203 & -0.6275 \\ 0.1980 & -0.2336 & 0.5680 & -0.5124 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0100 & 0.2294 \\ -0.1066 & 0.0105 \\ -0.0902 & -0.1139 \\ -0.0019 & -0.0487 \end{bmatrix}$$

$$C = \begin{bmatrix} -10.8002 & 10.8901 & -8.3741 & 2.3866 \\ -10.1480 & -4.8917 & 2.5310 & -3.8284 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

MOESP modificado:

$$A = \begin{bmatrix} 0.4789 & 0.5542 & 0.3520 & 0.1717 \\ -0.6669 & 0.3538 & 0.5167 & -0.0316 \\ 0.1839 & 0.5034 & -0.3203 & -0.6275 \\ 0.1980 & -0.2336 & 0.5680 & -0.5124 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0172 & 0.2279 \\ -0.1089 & 0.0116 \\ -0.0989 & -0.1151 \\ -0.0024 & -0.0458 \end{bmatrix}$$

$$C = \begin{bmatrix} -10.8002 & 10.8901 & -8.3741 & 2.3866 \\ -10.1480 & -4.8917 & 2.5310 & -3.8284 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.2094 & -1.3143 \\ -1.2804 & 1.0238 \end{bmatrix}$$

N4SID referência:

$$A = \begin{bmatrix} 0.4406 & 0.5006 & 0.4245 & 0.2098 \\ -0.7059 & 0.2423 & 0.5504 & 0.0145 \\ 0.1579 & 0.5843 & -0.1772 & -0.6217 \\ 0.1984 & -0.2889 & 0.5512 & -0.5058 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0417 & 1.9911 \\ -0.8256 & 0.1843 \\ -0.8526 & -0.8093 \\ -0.0085 & -0.3718 \end{bmatrix}$$

$$C = \begin{bmatrix} -1.2295 & 1.3948 & -0.9386 & 0.2536 \\ -1.1931 & -0.6287 & 0.2009 & -0.4645 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

N4SID modificado:

$$A = \begin{bmatrix} 0.1380 & -0.5142 & 0.3381 & 0.1539 \\ -0.3897 & 0.1524 & -0.3397 & 0.4456 \\ -1.0519 & -0.5530 & -0.0562 & -0.1067 \\ 0.8631 & -0.6541 & -0.7111 & -0.2343 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0071 & 2.4884 \\ 1.3268 & 0.4074 \\ -0.9632 & -0.0111 \\ 0.7578 & 0.4696 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.6956 & 0.6269 & 1.1092 & -0.0802 \\ -0.9739 & 0.1151 & -0.4488 & -0.2331 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

Com cada algoritmo obtemos um modelo, ou seja, a quádrupla de matrizes **A**, **B**, **C**, e **D** que vamos representar o sistema no espaço de estado; podemos observar que os modelos entregados pelos métodos MOESP baseado em Katayama e MOESP modificado têm as matrizes **A** e **C** idênticas, pois o procedimento para obtê-las corresponde ao mesmo (usando a matriz de observabilidade); mas para

Algoritmo	Média	Variância
MOESP 1	0.6802	0.0073
MOESP 2	0.2635	0.0010
MOESP 3	0.2101	0.0015

Tab. 3.1: Tempos (em segundos) para os algoritmos MOESP com dados sem ruído.

as matrizes **B** e **D** temos umas variações, devido às modificações feitas no algoritmo. Nos modelos obtidos pelos métodos N4SID, só a matriz **D** é igual, matriz que corresponde ao parâmetro de Markov no instante $k = 0$.

Baseados nas referências [19], [16], [27], consideramos, a primeira vista, medir a complexidade computacional calculando o número de operações em ponto flutuante (Floating point Operations Per Second, FLOPS) requeridas por cada algoritmo implementado, sendo esta função uma das mais comuns na hora de comparar algoritmos em testes computacionais; mas desde o Matlab versão 6.0, a função usada para dito objetivo, *flops*, tem sido desativada devido às novas arquiteturas dos computadores modernos (arquiteturas paralelas). Existem algumas formas de contar estas operações, por exemplo, o projeto ATLAS (Automatically Tuned Linear Algebra Software) da Universidade de Tennessee, dirigido pelo professor Dongarra, que tem um método para obter o número de *flops*, mas que é não trivial, pois utiliza instruções de hardware disponíveis só em alguns chips. Esta ainda não é uma boa estimativa, por ser uma função degradada da antiga função *flops*. Assim contar estas operações não é hoje uma boa predição do tempo de execução; [35], [42], [46]. Devido a estes inconvenientes, decidimos tomar a quantidade de tempo de CPU, em segundos, requerida por cada algoritmo como uma das variáveis para as comparações dos algoritmos. Para isto usamos a função *cputime* de Matlab, que entrega o tempo em segundos desde que inicializamos Matlab; para obter o tempo de execução, armazenamos o tempo antes de começar o algoritmo, executamos ele, e tomamos de novo o tempo. A diferença entre os dois tempos corresponde ao tempo de nosso algoritmo.

Na tabela 3.1 temos os valores de média e variância dos tempos de execução para cada um dos algoritmos, para 24 horas de experimentos. A idéia de fazer os experimentos durante este período é para cobrir o tempo todo em que o sistema operacional Windows realiza suas diversas tarefas, e com isto tentar minimizar o efeito no tempo de execução produzido por diferentes processos que se executam ao longo do dia. Nesta tabela, MOESP 1 que corresponde ao algoritmo referência, [22], MOESP 2 ao algoritmo baseado em Katayama, [17], e MOESP 3 ao algoritmo modificado, capítulo 2. Aqui tem-se os tempos de processamento requeridos por cada um dos algoritmos. Podemos observar uma diferença notória nos tempos do algoritmo de referência e os outros dois algoritmos MOESP: o algoritmo baseado em Katayama, implementado via decomposição LQ, é mais rápido do que o algoritmo de referência; e o algoritmo MOESP modificado é ainda mais rápido do que o algoritmo de

Algoritmo	Média	Variância
N4SID 1	0.5832	0.0025
N4SID 2	0.3594	0.0020

Tab. 3.2: Tempos (em segundos) para os algoritmos N4SID com dados sem ruído.

referência.

Para os algoritmos N4SID temos dois: N4SID 1 que corresponde ao algoritmo referência, [10], e N4SID 2 ao algoritmo modificado, capítulo 2. Na tabela 3.2 apresentamos esta outra comparação dos tempos de execução requeridos para cada um dos algoritmos, onde observa-se que o algoritmo modificado é mais rápido que o algoritmo referência. Cabe lembrar que como não conseguimos implementar o algoritmo N4SID baseado em Katayama, não podemos fazer comparações entre ele, o algoritmo de referência, e o algoritmo modificado.

3.2.5 Validação do modelo

Outra característica que vamos utilizar para examinar o desempenho (**performance**) dos algoritmos, além do tempo requerido para sua execução, é a qualidade da identificação realizada. Neste caso, onde trabalhamos com o modelo real do sistema (benchmark) para gerar dados que vamos utilizar depois nos algoritmos, podemos tomar os parâmetros de Markov do benchmark e do sistema identificado, e compará-los.

Olhando os modelos obtidos com cada um dos algoritmos, ou seja, as quádruplas de matrizes **A**, **B**, **C** e **D**, temos que estas matrizes obtidas são diferentes das matrizes do benchmark, mas fazendo uma comparação dos parâmetros de Markov, temos que estes para o caso do algoritmo MOESP referência e MOESP baseado em Katayama são iguais, e que variam levemente para o algoritmo MOESP modificado. Isto indica que os modelos obtidos são uma transformação de similaridade do modelo benchmark, e podem representar adequadamente os conjuntos de dados de entrada - saída com ele gerados. E para o caso dos algoritmos N4SID, tanto para o algoritmo referência como para o algoritmo modificado, os valores dos parâmetros de Markov são idênticos. A seguir são apresentados os parâmetros de Markov do benchmark e de cada um dos algoritmos:

Benchmark:

$$\mathbf{D} = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3022 & -1.5256 \\ 0.4015 & -2.4812 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3378 & -3.9271 \\ 1.3938 & 0.6063 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6327 & 1.1968 \\ 0.8477 & -0.0996 \end{bmatrix}$$

MOESP referência:

$$\mathbf{D} = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3022 & -1.5256 \\ 0.4015 & -2.4812 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3378 & -3.9271 \\ 1.3938 & 0.6063 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6327 & 1.1968 \\ 0.8477 & -0.0996 \end{bmatrix}$$

MOESP Katayama:

$$\mathbf{D} = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3022 & -1.5256 \\ 0.4015 & -2.4812 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3378 & -3.9271 \\ 1.3938 & 0.6063 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6327 & 1.1968 \\ 0.8477 & -0.0996 \end{bmatrix}$$

MOESP modificado:

$$\mathbf{D} = \begin{bmatrix} -0.2094 & -1.3143 \\ -1.2804 & 1.0238 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.1783 & -1.4797 \\ 0.4659 & -2.4852 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3994 & -3.9159 \\ 1.4992 & 0.6104 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.7359 & 1.1804 \\ 0.8666 & -0.0898 \end{bmatrix}$$

N4SID referência:

$$\mathbf{D} = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3022 & -1.5256 \\ 0.4015 & -2.4812 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3378 & -3.9271 \\ 1.3938 & 0.6063 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6327 & 1.1968 \\ 0.8477 & -0.0996 \end{bmatrix}$$

N4SID modificado:

$$\mathbf{D} = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3022 & -1.5256 \\ 0.4015 & -2.4812 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3378 & -3.9271 \\ 1.3938 & 0.6063 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6327 & 1.1968 \\ 0.8477 & -0.0996 \end{bmatrix}$$

Com o subconjunto de dados de validação podemos obter um erro que vamos chamar erro de validação, que é outro fator também importante, e que indica quão bom é o modelo ao aplicá-lo a uma parte dos dados que não foi utilizada no procedimento de identificação; este erro de validação está definido como:

$$e = \frac{1}{l} \sum_{c=1}^l \left[\sqrt{\frac{\sum_{k=1}^s ((y_k)_c - (y_k^s)_c)^2}{\sum_{k=1}^s ((y_k^s)_c)}} \right] 100\% \quad (3.1)$$

onde $(y_k)_c$ denota a c -ésima saída no instante de tempo k do conjunto de dados de validação, e $(y_k^s)_c$ a c -ésima saída, também no instante k , mas do conjunto de dados simulados. Nas tabelas apresentadas a seguir temos o valor como a média das saídas (no caso deste benchmark, das duas saídas).

Algoritmo	y_1	y_2
MOESP 1	4.06	3.55
MOESP 2	4.06	3.55
MOESP 3	6.05	5.29

Tab. 3.3: Erro de validação (em porcentagem) para os algoritmos MOESP com dados sem ruído.

Algoritmo	y_1	y_2
N4SID 1	3.68	2.35
N4SID 2	3.68	2.35

Tab. 3.4: Erro de validação (em porcentagem) para os algoritmos N4SID com dados sem ruído.

Na tabela 3.3 para os três algoritmos MOESP: o algoritmo referência, o algoritmo baseado em Katayama, e o algoritmo modificado, apresentamos os erros de validação resultantes. Podemos observar que os valores para o algoritmo referência e para o algoritmo baseado em Katayama são iguais; e para o último algoritmo, o modificado, temos um erro de validação um pouco maior. Estes valores de erros podem ser considerados aceitáveis, pois são similares e não muito grandes.

Na tabela 3.4 apresentamos os erros de validação para os dois algoritmos N4SID: o algoritmo referência, e o algoritmo modificado. Aqui a diferença entre os dois algoritmos está na velocidade de cada um, pois os erros de validação são idênticos para as duas implementações.

Usamos a função *compare* do *System Identification Toolbox* de Matlab, função que tem como entradas o conjunto de dados de validação e o modelo no espaço de estado (a quádrupla de matrizes), e que entrega um gráfico da saída real do modelo (conjunto de dados de validação) versus a saída experimental, e além disso entrega o valor de encaixe do modelo (*fitness*), como pode-se observar nas figuras 3.8, 3.10, 3.12, 3.14 e 3.16. Nestas figuras temos saída real versus saída experimental para cada uma das saídas do modelo (neste caso, para y_1 e y_2), para cada um dos três algoritmos MOESP e dois N4SID. O encaixe do modelo obtido com o algoritmo MOESP referência é igual ao obtido com o MOESP baseado em Katayama, com valor de 91.75% para a saída y_1 e de 94.53% para a saída y_2 ; e para o MOESP modificado temos os valores de 90.53% e 93.06% para as saídas y_1 e y_2 , respectivamente. Estes valores, ainda que sejam menores (isto é, que o modelo obtido com este método encaixa um pouco menos que com os outros dois), indicam que o modelo é uma boa estimativa. Agora, temos para os dois algoritmos N4SID: o algoritmo referência, e o algoritmo modificado, que os valores de encaixe dos modelos são idênticos para as duas implementações, 92.84% para a saída y_1 e de 93.96% para a saída y_2 .

Como no caso do erro de validação, estes valores indicam que todos os métodos aqui tratados podem

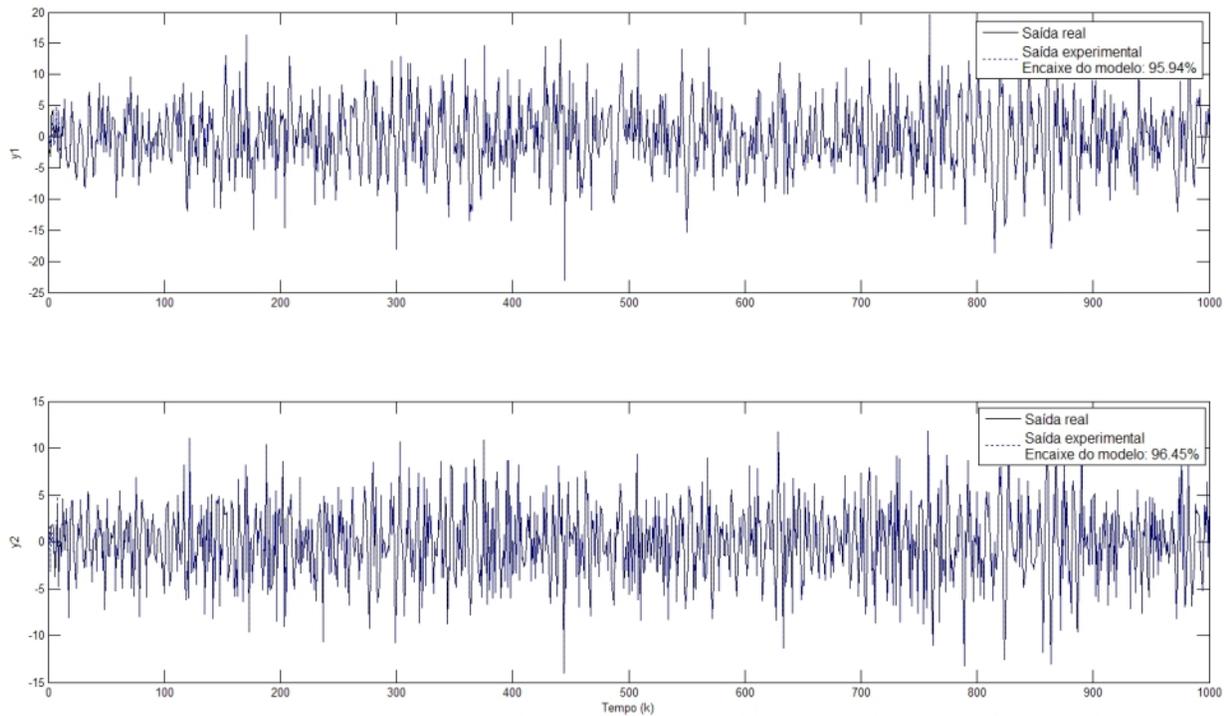


Fig. 3.8: Gráfico de encaixe do modelo para o algoritmo MOESP 1.

obter um modelo que represente adequadamente o conjunto de dados de entrada - saída. Nas figuras 3.9, 3.11, 3.13, 3.15 e 3.17 temos um zoom dos gráficos da saída real versus a saída experimental, para tentar observar mais detalhadamente a diferença entre as duas saídas.

3.2.6 Conclusões

Fazendo uso dos dados gerados utilizando o benchmark, obtemos as matrizes do sistema, **A**, **B**, **C** e **D**, com os métodos MOESP e N4SID (três algoritmos diferentes para o MOESP: referência, baseado em Katayama e modificado, e dois para o N4SID: referência e modificado); estas matrizes obtidas são diferentes das matrizes do benchmark, mas fazendo uma comparação dos parâmetros de Markov, temos que estes últimos são similares. Isto indica que os modelos obtidos são uma transformação de similaridade do modelo benchmark, e podem representar adequadamente os conjuntos de dados de entrada - saída com ele gerados.

Os Métodos de Subespaços requerem um parâmetro, k , que indica a dimensão das matrizes bloco de Hankel; esta é uma das vantagens destes métodos, pois a informação a priori necessária é mínima, só dois valores: a ordem das matrizes bloco de Hankel, e a ordem do sistema a identificar. Esta última ordem, n , pode-se observar nos algoritmos de subespaços no passo onde se faz a decomposição em

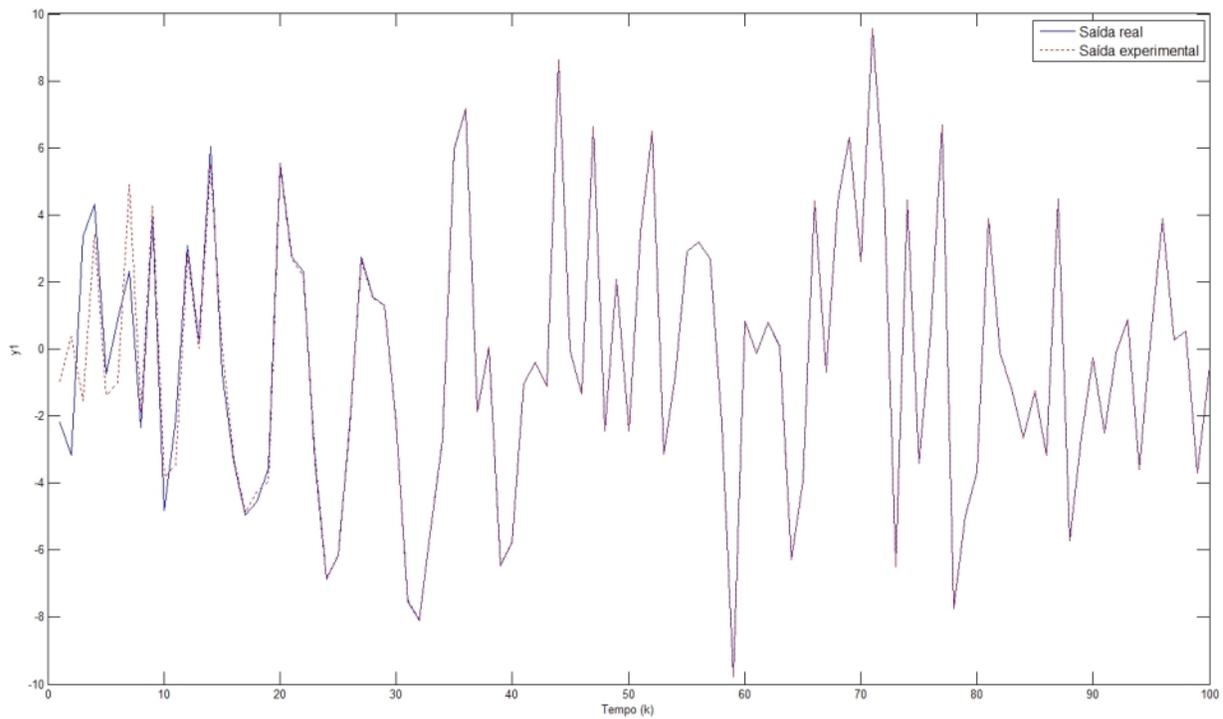


Fig. 3.9: Zoom no gráfico de encaixe do modelo para o algoritmo MOESP 1.

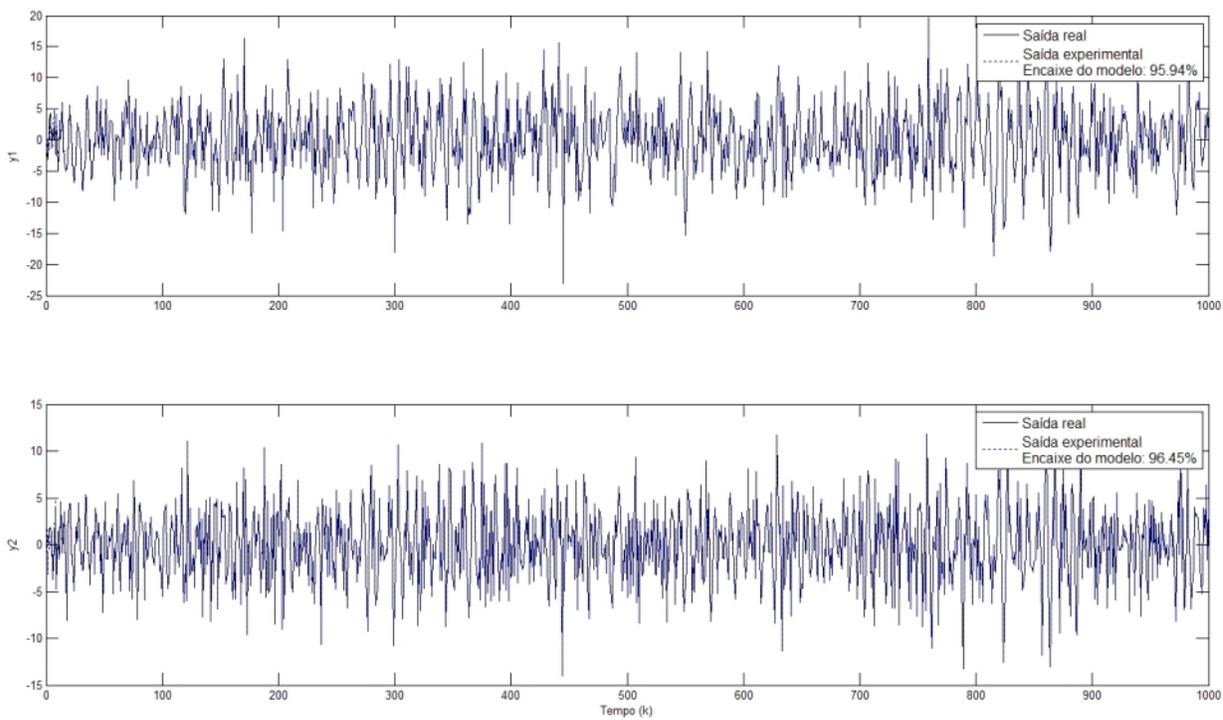


Fig. 3.10: Gráfico de encaixe do modelo para o algoritmo MOESP 2.

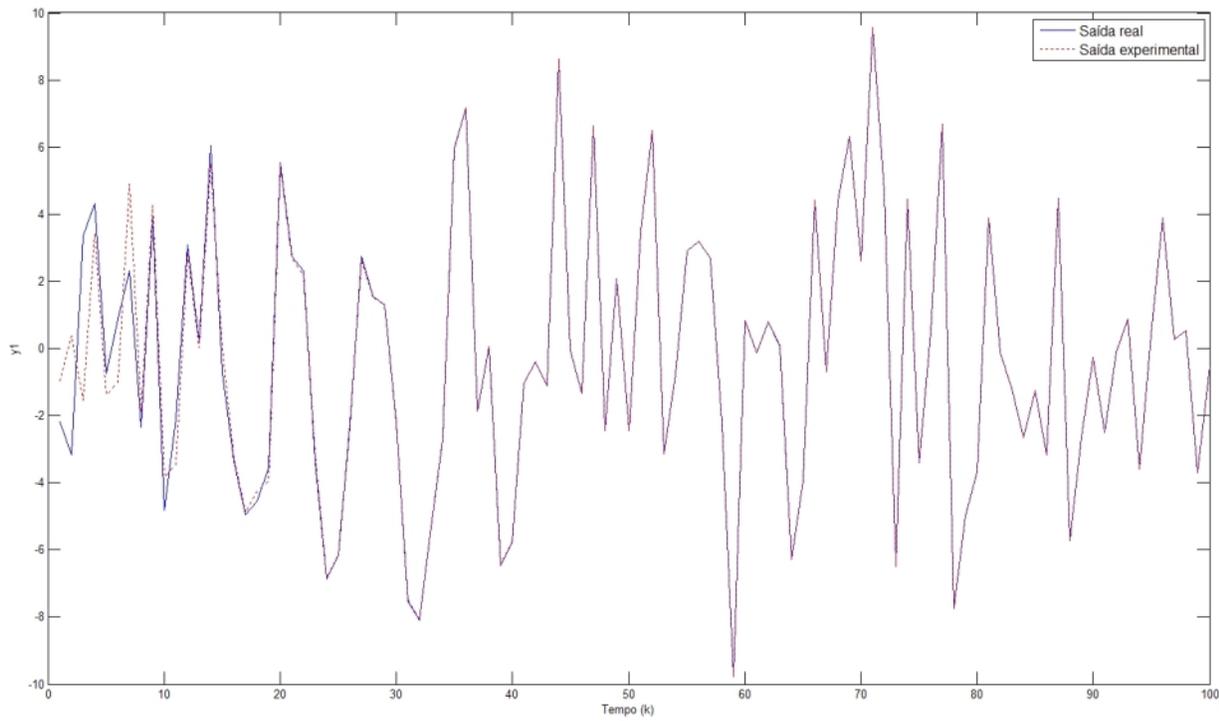


Fig. 3.11: Zoom no gráfico de encaixe do modelo para o algoritmo MOESP 2.

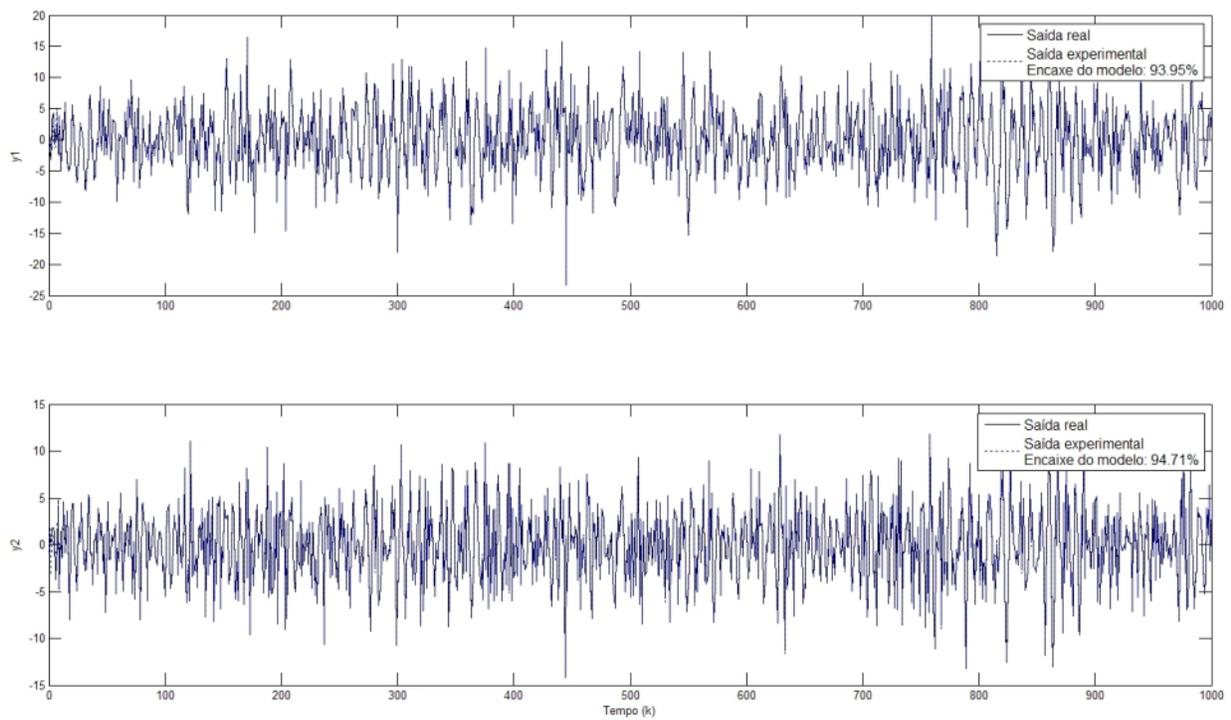


Fig. 3.12: Gráfico de encaixe do modelo para o algoritmo MOESP 3.

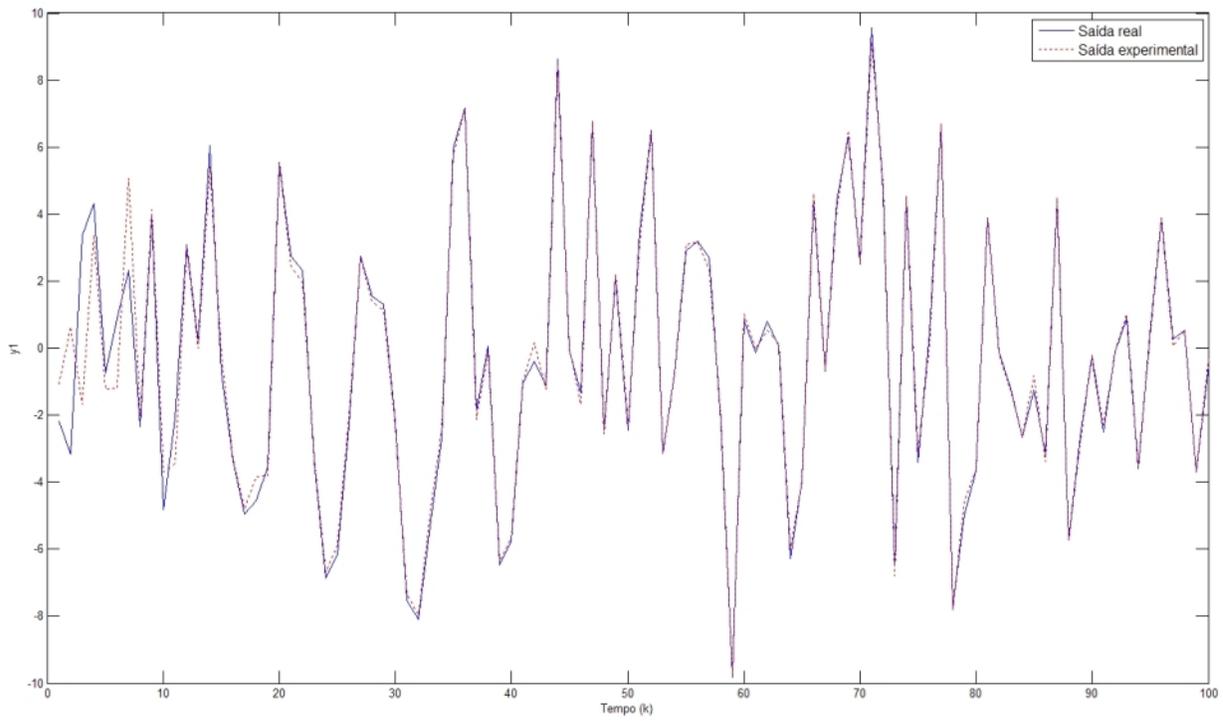


Fig. 3.13: Zoom no gráfico de encaixe do modelo para o algoritmo MOESP 3.

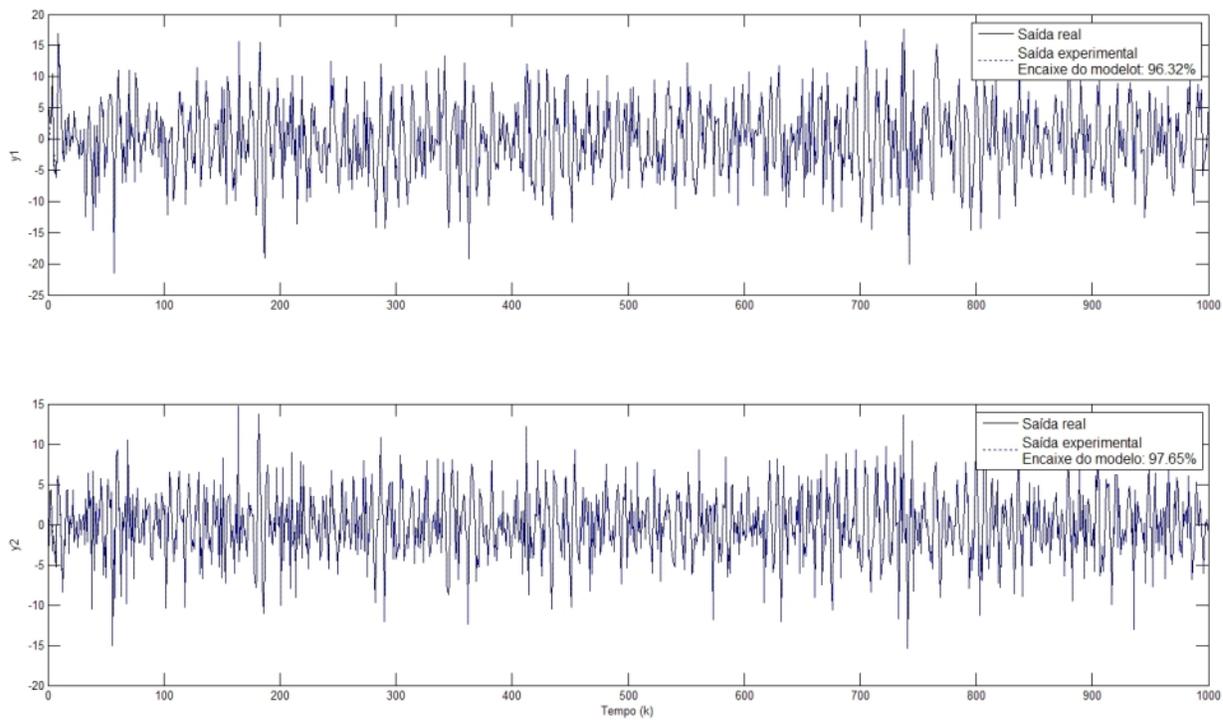


Fig. 3.14: Gráfico de encaixe do modelo para o algoritmo N4SID 1.

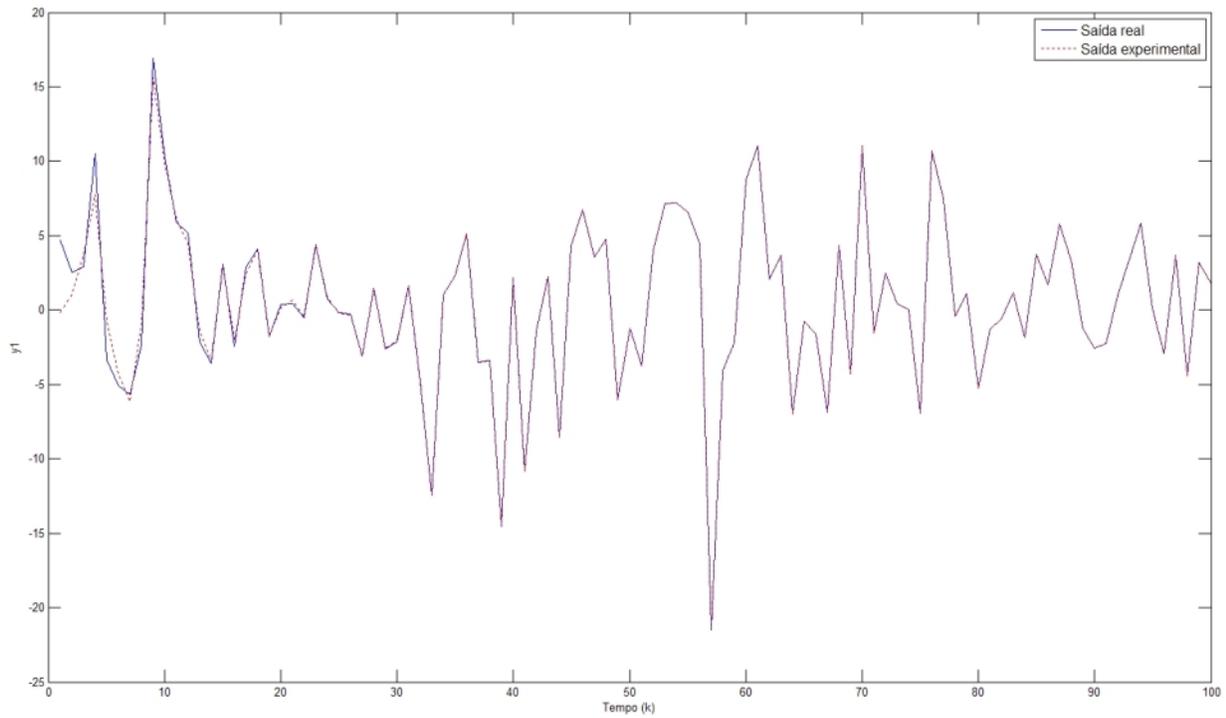


Fig. 3.15: Zoom no gráfico de encaixe do modelo para o algoritmo N4SID 1.

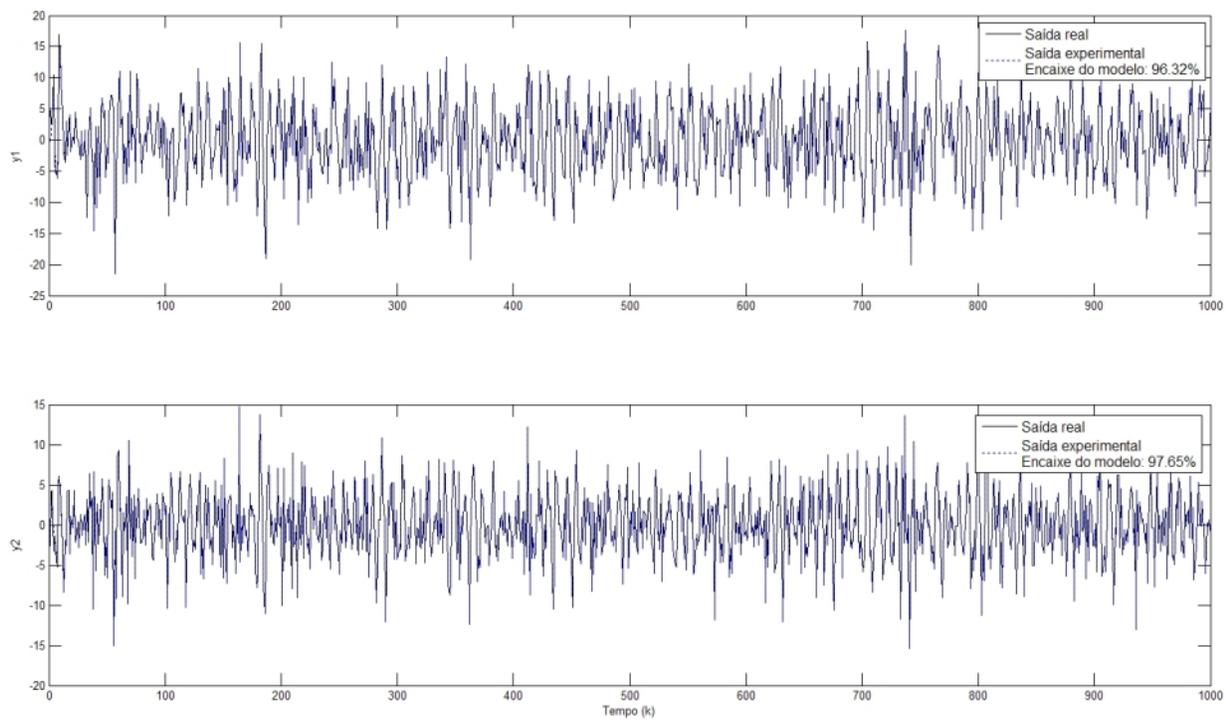


Fig. 3.16: Gráfico de encaixe do modelo para o algoritmo N4SID 2.

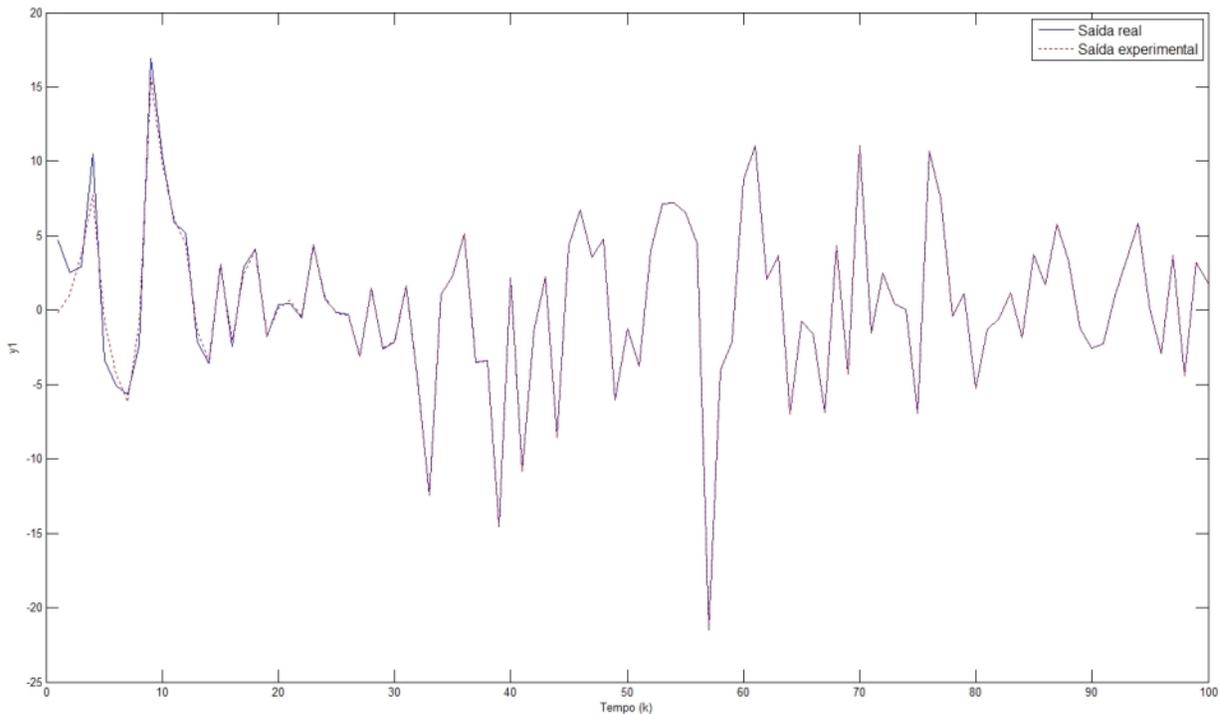


Fig. 3.17: Zoom no gráfico de encaixe do modelo para o algoritmo N4SID 2.

valores singulares. Estes valores selecionados têm que satisfazer o critério $kl > n$. Os Métodos de Subespaços trabalham com modelos no espaço de estado, os quais requerem cálculos com matrizes, cálculos que podem ser bem manipulados usando Matlab.

Medir a complexidade computacional calculando o número de operações em ponto flutuante, *flops*, não é tão trivial devido as novas arquiteturas dos computadores modernos, por isso optamos por tomar o tempo de processamento como uma das variáveis de interesse para a comparação entre os algoritmos. Por este ponto de vista, o algoritmo MOESP modificado é mais rápido do que o algoritmo referência. E o algoritmo N4SID modificado é mais rápido do que o algoritmo referência. Para ter uma boa estimativa destes tempos de execução fizemos 24 horas de experimentação para cada algoritmo. A idéia de fazer os experimentos durante este período é para tentar minimizar os efeitos produzidos por diferentes processos que são executados ao longo do dia no sistema operacional Windows. A outra variável que foi tomada para fazer as comparações entre os algoritmos foi o erro de validação, que indica quão bom é o modelo obtido quando aplicado a uma parte dos dados que não foi utilizada no procedimento de identificação (os chamados dados de validação). O algoritmo MOESP modificado apresenta um erro de validação um pouco maior, mas tomando em conta que este erro não é tão grande, e que este algoritmo é mais rápido que os outros dois (os quais apresentam igual erro de validação), podemos considerar esta velocidade como uma vantagem. Ou seja, dependendo

do caso, se precisa-se de velocidade este algoritmo modificado é melhor que os outros dois MOESP aqui tratados. Para o algoritmo N4SID, o algoritmo modificado apresenta o mesmo erro de validação que o algoritmo N4SID referência, mas é mais rápido.

Usando a função *compare* do *System Identification Toolbox* de Matlab, podemos obter uma idéia de quão boa é a identificação feita com cada algoritmo, mediante um gráfico que compara a saída experimental (ou seja, a saída do modelo obtido) com a saída real (do conjunto de dados de validação). Para o caso dos algoritmos N4SID, estes apresentam valores iguais, donde o algoritmo modificado pode ser considerado melhor que o algoritmo de referência, pois são idênticos em exatidão, mas o algoritmo modificado é mais rápido em quanto a tempo de processamento. Para os algoritmos MOESP, o algoritmo modificado apresenta menor exatidão que os outros dois algoritmos, desvantagem que pode ser compensada pela sua velocidade de execução.

Cabe ressaltar que em todos os casos, os erros de validação tem valores aceitáveis, mesmo que seja maior (para o algoritmo modificado), pois a sua diferença entre os métodos não é considerável.

3.3 Benchmark 1 com ruído

3.3.1 Introdução

Os testes anteriores são para o modelo ideal, sem ruído; mas como na prática é comum ter sinais que afetam o sistema de forma incontrolável, para analisar o comportamento dos algoritmos vamos aplicar a estes dados um ruído branco gaussiano. Para este caso do sistema com ruído, o conjunto de dados de entrada - saída usados no procedimento de identificação são apresentados nas figuras 3.18 e 3.19 (onde u_1 e u_2 correspondem as entradas, e y_1 e y_2 as saídas).

3.3.2 Algoritmos

Temos, de novo, os algoritmos de referência MOESP e N4SID, o algoritmo MOESP baseado em Katayama e os algoritmos modificados MOESP e N4SID, para serem aplicados aos dados de entradas - saídas e estimar em um modelo no espaço de estado.

3.3.3 Parâmetros de identificação

Os valores dos parâmetros para realizar a identificação por meio de Métodos de Subespaços, de novo foram $k = 10$ e $n = 4$; este ultimo valor é a ordem real do sistema original (*benchmark*). Esta ordem pode-se observar na figura 3.20, que corresponde ao gráfico de barras obtido no passo da decomposição em valores singulares do algoritmo MOESP baseado em Katayama.

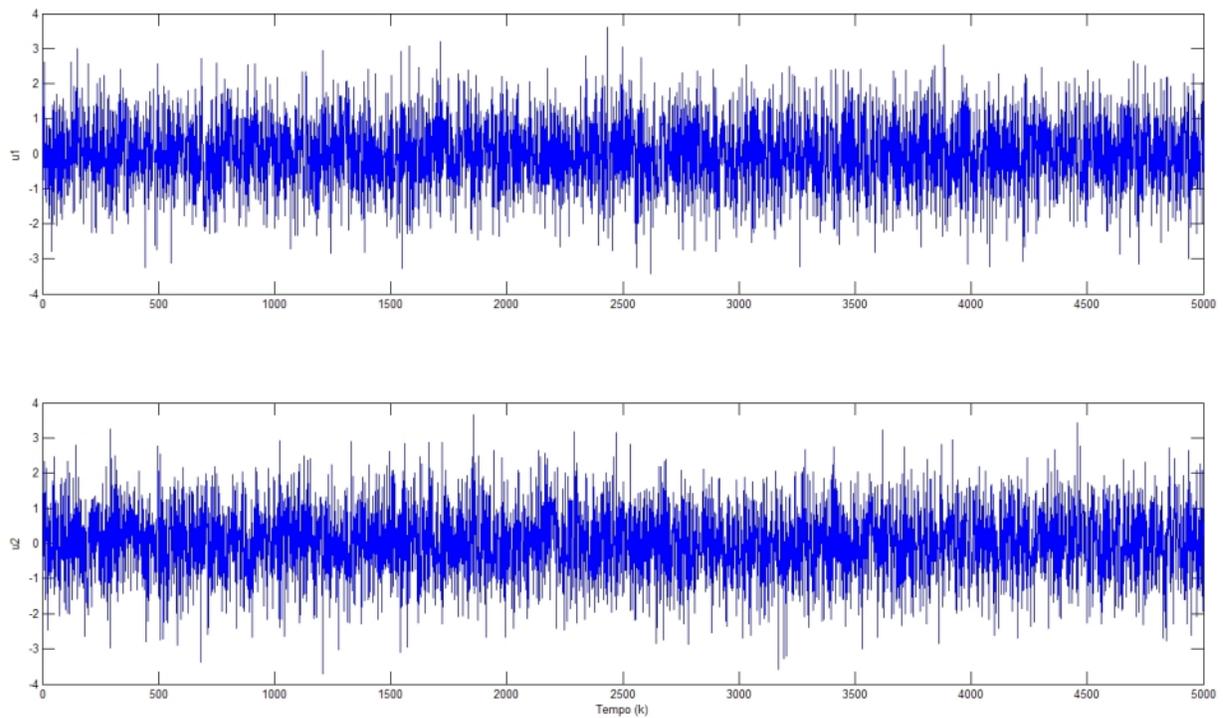


Fig. 3.18: Sequência de dados de entrada.

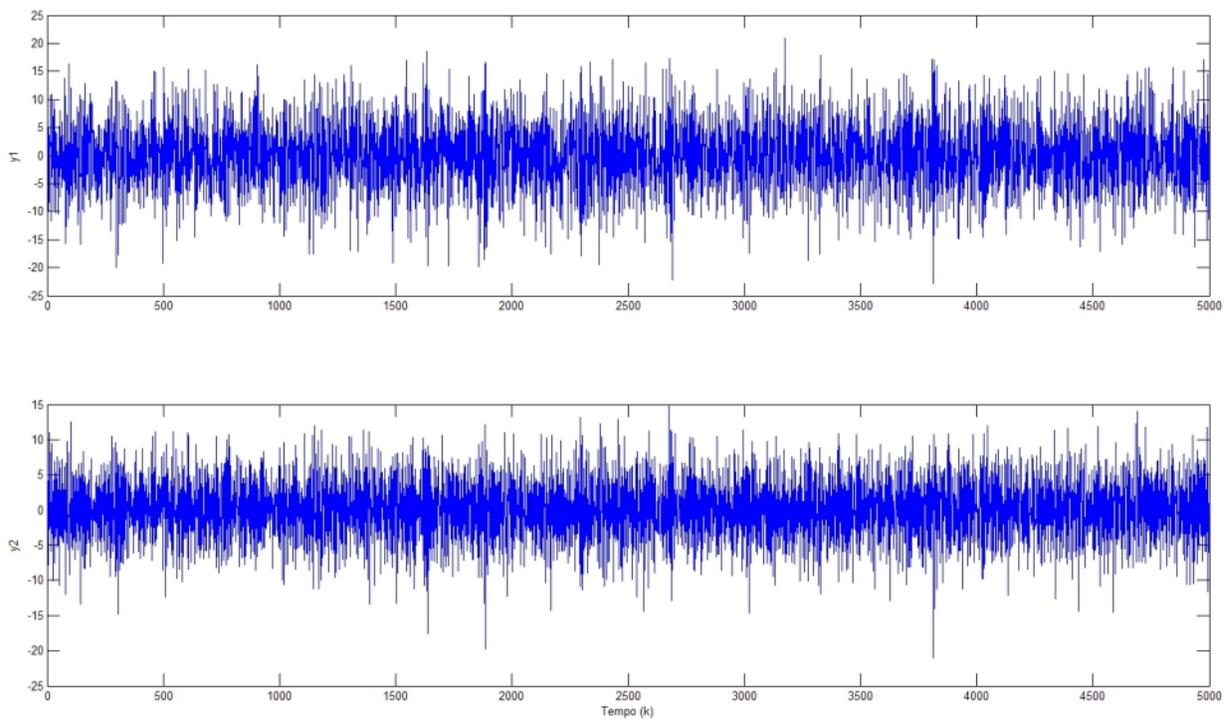


Fig. 3.19: Sequência de dados de saída.

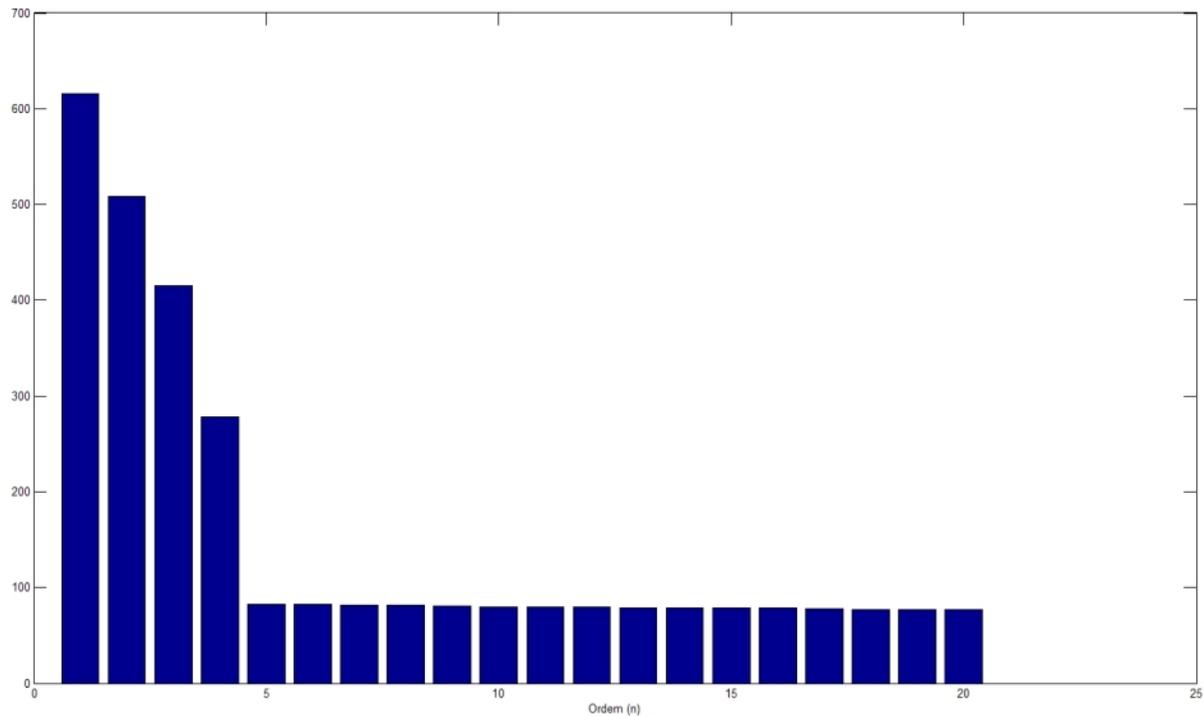


Fig. 3.20: Gráfico de barras dos valores singulares para o MOESP de Katayama.

3.3.4 Resultados

Os modelos obtidos para o sistema com ruído são dados pelas matrizes:

MOESP referência

$$A = \begin{bmatrix} 0.5099 & -0.6403 & 0.3751 & 0.1980 \\ -0.5551 & 0.1155 & 0.8088 & 0.1511 \\ -0.0272 & 0.5181 & -0.1316 & -0.7897 \\ 0.1776 & -0.1909 & 0.4150 & -0.4939 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.1153 & 4.9027 \\ -1.8315 & 1.4213 \\ -1.9069 & -1.9514 \\ -0.0194 & -0.6539 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.5707 & 0.4797 & -0.3323 & 0.1287 \\ -0.3912 & -0.3525 & 0.1051 & -0.2526 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1424 & -1.2642 \\ -1.3668 & 0.9867 \end{bmatrix}$$

MOESP Katayama:

$$A = \begin{bmatrix} 0.4631 & 0.5353 & 0.3548 & 0.1973 \\ -0.6785 & 0.2984 & 0.5609 & 0.0051 \\ 0.1400 & 0.5636 & -0.2564 & -0.6340 \\ 0.2111 & -0.2355 & 0.5631 & -0.5031 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0017 & 0.2291 \\ -0.1024 & 0.0312 \\ -0.0946 & -0.1055 \\ -0.0004 & -0.0402 \end{bmatrix}$$

$$C = \begin{bmatrix} -11.5919 & 10.7598 & -8.1028 & 2.4641 \\ -9.7274 & -5.6753 & 2.0636 & -4.1776 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1390 & -1.2670 \\ -1.3633 & 0.9754 \end{bmatrix}$$

MOESP modificado:

$$A = \begin{bmatrix} 0.4631 & 0.5353 & 0.3548 & 0.1973 \\ -0.6785 & 0.2984 & 0.5609 & 0.0051 \\ 0.1400 & 0.5636 & -0.2564 & -0.6340 \\ 0.2111 & -0.2355 & 0.5631 & -0.5031 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0086 & 0.2342 \\ -0.0963 & 0.0268 \\ -0.0949 & -0.1036 \\ -0.0011 & -0.0424 \end{bmatrix}$$

$$C = \begin{bmatrix} -11.5919 & 10.7598 & -8.1028 & 2.4641 \\ -9.7274 & -5.6753 & 2.0636 & -4.1776 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.0162 & -1.2459 \\ -1.2324 & 0.8683 \end{bmatrix}$$

N4SID referência:

$$A = \begin{bmatrix} 0.4532 & 0.5103 & 0.3852 & 0.2056 \\ -0.6925 & 0.2553 & 0.5674 & 0.0199 \\ 0.1323 & 0.5883 & -0.2044 & -0.6232 \\ 0.2101 & -0.2607 & 0.5607 & -0.5045 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0245 & 1.9495 \\ -0.8374 & 0.2566 \\ -0.8534 & -0.8758 \\ -0.0256 & -0.3507 \end{bmatrix}$$

$$C = \begin{bmatrix} -1.3250 & 1.3214 & -0.9243 & 0.2816 \\ -1.1724 & -0.6595 & 0.2196 & -0.4629 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1626 & -1.2786 \\ -1.3546 & 0.9925 \end{bmatrix}$$

N4SID modificado:

$$A = \begin{bmatrix} 0.4151 & 0.5038 & 0.3964 & 0.2255 \\ -0.7157 & 0.2486 & 0.5411 & -0.0043 \\ 0.1844 & 0.6152 & -0.1875 & -0.5975 \\ 0.2297 & -0.2770 & 0.5866 & -0.4847 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0030 & 2.0777 \\ -0.8629 & 0.2883 \\ -0.8896 & -0.8312 \\ -0.0506 & -0.3098 \end{bmatrix}$$

$$C = \begin{bmatrix} -1.2341 & 1.2673 & -0.9065 & 0.2586 \\ -1.1183 & -0.6184 & 0.1491 & -0.4423 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1605 & -1.2747 \\ -1.3546 & 0.9879 \end{bmatrix}$$

Como observado anteriormente, as matrizes **A**, **C** para os algoritmos MOESP baseado em Katayama e modificado são iguais pelo procedimento. Neste caso não nos interessa analisar os tempos de execução para o sistema com ruído, pois isso foi feito de forma rigorosa com o benchmark 1 sem ruído, onde observamos que, para 24 horas de experimentos, o algoritmo MOESP modificado é mais rápido

que os outros dois algoritmos MOESP, e o N4SID modificado também é mais rápido que o N4SID referência.

3.3.5 Validação do modelo

Para o caso do sistema com ruído, os parâmetros de Markov são dados pelas matrizes:

Benchmark:

$$\mathbf{D} = \begin{bmatrix} -0.1356 & -1.2704 \\ -1.3493 & 0.9846 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3022 & -1.5256 \\ 0.4015 & -2.4812 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3378 & -3.9271 \\ 1.3938 & 0.6063 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6327 & 1.1968 \\ 0.8477 & -0.0996 \end{bmatrix}$$

MOESP referência:

$$\mathbf{D} = \begin{bmatrix} -0.1424 & -1.2642 \\ -1.3668 & 0.9867 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3132 & -1.5519 \\ 0.4050 & -2.4587 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3477 & -3.9289 \\ 1.3908 & 0.6121 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6348 & 1.1978 \\ 0.8517 & -0.1007 \end{bmatrix}$$

MOESP Katayama:

$$\mathbf{D} = \begin{bmatrix} -0.1390 & -1.2670 \\ -1.3633 & 0.9754 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.3168 & -1.5538 \\ 0.4050 & -2.4552 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3471 & -3.9327 \\ 1.3894 & 0.6159 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6279 & 1.1980 \\ 0.8554 & -0.0947 \end{bmatrix}$$

MOESP modificado:

$$\mathbf{D} = \begin{bmatrix} -0.0162 & -1.2459 \\ -1.2324 & 0.8683 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.1696 & -1.6909 \\ 0.4390 & -2.4670 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3853 & -3.9583 \\ 1.3724 & 0.6156 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.5898 & 1.2281 \\ 0.7997 & -0.0736 \end{bmatrix}$$

N4SID referência:

$$\mathbf{D} = \begin{bmatrix} -0.1626 & -1.2786 \\ -1.3546 & 0.9925 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.2925 & -1.5333 \\ 0.4053 & -2.4849 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3340 & -3.9020 \\ 1.4049 & 0.6340 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6444 & 1.2328 \\ 0.8477 & -0.0781 \end{bmatrix}$$

N4SID modificado:

$$\mathbf{D} = \begin{bmatrix} -0.1605 & -1.2747 \\ -1.3546 & 0.9879 \end{bmatrix}$$

$$\mathbf{CB} = \begin{bmatrix} -0.2966 & -1.525 \\ 0.4257 & -2.4949 \end{bmatrix}$$

$$\mathbf{CAB} = \begin{bmatrix} 0.3446 & -3.9144 \\ 1.3831 & 0.5813 \end{bmatrix}$$

$$\mathbf{CA}^2\mathbf{B} = \begin{bmatrix} 1.6633 & 1.2905 \\ 0.8125 & -0.0658 \end{bmatrix}$$

Algoritmo	y_1	y_2
MOESP 1	20.14	25.19
MOESP 2	20.13	25.18
MOESP 3	20.34	25.74

Tab. 3.5: Erro de validação (em porcentagem) para os algoritmos MOESP com dados com ruído.

Algoritmo	y_1	y_2
N4SID 1	20.08	27.14
N4SID 2	20.22	27.20

Tab. 3.6: Erro de validação (em porcentagem) para os algoritmos N4SID com dados com ruído.

Neste caso os parâmetros de Markov variam um pouco, dependendo do método, mas os valores ficam ainda similares.

Na tabela 3.5, temos que analisando os valores de erro de validação para os algoritmos aplicados aos dados com ruído, estes variam levemente entre os algoritmos MOESP referência e MOESP baseado em Katayama (que para o caso sem ruído eram iguais), e para o caso do algoritmo modificado a diferença é um pouco maior, mas pode ser considerado como um erro aceitável, dependendo da exatidão requerida.

Na tabela 3.6 temos os dois algoritmos N4SID: o algoritmo referência, [10], e o algoritmo modificado, capítulo 2. Neste caso o algoritmo baseado em Katayama é um pouco melhor, pois apresenta uma sutil diferença no erro de validação (um pouco menor), mas os valores são próximos; para o caso com ruído, os algoritmos apresentam erros de validação diferentes, contrario ao que aconteceu para o sistema sem ruído.

Usando a função *compare* do *System Identification Toolbox* de Matlab, obtemos o encaixe do modelo (*fitness*), como pode-se observar nas figuras C.1 à C.5 do apêndice C. Nestas figuras temos saída real versus saída experimental para as duas saídas do modelo (y_1 e y_2), para cada um dos três algoritmos MOESP e dois N4SID.

3.3.6 Conclusões

A diferença com o sistema sem ruído, aqui temos que os algoritmos MOESP referência e MOESP baseado em Katayama por um lado, e N4SID referência e N4SID modificado por outro têm uma variação no erro de validação (erro que para o caso sem ruído era igual; erro de validação MOESP referência igual ao erro do MOESP baseado em Katayama, e erro do N4SID referência igual ao erro do N4SID modificado). Para este caso particular, os algoritmos MOESP baseado em Katayama e

N4SID referência são melhores para trabalhar com seqüência de dados afeitados por ruído. De novo, podemos ressaltar que os valores dos erros de validação para todos os casos, podem ser aceitados, pois a sua diferença não é significativa. Não tomamos tempos de execução, pois isso foi feito no experimento anterior. Aqui a idéia era observar o comportamento dos algoritmos para conjuntos de dados perturbados.

3.4 Benchmark 2

3.4.1 Introdução

Para um segundo caso, outro benchmark, utilizando a referência [3], da mesma ordem, 4, mas para um sistema um pouco mais complexo, de 3 entradas, 2 saídas, é usado para realizar experimentos neste trabalho. De novo, os dois conjuntos (para identificação e para validação) são gerados. Este segundo benchmark é dado pelas matrizes:

$$A = \begin{bmatrix} 0.2128 & 0.1360 & 0.1979 & -0.0836 \\ 0.1808 & 0.4420 & -0.3279 & 0.2344 \\ -0.5182 & 0.1728 & -0.5448 & -0.3083 \\ 0.2252 & -0.0541 & -0.4679 & 0.8290 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0101 & 0.0317 & -0.9347 \\ -0.0600 & 0.5621 & 0.1657 \\ -0.3310 & -0.3712 & -0.5846 \\ -0.2655 & 0.4255 & 0.2204 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.6557 & -0.2502 & -0.5188 & -0.1229 \\ 0.6532 & -0.1583 & -0.0550 & -0.2497 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.4326 & 0.1253 & -1.1465 \\ -1.6656 & 0.2877 & 1.1906 \end{bmatrix}$$

3.4.2 Algoritmos

Ao conjunto de dados de entrada - saída dedicados para o procedimento de identificação (seqüência de dados de entrada, figura 3.21, e seqüência de dados de saída, figura 3.22) vamos aplicar os algoritmos MOESP referência, MOESP baseado em Katayama, MOESP modificado, N4SID referência e N4SID modificado.

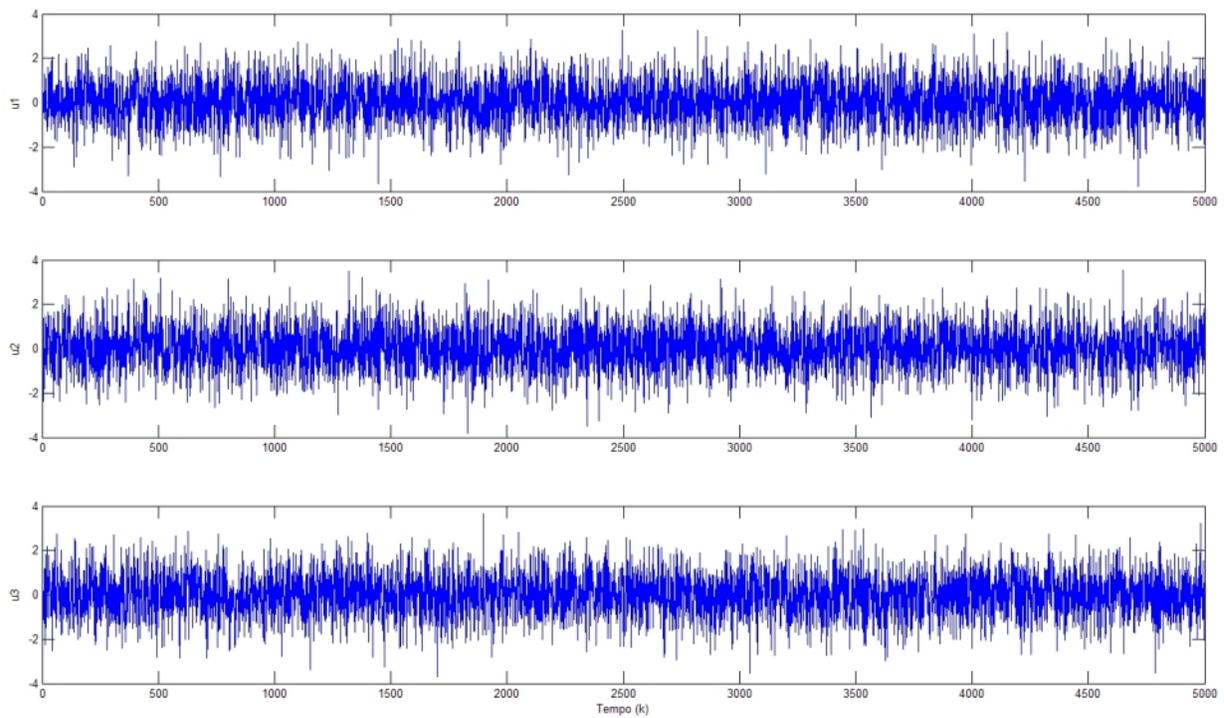


Fig. 3.21: Seqüência de dados de entrada.

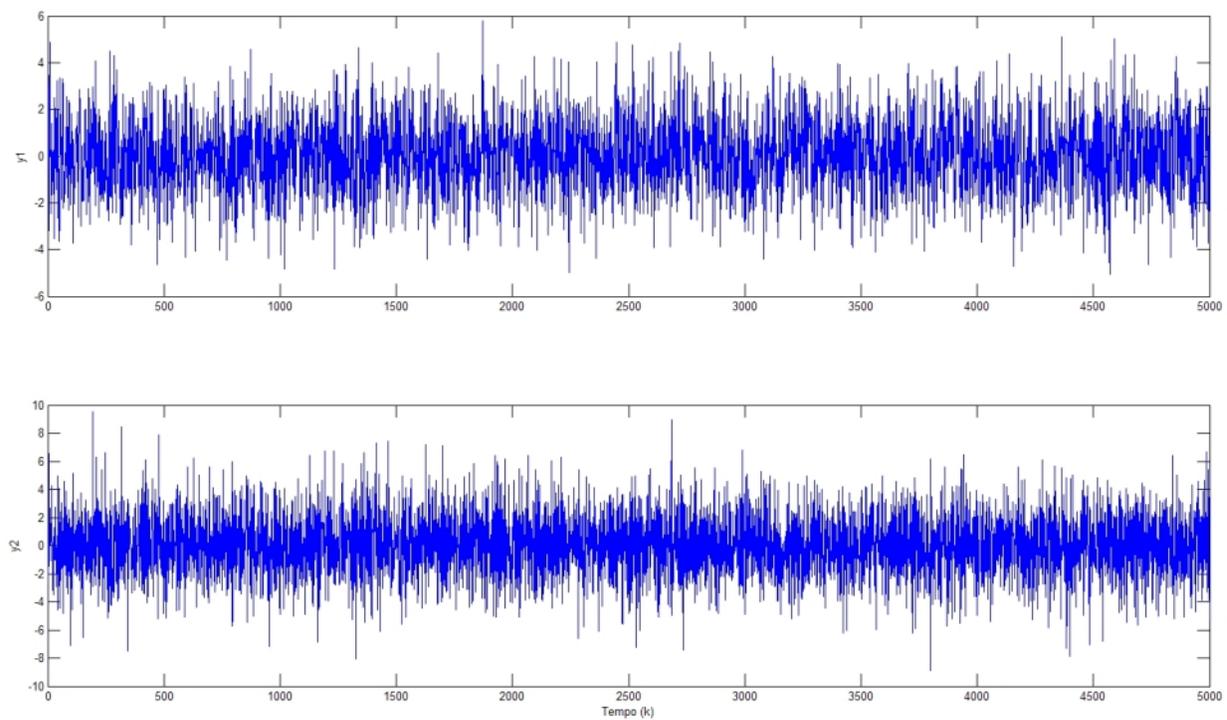


Fig. 3.22: Seqüência de dados de saída.

3.4.3 Parâmetros de identificação

Como o sistema é de igual ordem ao trabalhado nos experimentos anteriores, $n = 4$, vamos tomar também o parâmetro k de igual valor $k = 10$. A figura C.6 do apêndice C, corresponde ao gráfico dos valores singulares, obtido pelo algoritmo MOESP baseado em Katayama.

3.4.4 Resultados

As matrizes obtidas com cada um dos algoritmos são mostradas no apêndice C. De novo, estas matrizes variam do modelo original (benchmark), mas podem se analisar os parâmetros de Markov e os erros de validação para observar se as quádruplas de matrizes podem representar bem o conjunto de entradas - saídas.

3.4.5 Validação do modelo

Além dos parâmetros de Markov do benchmark e do sistema identificado por cada algoritmo, os erros de validação (tabelas C.1 e C.2) e gráficos de encaixe são mostrados no apêndice C (figuras C.7 à C.11).

3.4.6 Conclusões

Estes experimentos foram feitos de maneira menos rigorosa do que os experimentos correspondentes ao benchmark 1 sem perturbações; a idéia de realizar os experimentos com este outro benchmark é exemplificar o procedimento de identificação por Métodos de Subespaços para outro benchmark, e avaliar os algoritmos modificados apresentados neste trabalho. Como para o caso do benchmark 1 sem perturbações, os parâmetros de Markov ficam similares, e os erros de validação são pequenos; em geral, todos os algoritmos tratados permitem obter um modelo no espaço de estado que representa bem os conjuntos de dados de entrada - saída.

Capítulo 4

COMENTÁRIOS E CONCLUSÕES

Tratamos três itens principais nesta tese: teoria, implementação e aplicação de métodos de subespaços para identificação de sistemas lineares invariantes no tempo.

- A teoria para identificação baseada em métodos de subespaços foi apresentada, detalhando aspectos importantes como o teorema principal dos métodos de subespaços e suas características para cada um dos algoritmos tratados. Como analisado, estes métodos requerem mínima informação a priori do sistema em questão, além de ser métodos baseados em álgebra linear numérica, em algoritmos como a decomposição LQ, e a decomposição em valores singulares (SVD). São métodos não iterativos, pelo qual são rápidos e precisam pouco esforço computacional, pois não contêm técnicas de otimização não linear.
- As implementações dos algoritmos fazendo uso da decomposição LQ usada para obter uma projeção ortogonal (como observamos para o algoritmo MOESP), ou projeção oblíqua (como observamos para o algoritmo N4SID), e decomposição SVD utilizada para obter uma estimativa da ordem do sistema. Algumas modificações são propostas para os algoritmos, modificações que logo avaliadas. No apêndice podem-se encontrar os M - files, arquivos de Matlab com os algoritmos implementados.
- As aplicações dos algoritmos a diferentes *benchmarks* (conhecidos da literatura) são apresentados. Estes experimentos permitem ilustrar o procedimento de identificação via métodos de subespaços, e também realizar comparações entre os diferentes algoritmos aqui tratados. Os métodos de subespaços são excelentes para trabalhar com sistemas de múltiplas entradas - múltiplas saídas (MIMO); estes métodos requerem cálculos matriciais que podem ser bem manipulados usando ferramentas matemáticas como Matlab.

Outro importante resultado que cabe ressaltar da elaboração deste estudo é que pode ser uma referência útil para pesquisadores na área de modelagem, controle e processamento de sinais, de

sistemas complexos.

A continuação deste trabalho deve envolver a aplicação dos algoritmos propostos em dados reais, em conjuntos de dados entrada - saída obtidos de um processo industrial, por exemplo da industria aeronáutica.

Como sugestão para trabalhos futuros sugerimos:

- detalhar os efeitos produzidos pelo ruído em cada algoritmo,
- procurar alternativas para a medição da complexidade computacional, por exemplo, fazer a conta manual do número de operações em ponto flutuante, para corroborar os resultados aqui obtidos.

Apêndice A

PROGRAMAS COMPUTACIONAIS

A.1 Algoritmo MOESP baseado em Katayama:

```
% Algoritmo do metodo MOESP baseado em Katayama, com dados de entrada u,  
% saida y, e k a ordem das bloco de Hankel, no gráfico dos valores  
% singulares se pode obter a ordem do sistema
```

```
function [A,B,C,D] = moespkata(u,y,k)
```

```
% Numero de entradas, m, e saidas, l
```

```
[m,nu] = size(u);if (nu < m);u = u';[m,nu] = size(u);end
```

```
[l,ny] = size(y);if (ny < l);y = y';[l,ny] = size(y);end
```

```
% Determinar o numero de columns para as matrizes de Hankel
```

```
N = ny-2*k+1;
```

```
% Construir as matrizes de dados, bloco de Hankel
```

```
U = blkhank(u,k,N);
```

```
Y = blkhank(y,k,N);
```

```
km = size(U,1);
```

```
kl = size(Y,1);
```

```
% ***** ALGORITMO *****
```

```
% Passo 1
```

```
% decomposicao LQ
```

```
[Q,L] = qr([U;Y]',0);
```

```
Q=Q'; L=L';
```

```

L11 = L(1:km,1:km);
L21 = L(km+1:km+k1,1:km);
L22 = L(km+1:km+k1,km+1:km+k1);

% Passo 2
% decomposicao SVD
[UU,SS,VV] = svd(L22);
ss = diag(SS);
hold off
figure(1)
title('Valores singulares');
xlabel('Ordem');
plot(ss)
pause;
figure(2)
title('Valores singulares');
xlabel('Ordem');
bar(ss)
    n = input('    Ordem do sistema ? ');
while isempty(n)
    n = input('    Ordem do sistema ? ');
end
U1 = UU(:,1:n);
S1=SS(1:n,1:n);
On = U1*sqrtm(S1);

% Passo 3
% Obter as matrizes A e C do sistema
C = On(1:l,1:n);
A = pinv(On(1:l*(k-1),1:n))*On(1+1:l*k,1:n);

% Passo 4
% Obter as matrizes B e D do sistema
U2 = UU(:,n+1:size(UU',1));
Z = U2'*L21/L11;
XX = [];
RR = [];

for j=1:k
    XX = [XX; Z(:,m*(j-1)+1:m*j)];
    Onj = On(1:l*(k-j),:);
    Rj = [zeros(1*(j-1),l) zeros(1*(j-1),n);
          eye(l) zeros(l,n); zeros(1*(k-j),l) Onj];

```

```

    RR = [RR; U2'*Rj];
end
DB = pinv(RR)*XX;
D = DB(1:l,:);
B = DB(l+1:size(DB,1),:);

```

A.2 Algoritmo MOESP modificado:

```

% Algoritmo do metodo MOESP modificado, com dados de entrada u,
% saida y, e k a ordem das bloco de Hankel, no gráfico dos valores
% singulares se pode obter a ordem do sistema

```

```
function [A,B,C,D] = moespkatamod(u,y,k)
```

```

[l,ny] = size(y);if (ny < l);y = y';[l,ny] = size(y);end
[m,nu] = size(u);if (nu < m);u = u';[m,nu] = size(u);end

```

```
% Determinar o numero de columns para as matrizes de Hankel
```

```
N = ny-2*k+1;
```

```
%Construir as matrizes de dados, bloco de Hankel
```

```
U = blkhank(u,k,N);
```

```
Y = blkhank(y,k,N);
```

```
km = size(U,1);
```

```
kl = size(Y,1);
```

```
% ***** ALGORITMO *****
```

```
%Passo 1
```

```
%decomposicao LQ
```

```
[Q,L] = qr([U;Y]',0);
```

```
Q=Q'; L=L';
```

```
L11 = L(1:km,1:km);
```

```
L21 = L(km+1:km+kl,1:km);
```

```
L22 = L(km+1:km+kl,km+1:km+kl);
```

```
%Passo 2
```

```
[UU,SS,VV] = svd(L22);
```

```
ss = diag(SS);
```

```
hold off
```

```
figure(1)
```

```

title('Valores singulares');
xlabel('Ordem');
plot(ss)
pause;
figure(2)
title('Valores singulares');
xlabel('Ordem');
bar(ss)
    n = input('    Ordem do sistema ? ');
while isempty(n)
    n = input('    Ordem do sistema ? ');
end
U1 = UU(:,1:n);
S1=SS(1:n,1:n);
On = U1*sqrtm(S1);

%Passo 3
%Matrizes A e C
C = On(1:l,1:n);
A = pinv(On(1:l*(k-1),1:n))*On(1+1:l*k,1:n);

%Passo 4
%Matrizes B e D
TOEP = L21/L11;
G = TOEP(:,1:m);
G0 = G(1:l,:);
G1 = G(1+1:2*l,:);
G2 = G(2*l+1:3*l,:);
G3 = G(3*l+1:4*l,:);
G4 = G(4*l+1:5*l,:);
D = G0;

Hk = [G1 G2;G2 G3;G3 G4];
On1 = On(1:3*l,:);
Ck = pinv(On1)*Hk;
B = Ck(:,1:m);

```

A.3 Algoritmo N4SID modificado:

```

% Algoritmo do metodo N4SID modificado, com dados de entrada u,
% saida y, e k a ordem das bloco de Hankel, no gráfico dos valores

```

```

% singulares se pode obter a ordem do sistema

function [A,B,C,D,n] = n4sidkatamod(u,y,k)

[l,ny] = size(y);if (ny < 1);y = y';[l,ny] = size(y);end
[m,nu] = size(u);if (nu < m);u = u';[m,nu] = size(u);end

% Determinar o numero de columns para as matrizes de Hankel
N = ny-2*k+1;

%construir as matrizes de dados, bloco de Hankel
U = blkhank(u,2*k,N);
Y = blkhank(y,2*k,N);
Up=U(1:k*m,:);
Uf=U(k*m+1:2*k*m,:);
Yp=Y(1:k*1,:);
Yf=Y(k*1+1:2*k*1,:);

km = size(Up,1);
kl = size(Yp,1);

Wp = [Up;Yp];

% ***** ALGORITMO *****
%Passo 1
%decomposicao LQ
[Q,L] = qr([Uf;Up;Yp;Yf]',0);
Q=Q'; L=L';
L11 = L(1:km,1:km);
L21 = L(km+1:2*km,1:km);
L22 = L(km+1:2*km,km+1:2*km);
L31 = L(2*km+1:2*km+kl,1:km);
L32 = L(2*km+1:2*km+kl,km+1:2*km);
L33 = L(2*km+1:2*km+kl,2*km+1:2*km+kl);
L41 = L(2*km+kl+1:2*km+2*kl,1:km);
L42 = L(2*km+kl+1:2*km+2*kl,km+1:2*km);
L43 = L(2*km+kl+1:2*km+2*kl,2*km+1:2*km+kl);
L44 = L(2*km+kl+1:2*km+2*kl,2*km+kl+1:2*km+2*kl);
R11 = L11;
R21 = [L21;L31];
R22 = [L22 zeros(km,kl); L32 L33];
R31 = L41;
R32 = [L42 L43];

```

```

xi = R32*pinv(R22);

%Passo 2
[UU,SS,VV] = svd(xi);
ss = diag(SS);
hold off
figure(1)
title('Valores singulares');
xlabel('Ordem');
plot(ss)
pause;
figure(2)
title('Valores singulares');
xlabel('Ordem');
bar(ss)
    n = input('    Ordem do sistema ? ');
while isempty(n)
    n = input('    Ordem do sistema ? ');
end
U1 = UU(:,1:n);
S1 = SS(1:n,1:n);
V1 = VV(1:n,:);

%Matrizes A e C
Ok = U1*sqrtm(S1);
C = Ok(1:1,1:n);
A = pinv(Ok(1:1*(k-1),1:n))*Ok(1+1:1*k,1:n);

%Passo 3
%Matrizes B e D
TOEP = (R31 - R32*pinv(R22)*R21)*inv(R11);
G = TOEP(:,1:m);
G0 = G(1:1,:);
G1 = G(1+1:2*1,:);
G2 = G(2*1+1:3*1,:);
G3 = G(3*1+1:4*1,:);
G4 = G(4*1+1:5*1,:);
D = G0;

Hk = [G1 G2;G2 G3;G3 G4];
Ok1 = Ok(1:3*1,:);
Ck = pinv(Ok1)*Hk;
B = Ck(:,1:m);

```

Apêndice B

ÁLGEBRA LINEAR

B.1 Projeções ortogonais:

Se \mathbf{P}_B denota o operador que projeta o espaço linha de uma matriz no espaço linha da matriz \mathbf{B} , definido como:

$$\mathbf{P}_F := \mathbf{F}^T(\mathbf{F}\mathbf{F}^T)^\dagger\mathbf{F} \quad (\text{B.1})$$

onde \dagger denota a pseudo-inversa.

A expressão $\mathbf{A}|\mathbf{F}$ indica a projeção ortogonal do espaço linha da matriz \mathbf{A} no espaço linha da matriz \mathbf{F} , dada por:

$$\mathbf{A}|\mathbf{F} := \mathbf{A}\mathbf{P}_F = \mathbf{A}\mathbf{F}^T(\mathbf{F}\mathbf{F}^T)^\dagger\mathbf{F} \quad (\text{B.2})$$

E \mathbf{P}_F^\perp denota o operador que projeta o espaço linha de uma matriz no complemento ortogonal do espaço linha da matriz \mathbf{F} , tal que:

$$\mathbf{P}_F^\perp := \mathbf{I} - \mathbf{P}_F \quad (\text{B.3})$$

A projeção ortogonal do espaço linha da matriz \mathbf{A} no complemento ortogonal do espaço linha da matriz \mathbf{F} , é dada por:

$$\mathbf{A}|\mathbf{F}^\perp := \mathbf{A}\mathbf{P}_F^\perp \quad (\text{B.4})$$

A combinação destes dois operadores, $\mathbf{A}|\mathbf{F}$ e $\mathbf{A}|\mathbf{F}^\perp$, decompõe uma matriz \mathbf{A} em duas matrizes que têm espaços linha ortogonais, figura B.1, assim:

$$\mathbf{A} = \mathbf{A}\mathbf{P}_F + \mathbf{A}\mathbf{P}_F^\perp = \mathbf{A}|\mathbf{F} + \mathbf{A}|\mathbf{F}^\perp \quad (\text{B.5})$$

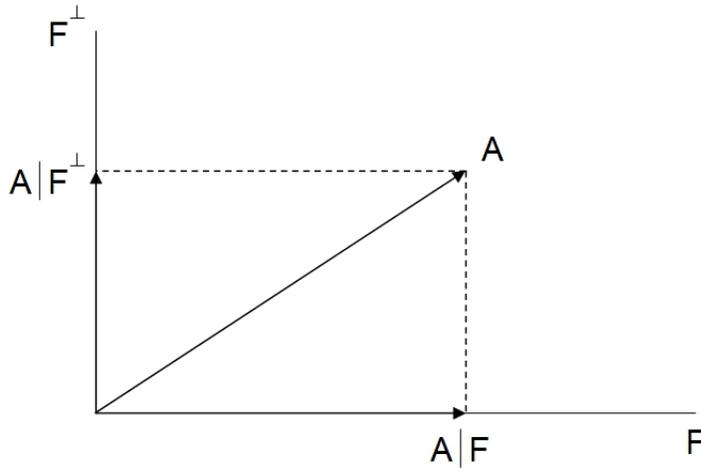


Fig. B.1: Projeção ortogonal.

B.2 Projeções oblíquas:

Se $\mathbf{A}/_G\mathbf{F}$ denota a projeção oblíqua do espaço linha de \mathbf{A} sobre o espaço linha de \mathbf{F} ao longo do espaço linha de \mathbf{G} , e $\mathbf{A}/_F\mathbf{G}$ denota a projeção oblíqua do espaço linha de \mathbf{A} sobre o espaço linha de \mathbf{G} ao longo do espaço linha de \mathbf{F} , figura B.2, tem-se:

$$\mathbf{A} = \mathbf{A}/_G\mathbf{F} + \mathbf{A}/_F\mathbf{G} \quad (\text{B.6})$$

Se $\mathbf{P}_{G,F}$ denota o projetor oblíquo sobre \mathbf{G} ao longo de \mathbf{F} , tem-se:

$$\mathbf{A}\mathbf{P}_{G,F} := \mathbf{A}/_F\mathbf{G} \quad (\text{B.7})$$

Para a projeção oblíqua de \mathbf{A} sobre \mathbf{F} ao longo de \mathbf{G} , tem-se a seguinte expressão:

$$\mathbf{A}/_G\mathbf{F} = [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^\dagger\mathbf{F} \quad (\text{B.8})$$

B.3 Prova equação 2.14:

$$\mathbf{W}_1\bar{\mathbf{P}}_k\mathbf{W}_2 = [\mathbf{Y}_f|\mathbf{U}_f^\perp]\mathbf{P}[\mathbf{W}_p|\mathbf{U}_f^\perp] \quad (\text{B.9})$$

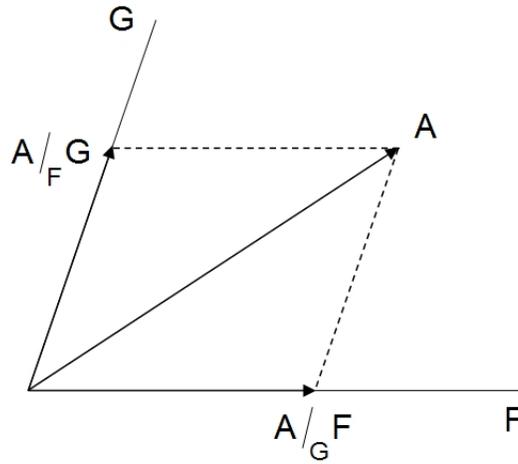


Fig. B.2: Projeção oblíqua.

Tomando a seguinte notação por simplicidade:

$$\mathbf{A} = \mathbf{Y}_f$$

$$\mathbf{F} = \mathbf{W}_p$$

$$\mathbf{G} = \mathbf{U}_f$$

podemos reescrever a equação (B.9) como:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = [\mathbf{A}/_G \mathbf{F}] \mathbf{P}_G^\perp$$

Com a equação (B.8) temos:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = [\mathbf{A} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^\dagger \mathbf{F} \mathbf{P}_G^\perp$$

Da definição de matriz pseudo-inversa, $\mathbf{B}^\dagger = \mathbf{B}^T (\mathbf{B} \mathbf{B}^T)^{-1}$, segue que:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = [\mathbf{A} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^T ([\mathbf{F} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^T)^{-1} \mathbf{F} \mathbf{P}_G^\perp$$

Fazendo, $\mathbf{I} = ([\mathbf{F} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^T) ([\mathbf{F} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^T)^{-1}$, e rearrumando, segue que:

$$\mathbf{W}_1 \bar{\mathbf{P}}_k \mathbf{W}_2 = [\mathbf{A} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^T \mathbf{I} ([\mathbf{F} | \mathbf{G}^\perp] [\mathbf{F} | \mathbf{G}^\perp]^T)^{-1} \mathbf{F} \mathbf{P}_G^\perp$$

$$\begin{aligned}
&= [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)^{-1}([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)^{-1}\mathbf{F}\mathbf{P}_G^\perp \\
&= [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T[\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T[\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)^{-1}\mathbf{F}\mathbf{P}_G^\perp
\end{aligned}$$

Agora, fazendo $\mathbf{B} = [\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T$, e como:

$$\mathbf{B}^T = ([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)^T = ([\mathbf{F}|\mathbf{G}^\perp]^T)^T[\mathbf{F}|\mathbf{G}^\perp]^T = [\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T = \mathbf{B}$$

pelas propriedades da transposição $(\mathbf{A}^T)^T = \mathbf{A}$, e $(\mathbf{AB})^T = \mathbf{A}^T\mathbf{B}^T$. Voltando para a equação principal, e substituindo nesta a nomenclatura de \mathbf{B} obtemos:

$$\begin{aligned}
\mathbf{W}_1\bar{\mathbf{P}}_k\mathbf{W}_2 &= [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T\mathbf{B}^T(\mathbf{BB}^T)^{-1}\mathbf{F}\mathbf{P}_G^\perp \\
&= [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T\mathbf{B}^\dagger\mathbf{F}\mathbf{P}_G^\perp
\end{aligned}$$

Substituindo \mathbf{B} :

$$\mathbf{W}_1\bar{\mathbf{P}}_k\mathbf{W}_2 = [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)^\dagger\mathbf{F}\mathbf{P}_G^\perp$$

e da equação (B.2) finalmente temos:

$$\begin{aligned}
\mathbf{W}_1\bar{\mathbf{P}}_k\mathbf{W}_2 &= [\mathbf{A}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T([\mathbf{F}|\mathbf{G}^\perp][\mathbf{F}|\mathbf{G}^\perp]^T)^\dagger\mathbf{F}|\mathbf{G}^\perp \\
&= [\mathbf{A}|\mathbf{G}^\perp]\mathbf{P}_{[\mathbf{F}|\mathbf{G}^\perp]}
\end{aligned}$$

Que corresponde, pela notação tomada, a:

$$\mathbf{W}_1\bar{\mathbf{P}}_k\mathbf{W}_2 = [\mathbf{Y}_f|\mathbf{U}_f^\perp]\mathbf{P}_{[\mathbf{W}_p|\mathbf{U}_f^\perp]}$$

B.4 Prova equação 2.31:

$$\mathbf{Y}_p | \mathbf{U}_p^\perp = \mathbf{L}_{22} \mathbf{Q}_2^T \quad (\text{B.10})$$

Supondo a decomposição LQ dada por:

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \quad (\text{B.11})$$

temos, da equação (B.2), que:

$$\mathbf{Y}_p | \mathbf{U}_p = \mathbf{Y}_p \mathbf{U}_p^T (\mathbf{U}_p \mathbf{U}_p^T)^\dagger \mathbf{U}_p \quad (\text{B.12})$$

Da definição de matriz pseudo-inversa, $\mathbf{B}^\dagger = \mathbf{B}^T (\mathbf{B} \mathbf{B}^T)^{-1}$, segue que:

$$\mathbf{Y}_p | \mathbf{U}_p = \mathbf{Y}_p \mathbf{U}_p^T (\mathbf{U}_p \mathbf{U}_p^T)^T ((\mathbf{U}_p \mathbf{U}_p^T) (\mathbf{U}_p \mathbf{U}_p^T)^T)^{-1} \mathbf{U}_p$$

e como $(\mathbf{U}_p \mathbf{U}_p^T)^T = (\mathbf{U}_p \mathbf{U}_p^T)$, e simplificando obtemos que:

$$\mathbf{Y}_p | \mathbf{U}_p = \mathbf{Y}_p \mathbf{U}_p^T (\mathbf{U}_p \mathbf{U}_p^T)^{-1} \mathbf{U}_p$$

Agora, substituindo os valores de \mathbf{Y}_p e \mathbf{U}_p da equação (B.11), obtemos:

$$\begin{aligned} \mathbf{Y}_p | \mathbf{U}_p &= (\mathbf{L}_{21} \mathbf{Q}_1^T + \mathbf{L}_{22} \mathbf{Q}_2^T) \mathbf{Q}_1 \mathbf{L}_{11}^T (\mathbf{L}_{11} \mathbf{Q}_1^T (\mathbf{L}_{11} \mathbf{Q}_1^T)^T)^{-1} \mathbf{L}_{11} \mathbf{Q}_1^T \\ &= \mathbf{L}_{21} \mathbf{Q}_1^T \mathbf{Q}_1 \mathbf{L}_{11}^T (\mathbf{L}_{11} \mathbf{Q}_1^T \mathbf{Q}_1 \mathbf{L}_{11}^T)^{-1} \mathbf{L}_{11} \mathbf{Q}_1^T + \mathbf{L}_{22} \mathbf{Q}_2^T \mathbf{Q}_1 \mathbf{L}_{11}^T (\mathbf{L}_{11} \mathbf{Q}_1^T \mathbf{Q}_1 \mathbf{L}_{11}^T)^{-1} \mathbf{L}_{11} \mathbf{Q}_1^T \\ &= \mathbf{L}_{21} \mathbf{L}_{11}^T (\mathbf{L}_{11} \mathbf{L}_{11}^T)^{-1} \mathbf{L}_{11} \mathbf{Q}_1^T \end{aligned}$$

pois $\mathbf{Q}_1^T \mathbf{Q}_1 = \mathbf{I}$, e $\mathbf{Q}_2^T \mathbf{Q}_1 = 0$. Da definição de matriz pseudo-inversa tem-se:

$$\mathbf{Y}_p | \mathbf{U}_p = \mathbf{L}_{21} \mathbf{L}_{11}^\dagger \mathbf{L}_{11} \mathbf{Q}_1^T = \mathbf{L}_{21} \mathbf{Q}_1^T \quad (\text{B.13})$$

Lembrando a equação (B.5) da projeção ortogonal, temos:

$$\mathbf{Y}_p = \mathbf{Y}_p | \mathbf{U}_p + \mathbf{Y}_p | \mathbf{U}_p^\perp$$

comparando-a com a equação (B.11):

$$\mathbf{Y}_p = \mathbf{L}_{21}\mathbf{Q}_1^T + \mathbf{L}_{22}\mathbf{Q}_2^T$$

finalmente, com as equações (B.5) e (B.13), concluímos que:

$$\mathbf{Y}_p|_{\mathbf{U}_p^\perp} = \mathbf{L}_{22}\mathbf{Q}_2^T \tag{B.14}$$

Apêndice C

EXPERIMENTOS

C.1 Benchmark 1 com ruído

C.1.1 Validação do modelo

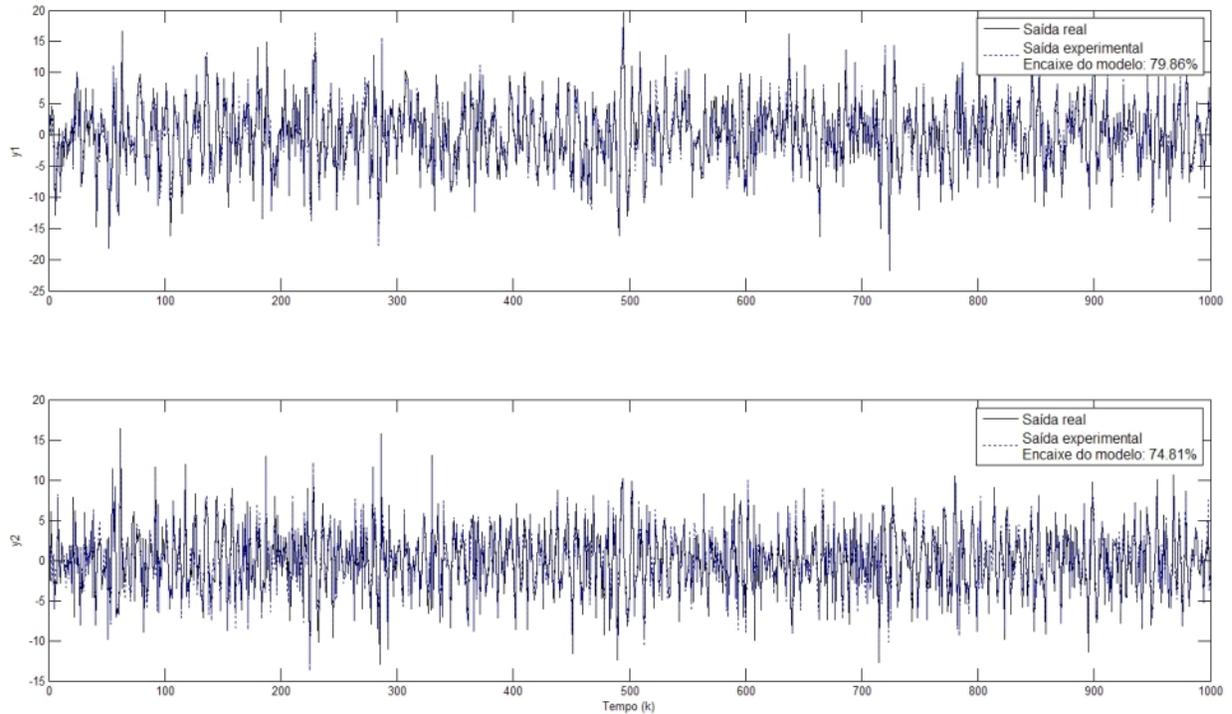


Fig. C.1: Gráfico de encaixe do modelo para o algoritmo MOESP 1.

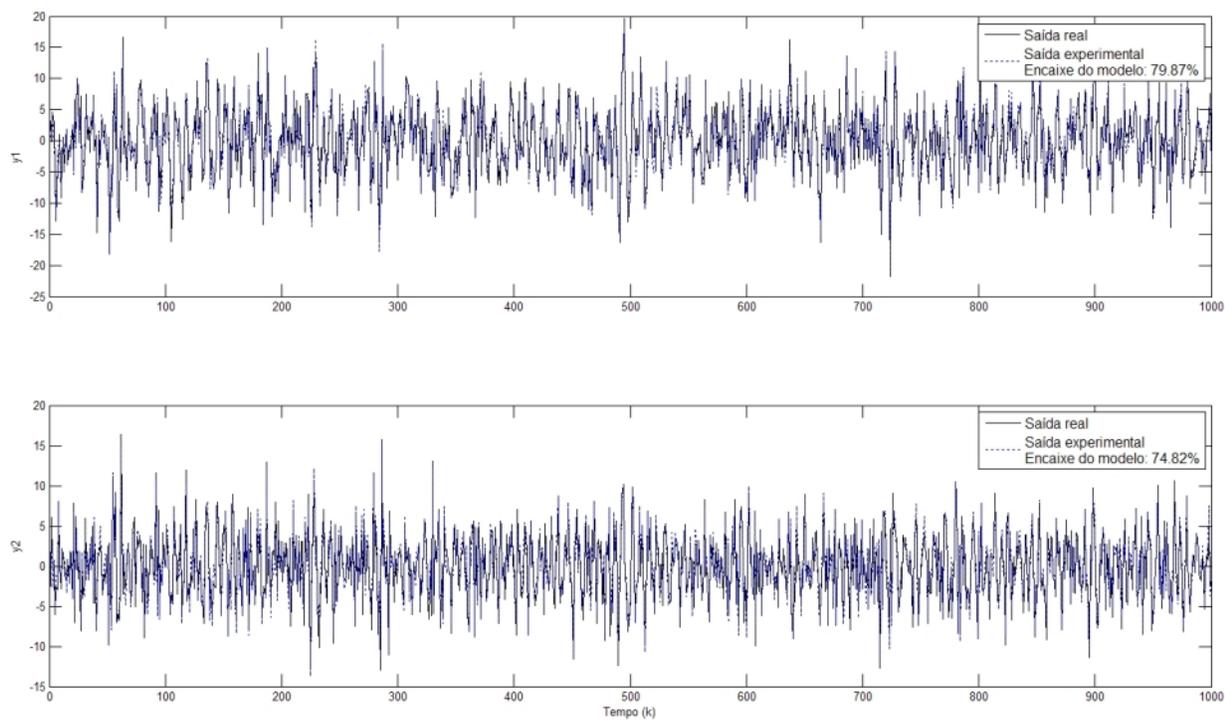


Fig. C.2: Gráfico de encaixe do modelo para o algoritmo MOESP 2.

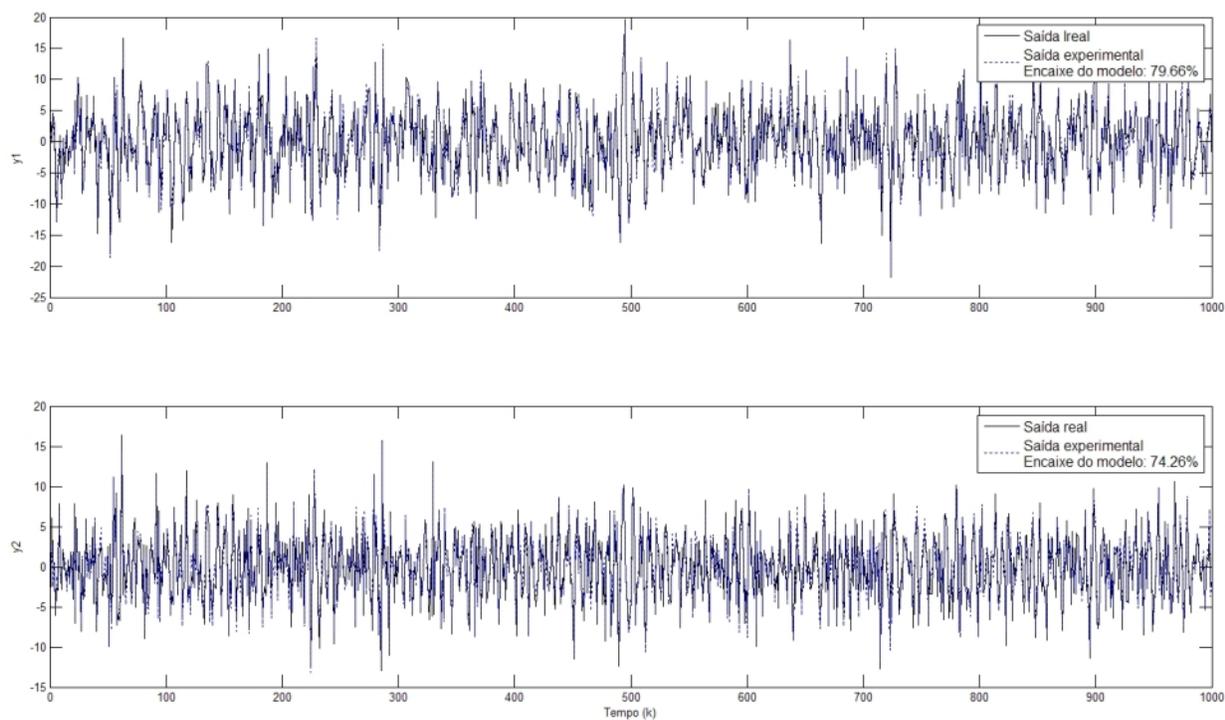


Fig. C.3: Gráfico de encaixe do modelo para o algoritmo MOESP 3.

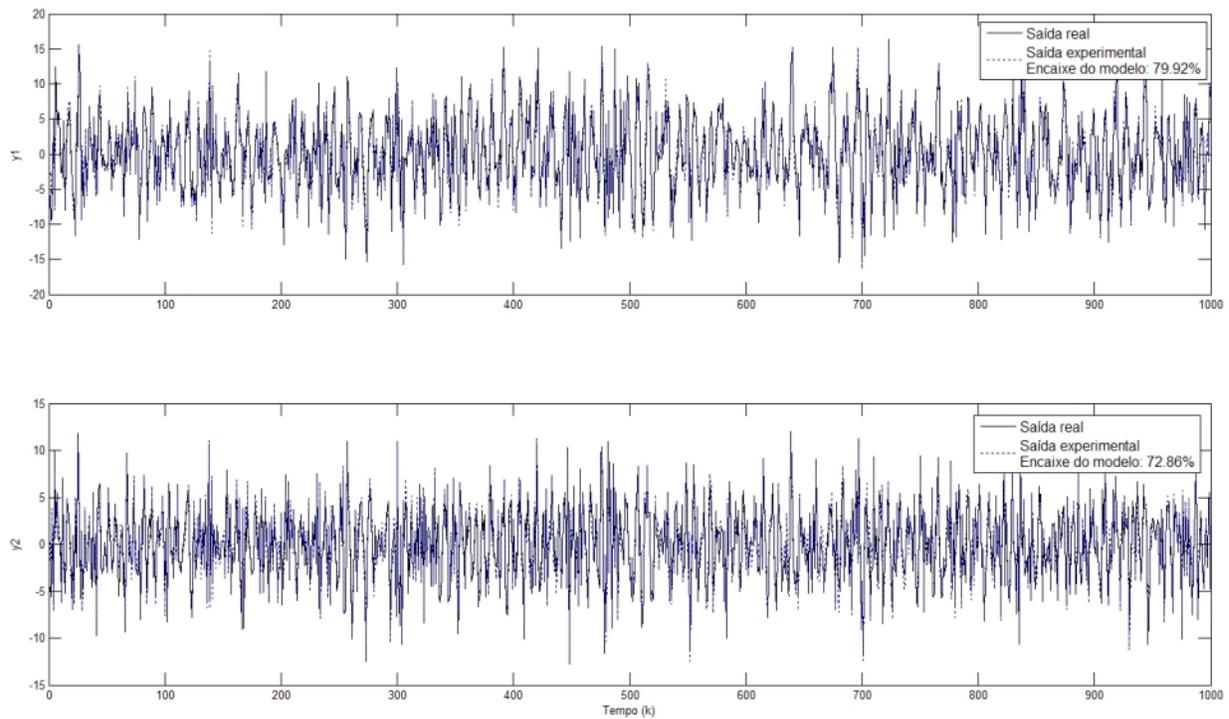


Fig. C.4: Gráfico de encaixe do modelo para o algoritmo N4SID 1.

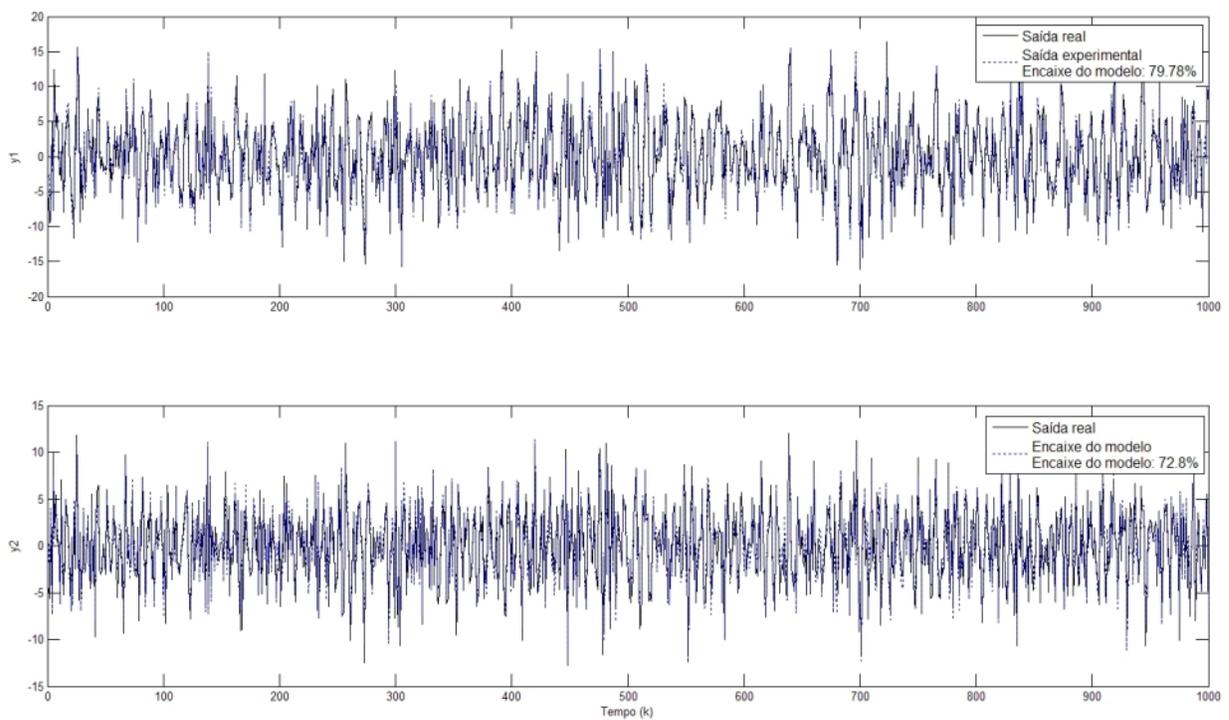


Fig. C.5: Gráfico de encaixe do modelo para o algoritmo N4SID 2.

C.2 Benchmark 2

C.2.1 Parâmetros de identificação

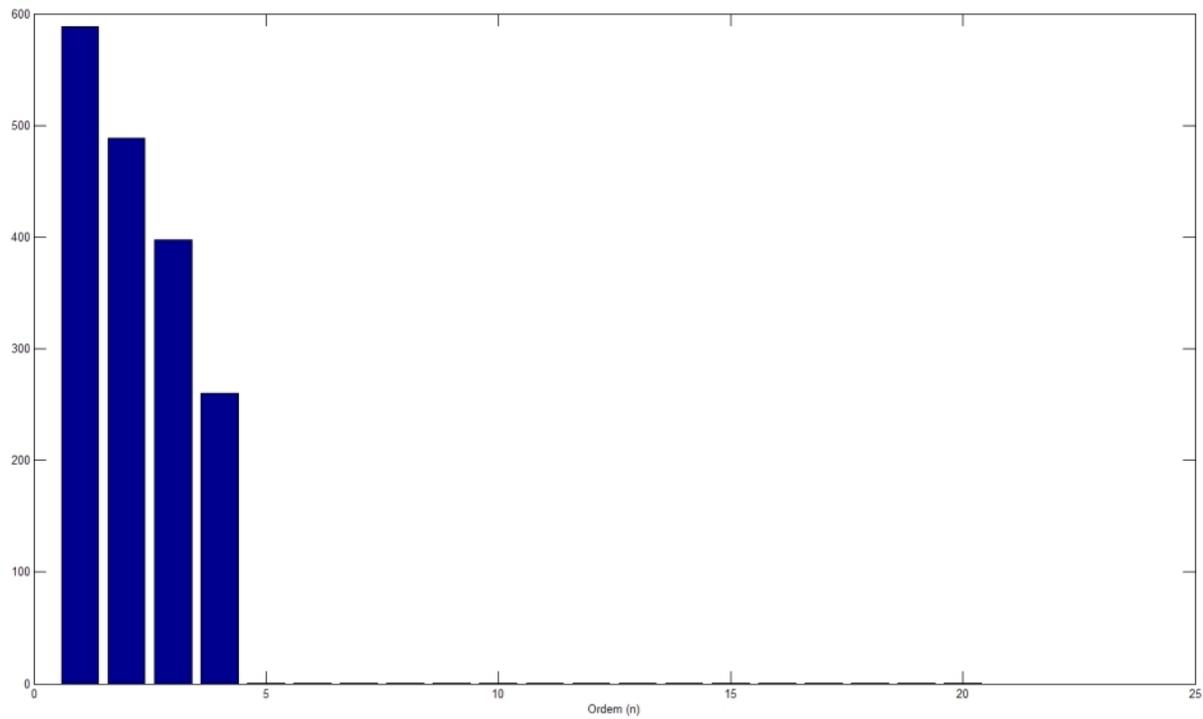


Fig. C.6: Gráfico de barras dos valores singulares para o MOESP de Katayama.

C.2.2 Resultados

MODELO OBTIDO POR MOESP 1

A =

0.3665	0.3933	-0.5739	0.1109
-0.3749	-0.6793	-0.4147	0.4063
-0.0753	0.0212	0.7710	0.5434
0.0001	-0.0048	-0.0097	0.4809

B =

-0.0852	0.2928	1.0418
0.3031	0.2542	0.3479
0.1121	-0.2118	0.2297

-0.0096 -0.0124 0.0029

C =

-0.5395 0.6125 -0.1322 0.4265
-0.6574 0.0981 -0.0317 -0.5421

D =

-0.4326 0.1253 -1.1465
-1.6656 0.2877 1.1906

MODELO OBTIDO POR MOESP 2

A =

0.3767 0.3445 -0.3600 0.0315
-0.4208 -0.6756 -0.3200 0.0635
-0.1349 0.0214 0.7540 0.0849
0.0016 -0.0373 -0.0526 0.4839

B =

-0.0094 0.0326 0.1082
0.0356 0.0304 0.0397
0.0191 -0.0366 0.0423
-0.0089 -0.0114 0.0026

C =

-5.0811 5.2334 -0.8790 0.4915
-6.2405 0.8830 -0.4174 -0.6101

D =

-0.4326 0.1253 -1.1465
-1.6656 0.2877 1.1906

MODELO OBTIDO POR MOESP 3

A =

0.3767 0.3445 -0.3600 0.0315
-0.4208 -0.6756 -0.3200 0.0635
-0.1349 0.0214 0.7540 0.0849
0.0016 -0.0373 -0.0526 0.4839

B =

-0.0078 0.0337 0.1068
0.0333 0.0296 0.0415

0.0186	-0.0372	0.0429
-0.0091	-0.0113	0.0027

C =

-5.0811	5.2334	-0.8790	0.4915
-6.2405	0.8830	-0.4174	-0.6101

D =

-0.4253	0.1015	-1.1570
-1.6715	0.2637	1.1925

MODELO OBTIDO POR N4SID 1

A =

0.3991	0.3549	-0.3664	0.0282
-0.3937	-0.6706	-0.3578	0.0633
-0.1579	0.0172	0.7255	0.0858
0.0021	-0.0388	-0.0514	0.4851

B =

-0.0850	0.2752	0.8851
0.2972	0.2629	0.3430
0.1568	-0.2825	0.3987
-0.0717	-0.0987	0.0269

C =

-0.6147	0.6163	-0.1180	0.0586
-0.7502	0.0981	-0.0685	-0.0728

D =

-0.4326	0.1253	-1.1465
-1.6656	0.2877	1.1906

MODELO OBTIDO POR N4SID 2

A =

0.7565	0.2993	-0.3183	-0.0069
-0.0973	-0.6146	-0.6236	0.0614
-0.1448	0.0961	0.3023	0.1104
0.0122	-0.0524	-0.0575	0.4947

B =

-0.1319	0.3535	0.6506
0.3442	0.3134	0.2781
0.1293	-0.0080	1.0769

-0.0491 -0.1063 0.0877

C =

-0.4230 0.5562 -0.2440 0.0592
 -0.5799 0.1300 -0.2897 -0.0740

D =

-0.4326 0.1253 -1.1465
 -1.6656 0.2877 1.1906

C.2.3 Validação do modelo

PARAMETROS DE MARKOV DO BENCHMARK

d =

-0.4326 0.1253 -1.1465
 -1.6656 0.2877 1.1906

cb =

0.2127 0.0204 -0.3781
 0.0874 -0.1541 -0.6597

cab =

-0.1651 -0.2765 -0.6657
 -0.0360 -0.2262 -0.3273

ca2b =

0.1256 -0.0024 0.2887
 0.0984 -0.0833 0.1614

PARAMETROS DE MARKOV DO MOESP 1

D =

-0.4326 0.1253 -1.1465
 -1.6656 0.2877 1.1906

CB =

0.2127 0.0204 -0.3781
 0.0874 -0.1541 -0.6597

CAB =

-0.1651	-0.2765	-0.6657
-0.0360	-0.2262	-0.3273

CA2B =

0.1256	-0.0024	0.2887
0.0984	-0.0833	0.1614

PARAMETROS DE MARKOV DO MOESP 2

D =

-0.4326	0.1253	-1.1465
-1.6656	0.2877	1.1906

CB =

0.2127	0.0204	-0.3781
0.0874	-0.1541	-0.6597

CAB =

-0.1651	-0.2765	-0.6657
-0.0360	-0.2262	-0.3273

CA2B =

0.1256	-0.0024	0.2887
0.0984	-0.0833	0.1614

PARAMETROS DE MARKOV DO MOESP 3

D =

-0.4253	0.1015	-1.1570
-1.6715	0.2637	1.1925

CB =

0.1929	0.0108	-0.3618
0.0758	-0.1616	-0.6493

CAB =

-0.1590	-0.2759	-0.6699
-0.0346	-0.2277	-0.3274

CA2B =

0.1206	-0.0051	0.2933
--------	---------	--------

0.0946 -0.0861 0.1653

PARAMETROS DE MARKOV DO N4SID 1

D =

-0.4326 0.1253 -1.1465
-1.6656 0.2877 1.1906

CB =

0.2127 0.0204 -0.3781
0.0874 -0.1541 -0.6597

CAB =

-0.1651 -0.2765 -0.6657
-0.0360 -0.2262 -0.3273

CA2B =

0.1256 -0.0024 0.2887
0.0984 -0.0833 0.1614

PARAMETROS DE MARKOV DO N4SID 2

D =

-0.4326 0.1253 -1.1465
-1.6656 0.2877 1.1906

CB =

0.2127 0.0204 -0.3781
0.0874 -0.1541 -0.6597

CAB =

-0.1651 -0.2765 -0.6657
-0.0360 -0.2262 -0.3273

CA2B =

0.1256 -0.0024 0.2887
0.0984 -0.0833 0.1614

Algoritmo	y_1	y_2
MOESP 1	0.68	0.59
MOESP 2	0.68	0.59
MOESP 3	2.71	1.48

Tab. C.1: Erro de validação (em porcentagem) para os algoritmos MOESP.

Algoritmo	y_1	y_2
N4SID 1	3.45	3.45
N4SID 2	3.45	3.45

Tab. C.2: Erro de validação (em porcentagem) para os algoritmos N4SID.

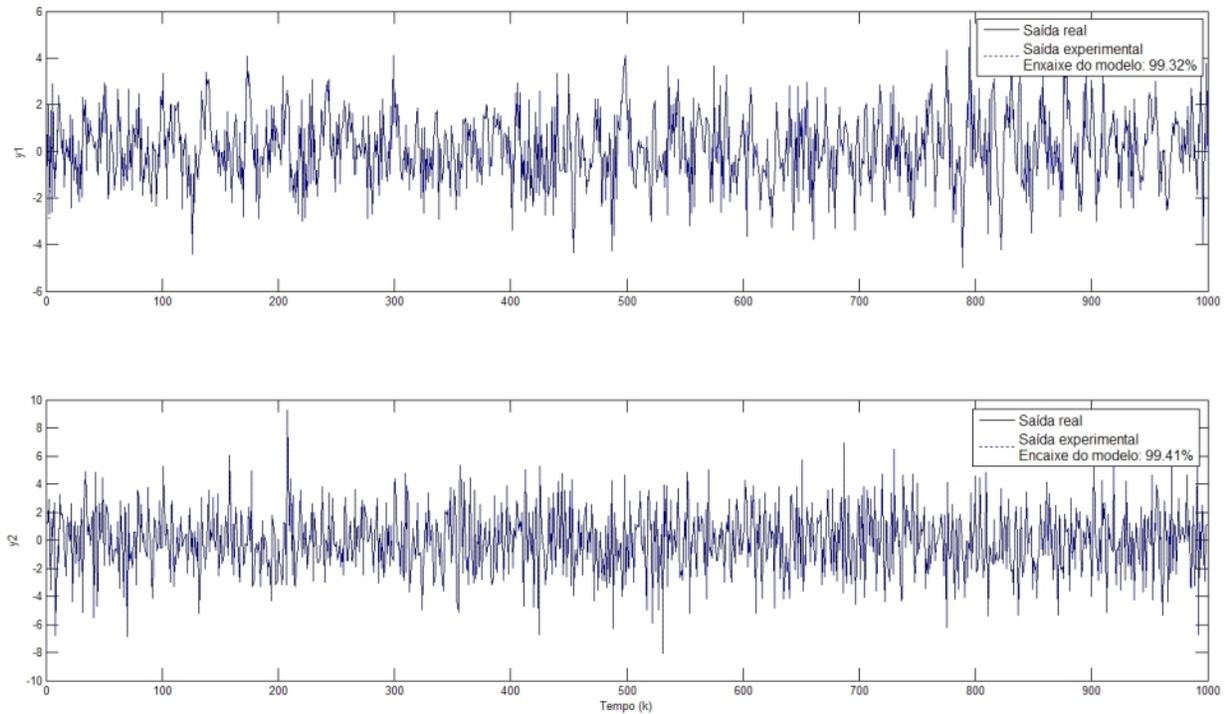


Fig. C.7: Gráfico de encaixe do modelo para o algoritmo MOESP 1.

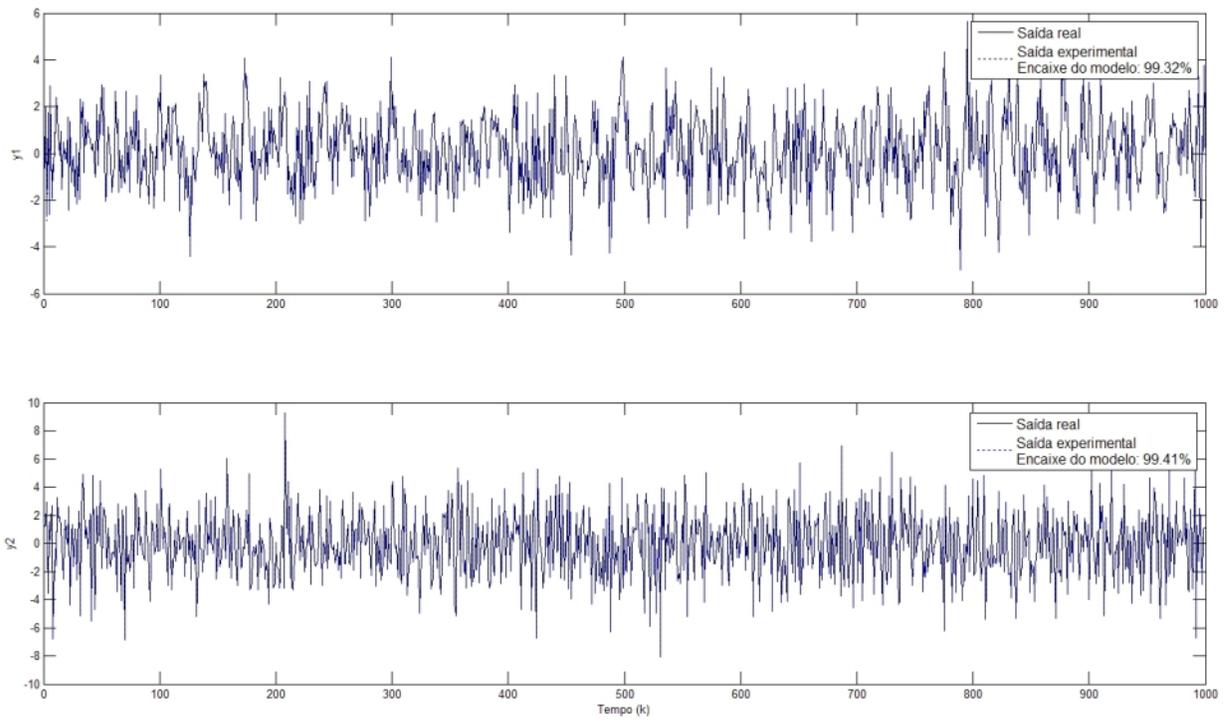


Fig. C.8: Gráfico de encaixe do modelo para o algoritmo MOESP 2.

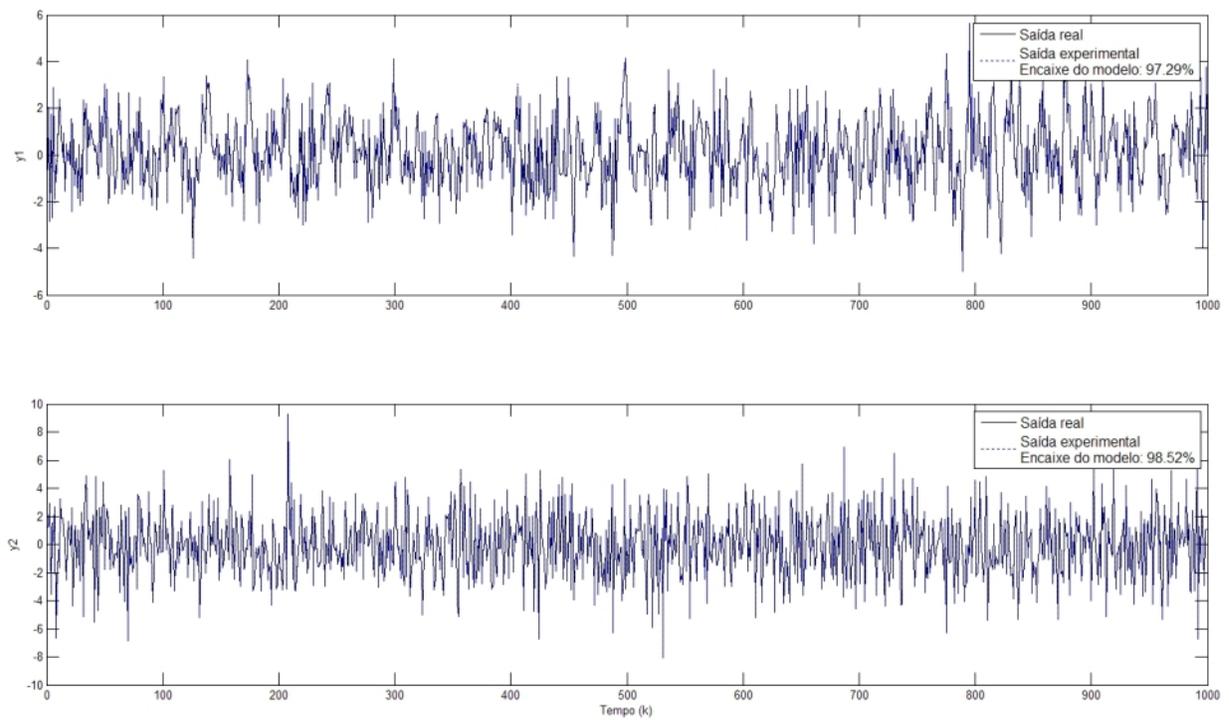


Fig. C.9: Gráfico de encaixe do modelo para o algoritmo MOESP 3.

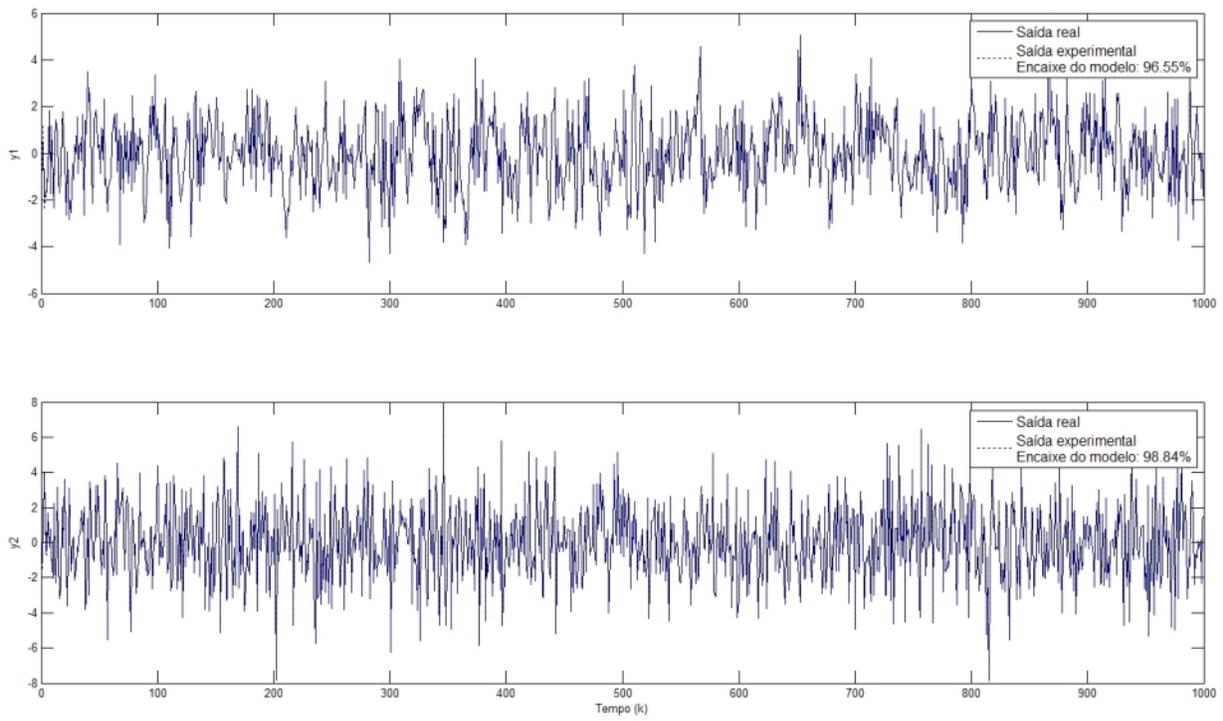


Fig. C.10: Gráfico de encaixe do modelo para o algoritmo N4SID 1.

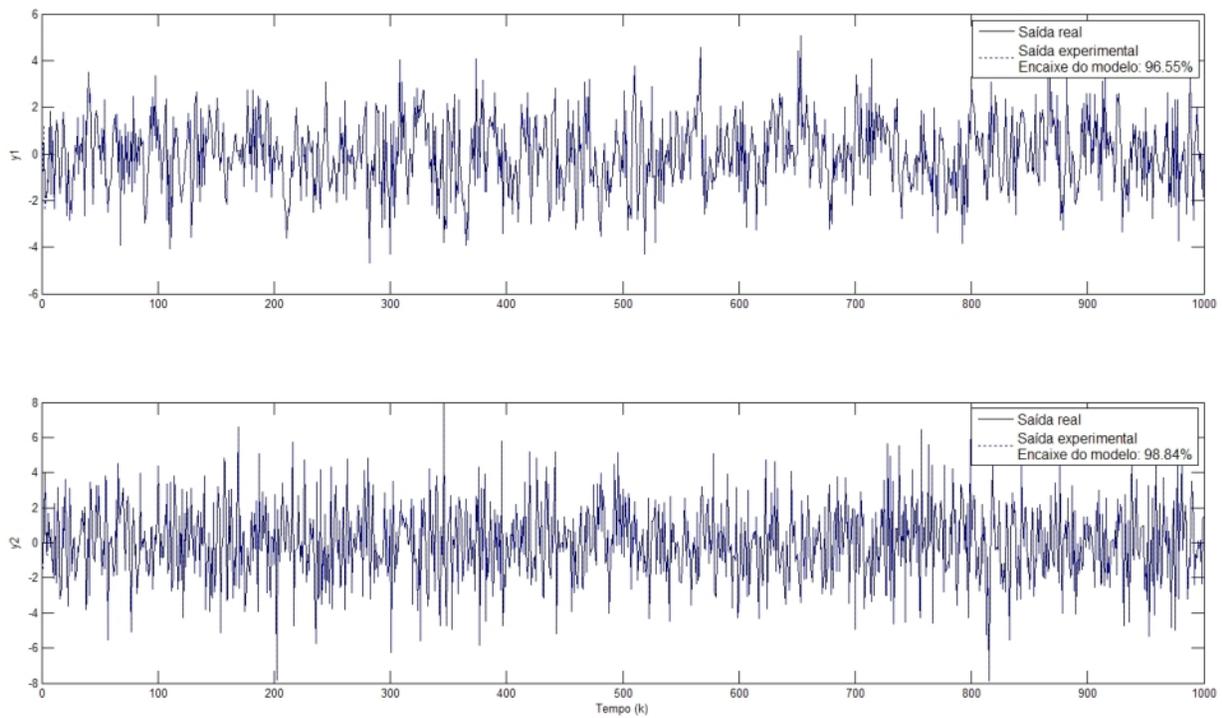


Fig. C.11: Gráfico de encaixe do modelo para o algoritmo N4SID 2.

Referências Bibliográficas

- [1] Aström, Karl J and Eykhoff P. *System Identification - A Survey*. *Automatica*, 7(2):123–162, Março 1971.
- [2] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel. *DAISY: Database for Identification of Systems*. *ESAT-SISTA, K.U.Leuven, Belgium, Journal A*, 38(3):4–5, 1997.
- [3] Barreto, Gilmar. *Modelagem Computacional Distribuída e Paralela de Sistemas e de Séries Temporais Multivariáveis no Espaço de Estado*. *Faculdade de Engenharia Elétrica e de Computação, UNICAMP*, Junho 2002.
- [4] Bottura, Celso P. *Análise Linear de Sistemas*. *Guanabara Dois*, 1982.
- [5] Elizabeth Bradley and Reinhard Easley, Matthew e Stolle. *Reasoning about Nonlinear System Identification*. *Artificial Intelligence*, 133:139–188, December 2001.
- [6] Bremer, W.A.P. van den and Marechal, T.M.A. and Vugts, R.P.A. *Linear Identification of Non-linear Systems*. *Dynamics and Control Technology Group, Departament Mechanical Engineering, Technische Universitei Eindhoven*, June 2006.
- [7] Chiuso, Alessandro and Picci Giorgio. *Some Algorithmic Aspect of Subspace Identification with Inputs*. *International Journal of Applied Mathematics and Computer Science*, 11(1):55–75, 2001.
- [8] Corteletti, R. and Barros, P.R. and Lima, A.M.N. *Parameters estimation of induction motor using subspace methods*. *ISIE '03, IEEE International Symposium on Industrial Electronics*, 2:667–671, Junho 2003.
- [9] De Moor, B. and Van Overschee, P. and Favoreel, W. *Numerical Algorithms for Subspace State-Space System Identification - An Overview*. *Applied and Computational Control, Signals and Circuits, Boston*, 1(6):385–422, 1998.

- [10] De Moor, Bart and Overschee, Peter Van. *Subspace Identification for Linear Systems: Theory, Implementations, Applications*. Kluwer Academic Publishers, 1996.
- [11] Del Real Tamariz, Annabell. *Modelagem Computacional de Dados e Controle Inteligente no Espaço de Estado*. Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Julho 2005.
- [12] Desai, Prasun N. and Knocke, Philip C. *Mars Exploration Rovers Entry, Descent, and Landing Trajectory Analysis*. 42nd AIAA Aerospace Science Meeting, 2004.
- [13] Elkaim, Gabriel H. *System Identification for Precision Control of a Wingsailed GPS-Guided Catamaran*. Department of Aeronautics and Astronautics, Stanford University, Dezembro 2002.
- [14] Favoreel, W. and Van Huffel, S. and De Moor, B. and Sima, V. and Verhaegen, M. *Comparative Study between three Subspace Identification Algorithms*. Proceedings of European Control Conference, Agosto-Setembro 1999.
- [15] Flint, T. and Vaccaro, R. *Performance Analysis of N4SID State-Space System Identification*. Proceedings of the American Control Conference, June 1998.
- [16] Haverkamp, B.R.J. *Efficient implementation of Subspace method identification Algorithms*. NICONET Report, 1999.
- [17] Katayama, Tohru. *Subspace Methods for System Identification*. Springer, 2005.
- [18] Katayama, Tohru. *A System Theoretic Interpretation of LQ Decomposition in Subspace Identification Methods*. Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, pages 1089–1095, Julho 2006.
- [19] Kawauchi, H. and Chiuso, A. and Katayama, T. and Picci, G. *A Comparison of Two Stochastic Subspace System Identification Methods*. Technical Report, Department of Applied Mathematics and Physics, Kyoto University, 2000.
- [20] Kay, H. and Rinner, B. and Kuipers, B. *Semi-Quantitative System Identification*, journal = Artificial Intelligence, year = 2000, optkey = , volume = 119, number = , pages = 103–140, month = august, optnote = citeseer.ist.psu.edu/article/kay99semiquantitative.html, optannote = .
- [21] Kwakemaak, H., and Sivan, R. *Linear Optimal Control Systems*. Wiley-Interscience, 1972.
- [22] Ljung, L. *System Identification Toolbox: User's Guide*. The MathWorks, (2), 2004.

- [23] Ljung, Lennart. *Development of System Identification*. Department of Electrical Engineering, Linköping University, Sweden, 1996.
- [24] Ljung, Lennart. *System Identification: Theory for the User*. Prentice Hall, (2), 1999.
- [25] Luna, Ivette. *Análise de Séries Temporais e Modelagem Baseada em Regras Nebulosas*. Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Outubro 2007.
- [26] Magnus, Jansson. *On Subspace Methods in System Identification and Sensor Array Processing*. Automatic Control Department of Signals, Sensors and Systems, Royal Institute of Technology, June 1997.
- [27] Markovsky, I. and Willems, J. and De Moor, B. *Comparison of Identification Algorithms on the Database DAISY*. *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 2858–2869, Decembro 2005.
- [28] Mhod-Mokhtar, Rosmiwati and Wang, Liuping. *Continue time System Identification Using Subspace Methods*. *Proceedings of the 7th Biennial Engineering Mathematics and Applications Conference, EMAC-2005*, 47:C712–C732, Junho 1971.
- [29] NASA. *PDS: Planetary Data System*. 2008.
- [30] Nitta, Masuhiro. *Design of Compensation Input Based on LQ Decomposition in N4SID Method*. *SICE Annual Conference*, pages 2530–2534, Setembro 2007.
- [31] Ogata, Katsuhiko. *Ingeniería de Control Moderna*. Prentice Hall, (3), 1998.
- [32] Giorgio Picci and Tohru Katayama. *Stochastic realization with exogenous inputs and subspace-methods identification*. *Signal Process*, 52(2):145–160, 1996.
- [33] Serra, Ginalber L. De Oliveira. *Propostas de Metodologias para Identificação e Controle Inteligente*. Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Setembro 2005.
- [34] Åström, Karl J. and Wittenmark, Björn. *Computer-Controlled Systems: Theory and Design*. Prentice Hall, (3), 1997.
- [35] The MathWors. *Matlab 6.5.1 Release Notes*.
- [36] Van Der Veen, A. and Deprettere, E. and Swindlehurst, A. *Subspace-Based Signal Analysis Using Singular Value Decomposition*. *Proceedings of the IEEE*, 81(9):1277–1308, September 1993.

- [37] Peter Van Overschee and Bart De Moor. *N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems*. *Automatica*, 30(1):75–93, 1994.
- [38] Peter Van Overschee and Bart De Moor. *A unifying theorem for three subspace system identification algorithms*. *Automatica*, 31:1853–1864, 1995.
- [39] Van Overschee, Peter and De Moor, Bart. *Subspace Identification of a Glass Tube*. *desconhecido*, 1998.
- [40] Van Overschee, Peter and De Moor, Bart and Favoreel, W. *Numerical algorithms for subspace state space system identification - An Overview*. *Applied and Computational Control, Signals and Circuits*, pages 385–422, 1998.
- [41] Van Overschee, Peter and De Moor, Bart and Favoreel, W. *Subspace State Space System Identification for Industrial Processes*. *Journal of Process Control*, vol. 10, pages 149–155, 2000.
- [42] Vavasis, Stephen. *Exchanges in NA Digest Regarding flops*. *CME 302/CS 237A, Numerical Linear Algebra*, 2005.
- [43] Verhaegen, M. and Dewilde, P. *Subspace model identification part 1. The output-error state-space model identification class of algorithms*. *International journal of control*, 56(5):1187–1210, November 1992.
- [44] Verhaegen, M. and Dewilde, P. *Subspace model identification part 2. Analysis of the elementary output-error state-space model identification algorithm*. *International journal of control*, 56(5):1211–1241, November 1992.
- [45] Verhaegen, Michel. *Identification of the deterministic part of MIMO state space models given in innovations form from input-output data*. *Automatica*, 30(1):61–74, 1994.
- [46] Whaley, R. and Dongarra, J. *Automated Empirical Optimizations of Software and the ATLAS project*.