

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Identificação de Sistemas Utilizando a Parametrização MOLI

Patrícia Gomes Saraiva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Paulo Lopes dos Santos

17 de Junho de 2018

Resumo

Nesta dissertação, são propostos novos algoritmos para identificação de sistemas lineares e invariantes no tempo com múltiplas entradas e uma só saída. Nestes algoritmos, o sistema é descrito por um modelo de espaço de estados na forma canônica observável, sendo este sistema representado por um Observador de Luenberger com uma matriz de estado conhecida. Desse modo, o problema de identificação é reduzido ao da estimação da matriz de entrada e do ganho do observador. A solução passa, assim, por aplicar um simples Estimador de Mínimos Quadrados. A qualidade do estimador depende da matriz de estado do observador do sistema, mais concretamente, do seu polinômio característico. Propõem-se, então, os seguintes algoritmos de identificação: o Baricentro Iterativo, o Baricentro Iterativo Modificado e o *MOLIZOE*. O Baricentro Iterativo estima os parâmetros do polinômio característico através de uma média pesada. Para calcular esta média, é gerado, em cada iteração, um polinômio, designado por "curiosidade", ao qual é atribuído um peso (massa) que cresce inversamente com a função custo. Os parâmetros estimados são a média pesada, o baricentro ou o centro de massa de todas as "curiosidades" geradas até ao momento. Esta média é calculada recursivamente em cada iteração. Já o Baricentro Iterativo Modificado está baseado numa interpretação bayesiana do Baricentro. Este algoritmo encara cada iteração como um Estimador de Bayes, o qual vai tendo cada vez mais informação sobre o sistema a estimar. Formula-se, ainda, o *MOLIZOE*, o algoritmo das elipses, que minimiza a função custo ao calcular sucessivamente o baricentro de "curiosidades" situadas em elipsoides convenientemente geradas.

No estudo de caso, que consiste em três simulações de Monte Carlo de 100 experiências de identificação de um sistema de ordem 2 com relações sinal-ruído de 20dB, 10dB e 0dB, testam-se e comparam-se os algoritmos de identificação com outros já existentes. Todos os algoritmos geram estimativas com um grau de exatidão aceitável, sendo as estimativas do *MOLIZOE* exatamente iguais às de um estimador do erro de previsão.

Abstract

In this dissertation, new system identification algorithms are proposed for linear and time invariant systems with multiple input and a single output. In these algorithms, the system is described by a state space model in the canonical observable form and represented by a Luenberger Observer with a known state matrix. Thence, the identification problem is reduced to the estimation of the system input matrix and of the observer gain which can be performed by a simple Least Squares Estimator. The quality of the estimator depends on the observer state matrix, i.e., on the observer characteristic polynomial. Three algorithms were proposed: The Iterative Barycenter, The Modified Iterative Barycenter and The *MOLIZOE*. The Iterative Barycenter estimates the parameters of the characteristic polynomial, through a weighted mean. A polynomial named as "curiosity" is generated in every iteration, in order to calculate this mean. Furthermore, a non negative weight that increases inversely with the cost function is assigned to each curiosity. The estimated parameters are the weighted mean of all "curiosities" generated so far. This mean is equivalent to a barycenter or a center of mass and it is calculated recursively in every iteration. On the other hand, the Modified Iterative Barycenter is based on a bayesian approach of the Barycenter. This algorithm faces each iteration as a Bayes Estimator with more and more information about the system. The *MOLIZOE*, or the ellipses algorithm, calculates the barycenter of a set of "curiosities" in order to minimize the cost function. Those "curiosities" are properly generated in ellipsoids.

These algorithms are tested and compared here with others in a case study. This case study consists in three Monte Carlo simulations of 100 experiments to identify a second order system with signal-to-noise ratios of 20dB, 10dB and 0dB. All algorithms generate good estimates with an acceptable accuracy. The *MOLIZOE* estimates are equal to the ones given by the Prediction Error Estimator.

Agradecimentos

Deixo aqui o meu sincero agradecimento a todos os que contribuíram para a realização desta dissertação:

- Ao meu orientador, o Professor Doutor Paulo Lopes dos Santos, pela disponibilidade, colaboração e sabedoria;
- Às Professoras Doutoras Maria do Rosário Pinho e Maria Margarida Ferreira por todo o apoio disponibilizado;
- Aos meus pais e aos meus amigos que sempre me acompanharam no meu percurso.

Um agradecimento muito especial a todos aqueles que dedicaram e transmitiram o incentivo, a motivação e a clareza necessárias para a realização deste trabalho.

*“Há sem dúvida quem ame o infinito,
Há sem dúvida quem deseje o impossível,
Há sem dúvida quem não queira nada —
Três tipos de idealistas, e eu nenhum deles:
Porque eu amo infinitamente o finito,
Porque eu desejo impossivelmente o possível,
Porque quero tudo, ou um pouco mais, se puder ser,
Ou até se não puder ser...”*

Álvaro de Campos

Conteúdo

1	Introdução	1
2	Revisão Bibliográfica	5
2.1	Identificação de Sistemas	5
2.2	Identificação no Subespaço de Estados	5
2.2.1	Teoria da Realização Determinística	5
2.2.2	Teoria da Realização Estocástica	7
2.2.3	Métodos de Identificação de Subespaço de Estados	7
2.3	Métodos de Otimização do Erro de Previsão (PEM)	9
2.3.1	Modelos de Função de Transferência	9
2.3.2	Métodos do Erro Previsão (PEM) para Modelos no Espaço de Estados	11
2.4	Identificação de Sistemas em Malha Fechada (<i>Closed Loop</i>)	12
2.5	Métodos MOLI-ZOFT	13
3	Matchable Observable Linear Identification (MOLI)	15
3.1	Introdução	15
3.2	Parametrização MOLI para Sistemas MISO	15
3.2.1	Formulação do Previsor	18
3.2.2	Estimação dos Parâmetros do Modelo	18
3.3	Abordagens MOLI para Identificação de Sistemas MISO	19
3.3.1	ARX MOLI	20
3.3.2	The Output MOLI	20
3.3.3	PEM MOLI	21
3.3.4	Barycenter MOLI-ZOFT	23
4	Algoritmo do Baricentro Iterativo	25
4.1	Introdução	25
4.2	Método do Baricentro Discreto	25
4.2.1	Fator de Esquecimento	27
4.2.2	Inicialização	28
5	Interpretação Bayesiana do Baricentro	29
5.1	Introdução	29
5.2	Estimador de Bayes de Erro Quadrático Médio Mínimo	29
5.2.1	O Baricentro como MMSE	31
5.3	Algoritmo do Baricentro Iterativo Modificado	33

6	Algoritmo MOLI-ZOE	37
6.1	Introdução	37
6.2	Geração das "Curiosidades"	39
6.2.1	Inicialização	41
6.2.2	Pormenores de Implementação	42
7	Estudo de Caso	47
7.1	Formulação	47
7.2	Considerações	48
7.3	Análise de Resultados	48
8	Conclusões	53
8.1	Contribuições	53
8.2	Trabalho Futuro	54
	Referências	57

Lista de Figuras

2.1	Sistema em Malha Fechada.	12
3.1	Formulação do Previsor.	16
3.2	Sistema de filtragem da saída e da entrada.	17
3.3	Sistema de filtragem do erro.	17
3.4	Modelo ARX.	20
3.5	Modelo <i>Output Error</i>	20
6.1	Função de densidade de probabilidade de uma distribuição normal a duas dimensões.	37
6.2	Superfícies de nível de uma função de densidade de probabilidade gaussiana a duas dimensões.	37
6.3	Elipsoide de ordem 2.	39
6.4	Inicialização das "curiosidades".	41
6.5	Elipse de ordem 2.	42
6.6	Função $p(Y \alpha)$, para um conjunto de dados Y com $n_x = 1$	43
6.7	Funções $p(Y \alpha)$, para $n_x = 1$ com diferentes valores de σ^2	43
6.8	Elipsoide de ordem 2 a ser deslocada.	45
6.9	MOLIZOE: iteração 0.	45
6.10	MOLIZOE: iteração 1.	45
6.11	MOLIZOE: iteração 4.	46
6.12	MOLIZOE: iteração 6.	46
6.13	MOLIZOE: iteração 8.	46
6.14	MOLIZOE: iteração 25.	46

Lista de Tabelas

7.1	Resultados: SNR = 20 dB.	50
7.2	Resultados: SNR = 10 dB.	51
7.3	Resultados: SNR = 0 dB.	52

Abreviaturas e Símbolos

ARMAX	Auto-Regressive Medium Average with Exogenous Input
ARX	Auto-Regressive with Exogenous Input
LPV	Linear Parameter-Varying
LTI	Linear and Time Invariant
MC	Monte Carlo
MIMO	Multi-Input Multi-Output
MISO	Multi-Input Single-Output
MMSE	Minimum Median Square Estimator
MOLI	Matchable Observable Linear Identification
MSE	Mean Squared Error
PEM	Prediction Error Method
SISO	Single-Input Single-Output
SNR	Signal to Noise Ratio
UKF	Unscented Kalman Filter
ZOFT	Zero-Order Oracle Filter Tuning

Capítulo 1

Introdução

Conhecer, relacionar e formular as diferentes variáveis que determinam a dinâmica de um sistema pode tornar-se um processo complexo, demorado e por vezes de difícil conceção. Acontece que frequentemente não se consegue conhecer a totalidade das leis e dos princípios matemáticos que regem a dinâmica de um sistema, o que nos coloca a necessidade de procurar novos métodos para se conseguir obter um modelo aproximado do comportamento desse sistema. É, então, que surge a Identificação de Sistemas, que consiste em estimar o modelo matemático de um sistema a partir dos dados de entrada e de saída, recolhidos em experiências previamente planeadas. Esta área do saber, que tem por base a teoria dos processos estatísticos e econométricos, desenvolveu-se, maioritariamente, em meados do século XX, após um crescimento acentuado das técnicas de projeto de sistemas de controlo baseadas em modelos. Essas técnicas de controlo assentam no desenvolvimento de modelos matemáticos específicos dos processos que estão envolvidos no funcionamento de um sistema, o que explica o grande desenvolvimento das metodologias de Identificação de Sistemas no século passado.

Um dos aspetos mais relevantes no processo de identificação de sistemas prende-se com a parametrização do modelo estimado, pelo que é, muitas vezes, essencial a utilização de uma parametrização que auxilie a estimação dos parâmetros do modelo. Romano *et al.* [1] constataram que uma parametrização previamente proposta por Morse e Pait [2] é especialmente adequada para a estimação de sistemas multivariáveis, isto é, sistemas com múltiplas entradas e múltiplas saídas (MIMO) [1]. Cada um desses sistemas é descrito por uma representação no espaço de estados, a qual é muito próxima da forma canónica observável. Por sua vez, cada uma das saídas é prevista por um Observador de Luenberger com matriz de estado conhecida, designando-se esta parametrização por MOLI, o acrónimo de *Matchable Observable Linear Identification*¹. Por seu turno, esta parametrização permite utilizar um Estimador de Mínimos Quadrados para o cálculo dos parâmetros desconhecidos do observador. Tal só é possível porque o erro de previsão do modelo é

¹*Matchable* querendo sublinhar que, quando numa representação no espaço de estados se fixa a matriz de estado do observador e se fixa também um fator da matriz de saída, é gerada uma qualquer matriz de transferência. *Observable* porque se trata de uma representação observável muito próxima da forma canónica observável. *Linear Identification* porque a parametrização é adequada para a identificação de sistemas lineares.

uma função linear desses parâmetros. Por outro lado, dado que a qualidade das estimativas produzidas pelo Estimador de Mínimos Quadrados depende do polinómio característico do observador, este também deve ser objeto de otimização. Além do mais, as estimativas de mínimos quadrados são calculadas numa só iteração, pelo que podem ser vistas como uma função dos parâmetros do polinómio do observador, o que conduz a uma redução significativa do espaço das variáveis de otimização.

Em [1] foi utilizado o método do Baricentro para otimizar o polinómio característico do observador. Neste método, são gerados vários polinómios característicos que se designam por "curiosidades". A cada "curiosidade" é associado um peso que cresce com o inverso do valor da função custo a minimizar, que, por sua vez, é a soma dos quadrados dos resíduos do Estimador de Mínimos Quadrados. O baricentro dessas "curiosidades" é, então, considerado o valor ótimo do vetor de parâmetros do polinómio característico. O método do Baricentro surge, assim, como um método de otimização computacionalmente leve e livre de cálculos complexos de derivadas. Romano e Paít designaram todo o algoritmo de identificação aqui descrito por MOLI-ZOFT. MOLI decorre da parametrização e ZOFT de *Zero-Order Oracle Filter Tuning*, já que a otimização do polinómio característico é vista como a sintonização de um filtro. Esta sintonização é realizada pelo método do Baricentro, isto é, um método sem derivadas que, no contexto da otimização, é visto como um algoritmo de ordem zero. A partir da parametrização MOLI, conceberam-se novos algoritmos para a identificação de sistemas com múltiplas entradas e uma só saída (MISO), surgindo, assim, duas vertentes: algoritmos lineares de parâmetros variantes (LPV), descritos em [3] e em [4], e algoritmos lineares invariantes no tempo (LTI), apresentados em [5]. Em [5] compararam-se diferentes abordagens, nomeadamente, a de ARX, a de Baricentro, a de otimização do Erro de Saída (*Output Error*) e a de otimização do erro de previsão (PEM). Através do método dos Mínimos Quadrados Separáveis, estas abordagens foram aplicadas no sentido da identificação de modelos dinâmicos de atividade física a partir dos dados recolhidos na intervenção *Just Walk* [6].

Neste enquadramento, propõem-se, nesta dissertação, novos algoritmos de identificação em tempo discreto para sistemas MISO LTI, que têm por base não só a parametrização MOLI, mas também métodos de otimização sem derivadas. Estes novos algoritmos são baseados no conceito de baricentro e exploram o facto desta parametrização reduzir o problema da identificação a um outro problema: o da estimação dos coeficientes da equação característica da matriz de um observador de estado, o que torna a dimensão do espaço da variável de otimização significativamente menor. Os algoritmos de identificação que serão então formulados têm desempenhos semelhantes aos dos melhores algoritmos de identificação até agora conhecidos. A aplicação iterativa do método do Baricentro surge como uma das ideias inovadoras neste trabalho, o que possibilitou a formulação de três algoritmos de identificação. Por outro lado, uma análise mais atenta do comportamento deste método permitiu formular outros algoritmos, baseados numa abordagem bayesiana do Baricentro. Todas estes algoritmos serão comparados e aplicados a um estudo de caso específico.

Esta dissertação está estruturada em oito capítulos, incluindo a introdução e a conclusão. No capítulo dois, apresenta-se uma perspetiva e uma breve descrição das principais abordagens relevantes para o problema de identificação explorado e tratado neste trabalho. No capítulo três é

descrita a parametrização MOLI para sistemas com várias entradas e uma saída (MISO) e ainda se descrevem alguns algoritmos de identificação recentemente propostos que utilizam esta parametrização. O capítulo quatro apresenta o algoritmo do Baricentro Iterativo. Nos capítulos cinco e seis, definem-se novas abordagens de identificação, baseadas nos resultados obtidos após a simulação e o teste do primeiro algoritmo implementado. Por um lado, no capítulo cinco apresenta-se uma das contribuições mais importantes deste trabalho, e a qual consiste na demonstração de que em determinadas condições o método do Baricentro se aproxima de um Estimador de Bayes, por outro, no capítulo seis, propõe-se um novo método de otimização do erro de previsão. Por seu turno, o capítulo sete apresenta o estudo de caso necessário para comparação e validação dos algoritmos de identificação que este trabalho propõe.

Capítulo 2

Revisão Bibliográfica

2.1 Identificação de Sistemas

Os primeiros estudos e abordagens teóricas na área da Identificação e Estimação remetem para os domínios da estatística e da econometria, e para a análise de séries temporais. Gauss [7] e Fisher [8] desenvolveram estudos estatísticos e matemáticos e acabaram, assim, por definir alguns dos primeiros conceitos de identificação. Tais conceitos vieram a ser utilizados, mais tarde, para formular a Teoria da Realização Estocástica. No entanto, foi a partir dos anos 60, e em consequência do forte crescimento das técnicas de projeto de sistemas de controlo baseadas em modelos, que esta área ficou reconhecida como uma nova vertente da área do controlo, que se designa por Identificação de Sistemas. Na verdade, o desenvolvimento da Teoria da Identificação surge exatamente da ambição em explorar e expandir a aplicação dessas novas técnicas de controlo no plano das ciências tecnológicas. De entre os vários estudos desenvolvidos ao longo da década de 60, há dois que se destacam pelas diferentes técnicas de identificação que apresentam e que deram origem às duas grandes metodologias de identificação ainda hoje utilizadas no âmbito da estimação de sistemas. Por um lado, o de Ho e Kalman [9], que aponta para a identificação em subespaços de estados e que tem por base projeções no espaço. Por outro, o de Åström e Bohlin [10], cuja investigação assenta na estimação de parâmetros de modelos através de métodos de otimização do erro de previsão.

2.2 Identificação no Subespaço de Estados

2.2.1 Teoria da Realização Determinística

Ho e Kalman [9] deram resposta para o que ficou conhecido como o problema da realização em espaço de estados. Tendo em conta o modelo de entrada-saída

$$y(k) = \sum_{s=0}^{\infty} h(s)u(k-s), \quad (2.1)$$

de um sistema com n_u entradas ($u(k) \in \mathbb{R}^{n_u}$) e n_y saídas ($y(k) \in \mathbb{R}^{n_y}$), em que $h(k) \in \mathbb{R}^{n_y \times n_u}$ é a resposta impulsional do modelo. Sabe-se que

$$H(z) = \sum_{k=1}^{\infty} h(k)z^{-k} \quad (2.2)$$

é a função de transferência do sistema. O problema surge do facto de $H(z)$ ser descrita por um somatório infinito. A solução deste problema consiste em determinar uma realização (determinística) mínima no espaço de estados

$$x(k+1) = Ax(k) + B(k) \quad (2.3)$$

$$y(k) = Cx(k) + Du(k), \quad (2.4)$$

tal que

$$H(z) = C(zI - A)^{-1}B + D. \quad (2.5)$$

Define-se, então, a matriz \mathcal{H} de blocos de Hankel, igual ao produto entre as matrizes de observabilidade e de controlabilidade

$$\mathcal{H} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix} \quad (2.6)$$

$$= \begin{bmatrix} h(1) & h(2) & h(3) & \dots \\ h(2) & h(3) & h(4) & \dots \\ h(3) & h(4) & h(5) & \dots \\ \vdots & \vdots & \vdots & \dots \end{bmatrix}, \quad (2.7)$$

em que $h(k) = CA^{k-1}B$, $k = 1, 2, \dots$, são os parâmetros de Markov, isto é, os valores da resposta impulsional. Se a característica desta matriz for igual $n_x < \infty$, então existem $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ e $C \in \mathbb{R}^{n_y \times n_x}$, tal que $H(z) = C(zI - A)^{-1}B + D$ e

$$x(k+1) = Ax(k) + Bu(k) \quad (2.8)$$

$$y(k) = Cx(k) + Du(k) \quad (2.9)$$

com $x(k) \in \mathbb{R}^{n_x}$ e $D = h(0)$.

A aplicação exata e numérica do algoritmo de identificação aqui formulado só foi, porém, conseguida por Jong em 1975 [11] e em 1978 [12].

2.2.2 Teoria da Realização Estocástica

Mais tarde, a determinação da relação entre a fatorização de uma matriz de Hankel

$$\mathcal{H}_{+-} = \mathbf{E} = \left\{ \begin{bmatrix} y(i) \\ y(i+1) \\ y(i+2) \\ \vdots \end{bmatrix} \begin{bmatrix} y(i-1) & y(i-2) & y(i-3) & \dots \end{bmatrix} \right\} = \quad (2.10)$$

$$\begin{bmatrix} \mathbf{E}\{y(i)y(i-1)\} & \mathbf{E}\{y(i)y(i-2)\} & \mathbf{E}\{y(i)y(i-3)\} & \dots \\ \mathbf{E}\{y(i+1)y(i-1)\} & \mathbf{E}\{y(i+1)y(i-2)\} & \mathbf{E}\{y(i+1)y(i-3)\} & \dots \\ \mathbf{E}\{y(i+2)y(i-1)\} & \mathbf{E}\{y(i+2)y(i-2)\} & \mathbf{E}\{y(i+2)y(i-3)\} & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} =$$

$$\begin{bmatrix} \lambda_y(1) & \lambda_y(2) & \lambda_y(3) & \dots \\ \lambda_y(2) & \lambda_y(3) & \lambda_y(4) & \dots \\ \lambda_y(3) & \lambda_y(4) & \lambda_y(5) & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

formada pelos elementos da sequência de autocovariância

$$\lambda_y(\tau) = \mathbf{E}\{y(k)y(k-\tau)\} \quad (2.11)$$

de um processo estocástico estacionário $\{y(k)\}$, e a sua representação na forma

$$x_p(k+1) = Ax_p(k) + Ke(k) \quad (2.12)$$

$$y_p(k) = Cx_p(k) + e(k), \quad (2.13)$$

em que $e(k)$ é ruído branco, com covariância $\lambda_e(0) = \mathbf{E}\{e(k)e^T(k)\} = R$, permitiu formular a Teoria da Realização Estocástica para realizações mínimas [13, 14, 15]. O problema resume-se a encontrar as matrizes A , C , K e R , que configurem a menor dimensão possível para A , de forma a que $\lambda_{y_p}(\tau) = \lambda_y(\tau)$.

2.2.3 Métodos de Identificação de Subespaço de Estados

A ideia que está na base dos métodos de subespaço de estados é estimar sequências de vetores de estados do sistema, $\hat{X}_i = [\hat{x}(i) \ \dots \ \hat{x}(i+j-1)] \in \mathbb{R}^{n_x \times j}$ e $\hat{X}_{i+1} = [\hat{x}(i+1) \ \dots \ \hat{x}(i+j)] \in \mathbb{R}^{n_x \times j}$, a partir dos dados de entrada-saída. Assim, tem-se

$$\begin{bmatrix} \hat{X}_{i+1} \\ Y_i \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}_i \\ U_i \end{bmatrix} + \begin{bmatrix} \eta_i \\ v_i \end{bmatrix}, \quad (2.14)$$

onde $U_i \in \mathbb{R}^{n_u \times j}$ e $Y_i \in \mathbb{R}^{n_y \times j}$ são, respetivamente, sequências de entrada e de saída, e $\begin{bmatrix} \eta_i \\ v_i \end{bmatrix}$ são os resíduos. Dado que todas as variáveis são conhecidas, a equação 2.17 é um modelo de regressão dos parâmetros do sistema $\theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathbb{R}^{(n_x+n_y) \times (n_x+n_u)}$. Estes podem ser determinados pelo Estimador de Mínimos Quadrados

$$\hat{\theta} = \begin{bmatrix} \hat{X}_{i+1} \\ Y_i \end{bmatrix} \begin{bmatrix} \hat{X}_i^T & U_i^T \end{bmatrix} \left(\begin{bmatrix} \hat{X}_i \\ U_i \end{bmatrix} \begin{bmatrix} \hat{X}_i^T & U_i^T \end{bmatrix} \right)^{-1}. \quad (2.15)$$

Os métodos que utilizam esta abordagem são designados por Métodos Diretos N4SID [16], [17]. Percebe-se que a estimativa de θ é única se a característica da matriz $\begin{bmatrix} \hat{X}_i \\ U_i \end{bmatrix}$ for igual a $n_x + n_u$. Já a matriz de covariância dos resíduos define-se como

$$\lambda_r = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \frac{1}{j} \begin{bmatrix} \eta_i \\ v_i \end{bmatrix} \begin{bmatrix} \eta_i^T & v_i^T \end{bmatrix}. \quad (2.16)$$

Resolvendo-se uma equação algébrica de Riccati específica, consegue chegar-se à forma do modelo de inovação, tal como já foi caracterizada em 2.12 e 2.13:

$$\begin{bmatrix} \hat{x}(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} K \\ I_p \end{bmatrix} \hat{e}(t), \quad (2.17)$$

em que K é o ganho do filtro de Kalman estacionário, \hat{x} é a estimativa do vetor de estado e $\hat{e}(t)$ é a estimativa do processo de inovação.

Mais concretamente, nesta abordagem, define-se $k = i$ como sendo o instante de tempo presente. O estado é estimado através de projeções ortogonais ou oblíquas de subespaços gerados pelas linhas de matrizes de Hankel. O subespaço gerado pelas saídas futuras é projetado no subespaço que é formado não só pelas saídas e entradas passadas, mas também pelas próprias saídas futuras. A partir destas projeções obtêm-se os produtos $\mathcal{O}_i = \Gamma_i \hat{X}_i$ e $\mathcal{O}_{i-1} = \Gamma_{i-1} \hat{X}_{i+1}$, em que Γ_i é a matriz de observabilidade alargada

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} \quad (2.18)$$

e \hat{X}_i é a sequência de estados

$$\hat{X}_i = \begin{bmatrix} \hat{x}(i) & \hat{x}(i+1) & \cdots & \hat{x}(i+j-1) \end{bmatrix}. \quad (2.19)$$

A ordem do sistema n_x é extraída juntamente com Γ_i e \hat{X}_i através da Decomposição em Valores Singulares de \mathcal{O}_i . Γ_{i-1} é determinado a partir de Γ_i , retirando-lhe as suas n_y últimas linhas. \hat{X}_{i+1} é obtido através da resolução da equação $\mathcal{O}_{i-1} = \Gamma_{i-1}\hat{X}_{i+1}$, no sentido dos mínimos quadrados [18, 19, 20, 17]. Finalmente, as matrizes A , B , C e D são estimadas resolvendo a equação (2.15).

Na sua implementação, os métodos de subespaços de estados lidam com algoritmos numéricos, tais como a Decomposição QR e a Decomposição em Valores Singulares, não necessitando, portanto, de técnicas de otimização não lineares [18, 19, 20, 17]. Estes métodos podem ser aplicados tanto a sistemas MIMO como a sistemas SISO e não têm em conta qualquer forma canónica, pois não é necessário conhecer a forma do modelo do sistema. São métodos não iterativos que não se preocupam com as parametrizações do modelo, visto que são os dados a analisar que as irão definir. Nos algoritmos originalmente propostos, os modelos estimados têm mais parâmetros do que os necessários para descrever o comportamento do sistema: modelos sobreparametrizados. Este tipo de modelos tem vantagens numéricas. No entanto, os métodos para a sua estimação só funcionam corretamente em condições de excitabilidade exigentes do sinal de entrada do sistema. Recentemente, foi proposta uma abordagem no subespaço de estados que estima modelos numa forma canónica com um número mínimo de parâmetros [21].

Sendo baseados em Estimadores de Mínimos Quadrados e em projeções ortogonais e oblíquas, os métodos de subespaço de estados são simples e numericamente eficientes. Contudo, produzem estimativas menos precisas que os métodos PEM [22]. Por esta razão, os métodos PEM são frequentemente utilizados para inicializar as estimativas dos métodos de subespaço de estado, os quais minimizam funções não convexas com vários mínimos locais e pontos em sela [23, 24].

2.3 Métodos de Otimização do Erro de Previsão (PEM)

2.3.1 Modelos de Função de Transferência

Åström e Bohlin [10] introduziram o método de máxima verosimilhança para estimação dos parâmetros de um modelo de entrada-saída univariável (com uma entrada e uma saída) descrito como

$$A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})e(k), \quad (2.20)$$

onde q^{-1} é o operador de atraso ($q^{-1}y(t) = y(t-1)$), $u(k)$ é a entrada, $y(k)$ é a saída, $e(k)$ é ruído branco de média nula e

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad (2.21)$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad (2.22)$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{n_c}q^{-n_c}. \quad (2.23)$$

Este é o modelo ARMAX (*Auto-Regressive Moving Average with Exogenous Input*) e a estimação dos seus parâmetros, aglomerados no vetor

$$\theta = \left[a_1 \quad \cdots \quad a_{n_a} \quad b_0 \quad b_1 \quad \cdots \quad b_{n_b} \quad c_1 \quad \cdots \quad c_{n_c} \right]^T, \quad (2.24)$$

foi um dos passos iniciais no sentido da identificação paramétrica, o que permitiu estabelecer os primeiros métodos de erro de previsão. De facto, partindo deste modelo ARMAX e supondo que $e(k)$ é ruído branco gaussiano, mostrou-se que maximizar a função de verosimilhança das observações é equivalente a minimizar o critério quadrático

$$V(\theta) = \frac{1}{2} \sum_{t=1}^N \hat{e}(k, \theta)^2, \quad (2.25)$$

onde $\hat{e}(k)$ é o erro de previsão, dado por

$$\hat{e}(k, \theta) = y(k) - \hat{y}(k, \theta), \quad (2.26)$$

em que

$$\hat{y}(k, \theta) = \frac{(C(q^{-1}) - A(q^{-1}))y(k) + B(q^{-1})u(k)}{C(q^{-1})}. \quad (2.27)$$

é a previsão de $y(k)$ no instante $k - 1$, sendo, então,

$$\hat{\theta} = \arg \min V(\theta) \quad (2.28)$$

o Estimador de Máxima Verosimilhança. Mesmo que $e(k)$ não seja gaussiano, poder-se-á também estimar θ através da minimização de $V(\theta)$. Os métodos que minimizam este critério quadrático do erro de previsão são conhecidos por Métodos do Erro de Previsão (*Prediction Error Methods - PEM*). Por outro lado, seja qual for a distribuição de $e(k)$, se este sinal for ruído branco de média nula, $\hat{\theta} \rightarrow \theta^*$ quando o número de observações tende para infinito. O valor de θ^* é

$$\theta^* = \arg \min \mathbf{E} \{ V(\theta) \}, \quad (2.29)$$

onde $\mathbf{E} \{ \cdot \}$ é o operador valor esperado [25]. Se o sinal de entrada tiver excitação suficiente e se a estrutura do modelo estimado for igual à do verdadeiro, então θ^* é o valor verdadeiro de θ . A distribuição de $\hat{\theta}$ tende para uma distribuição normal com valor esperado θ^* e covariância [25]

$$P_{\theta} = \sigma^2 (\psi(k) \psi(k)^T), \quad (2.30)$$

em que

$$\sigma^2 = \mathbf{E} \{ e(k)^2 \} \quad (2.31)$$

$$\psi(k) = \left. \frac{d\hat{y}(k)}{d\theta} \right|_{\theta=\theta^*}. \quad (2.32)$$

No caso de $e(k)$ ter uma distribuição normal, em que $\hat{\theta}$ é o Estimador de Máxima Verosimilhança, a covariância P_θ converge para limite de Cramer-Rao [25], isto é, o limite mínimo teórico de qualquer estimador [26]. Por outras palavras, quando as perturbações do sistema são gaussianas, os estimadores de erro de previsão são assintoticamente ótimos.

Após inúmeros estudos nesta área, houve a necessidade de unificar toda a teoria dos métodos PEM. As maiores contribuições nesta unificação foram as de Lennart Ljung [27], que distinguiu dois conceitos que se sobrepunham até à altura: a escolha de uma parametrização para a estrutura do modelo e a escolha de um critério de identificação. Foi Ljung [27] o responsável pelo desenvolvimento da *toolbox* de identificação do MATLAB e pela formulação de um modelo genérico de entrada-saída. A sua pesquisa está totalmente reunida num livro [28], que ainda hoje é considerado uma das maiores referências para a comunidade científica de Identificação de Sistemas.

2.3.2 Métodos do Erro Previsão (PEM) para Modelos no Espaço de Estados

Tendo em conta o modelo de inovação de um sistema LTI, e agora considerando que o sistema tem múltiplas entradas e saídas (MIMO) [20]

$$x(k+1) = Ax(k) + Bu(k) + Ke(k) \quad (2.33)$$

$$y(t) = Cx(k) + Du(k) + e(k), \quad (2.34)$$

em que $e(k)$, também designado de parâmetro de inovação, tem média nula e a sua matriz de covariância é $\lambda_e > 0$. As matrizes A, B, C e K têm dimensões adequadas e os parâmetros desconhecidos do modelo no espaço de estados estão presentes nessas matrizes, bem como na matriz λ_e .

A aplicação de um método PEM a este modelo, consiste na minimização do critério

$$V(\theta) = \frac{1}{N} \sum_{k=0}^{N-1} \|\hat{e}(k, \theta)\|^2, \quad (2.35)$$

em que o erro de previsão $\hat{e}(k, \theta)$, é dado por

$$\hat{x}(k+1, \theta) = [A(\theta) - K(\theta)C]\hat{x}(k, \theta) + B(\theta)u(t) + K(\theta)y(k) \quad (2.36)$$

$$\hat{e}(k, \theta) = -C\hat{x}(k, \theta) - D(\theta)u(k) + y(k). \quad (2.37)$$

$V(\theta)$ é normalmente minimizado por um método que procura anular a sua derivada: método de gradiente. No entanto, a aplicação de métodos de otimização requer sempre que o modelo do sistema seja descrito por uma parametrização específica. Essa parametrização é frequentemente difícil de estabelecer, já que não existe apenas uma possibilidade de parametrização para um sistema multivariável. Na verdade, para uma dada ordem do modelo, existe uma infinidade de representações para um só sistema, sendo necessário escolher uma parametrização adequada para evitar problemas numéricos [29] no processo de minimização de um critério de desempenho. Normalmente, são escolhidas representações canónicas, embora, por vezes, se torne difícil saber à priori

qual a representação canônica mais adequada para estimar um modelo a partir de um determinado conjunto de dados, sendo mesmo necessário ajustar a parametrização ao longo da implementação do método de otimização [30, 18, 31].

Nesse sentido, Bergboer *et al.* [32] e Will e Ninness [33] propuseram métodos que conseguem ultrapassar os problemas da parametrização. No entanto, esses métodos, além de serem muito pesados em termos computacionais, encalham facilmente em mínimos locais ou em pontos em sela.

2.4 Identificação de Sistemas em Malha Fechada (*Closed Loop*)

Os primeiros estudos realizados nesta área, que surgiram no contexto da teoria da realização estocástica [34] e do processo de identificação a partir de dados espectrais e de covariância [35], trouxeram a visão de que os dados recolhidos de um sistema em malha fechada, com uma configuração idêntica ao da Figura 2.1, introduziam erros e enviesamentos na estimativa do modelo da planta do sistema, que na figura que se segue está representada pelo bloco $P(z)$. Esse problema prevaleceu ao longo de vários anos, uma vez que não se conseguia perceber ao certo a influência da correlação entre o sinal de entrada da planta e o ruído na estimativa do modelo. Uma das primeiras soluções foi formulada por Söderström, Ljung e Gustavsson [36], que estudaram a utilização de diferentes métodos de identificação em sistemas realimentados. Por um lado, provaram

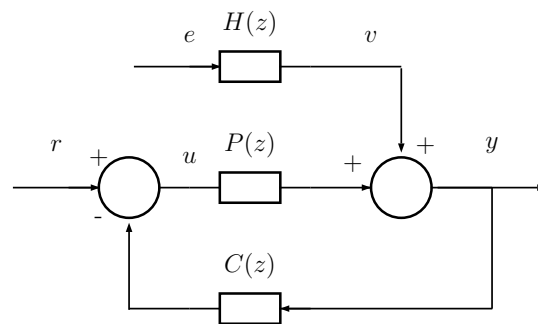


Figura 2.1: Sistema em Malha Fechada.

que a aplicação direta de métodos de erro de previsão a dados de entrada-saída permite, em certos casos práticos, chegar ao modelo da planta, apesar da presença de realimentação no sistema. Por outro lado, também concluíram que identificando, em primeiro lugar, a função de transferência em malha fechada do sistema, poder-se-ia posteriormente chegar ao modelo da planta, tendo em conta a ação de um controlador $C(z)$. Formularam ainda uma última abordagem de identificação, designada por *Joint Input-Output*, que consiste em identificar, em primeiro lugar, as funções de transferência $T_{yr}(z) = \frac{Y(z)}{R(z)}$ e $T_{ur}(z) = \frac{U(z)}{R(z)}$, que é um problema de identificação em malha aberta,

e a seguir determinar $P(z)$ [37, 38, 39, 40] através de

$$P(z) = \frac{Y(z)}{U(z)} = \frac{Y(z)/R(z)}{U(z)/R(z)} = \frac{T_{yr}(z)}{T_{ur}(z)}. \quad (2.38)$$

A aplicabilidade destas abordagens torna-se essencial quando os sistemas a estimar são instáveis ou então quando a realimentação é uma característica intrínseca do sistema [14, 41, 42]. Adicionalmente, quando fatores de qualidade e de segurança de sistemas de produção entram em consideração, entende-se que a aplicação de métodos de identificação em malha fechada é quase obrigatória.

2.5 Métodos MOLI-ZOFT

Ao longo das últimas décadas, grande parte da investigação realizada em torno da área de Identificação de Sistemas tem tido o seu foco na conceção de novos algoritmos para identificação de modelos no espaço de estados. Para esse efeito, foram propostos não só vários algoritmos de subespaço de estados, mas também outros que minimizam o erro de previsão (PEM). Como foi referido anteriormente, apesar de serem muito atrativos, os métodos de subespaço de estados exigem sinais com nível de excitação muito elevado. Por outro lado, os métodos PEM são numericamente pesados e encaham frequentemente em mínimos locais e em pontos em sela.

A abordagem que aqui se apresenta foi proposta originalmente para a identificação de sistemas multivariáveis [1], tendo também sido adaptada para a identificação de sistemas com várias entradas e uma saída (sistemas MISO), lineares e invariantes no tempo (LTI) [5] e lineares de parâmetros variantes (LPV) [3]. Esta abordagem, designada por MOLI, reduz o problema da identificação do modelo no espaço de estados ao da estimação dos parâmetros da equação característica de um observador ótimo, uma vez que conhecida esta equação, os outros parâmetros do sistema podem ser estimados por um simples Estimador de Mínimos Quadrados. Para estimar os parâmetros da equação característica do observador, Romano *et.al* [1] propuseram um método de otimização sem derivadas, tendo designado todo o algoritmo por MOLI-ZOFT. MOLI da parametrização e ZOFT de *Zero-Order Oracle Filter Tuning*, uma vez que o método de otimização do polinómio do observador é visto como a sintonização de um filtro.

Capítulo 3

Matchable Observable Linear Identification (MOLI)

3.1 Introdução

É no contexto da necessidade de se explorar novas abordagens de identificação que ultrapassassem as dificuldades dos métodos já conhecidos, que surgiu a parametrização MOLI.

MOLI combina uma estrutura do modelo baseada numa representação muito próxima da forma canónica observável com a formulação de um observador de estado que prevê a saída do sistema. Porém, no contexto muito particular dos sistemas MISO a representação em espaço de estados não é propriamente próxima da forma canónica observável, mas é sim exatamente a forma canónica observável. A utilização da forma quase canónica traz vantagens ao processo de identificação, uma vez que reduz o número de parâmetros a estimar. Contudo, por outro lado, a escolha de uma representação próxima da realização canónica observável permite fixar o valor de uma parte da matriz C - ou até mesmo da totalidade da matriz C em sistemas MISO - na formulação em espaço de estados, o que também facilita a estimação dos parâmetros do modelo. Neste capítulo, é apresentada a parametrização MOLI para sistemas MISO [5], já anteriormente utilizada na identificação de modelos de atividade física a partir dos dados recolhidos na intervenção *JustWalk* [6].

3.2 Parametrização MOLI para Sistemas MISO

Considera-se a seguinte realização no espaço de estados de um sistema MISO LTI,

$$x(k+1) = Ax(k) + Bu(k) + Ke(k) \quad (3.1)$$

$$y(k) = Cx(k) + Du(k) + e(k), \quad (3.2)$$

em que $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$, $y(k) \in \mathbb{R}$, $e(k) \in \mathbb{R}$ é ruído branco com média nula, $A \in \mathbb{R}^{n_x \times n_x}$,

$$B = \begin{bmatrix} B_1 & B_2 & \cdots & B_{n_u} \end{bmatrix} \in \mathbb{R}^{n_x \times n_u}, \quad (3.3)$$

$C \in \mathbb{R}^{1 \times n_x}$, $D \in \mathbb{R}^{1 \times n_u}$ e $K \in \mathbb{R}^{n_x}$ é o ganho do previsor de Kalman. Conhecendo à partida um valor de C e um valor estável de $A_0 \in \mathbb{R}^{n_x \times n_x}$, e tendo em conta que (A_0, C) é observável, a matriz A é dada por

$$A = A_0 + LC, \quad (3.4)$$

em que $L \in \mathbb{R}^{n_x}$. A é a matriz de estado de uma realização que corresponde a qualquer matriz de transferência racional

$$G(z) = \frac{b_0^T z^{n_x} + b_1^T z^{n_x-1} + \cdots + b_{n_x}^T}{z^{n_x} + a_1 z^{n_x-1} + \cdots + a_{n_x}} \in \mathbb{R}^{1 \times n_u} \quad (3.5)$$

de um sistema MISO, sendo $a_i \in \mathbb{R}$, $i = 1, \dots, n_x$, os coeficientes do polinómio característico de A , $b_0^T = D$ e $b_i^T \in \mathbb{R}^{1 \times n_u}$, $i = 1, \dots, n_x$ a $(n_x - i)$ ésima linha de B . Assim, a equação (3.1) pode ser reescrita como

$$x(k+1) = (A_0 + LC)x(k) + Bu(k) + Ke(k). \quad (3.6)$$

A partir de (3.2) retira-se que $Cx(k) = y(k) - Du(k) - e(k)$ e a equação (3.6) resulta em

$$x(k+1) = A_0x(k) + Ly(k) + (B - LD)u(k) + (K - L)e(k). \quad (3.7)$$

Como $y(k)$ é uma variável de medição e, por isso, conhecida, pode ser vista como uma entrada

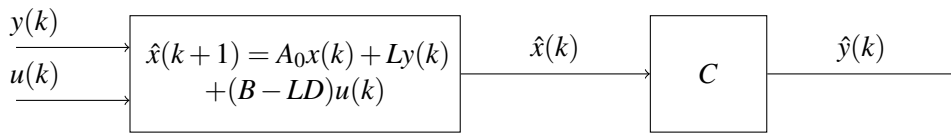


Figura 3.1: Formulação do Previsor.

adicional do sistema, tal como está representado na Figura 3.1. Deste modo, $x(k)$ é dado por

$$x(k) = A_0^{k-1}x(1) + (qI - A_0)^{-1}[Ly(k) + (B - LD)u(k) + (K - L)e(k)], \quad (3.8)$$

onde q é o operador de avanço, ou seja, $qx(k) = x(k+1)$. Definindo

$$B_a = B - LD \quad (3.9)$$

e substituindo $x(k)$ na equação de saída do sistema, $y(k)$ pode ser redefinido como

$$\begin{aligned} y(k) &= CA_0^{k-1}x(1) + C(qI - A_0)^{-1}[Ly(k) + B_a u(k) + (K - L)e(k)] + Du(k) + e(k) \\ &= CA_0^{k-1}x(1) + C(qI - A_0)^{-1}Ly(k) + C(qI - A_0)^{-1}(K - L)e(k) + Du(k) + e(k) \\ &\quad + \sum_{i=1}^{n_u} C(qI - A_0)^{-1}B_{a_i}u_i(k). \end{aligned} \quad (3.10)$$

Dado que $y(k)$ e $u_i(k)$, $i = 1, \dots, n_u$, são escalares, é possível escrever esta última equação como

$$y(k) = CA_0^{k-1}x(1) + y_f^T(k)L + e_f(k) + Du(k) + \sum_{i=1}^{n_u} u_{i_f}^T(k)B_{a_i}, \quad (3.11)$$

em que

$$y_f(k) = (qI - A_0^T)^{-1}C^T y(k) \in \mathbb{R}^{n_x} \quad (3.12)$$

$$u_{i_f}(k) = (qI - A_0^T)^{-1}C^T u_i(k) \in \mathbb{R}^{n_x}, \quad i = 1, \dots, n_x, \quad (3.13)$$

são as saídas do sistema representado na Figura 3.2 quando $\mathcal{U}(k) = y(k)$ ou $\mathcal{U}(k) = u_i(k)$ e as condições iniciais são nulas.

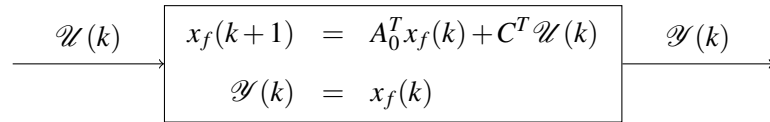


Figura 3.2: Sistema de filtragem da saída e da entrada.

O sinal

$$e_f(k) = [1 + C(qI - A_0)^{-1}(K - L)]e(k) \quad (3.14)$$

é a saída do sistema estocástico representado na Figura 3.3.

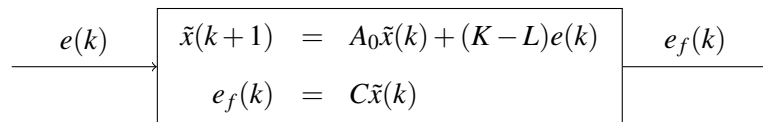


Figura 3.3: Sistema de filtragem do erro.

3.2.1 Formulação do Previsor

Visto que $e_f(k)$ é desconhecido, mas tem um valor de média nulo, a saída pode ser estimada através do previsor

$$\hat{y}(k) = CA_0^{k-1}x(1) + y_f^T(k)L + Du(k) + \sum_{i=1}^{n_u} u_{fi}^T(k)B_{a_i}. \quad (3.15)$$

Percebe-se que o previsor que aqui se formula é, na verdade, a saída do observador de estado

$$\begin{aligned} \hat{x}(k+1) &= A_0\hat{x}(k) + Ly(k) + B_a u(k) \\ \hat{y}(k) &= C\hat{x}(k) + Du(k), \end{aligned} \quad (3.16)$$

que está representado na Figura 3.1 e em que $A_0 = A - LC$ e $B_a = B - LD$. Uma vez que A_0 e C são conhecidos, é possível utilizar o filtro da Figura 3.1 para gerar os sinais $y_f(k)$ e $u_{if}(k)$, $i = 1, \dots, n_u$. Logo, $\hat{y}(k)$, definido em (3.15), é uma função linear dos seus parâmetros desconhecidos ($x(1)$, L , D e B_{a_i}), e estes podem ser estimados através de um simples Estimador de Mínimos Quadrados. Já as matrizes de estado e de controlo do sistema (A e B) podem ser extraídas através das equações (3.4) e (3.9), ficando agora conhecidos todos os parâmetros do modelo (3.1)-(3.2).

3.2.2 Estimação dos Parâmetros do Modelo

Unindo as variáveis CA_0^{k-1} , $y_f^T(k)$ e $u_{if}^T(k)$ no regressor

$$\varphi(k) = \left[CA_0^{k-1} \quad y_f^T(k) \quad u_{1f}^T(k) \quad \dots \quad u_{n_{uf}}^T(k) \right] \quad (3.17)$$

e $x(1)$, L e B_{a_i} e D no vetor de parâmetros

$$\theta = \left[x^T(1) \quad L^T \quad B_{a_1}^T \quad \dots \quad B_{a_{n_u}}^T \quad D \right]^T, \quad (3.18)$$

a equação (3.11) pode então ser escrita como

$$y(k) = \varphi(k)\theta + e_f(k). \quad (3.19)$$

Mais uma vez percebe-se que o previsor da saída, $\hat{y}(k) = \varphi(k)\theta$, é uma função linear dos parâmetros. Define-se para $k = 1, \dots, N$,

$$Y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \quad \Phi = \begin{bmatrix} \varphi(1) \\ \vdots \\ \varphi(N) \end{bmatrix} \quad \mathcal{E} = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix} \quad (3.20)$$

e Y , Φ e \mathcal{E} relacionam-se em

$$Y = \Phi\theta + \mathcal{E}. \quad (3.21)$$

Desta forma, o objetivo passa por determinar o valor do estimador θ de mínimos quadrados que minimiza a função custo

$$J = \frac{1}{2} (Y - \Phi\theta)^T (Y - \Phi\theta) = \frac{1}{2} \varepsilon^T \varepsilon. \quad (3.22)$$

Se $\Phi^T \Phi$ for não singular, o Estimador de Mínimos Quadrados é dado por

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y. \quad (3.23)$$

Por outro lado, o regressor $\varphi(k)$ contém o sinal $y_f(k)$, o que poderá indicar que $\hat{\theta}$ pode não ser um estimador consistente. No entanto, como $y_f(k)$ é obtido através da filtragem de $y(k)$ por $G(q) = (qI - A_0^T)^{-1} C^T$, uma escolha apropriada de um valor de A_0 pode reduzir consideravelmente valores de enviesamento e de variância de $\hat{\theta}$, conseguindo-se chegar a um bom estimador. Tendo ainda em conta a equação (3.11), e considerando valores de $L = K$, o erro de previsão do sistema passa a ser apenas uma sequência de ruído branco $e(k)$. Assim, a matriz de estado do previsor toma o valor $A_0 = A - KC$ que corresponde ao valor ótimo a que se pretende chegar. Os algoritmos que conseguem encontrar este valor ótimo são equivalentes aos Métodos do Erro de Previsão (PEM), já referidos no capítulo 2.

3.3 Abordagens MOLI para Identificação de Sistemas MISO

Em (3.2), percebe-se que a formulação de um previsor de estado é independente da realização de espaço de estados que se está a analisar. No entanto, a análise do sistema seria facilitada se houvesse uma relação conhecida entre a matriz de estado desse previsor e o seu polinómio característico. Em sistemas MISO essa relação pode ser estabelecida quando as matrizes A e A_0 , a matriz de estado do sistema e a matriz de estado do previsor, apresentam a mesma forma, o que só é possível quando a realização em espaço de estados está na forma canónica observável. A forma canónica observável é definida por

$$\begin{aligned} A &= \begin{bmatrix} 0 & \cdots & 0 & -a_{n_x} \\ 1 & \cdots & 0 & -a_{n_x-1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -a_1 \end{bmatrix} & B &= \begin{bmatrix} b_{n_x}^T \\ b_{n_x-1}^T \\ \vdots \\ b_1^T \end{bmatrix}, b_i \in \mathbb{R}^{n_u} \\ C &= [0 \quad \cdots \quad 0 \quad 1] & D &= b_0^T \\ A_0 &= \begin{bmatrix} 0 & \cdots & 0 & -\alpha_{n_x} \\ 1 & \cdots & 0 & -\alpha_{n_x-1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -\alpha_1 \end{bmatrix}. \end{aligned} \quad (3.24)$$

A estimação de uma realização do sistema no espaço de estados colocada na forma canónica observável permitiu formular vários algoritmos de identificação que tentam encontrar o melhor

valor inicial para A_0 . Salienta-se também que a utilização desta forma de representação do sistema possibilita a diminuição da dimensão do espaço de otimização, correspondendo este apenas à dimensão n_x do sistema em análise. O espaço de otimização é, então, reduzido ao número de coeficientes do polinómio característico do observador.

3.3.1 ARX MOLI

O algoritmo ARX MOLI utiliza um vetor $\alpha = [\alpha_{n_x} \ \cdots \ \alpha_1]^T$ nulo, formulando-se, então, a seguinte matriz de estado do observador:

$$A_0 = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}. \quad (3.25)$$

Consegue-se, assim, provar que $\hat{\theta}$, definido em (3.23), é o Estimador de Mínimos Quadrados do modelo ARX (*Auto-Regressive with Exogenous Input*) [5] representado na Figura 3.4. Este modelo corresponde ao modelo estimado pelo comando "arx" da *toolbox* de identificação de sistemas do MATLAB [43]. Foi assim que surgiu a designação ARX MOLI [5] para o algoritmo referido.

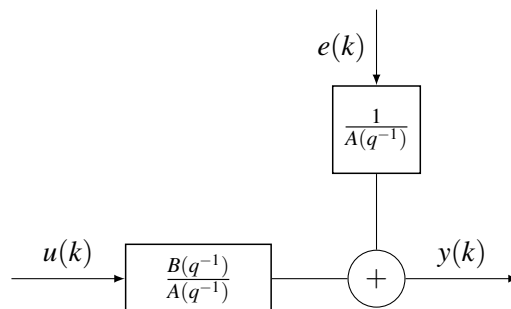


Figura 3.4: Modelo ARX.

3.3.2 The Output MOLI

Este algoritmo baseia-se no modelo *Output Error* considerando-se, assim, que o sistema em causa não é afetado por ruído de processo, sendo apenas influenciado por ruído de saída. Isto é,

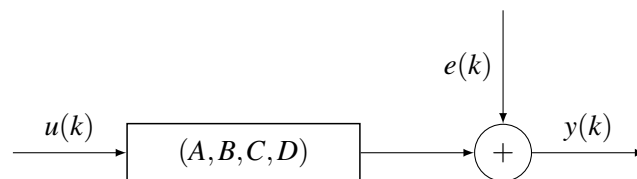


Figura 3.5: Modelo *Output Error*.

considera-se que o ganho K do modelo de inovação (3.1)-(3.2) é nulo, o que corresponde agora ao

modelo

$$x(k+1) = Ax(k) + Bu(k) \quad (3.26)$$

$$y(k) = Cx(k) + Du(k) + e(k), \quad (3.27)$$

onde $e(k)$ é ruído branco. Porém, e dado que K é nulo, o ganho L do observador terá de ter um valor igualmente nulo para que o ruído

$$e_f(k) = [1 + C(qI - A_0)^{-1}(K - L)]e(k) \Rightarrow e_f(k)|_{K=L} = e(k) \quad (3.28)$$

seja uma sequência pura de ruído branco. Consequentemente, o valor ótimo da matriz de estado do observador é $A_0 = A$.

Lopes dos Santos *et al.* [5] propuseram um algoritmo de ponto fixo que tenta chegar ao valor ótimo A_0 . Para tal, em cada iteração, modifica-se o valor da matriz A_0 para o valor de A estimado na iteração anterior:

$$A_0^{(i+1)} = A^{(i)}. \quad (3.29)$$

3.3.3 PEM MOLI

O algoritmo *PEM MOLI*, formulado em [5], é baseado na abordagem de mínimos quadrados separáveis [44, 45]. Conhecendo as matrizes A_0 e C , tanto o ganho L do observador como as matrizes B_a e D do sistema podem ser estimadas através da minimização da função custo J (3.22)

$$J(\alpha, \theta) = \frac{1}{2} [Y - \Phi(\alpha)\theta]^T [Y - \Phi(\alpha)\theta], \quad (3.30)$$

em que Φ é uma função de α . Φ depende, então, dos coeficientes da equação característica de A_0 , pois Φ é formado pelos produtos CA_0^{k-1} e pelos sinais $y_f(k)$ e $u_f(k)$, $k = 1, \dots, N$, sendo estes as saídas do filtro representado na Figura 3.2, cuja matriz de estado é A_0^T . Dado que $Y - \Phi(\alpha)\theta$ é uma função linear de θ , o valor de θ , que minimiza $J(\alpha, \theta)$, é também uma função de α dada por

$$\hat{\theta}(\alpha) = \Phi^\dagger(\alpha)Y, \quad (3.31)$$

em que \dagger é o operador pseudo inverso. Se na equação (3.30) θ for substituído pelo sua estimativa $\hat{\theta}(\alpha)$ (3.31), J torna-se uma função de apenas α dada por

$$J(\alpha) = \frac{1}{2} Y^T \Pi_\Phi^\perp(\alpha) Y, \quad (3.32)$$

em que

$$\Pi_\Phi^\perp = I_N - \Pi_\Phi(\alpha) = I_N - \Phi(\alpha) [\Phi(\alpha)^T \Phi(\alpha)]^{-1} \Phi(\alpha)^T \quad (3.33)$$

é o operador projeção no complemento ortogonal do espaço colunas de $\Phi(\alpha)$. Já $\Pi_{\Phi}(\alpha)$ corresponde ao operador projeção ortogonal:

$$\Pi_{\Phi}(\alpha) = \Phi(\alpha)\Phi(\alpha)^{\dagger}. \quad (3.34)$$

O algoritmo *PEM MOLI* é, então, uma abordagem de mínimos quadrados separáveis que tenta minimizar uma função custo $J(\alpha)$ através de um método de gradiente, neste caso, o método de Gauss-Newton.

3.3.3.1 Implementação do método de Gauss-Newton

O valor de α é gerado iterativamente através de

$$\alpha^{(i+1)} = \alpha^{(i)} - \left(\frac{d(\Pi_{\Phi}^{\perp}(\alpha)Y)}{d\alpha^T} \right)^{\dagger} \Pi_{\Phi}^{\perp}(\alpha)Y. \quad (3.35)$$

Prova-se que

$$\frac{d(\Pi_{\Phi}^{\perp}(\alpha)Y)}{d\alpha^T} = -\Pi_{\Phi}^{\perp}(\alpha) \frac{d\Phi(\alpha)}{d\alpha^T} (I_{n_x} \otimes \theta(\alpha)) - (\Phi^{\dagger}(\alpha))^T \frac{d\Phi(\alpha)^T}{d\alpha^T} (I_{n_x} \otimes \Pi_{\Phi}^{\perp}(\alpha)Y), \quad (3.36)$$

sendo que \otimes indica o produto de *Kronecker* e θ é dado pela expressão definida em (3.31). Para qualquer valor da matriz M e do vetor de parâmetros $\beta = [\beta_1 \ \beta_2 \ \dots \ \beta_n]^T \in \mathbb{R}^n$, define-se

$$\frac{dM}{d\beta^T} = \begin{bmatrix} \frac{dM}{d\beta_1} & \frac{dM}{d\beta_2} & \dots & \frac{dM}{d\beta_n} \end{bmatrix}. \quad (3.37)$$

A derivada de $J(\alpha)$ (3.32) corresponde a

$$\frac{dJ(\alpha)}{d\alpha} = \left(\frac{d\Pi_{\Phi}^{\perp}(\alpha)Y}{d\alpha^T} \right)^T \Pi_{\Phi}^{\perp}(\alpha)Y. \quad (3.38)$$

Uma vez que na equação (3.36) se verifica que o termo $\frac{d\Pi_{\Phi}^{\perp}(\alpha)Y}{d\alpha^T}$ é o complemento ortogonal no espaço colunas de $\Pi_{\Phi}^{\perp}(\alpha)Y$, o gradiente de $J(\alpha)$ pode também ser visto como

$$\frac{dJ(\alpha)}{d\alpha} = -\Pi_{\Phi}^{\perp}(\alpha) \frac{d\Phi(\alpha)}{d\alpha^T} (I_{n_x} \otimes \theta(\alpha)) \Pi_{\Phi}^{\perp}(\alpha)Y. \quad (3.39)$$

Deste modo, redefine-se a equação (3.36) apenas como

$$\frac{d(\Pi_{\Phi}^{\perp}(\alpha)Y)}{d\alpha^T} = -\Pi_{\Phi}^{\perp}(\alpha) \frac{d\Phi(\alpha)}{d\alpha^T} (I_{n_x} \otimes \theta(\alpha)), \quad (3.40)$$

podendo, assim, α ser gerado através de

$$\alpha_{i+1} = \alpha_i + \left(\Pi_{\Phi}^{\perp}(\alpha) \frac{d\Phi(\alpha)}{d\alpha^T} (I_{n_x} \otimes \theta(\alpha)) \right)^{\dagger} \Pi_{\Phi}^{\perp}(\alpha) Y \Big|_{\alpha=\alpha_i}. \quad (3.41)$$

Para que o algoritmo seja numericamente robusto, o cálculo de α_{i+1} deve ser efetuado por uma Decomposição QR. Dado que $\alpha \in \mathbb{R}^{n_x}$, a equação (3.37) permite perceber que $\frac{d\Phi(\alpha)}{d\alpha^T}$ é composto por n_x blocos de colunas, sendo cada posição j definida por

$$\frac{d\Phi(\alpha)}{d\alpha_j} = \begin{bmatrix} \frac{\partial \Gamma_N}{\partial a_j} & \frac{\partial Y_f}{\partial a_j} & \frac{\partial U_f}{\partial a_j} \end{bmatrix}, \quad (3.42)$$

em que

$$\left(\frac{\partial \Gamma_N}{\partial a_j} \right)_i = \begin{cases} 0_{(1 \times n_x)}, & i = 0 \\ \frac{\partial (CA_0^i)}{\partial a_j}, & i = 1, \dots, N-1. \end{cases} \quad (3.43)$$

$\frac{\partial A_0}{\partial a_i}$ é a matriz com o valor

$$\left(\frac{\partial A_0}{\partial a_i} \right)_{n_x, n_x+1-i} = -1 \quad (3.44)$$

e as restantes entradas são nulas. Mostra-se também que cada linha de $\frac{\partial Y_f}{\partial a_j}$ e de $\frac{\partial U_{f_i}}{\partial a_j}$ para $i = 1, \dots, n_u$ corresponde ao valor da saída transposta do sistema

$$\mathcal{X}(k+1) = A_0^T \mathcal{X}(k) - [0 \dots 1]^T v(k) \quad (3.45)$$

$$\psi(k) = \mathcal{X}(k), \quad (3.46)$$

cujas entradas são respetivamente $v(k) = y(k)$ e $v(k) = u_i(k)$, $i = 1, \dots, n_u$.

3.3.4 Barycenter MOLI-ZOFT

Barycenter MOLI-ZOFT é um algoritmo que utiliza um método sem derivadas que pode servir como alternativa ao método de gradiente para otimização da matriz A_0 , a matriz de estado do observador. Este algoritmo foi inicialmente proposto Romano e Pait [46] para a identificação de sistemas multivariáveis, e tenta encontrar os valores ótimos dos α parâmetros da matriz A_0 através da média pesada

$$\hat{\alpha} = \frac{\sum_{i=1}^n \alpha_i e^{-\mu J(\alpha_i)}}{\sum_{i=1}^n e^{-\mu J(\alpha_i)}}. \quad (3.47)$$

O baricentro $\hat{\alpha} = [\hat{\alpha}_{n_x} \dots \hat{\alpha}_1]^T$ corresponde, assim, ao centro de massa ($m_i = e^{-\mu J(\alpha_i)}$) de um conjunto de pontos de interesse designados por "curiosidades". Cada "curiosidade" é definida por

$$\alpha_i = [\alpha_{i_1} \dots \alpha_{i_{n_x}}]^T. \quad (3.48)$$

No cálculo do baricentro, a constante μ associada ao termo do peso $e^{-\mu J(\alpha_i)}$, dado a cada "curiosidade", tem a função de dar mais valor às "curiosidades" que apresentam um melhor desempenho, ou seja, que garantem um valor mínimo para a função custo.

Nesta abordagem MOLI foram propostas três formas diferentes de gerar as "curiosidades" [1, 46, 5]:

- **"Curiosidades" de Polo Dominante (Dominant Pole "Curiosities")** [1, 46] - as "curiosidades" são definidas pela otimização de dois parâmetros. Aqui α corresponde ao polinómio característico de um filtro passa-baixo com um par de pólos complexos dominantes que estão parametrizados em termos de frequência de corte w_c e fator de amortecimento ζ ;
- **"Curiosidades" Aleatórias (Random "Curiosities")** [5] - os valores próprios das "curiosidades" são reais, estáveis e gerados aleatoriamente entre um valor mínimo m e um valor máximo M ;
- **"Curiosidades" Politópicas (Polytopic "Curiosities")** [5] - os valores próprios das "curiosidades" são mais uma vez estáveis, mas podem ser agora reais

$$\lambda = M, |M| < 1 \vee \lambda = m, m < M, |m| < 1$$

ou complexos

$$\lambda = \frac{(m+M)}{2} \pm j\sqrt{M^2 - \frac{(m+M)^2}{4}}, \quad (3.49)$$

estando definidos nos vértices de um politopo.

Capítulo 4

Algoritmo do Baricentro Iterativo

4.1 Introdução

Os algoritmos *Barycenter MOLI-ZOFT*, apresentados no capítulo anterior, tentam minimizar uma determinada função objetivo através do cálculo iterativo de uma média pesada, designada por baricentro, ou centro de massa [1, 46, 3, 5]. Nestes algoritmos, o centro de massa é determinado a partir de "curiosidades" geradas segundo um determinado critério que tem em conta o conhecimento prévio de todos os pontos de interesse do sistema em análise [1, 46]. O processo de geração de "curiosidades" é, assim, um processo não iterativo. Contudo, intuitivamente percebe-se que o conhecimento sobre o sistema também pode ser adquirido durante esse processo. Conclui-se, então, que uma abordagem a ter em conta seria a geração de "curiosidades" de uma forma iterativa, o que deu origem ao algoritmo do Baricentro Iterativo. Este algoritmo, proposto em [47] por Saraiva *et al.*, está baseado no método do Baricentro Discreto e tenta encontrar o valor ótimo dessas "curiosidades" recursivamente. O Baricentro Iterativo pode ainda ser visto como um Processo de Markov que tem como estado o baricentro, pois, sendo este a média pesada de todas as "curiosidades" geradas até ao momento, funciona como memória do processo. Assim, em cada iteração, apenas é necessária a informação relativa à "curiosidade" que se está a analisar e ao valor atual do baricentro.

4.2 Método do Baricentro Discreto

Este método, proposto por Vicinansa *et al.* [48], foi inicialmente implementado em problemas de controlo ótimo, tendo sido, por sua vez, neste trabalho adaptado para problemas de otimização no contexto da Identificação de Sistemas. Trata-se de um método sem derivadas que procura o valor mínimo de uma determinada função não linear através de uma média pesada. O cálculo dessa média pesada é efetuado de forma iterativa e, em cada iteração, gera-se uma nova "curiosidade" α_{i+1} através de

$$\alpha_{i+1} = \hat{\alpha}_i + s_{i+1}, \quad (4.1)$$

em que $\hat{\alpha}_i$ é o valor do baricentro estimado na iteração anterior:

$$\hat{\alpha}_i = \frac{\sum_{j=1}^i \alpha_j e^{-\mu J(\alpha_j)}}{\sum_{j=1}^i e^{-\mu J(\alpha_j)}}, \quad (4.2)$$

e s_{i+1} é um número aleatório. Esta é a variável que garante a dinâmica de convergência do algoritmo, uma vez que s_{i+1} funciona como um passo na direção do valor ótimo a que se pretende chegar. Assim, s_{i+1} foi designado por Passo de Pesquisa (*Search Step*), definido por

$$s_{i+1} = r_i \left(\frac{J(\hat{\alpha}_i)}{\max_{j \in \{1, \dots, i\}} \{J(\hat{\alpha}_j)\}} \right)^\gamma, \quad (4.3)$$

e resulta da multiplicação de um número aleatório r_i pela componente

$$\left(\frac{J(\hat{\alpha}_i)}{\max_{j \in \{1, \dots, i\}} \{J(\hat{\alpha}_j)\}} \right)^\gamma, \quad (4.4)$$

definida em [48] como Fator de Forma (*Shape Factor*). O expoente γ é um parâmetro de projeto e apenas pode tomar valores no intervalo $0 < \gamma \leq 1$. Na verdade, o Fator de Forma tem como finalidade diminuir o valor de s_{i+1} , quando o algoritmo está a aproximar-se do valor mínimo.

Em [48] provou-se que o valor do baricentro também pode ser encontrado através de

$$\hat{\alpha}_i = \hat{\alpha}_0 + \sum_{j=1}^i \frac{s_j e^{-\mu J(\alpha_j)}}{S_j}, \text{ em que } S_j = \sum_{n=1}^j e^{-\mu J(\alpha_n)}, \quad (4.5)$$

em que $\hat{\alpha}_0$ é o seu valor inicial. A partir desta equação é possível definir uma expressão para o incremento do baricentro obtido entre cada iteração:

$$\hat{\alpha}_{i+1} - \hat{\alpha}_i = \frac{s_{i+1} e^{-\mu J(\alpha_{i+1})}}{S_{i+1}}. \quad (4.6)$$

Deste modo, se a função custo J for limitada no intervalo

$$L_m < J(\alpha_j) < L_M, \quad (4.7)$$

então

$$\sum_{j=1}^{i+1} e^{-\mu L_M} < \sum_{j=1}^{i+1} e^{-\mu J(\alpha_j)} < \sum_{j=1}^{i+1} e^{-\mu L_m}. \quad (4.8)$$

Assim,

$$\lim_{i \rightarrow \infty} S_{i+1} = \lim_{i \rightarrow \infty} \sum_{j=1}^{i+1} e^{-\mu J(\alpha_j)} = \infty \quad (4.9)$$

e

$$\lim_{i \rightarrow \infty} \|\hat{\alpha}_{i+1} - \hat{\alpha}_i\| = \lim_{i \rightarrow \infty} \frac{\|s_i\|}{S_{i+1}} = \frac{\|s_i\|}{\infty} = 0 \quad (4.10)$$

provando-se, então, que o algoritmo converge.

4.2.1 Fator de Esquecimento

A convergência do algoritmo deve-se sobretudo ao crescimento ilimitado do somatório dos pesos S_{i+1} . Surgiu, então, a ideia de se experimentar a ação de um Fator de Esquecimento ρ , associado ao cálculo do baricentro de todas as "curiosidades". Na abordagem que foi implementada, o baricentro resulta, assim, em

$$\hat{\alpha}_i = \frac{\sum_{j=1}^i \alpha_j e^{-\mu J(\alpha_j)} \rho^{i-j}}{\bar{S}_i}, \quad (4.11)$$

em que

$$\bar{S}_i = \sum_{j=1}^i e^{-\mu J(\alpha_j)} \rho^{i-j} = \rho \bar{S}_{i-1} + e^{-\mu J(\alpha_i)}. \quad (4.12)$$

ρ , definido no intervalo $0 < \rho < 1$, dá um peso maior às "curiosidades" geradas nas iterações mais recentes, que se supõem estar mais próximas do valor ótimo e esquece as "curiosidades" mais antigas, que estarão mais afastadas desse valor ótimo. Desta forma, espera-se obter uma estimativa melhor e mais exata do baricentro. A equação (4.6) pode ser, então, reescrita como

$$\hat{\alpha}_i - \hat{\alpha}_{i-1} = \frac{s_i e^{-\mu J(\alpha_i)}}{\bar{S}_i}. \quad (4.13)$$

Mais uma vez, com

$$L_m < J(\alpha_i) < L_M \quad (4.14)$$

obtém-se

$$\frac{e^{-\mu L_M}}{1 - \rho} < \lim_{i \rightarrow \infty} \bar{S}_i < \frac{e^{-\mu L_m}}{1 - \rho}. \quad (4.15)$$

No entanto, o somatório dos pesos \bar{S}_i toma agora um valor finito, o que impede o algoritmo de convergir. Verifica-se que o valor do baricentro se aproxima do valor mínimo a que se quer chegar, mas acaba por nunca o atingir. Por outro lado, os valores que o baricentro apresenta em torno desse mínimo são valores aleatórios dentro de uma hiper-esfera de raio definido por ρ e μ . De modo a restringir esse passeio aleatório, estabeleceu-se um limite para o número máximo de iterações que se pode atingir. Veja-se, então, um exemplo para o limite máximo de iterações:

$$I_{\max} = \frac{10}{1 - \rho}. \quad (4.16)$$

4.2.2 Inicialização

O valor do baricentro inicial é determinante para garantir uma convergência mais rápida do algoritmo. Foram, assim, implementadas três formas diferentes de inicialização:

- **Inicialização Simples (*Simple Initialization*)** - o valor inicial do baricentro é um ponto aleatoriamente gerado no conjunto de valores admissíveis de α , ou seja, valores que conduzem a matrizes de estado A_0 do observador estáveis;
- **Inicialização Aleatória (*Random Initialization*)** - o valor inicial do baricentro é gerado pela versão não iterativa do método do Baricentro, isto é, pelo algoritmo *Barycenter MOLI-ZOFT* com inicialização aleatória, referida no capítulo 3;
- **Inicialização Politópica (*Polytopic Initialization*)** - o valor inicial do baricentro é gerado pela versão não iterativa do método Baricentro, isto é, pelo algoritmo *Barycenter MOLI-ZOFT* com inicialização politópica, também apresentada no capítulo 3.

Capítulo 5

Interpretação Bayesiana do Baricentro

5.1 Introdução

Neste capítulo, mostra-se que em determinadas condições a estimação dos parâmetros de um sistema através de um método de Baricentro aproxima-se de um Estimador de Bayes.

Os Estimadores de Bayes consideram que os parâmetros do sistema são variáveis aleatórias com uma determinada função de densidade de probabilidade. É através da minimização de um determinado critério de desempenho que esses parâmetros são encontrados. Se esse critério for o Erro Quadrático Médio dos parâmetros do sistema, então, essa minimização é equivalente a uma média pesada, ou seja, a um baricentro. Daí se poder concluir que, com "curiosidades" e valores de peso adequados, o método do Baricentro é um estimador bayesiano que minimiza o Erro Quadrático Médio dos parâmetros.

5.2 Estimador de Bayes de Erro Quadrático Médio Mínimo

Ao considerarem os parâmetros $\alpha \in \mathbb{R}^{n_x}$ e as observações $Y \in \mathbb{R}^N$ como variáveis aleatórias definidas em espaços amostrais Ω_α e Ω_Y , os Estimadores de Bayes resolvem o problema:

$$\hat{\alpha} = \operatorname{argmin}_{\hat{\alpha}} \mathbf{E}\{C(\hat{\alpha} - \alpha)\} = \operatorname{argmin}_{\hat{\alpha}} \int_{\Omega_Y} \int_{\Omega_\alpha} C(\hat{\alpha} - \alpha) p(Y, \alpha) d\alpha dY, \quad (5.1)$$

em que $p(Y, \alpha)$ é a função de densidade de probabilidade conjunta de Y e α , e C representa a função custo a minimizar. C pode tomar várias formulações consoante o erro que se quer minimizar e o critério que se estabelece para fazê-lo. Se C for o erro quadrático dos parâmetros do sistema, isto é, se

$$C(\hat{\alpha} - \alpha) = (\hat{\alpha} - \alpha)^T (\hat{\alpha} - \alpha) = \|\hat{\alpha} - \alpha\|_2^2, \quad (5.2)$$

então

$$\hat{\alpha} = \mathbf{E}\{\alpha|Y\}, \quad (5.3)$$

ou seja, $\hat{\alpha}$ é o valor esperado de α condicionado às observações de Y [49]. Deste modo, percebe-se que o estimador de parâmetros $\hat{\alpha}$ (5.3) minimiza um critério quadrático do erro dos parâmetros, sendo, por isso, designado por Estimador de Erro Quadrático Médio Mínimo (*Minimum Mean Square Estimator* - MMSE) [49, 50]. Pode, então, definir-se a equação para MMSE como

$$\hat{\alpha} = \mathbf{E}\{\alpha|Y\} = \int_{\Omega_\alpha} \alpha p(\alpha|Y) d\alpha, \quad (5.4)$$

em que $p(\alpha|Y)$ é a função de densidade de probabilidade à posteriori. Por outro lado, sabe-se que quando se observa a saída de um sistema dinâmico, se consegue definir apenas uma função de densidade de probabilidade condicionada $p(Y|\alpha)$, dado que Y é função de α . Esta é também a função de verossimilhança maximizada pelo Estimador de Máxima Verossimilhança.

Aplicando a Regra de Bayes,

$$p(\alpha|Y) = \frac{p(Y|\alpha)p(\alpha)}{p(Y)}, \quad (5.5)$$

e tendo em conta as definições

$$\int_{\Omega_\alpha} p(\alpha) d\alpha = 1 \quad (5.6)$$

$$\int_{\Omega_\alpha} p(\alpha|Y) d\alpha = 1, \quad (5.7)$$

pode reescrever-se a equação anterior como

$$\begin{aligned} \int_{\Omega_\alpha} p(\alpha|Y) d\alpha = 1 &\Leftrightarrow \int_{\Omega_\alpha} \frac{p(Y|\alpha)p(\alpha)}{p(Y)} d\alpha = 1 \Leftrightarrow \frac{1}{p(Y)} \int_{\Omega_\alpha} p(Y|\alpha)p(\alpha) d\alpha = 1 \quad (5.8) \\ &\Leftrightarrow p(Y) = \int_{\Omega_\alpha} p(Y|\alpha)p(\alpha) d\alpha. \end{aligned}$$

Analisando agora a estimativa bayesiana definida em (5.4)

$$\hat{\alpha} = \int_{\Omega_\alpha} \alpha p(\alpha|Y) d\alpha = \int_{\Omega_\alpha} \alpha \frac{p(Y|\alpha)p(\alpha)}{p(Y)} d\alpha = \frac{1}{p(Y)} \int_{\Omega_\alpha} \alpha p(Y|\alpha)p(\alpha) d\alpha \quad (5.9)$$

e utilizando a equação (5.8), obtêm-se

$$\hat{\alpha} = \frac{\int_{\Omega_\alpha} \alpha p(Y|\alpha)p(\alpha) d\alpha}{\int_{\Omega_\alpha} p(Y|\alpha)p(\alpha) d\alpha} = \frac{\mathbf{E}_\alpha\{\alpha p(Y|\alpha)\}}{\mathbf{E}_\alpha\{p(Y|\alpha)\}}, \quad (5.10)$$

em que $\mathbf{E}_\alpha\{\cdot\}$ é o operador valor esperado em α .

Assume-se agora que Y condicionado a α tem a seguinte distribuição de probabilidade gaussiana:

$$p(Y|\alpha) = \frac{1}{(2\pi)^{N/2} \det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}[Y-\mu_Y(\alpha)]^T \Sigma^{-1} [Y-\mu_Y(\alpha)]}, \quad (5.11)$$

em que $\mu_Y(\alpha) = \mathbf{E}\{Y|\alpha\}$, ou seja, $\mu_Y(\alpha)$ é o valor esperado de Y condicionado a α , e Σ é a sua matriz de covariância. Se Y for um vetor de variáveis aleatórias independentes com variância σ^2 , então,

$$\Sigma = \mathbf{E}\{\varepsilon\varepsilon^T\} = \sigma^2 I, \quad (5.12)$$

e

$$p(Y|\alpha) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{1}{2\sigma^2}[Y-\mu_Y(\alpha)]^T[Y-\mu_Y(\alpha)]}. \quad (5.13)$$

Definindo a constante $K_p = \frac{1}{(2\pi\sigma^2)^{N/2}}$, a equação (5.10) pode ser reescrita como

$$\hat{\alpha} = \frac{\int_{\Omega_\alpha} \alpha K_p e^{-\frac{1}{2\sigma^2}[Y-\mu_Y(\alpha)]^T[Y-\mu_Y(\alpha)]} p(\alpha) d\alpha}{\int_{\Omega_\alpha} K_p e^{-\frac{1}{2\sigma^2}[Y-\mu_Y(\alpha)]^T[Y-\mu_Y(\alpha)]} p(\alpha) d\alpha} = \frac{\mathbf{E}_\alpha\{\alpha e^{-\frac{1}{2\sigma^2}[Y-\mu_Y(\alpha)]^T[Y-\mu_Y(\alpha)]}\}}{\mathbf{E}_\alpha\{e^{-\frac{1}{2\sigma^2}[Y-\mu_Y(\alpha)]^T[Y-\mu_Y(\alpha)]}\}}. \quad (5.14)$$

Definindo ainda $J = \frac{1}{2}[Y - \mu_Y(\alpha)]^T [Y - \mu_Y(\alpha)]$ e considerando que α pode tomar qualquer valor do conjunto de valores admissíveis com igual probabilidade, isto é, se $p(\alpha)$ for a função de densidade de probabilidade de uma distribuição uniforme, então

$$\hat{\alpha} = \frac{\int_{\Omega_\alpha} \alpha e^{-\frac{1}{\sigma^2}J} d\alpha}{\int_{\Omega_\alpha} e^{-\frac{1}{\sigma^2}J} d\alpha}. \quad (5.15)$$

Se se tiver uma amostra aleatória de α de dimensão N_c , então, quando $N_c \rightarrow \infty$,

$$\hat{\alpha} = \frac{\sum_{i=1}^{N_c} \alpha_i e^{-\frac{1}{\sigma^2}J}}{\sum_{i=1}^{N_c} e^{-\frac{1}{\sigma^2}J}} \rightarrow \mathbf{E}\{\alpha|Y\} \quad (5.16)$$

Fica, assim, provado que MMSE $\hat{\alpha}$ é uma média pesada.

5.2.1 O Baricentro como MMSE

No capítulo 3, viu-se que os algoritmos do Baricentro estimam o vetor α dos coeficientes do polinómio característico do observador, através da média pesada

$$\hat{\alpha} = \frac{\sum_{i=1}^{N_c} \alpha_i e^{-\mu J(\alpha_i)}}{\sum_{i=1}^{N_c} e^{-\mu J(\alpha_i)}}, \quad (5.17)$$

em que

$$J(\alpha) = \frac{1}{2} \sum_{k=1}^N e_f(k, \alpha)^2 \quad (5.18)$$

e $e_f(k, \alpha)$ é o erro de previsão dado pela equação (3.28). Se em (3.1) e (3.2) $e(k)$ for ruído branco gaussiano de média nula, então

$$p(Y|\alpha) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{\frac{1}{2\sigma^2}J(\alpha)}. \quad (5.19)$$

Nestas condições, se em (5.17) $\mu = \frac{1}{\sigma^2}$ e se as "curiosidades" forem uma amostra de α , $\hat{\alpha}$ aproxima-se do Estimador de Bayes MMSE. Se $\bar{\alpha} = \mathbf{E}\{\hat{\alpha}\}$ for o valor médio de um estimador de α com valor verdadeiro α_0 , então, o Erro Quadrático Médio (*Mean Square Error* - MSE) é

$$MSE_{\hat{\alpha}} = \mathbf{E}\left\{\|\hat{\alpha} - \alpha_0\|_2^2\right\} = \text{traço}P_{\hat{\alpha}} + \text{bias}(\hat{\alpha}), \quad (5.20)$$

em que

$$P_{\hat{\alpha}} = \mathbf{E}\left\{(\hat{\alpha} - \bar{\alpha})(\hat{\alpha} - \bar{\alpha})^T\right\} \quad (5.21)$$

é a matriz de covariância de $\hat{\alpha}$ e

$$\text{bias}(\hat{\alpha}) = \mathbf{E}\{\hat{\alpha} - \bar{\alpha}\} \quad (5.22)$$

é o seu enviesamento. Sendo $\alpha = [\alpha_1 \ \cdots \ \alpha_{n_x}] \in \mathbb{R}^{n_x}$, então

$$MSE_{\hat{\alpha}} = \sum_{i=1}^{n_x} [\text{var}(\hat{\alpha}_i) + \text{bias}(\hat{\alpha}_i)^2] = \sum_{i=1}^{n_x} MSE_{\hat{\alpha}_i}, \quad (5.23)$$

ou seja, MSE de um estimador é igual à soma dos erros médios quadráticos dos seus componentes. Apesar de enviesado¹, pode-se afirmar que este Estimador de Bayes é o que oferece estimativas mais próximas do valor verdadeiro de α , quando o erro a minimizar é o erro médio quadrático dos parâmetros. Porém, como o erro $e_f(k, \alpha)$ do predictor não é uma função linear dos parâmetros, a função de densidade de probabilidade $p(Y|\alpha)$, não é uma função simétrica em torno de $\bar{\alpha}$. No entanto, quando $N \rightarrow \infty$,

$$\frac{J(\alpha_0)}{N} \rightarrow \sigma^2, \quad (5.24)$$

e

$$p(Y|\alpha) \rightarrow \frac{1}{(2\pi\sigma^2)^{N/2}} e^{\frac{NJ(\alpha)}{J(\alpha_0)}}, \quad (5.25)$$

pelo que, a "largura" de $p(Y|\alpha)$ vai diminuindo em torno da sua moda e, conseqüentemente, $\hat{\alpha}$ converge para o valor da moda de $p(Y|\alpha)$. Mas como $p(Y|\alpha)$ é a função de verosimilhança, a sua moda, ou seja, o seu valor máximo, é, como já se referiu, o Estimador de Máxima Verosimilhança. Concluiu-se então, que, no caso do ruído ser gaussiano, o baricentro com $\mu = \frac{1}{\alpha^2}$ tende para o Estimador de Máxima Verosimilhança quando o número de observações dos pares de entrada

¹ $\mathbf{E}\{\hat{\alpha}\} \neq \alpha_0$.

e de saída do sistema tende para infinito. Como o Estimador de Máxima Verosimilhança tende assintoticamente para o estimador não enviesado de variância mínima, nestas condições o mesmo se pode afirmar para o Baricentro.

5.3 Algoritmo do Baricentro Iterativo Modificado

Uma das grandes vantagens dos estimadores bayesianos é permitir uma fácil integração do conhecimento que se tem à priori sobre os parâmetros do sistema. Essa integração materializa-se na distribuição à priori $p(\alpha)$.

Se, mais uma vez, se pretender estimar a matriz de estado do previsor do sistema de ordem n_x , formulado no capítulo 3, e, não se tendo qualquer conhecimento prévio do sistema, faz sentido considerar-se que todos os observadores estáveis do sistema são igualmente prováveis. Consequentemente, $\alpha = [\alpha_1 \ \cdots \ \alpha_{n_x}]^T \in \mathbb{R}^{n_x}$ tem uma distribuição uniforme no conjunto

$$\Omega_\alpha = \{ \alpha : \lambda^{n_x} + \alpha_1 \lambda^{n_x-1} + \cdots + \alpha_{n_x} = 0 \Rightarrow |\lambda| < 1 \}. \quad (5.26)$$

Porém, se o sistema for amostrado de acordo com uma frequência de amostragem adequada, o espaço amostral poderá ser reduzido a um conjunto de polinómios com zeros reais e positivos, que estão no interior do círculo unitário.

Sabe-se também que o conhecimento prévio dos parâmetros do sistema pode ter múltiplas origens. Entre essas origens podem estar os resultados obtidos através de um outro estimador que segue uma outra distribuição de probabilidades.

Numa perspetiva bayesiana, um algoritmo iterativo pode ser visto como um processo onde, em cada iteração, se vai obtendo mais informação sobre os parâmetros a estimar. Adotando esta perspetiva para o Baricentro Iterativo apresentado no capítulo anterior, considera-se que, no processo de geração das "curiosidades" através de

$$\alpha_{i+1} = \hat{\alpha}_i + s_{i+1}, \quad (5.27)$$

$\hat{\alpha}_i$ é o valor esperado da distribuição à priori de α e

$$P_{\hat{\alpha}_i} = \frac{\sum_{j=0}^i (\alpha_j - \hat{\alpha}_i) (\alpha_j - \hat{\alpha}_i)^T e^{-\mu J(\alpha_j)}}{\sum_{j=0}^i e^{-\mu J(\alpha_j)}} = \frac{\sum_{j=0}^i \alpha_j \alpha_j^T e^{-\mu J(\alpha_j)}}{\sum_{j=0}^i e^{-\mu J(\alpha_j)}} - \hat{\alpha}_i \hat{\alpha}_i^T \quad (5.28)$$

é a sua matriz de covariância.

Pelo Teorema do Limite Central, pode igualmente considerar-se que α tem uma distribuição

normal e que s_{i+1} é um vetor aleatório com distribuição $N(0, P_{\hat{\alpha}_i})$. Definindo

$$N_i = \sum_{j=0}^i \alpha_j e^{-\mu J(\alpha_j)} \quad (5.29)$$

$$N_i^{(2)} = \sum_{j=0}^i \alpha_j \alpha_j^T e^{-\mu J(\alpha_j)} \quad (5.30)$$

$$S_i = \sum_{j=0}^i e^{-\mu J(\alpha_j)}, \quad (5.31)$$

então

$$\hat{\alpha}_i = \frac{N_i}{S_i}$$

$$P_{\hat{\alpha}_i} = \frac{N_i^{(2)}}{S_i}.$$

Na iteração $i + 1$ obtém-se

$$N_{i+1} = N_i + \alpha_{i+1} e^{-\mu J(\alpha_{i+1})} \quad (5.32)$$

$$N_{i+1}^{(2)} = N_i^{(2)} + \alpha_{i+1} \alpha_{i+1}^T e^{-\mu J(\alpha_{i+1})} \quad (5.33)$$

$$S_{i+1} = S_i + e^{-\mu J(\alpha_{i+1})} \quad (5.34)$$

$$\hat{\alpha}_{i+1} = \frac{N_{i+1}}{S_{i+1}} \quad (5.35)$$

$$P_{\hat{\alpha}_{i+1}} = \frac{N_{i+1}^{(2)}}{S_{i+1}}. \quad (5.36)$$

Para que a abordagem bayesiana fique completa é necessário compreender que $\mu = \frac{1}{2\sigma^2}$. Se α_0 , isto é, o valor verdadeiro de α , for conhecido, pode considerar-se que σ^2 toma o valor $\frac{J(\alpha_0)}{N}$. Tal acontece uma vez que $\frac{J(\alpha_0)}{N} \rightarrow \sigma^2$ quando o número de observações $N \rightarrow \infty$, o que permite substituir a equação da função peso por

$$e^{-\frac{NJ(\alpha_i)}{2J(\alpha_0)}}. \quad (5.37)$$

De facto, o valor de α_0 não é nem nunca será conhecido, pelo que se terá de fazer uma aproximação a esse valor. Essa aproximação poderá ser uma estimativa anterior de $\hat{\alpha}_0$, efetuada por um outro estimador. A verdade é que a função $J(\alpha)$ é quase plana na vizinhança de α_0 , o que permite concluir que $J(\hat{\alpha}_0)$ pode ser uma boa estimativa de $J(\alpha_0)$. Deste modo, utiliza-se, neste trabalho, o algoritmo não iterativo *Barycenter MOLI-ZOFT* com inicialização politópica, referido no capítulo 3, para gerar esse mesmo valor de $\hat{\alpha}_0$.

Na formulação deste algoritmo, teve-se ainda em conta uma normalização do valor da função peso em torno de $J(\hat{\alpha}_i)$ de forma a evitar que esta função peso tome valores muito pequenos ou

muito grandes. Em cada iteraço, multiplica-se, ento, N_{i+1} , $N_{i+1}^{(2)}$ e S_{i+1} por $e^{J(\hat{\alpha}_i)}$, passando o baricentro e a respetiva matriz de covarincia a serem calculados atravs de

$$\hat{\alpha}_{i+1} = \frac{N_{i+1}}{S_{i+1}} \quad (5.38)$$

$$P_{\hat{\alpha}_{i+1}} = \frac{N_{i+1}^{(2)}}{S_{i+1}}, \quad (5.39)$$

em que

$$N_{i+1} = N_i e^{J(\hat{\alpha}_i)} + \alpha_{i+1} e^{-\mu[J(\alpha_{i+1}) - J(\hat{\alpha}_i)]} \quad (5.40)$$

$$N_{i+1}^{(2)} = N_i^{(2)} e^{J(\hat{\alpha}_i)} + \alpha_{i+1} \alpha_{i+1}^T e^{-\mu[J(\alpha_{i+1}) - J(\hat{\alpha}_i)]} \quad (5.41)$$

$$S_{i+1} = S_i e^{J(\hat{\alpha}_i)} + e^{-\mu[J(\alpha_{i+1}) - J(\hat{\alpha}_i)]} \quad (5.42)$$

Capítulo 6

Algoritmo MOLI-ZOE

6.1 Introdução

Com base na interpretação bayesiana do método do Baricentro, formula-se, neste capítulo, um novo algoritmo de identificação para a estimação dos coeficientes α do polinómio característico da matriz de estado do observador, obtidos segundo a parametrização MOLI. Este algoritmo é essencialmente baseado numa abordagem de Mínimos Quadrados Separáveis, que tenta minimizar a função custo definida em (3.32) e para tal utiliza um método sem derivadas.

Uma asserção razoável é considerar-se que $p(Y|\alpha)$ é a função de densidade de probabilidade de uma distribuição gaussiana e que as suas superfícies de nível são elipsoides de ordem n_x :

$$(\alpha - \alpha_0)^T (CP_\alpha)^{-1}(\alpha - \alpha_0) = 1, \quad (6.1)$$

em que α tem dimensão n_x , P_α é a covariância de α e C é uma constante positiva. O valor esperado $\mathbf{E}\{Y|\alpha\}$, isto é o baricentro, é dado por α_0 , que corresponde ao centro de uma elipsoide. As Figuras 6.1 e 6.2 mostram uma função de densidade de probabilidade gaussiana com $n_x = 2$ e as respetivas curvas de nível.

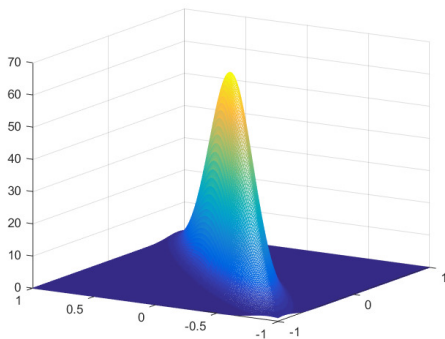


Figura 6.1: Função de densidade de probabilidade de uma distribuição normal a duas dimensões.

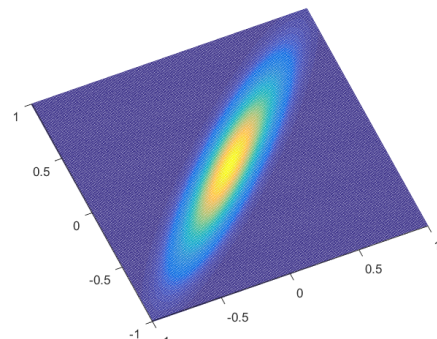


Figura 6.2: Superfícies de nível de uma função de densidade de probabilidade gaussiana a duas dimensões.

Ao considerar-se uma amostra de pontos α_i da elipsoide (6.1), sendo esses pontos simétricos relativamente ao centro dessa elipsoide, então, α_0 é o baricentro, isto é, é o valor médio destes pontos, e $\frac{CP_\alpha}{n_x}$ é a matriz de covariância [51]. Sabe-se que esse conjunto de pontos α_i , em que $i = 1, \dots, 2n_x$, é, na verdade, idêntico ao usado no Filtro de Kalman do tipo *Unscented Kalman Filter* (UKF), sendo cada um desses pontos designado por ponto-sigma [51].

Se não existir qualquer conhecimento prévio sobre a localização do baricentro, considera-se, então, que este se encontra numa região delimitada por uma das elipsoides visíveis em 6.2, que está centrada em $\hat{\alpha}_0$ e é definida pela matriz positiva P_0 . Considerando essa elipsoide como uma superfície de nível, uma estimativa natural de α é o baricentro $\hat{\alpha}_1$ de um conjunto de pontos-sigma dessa elipsoide. No entanto, como a elipsoide foi gerada sem qualquer conhecimento prévio de $p(Y|\alpha)$, esta tem probabilidade nula de ser uma superfície de nível e, por esse motivo, $\hat{\alpha}_1$ não estará exatamente no centro da elipsoide. Contudo, dado que a região delimitada pela superfície de nível é convexa, garante-se que $\hat{\alpha}_1$ está no seu interior e aproxima-se, assim, do valor α_0 verdadeiro.

O processo de geração de novos pontos-sigma sobre elipsoides é, assim, um processo iterativo, onde se geram elipsoides centradas nos sucessivos baricentros $\hat{\alpha}_i$, definidas pelas matrizes de covariância dos pontos-sigma gerados em torno de $\hat{\sigma}_{i-1}$. Se na iteração i ,

$$\hat{\alpha}_i = \alpha_0, \quad (6.2)$$

então, para $j > i$

$$\hat{\alpha}_j = \hat{\alpha}_i = \alpha_0 \quad (6.3)$$

e

$$P_j = \frac{P_i}{(n_x)^{j-i}}. \quad (6.4)$$

Nestas condições, se $n_x > 1$, as elipsoides convergem para o ponto α_0 . Se $\hat{\alpha}_i \neq \alpha_0$, também é expectável que os comprimentos dos eixos principais das elipsoides diminuam de iteração para iteração, convergindo, assim, para um ponto. Contudo, essa convergência pode ocorrer cedo de mais, havendo o perigo de o algoritmo não atingir α_0 . No sentido de se contornar este problema, sempre que o valor de $p(Y|\alpha)$ do centro da elipsoide não for superior a $p(Y|\alpha)$ de todos os pontos-sigma, o centro é deslocado para o ponto-sigma com o maior valor de $p(Y|\alpha)$. Verificou-se ainda que em UKF, a inclusão do centro das elipsoides no conjunto dos pontos-sigma reduz o erro de estimação, uma vez que na aproximação da função de propagação por uma Série de Taylor [51] se eliminam termos de ordem superior à primeira ordem. Por outro lado, a inclusão do centro da elipsoide aumenta o fator de redução da matriz de covariância, acelerando, assim, a convergência do algoritmo e tornando-o, por isso, mais eficiente.

Em suma, em cada iteração, o algoritmo que aqui se propõe gera "curiosidades" que não são mais do que pontos-sigma nas superfícies de nível de uma função de densidade de probabilidade gaussiana, centrada no baricentro encontrado na iteração anterior e definida pela matriz de covariância P , igualmente determinada na iteração anterior. Estas "curiosidades" estão no centro e

nas extremidades dos eixos principais da elipsoide, tal como é ilustrado na Figura 6.3. A geração das "curiosidades" é, por isso, um problema determinístico. Se a elipsoide tiver dimensão n_x , calculam-se $2n_x + 1$ "curiosidades", visto que:

- Se o sistema for de ordem 1, apenas se geram 3 pontos, que, quando unidos, tomam a forma de um segmento de reta;
- Se o sistema for de ordem 2 (Figura 6.3), a figura plana que se forma é uma elipse que tem 2 eixos, pelo que se geram 5 pontos, isto é, 2 em cada eixo principal da elipse e um no seu centro;

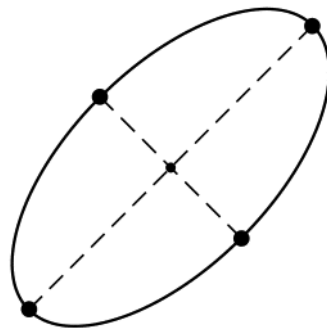


Figura 6.3: Elipsoide de ordem 2.

- Se o sistema for de ordem 3, a elipse deixa de ser plana e tem agora 3 eixos principais, onde se geram 6 pontos nos seus extremos e um outro no centro da elipsoide;

6.2 Geração das "Curiosidades"

Na iteração i , considera-se que α é uma variável aleatória gaussiana com valor médio $\hat{\alpha}_i$ e covariância P_i . A variável aleatória $z = \alpha - \hat{\alpha}_i$ é também gaussiana, tendo a mesma covariância

$$P_i = \mathbf{E} \{zz^T\} \quad (6.5)$$

e valor médio nulo. Aplicando uma decomposição em valores singulares à matriz P_i ¹,

$$P_i = USV^T = USU^T. \quad (6.6)$$

¹ P_i é uma matriz simétrica, ou seja, $P_i = P_i^T$, então $V = U$ na Decomposição em Valores Singulares $P = USV^T$.

Ao definir-se $w = U^T z$, então w é também uma variável aleatória gaussiana² com média nula e covariância

$$\mathbf{E}\{ww^T\} = \mathbf{E}\{U^T z (U^T z)^T\} = \mathbf{E}\{U^T z z^T U\} = U^T \mathbf{E}\{zz^T\} U \quad (6.7)$$

$$\begin{aligned} &= U^T P U = U^T U S U^T U = S, \\ S &= \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}. \end{aligned} \quad (6.8)$$

Como S é uma matriz diagonal, as componentes de w são variáveis aleatórias independentes. Estas são, então, também, os eixos de uma base canônica $\{e_1, e_2, \dots, e_{n_x}\}$ ³, ou seja, os eixos principais da elipsoide.

Ao ter em conta o espaço de w , quer-se agora gerar a elipsoide a partir de um valor médio 0, isto é, $w_0 = 0$. Logo, os pontos a gerar seriam

$$w_1 = \begin{bmatrix} \sqrt{\sigma_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sqrt{\sigma_1} e_1, \quad w_2 = \begin{bmatrix} -\sqrt{\sigma_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = -\sqrt{\sigma_1} e_1, \quad (6.9)$$

$$w_3 = \begin{bmatrix} 0 \\ \sqrt{\sigma_2} \\ \vdots \\ 0 \end{bmatrix} = \sqrt{\sigma_2} e_2, \quad w_4 = \begin{bmatrix} 0 \\ -\sqrt{\sigma_2} \\ \vdots \\ 0 \end{bmatrix} = -\sqrt{\sigma_2} e_2, \quad (6.10)$$

$$\vdots \quad \vdots \quad (6.11)$$

$$w_{2n_x-1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \sqrt{\sigma_{n_x}} \end{bmatrix} = \sqrt{\sigma_{n_x}} e_{n_x}, \quad w_{2n_x} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -\sqrt{\sigma_{n_x}} \end{bmatrix} = -\sqrt{\sigma_{n_x}} e_{n_x}, \quad (6.12)$$

em que σ_i , para $i = 1, \dots, n_x$, são os valores singulares de P (elementos da diagonal de S). Contudo, como $w = U^T z$, então $z = U^T w$, com $U = [u_1 \ u_2 \ \cdots \ u_{n_x}]$. Assim,

$$z_1 = U \sqrt{\sigma_1} e_1 = \sqrt{\sigma_1} U e_1, \quad (6.13)$$

pois $\sqrt{\sigma_1}$ é um escalar. A matriz U funciona, então, como um "cabide", visto que roda o vetor e_1 da base canônica, "pendurando-o" na sua primeira coluna

$$u_1 = U e_1, \quad (6.14)$$

²As combinações lineares de variáveis aleatórias gaussianas são também variáveis aleatórias gaussianas.

³Se e_1, e_2, \dots, e_n forem os vetores da base canônica, então $[e_1 \ e_2 \ \cdots \ e_{n_x}] = I_{n_x}$.

pelo que $z_1 = \sqrt{\sigma_1}u_1$. De igual forma,

$$z_2 = -\sqrt{\sigma_1}u_1, z_3 = \sqrt{\sigma_2}u_2, z_4 = -\sqrt{\sigma_1}u_2, \dots, z_{2n_x-1} = \sqrt{\sigma_{n_x}}u_{n_x}, z_{2n_x} = \sqrt{\sigma_{n_x}}u_{n_x}. \quad (6.15)$$

Por último, como $z = \alpha - \hat{\alpha}_i$, então $\alpha = \hat{\alpha}_i + z$:

$$\alpha_1 = x_0 + z_1 = \hat{\alpha}_i + \sqrt{\sigma_1}u_1, \quad \alpha_2 = x_0 + z_2 = \hat{\alpha}_i - \sqrt{\sigma_1}u_1, \quad (6.16)$$

$$\alpha_3 = x_0 + z_3 = \hat{\alpha}_i + \sqrt{\sigma_2}u_2, \quad \alpha_4 = x_0 + z_4 = \hat{\alpha}_i - \sqrt{\sigma_2}u_2,$$

$$\vdots \quad \quad \quad \vdots$$

$$\alpha_{2n_x-1} = \hat{\alpha}_i + z_{2n_x-1} = x_0 + \sqrt{\sigma_{n_x}}u_{n_x}, \quad \alpha_{2n_x} = \hat{\alpha}_i + z_{2n_x} = x_0 - \sqrt{\sigma_{n_x}}u_{n_x}. \quad (6.17)$$

Os pontos das elipsoides são, portanto, $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \dots, \alpha_{2n_x-1}, \alpha_{2n_x}$.

6.2.1 Inicialização

Primeiramente, gera-se um baricentro inicial qualquer com dimensão n_x . Até ao momento, tem-se utilizado o valor $\hat{\alpha}_0 = 0$. De seguida, gera-se uma elipsoide em torno desse baricentro, assumindo-se o centro da elipsoide e os pontos nas extremidades dos seus eixos como "curiosidades". Essa elipsoide deve ser suficientemente "larga" para cobrir todo o espaço amostral, que neste caso é o conjunto de coeficientes de polinómios estáveis, ou seja, de polinómios com raízes no interior do círculo unitário. Deste modo, tem-se utilizado coeficientes de polinómios com raízes, que ou são nulas ou estão nos limites do círculo unitário:

$$\alpha_{i_1} = \begin{bmatrix} 0.99 \\ 0 \\ \vdots \\ 0_{n_x} \end{bmatrix}, \alpha_{i_2} = \begin{bmatrix} -0.99 \\ 0 \\ \vdots \\ 0_{n_x} \end{bmatrix}, \alpha_{i_3} = \begin{bmatrix} 0 \\ 0.99 \\ \vdots \\ 0_{n_x} \end{bmatrix}, \alpha_{i_4} = \begin{bmatrix} 0 \\ -0.99 \\ \vdots \\ 0_{n_x} \end{bmatrix}, \dots, \alpha_{i_{2n_x}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -0.99 \end{bmatrix}, \quad (6.18)$$

tal como se mostra na Figura 6.4.

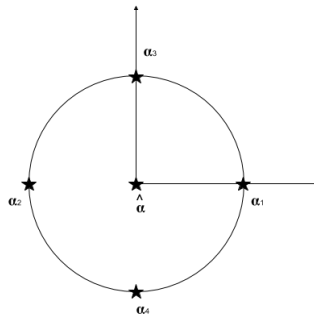


Figura 6.4: Inicialização das "curiosidades".

6.2.2 Pormenores de Implementação

Em cada iteração do algoritmo, calcula-se:

- O baricentro atual $\hat{\alpha}_i$, isto é, a média pesada dos pontos da elipsoide;
- O valor médio quadrático $\mathbf{E}\{\hat{\alpha}\hat{\alpha}^T\}_i$, ou seja, a média pesada dos quadrados dos pontos da elipsoide:

$$\mathbf{E}\{\hat{\alpha}\hat{\alpha}^T\}_i = \frac{\sum_{j=1}^{2n_x+1} \alpha_{i,j} \alpha_{i,j}^T e^{-\frac{1}{2\sigma^2}J(\alpha_{i,j})}}{\sum_{i=1}^N e^{-\frac{1}{2\sigma^2}J(\alpha_i)}}; \quad (6.19)$$

- A matriz P_i que corresponde à covariância de $\hat{\alpha}_i$, sendo que P_i é a matriz de desvio médio quadrático relativo ao novo baricentro encontrado, tal como foi demonstrado em (5.21):

$$P_i = \mathbf{E}\{\hat{\alpha}\hat{\alpha}^T\}_i - \hat{\alpha}_i \hat{\alpha}_i^T; \quad (6.20)$$

- O valor de s que corresponde à raiz quadrada do maior valor singular da matriz P_i^4 :

$$s = \sqrt{\sigma_1}, \quad (6.21)$$

que é encontrado através da Decomposição em Valores Singulares da matriz P_i .

Gera-se, assim, o conjunto de "curiosidades" da próxima iteração, constituído pelo baricentro $\hat{\alpha}_i$ e pelos extremos dos eixos principais da elipsoide centrada em $\hat{\alpha}_i$ e definida por P_i .

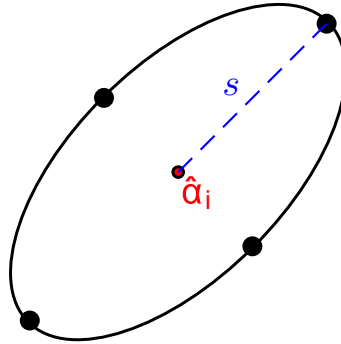


Figura 6.5: Elipse de ordem 2.

6.2.2.1 Função Peso

Considera-se que $p(Y|\alpha)$ é um função uni-modal e que as suas superfícies de nível são elipsoides centradas no seu máximo. Nestas condições, o baricentro é o máximo de $p(Y|\alpha)$ e o Estimador de Bayes ($\hat{\alpha} = \mathbf{E}\{\alpha|Y\}$) é também o Estimador de Máxima Verosimilhança. Por outro lado, como

⁴ $\|P_i\|_2 = \sigma_1$.

$p(Y|\alpha)$ é máximo quando $J(\alpha)$ for mínimo, o estimador $\hat{\alpha}$ também minimiza esta função. Porém, $J(\alpha)$ não é uma função quadrática de α e, conseqüentemente, as suas curvas de nível, que também são curvas de nível de $p(Y|\alpha)$ ⁵, não são precisamente elipses ou elipsoides. Mais ainda, como se ilustra na Figura 6.6, o baricentro $\hat{\alpha} = \mathbf{E}\{\alpha|Y\}$ nem sequer coincide com o máximo de $p(Y|\alpha)$ e,

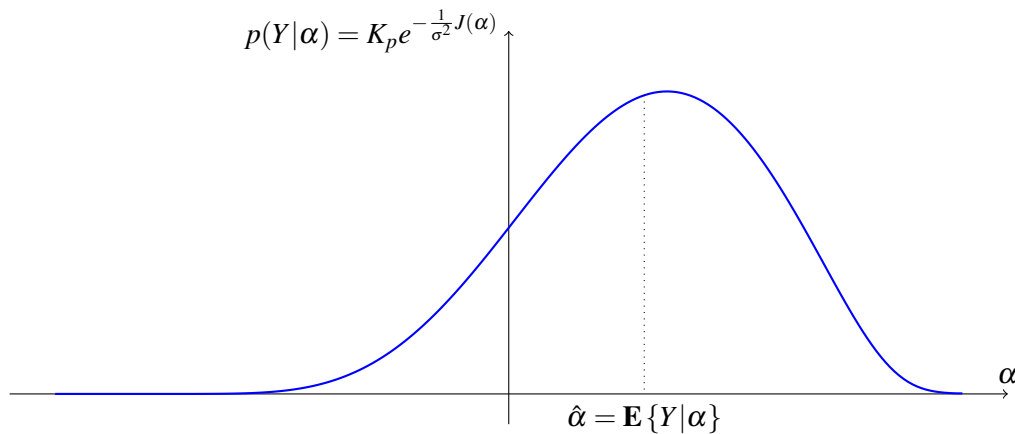


Figura 6.6: Função $p(Y|\alpha)$, para um conjunto de dados Y com $n_x = 1$.

por isso, não se pode garantir que o algoritmo proposto neste capítulo convirja para $\hat{\alpha}$, ou mesmo para $\alpha_{max} = \max\{p(Y|\alpha)\}$. No entanto, se σ^2 diminuir, a função $p(Y|\alpha)$ comprime-se em torno do seu máximo e $\hat{\alpha}$ aproxima-se de α_{max} (6.7).

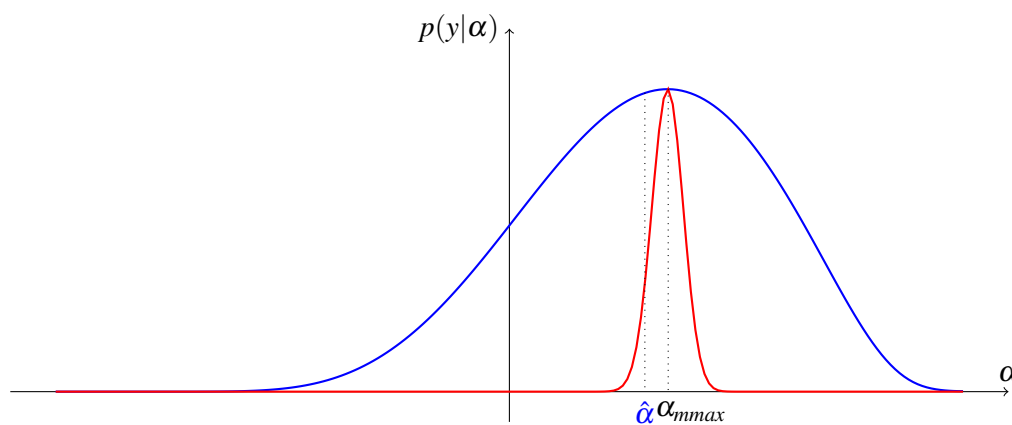


Figura 6.7: Funções $p(Y|\alpha)$, para $n_x = 1$ com diferentes valores de σ^2 .

⁵Como $p(Y|\alpha) = K_p e^{-\frac{1}{2\sigma^2} J(\alpha)}$ então, se $J(\alpha) = C \Rightarrow p(Y|\alpha) = K_p e^{-\frac{C}{2\sigma^2}} = \text{constante}$, pelo que as superfícies de nível da função $p(Y|\alpha)$ também são superfícies de nível $J(\alpha)$.

Comparam-se duas curvas $p(Y|\alpha)$ para $n_x = 1$ com diferentes valores de σ^2 . Observa-se que o baricentro da curva com maior valor de σ^2 , isto é, o da curva a vermelho, é praticamente coincidente com o seu máximo. Assim, uma forma de se forçar o algoritmo a convergir para o máximo é utilizar uma função $p(Y|\alpha)$ com um valor de σ^2 igual ao quadrado da norma quadrática induzida da matriz de covariância P_i . Como esta matriz tende para zero, σ^2 também tende para zero. Por sua vez, a função $p(Y|\alpha)$ tende para um impulso de Dirac quando $\alpha = \alpha_{max}$, pelo que o algoritmo converge para α_{max} . Tudo isto equivale a utilizar uma função peso diferente em cada iteração i , dada por

$$W_i(\alpha) = e^{-\frac{1}{\sigma_i^2} J(\alpha)}, \quad (6.22)$$

com $\sigma_i = \|P_{i-1}\|_2^2$, e em que $\|P_{i-1}\|_2$ é a norma quadrática induzida da matriz P_{i-1} ⁶.

Por último, para assegurar que a função peso toma valores próximos de 1, normaliza-se o seu valor, de forma a que na iteração i , o baricentro determinado na iteração anterior tenha um peso unitário. De acordo com esta normalização, a função peso passa a ser definida como

$$\bar{W}_i = e^{-\frac{1}{\sigma_i^2} [J(\alpha) - J(\hat{\alpha}_{i-1})]}. \quad (6.23)$$

6.2.2.2 Penalização das "Curiosidades Instáveis"

Exige-se que o predictor da parametrização MOLI seja um sistema estável. Logo, o seu polinómio característico tem também de ser "estável", isto é, tem de ter todas as suas raízes no interior do círculo unitário. Deste modo, restringe-se o conjunto admissível de pontos α aos pontos que correspondem a polinómios estáveis. Para assegurar que os baricentros $\hat{\alpha}_i$ se mantêm no conjunto de pontos admissíveis, as "curiosidades" que correspondem a polinómios instáveis, ou seja "curiosidades instáveis", são penalizadas com um custo muito grande:

$$J(\alpha_i) = e^{12}, \quad (6.24)$$

o que confere um valor de peso muito pequeno no cálculo do baricentro.

6.2.2.3 Deslocação da Elipsoide

O baricentro calculado em cada iteração é geralmente a "curiosidade" com menor valor de custo J , isto é, é a "curiosidade" que mais se aproxima do valor ótimo. No entanto, se esta condição não se verificar, percebe-se que a elipsoide que se gerou não circunda o valor ótimo a que se pretende chegar. Daí se necessitar de deslocar a elipsoide na direção da "curiosidade" de custo mínimo, tal como exemplificado na Figura (6.8).

⁶A norma quadrática induzida de uma matriz M genérica é dada por $\|M\|_2$ e é igual ao seu maior valor singular.

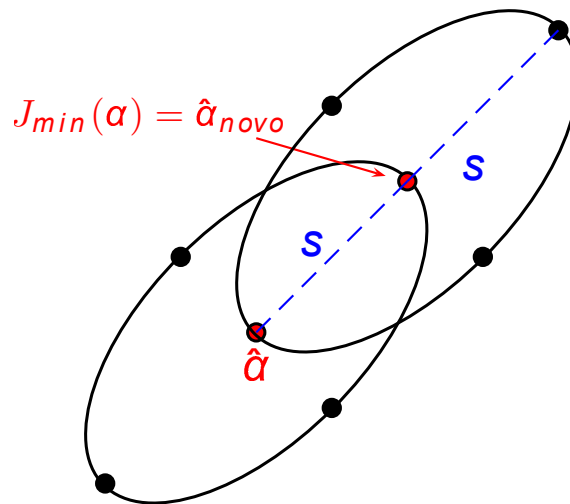


Figura 6.8: Elipsóide de ordem 2 a ser deslocada.

6.2.2.4 Convergência

Ao longo da execução do algoritmo, espera-se que o valor da matriz de covariância P_i decresça e portanto que a sua norma $\|P_i\|_2$ e o valor do parâmetro $s = \sqrt{\|P_i\|_2}$ sejam cada vez menores, tal como se mostra nas Figuras 6.9 a 6.14. Deste modo, considera-se que o algoritmo converge quando esse parâmetro s tem um valor muito pequeno:

$$s < e^{-4}. \quad (6.25)$$

É também importante referir que a convergência do algoritmo só é garantida quando a função a otimizar é uma função convexa, de forma a que o algoritmo consiga convergir para o ponto mínimo (6.14).

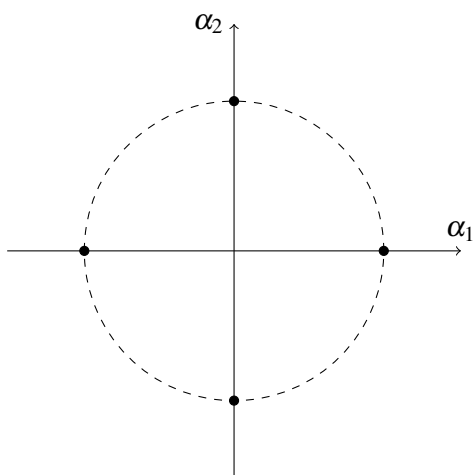


Figura 6.9: MOLIZOE: iteração 0.

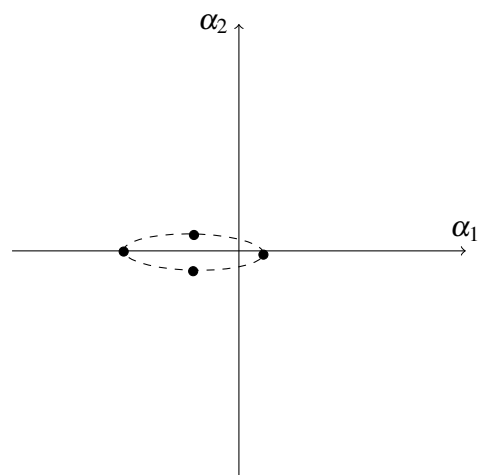


Figura 6.10: MOLIZOE: iteração 1.

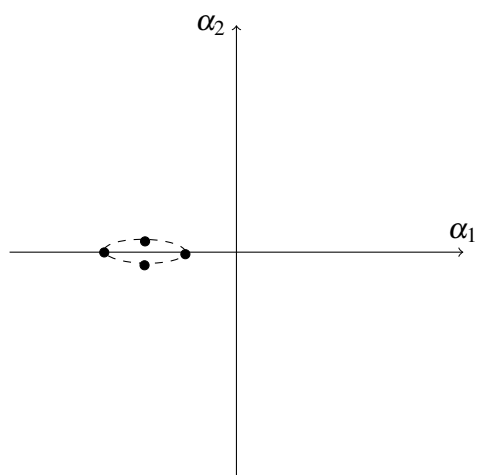


Figura 6.11: MOLIZOE: iteração 4.

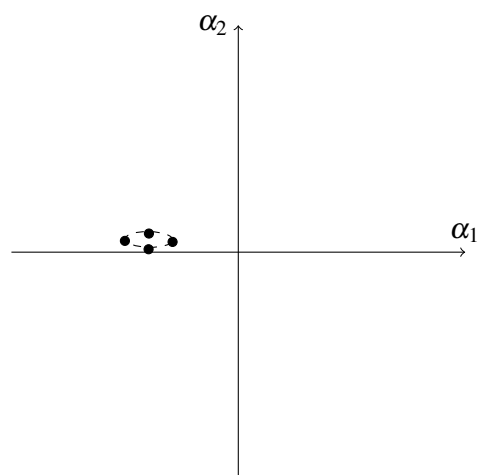


Figura 6.12: MOLIZOE: iteração 6.

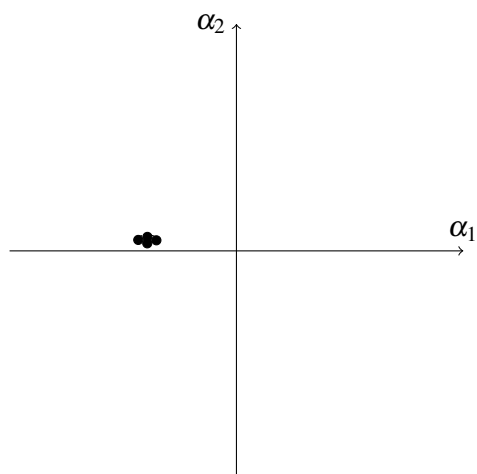


Figura 6.13: MOLIZOE: iteração 8.

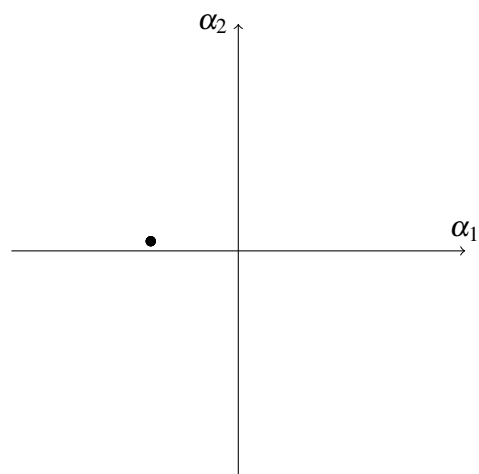


Figura 6.14: MOLIZOE: iteração 25.

Capítulo 7

Estudo de Caso

7.1 Formulação

Neste capítulo, testam-se os algoritmos de identificação propostos nesta dissertação e comparam-se os seus resultados com alguns algoritmos já apresentados no capítulo 3, nomeadamente o *ARX MOLI*, o *Output MOLI*, o *PEM MOLI* e o algoritmo *Barycenter MOLI ZOFT* com inicialização aleatória e politópica. Note-se que o *ARX MOLI* e o *PEM MOLI* são idênticos aos algoritmos executados pelos comandos "*arx*" e "*armax*" do MATLAB, que minimizam a soma dos quadrados dos erros de previsão de modelos ARX e ARMAX, respetivamente. Já o *Output MOLI* é um algoritmo de erro de saída equivalente a vários outros algoritmos como, por exemplo, o proposto por Landau em 1979 [52] e o que está implementado no comando "*oe*" do MATLAB. Estes são algoritmos de referência na área de Identificação de Sistemas e têm implementações diferentes.

Todos os algoritmos mencionados foram utilizados na identificação do sistema de 2ª ordem descrito pela seguinte realização no espaço de estados:

$$\begin{aligned}x(k+1) &= \begin{bmatrix} 0 & -\alpha_0 \\ 1 & -\alpha_1 \end{bmatrix} x(k) + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(k) + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} e(k) \text{ ,} \\ y(k) &= \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) + e(k)\end{aligned}$$

em que $e(k)$ é ruído branco gaussiano de média nula, $\alpha_0 = 0.7$, $\alpha_1 = -1.5$, $b_1 = -0.5$, $b_2 = 1$, $k_1 = -0.6$ e $k_2 = 1$. Este sistema foi identificado em três simulações de Monte Carlo (MC) com relações de sinal-ruído (*Sinal to Noise Ratio* - SNR) de 20dB, de 10dB e de 0dB, tendo, cada uma dessas simulações, 100 experiências de identificação. Utilizou-se ainda uma sequência binária aleatória como sinal de entrada e um conjunto de dados composto por 300 pares de entrada-saída.

Nestas simulações, ilustrou-se o desempenho dos algoritmos de identificação em situações de ruído moderado ($SNR = 20\text{dB}$), intenso ($SNR = 10\text{dB}$) e muito intenso ($SNR = 0\text{dB}$), onde a potência do sinal de ruído é igual à potência do sinal de saída.

7.2 Considerações

Os resultados das simulações MC estão apresentados em três tabelas, cada uma com um valor diferente de SNR: Tabela 7.1 com SNR = 20dB, Tabela 7.2 com SNR = 10dB e Tabela 7.3 com SNR = 0dB. Cada tabela é formada por 7 colunas. A primeira indica o algoritmo de identificação testado e as restantes mostram o valor médio e o desvio padrão - este último apresentado entre parêntesis - das estimativas dos parâmetros obtidas na respetiva simulação MC. A primeira linha de cada tabela apresenta os valores verdadeiros dos parâmetros estimados.

ITSimple, *ITRandom* e *ITPoly* referem-se ao algoritmo do Baricentro Iterativo com inicialização simples, aleatória e politópica, respetivamente. Por outro lado, *RandMZOFT* e *PolymZOFT* apontam para o algoritmo não iterativo *Barycenter MOLI-ZOFT* com inicialização aleatória e politópica. Por seu turno, *ARXMOLI*, *OutputMOLI* e *PemMOLI* são os algoritmos *ARX MOLI*, *Output MOLI* e *PEM MOLI*, mencionados no capítulo 3. Já *ITModified* corresponde ao algoritmo do Baricentro Iterativo Modificado, introduzido no capítulo 5. Na última linha da tabela, encontra-se ainda o algoritmo *MOLIZOE*, apresentado no capítulo 6.

7.3 Análise de Resultados

Uma das primeiras conclusões que se pode retirar dos resultados das simulações MC é que o Baricentro Iterativo Simples (*ITSimple*) apresenta estimativas muito razoáveis, com níveis de exatidão superiores às estimativas do *Output MOLI* e do *ARX MOLI*. Porém, foram necessárias cerca de 3000 iterações para o algoritmo convergir. Além do mais, a utilização de um Fator de Esquecimento ρ apenas permitiu melhorar ligeiramente as estimativas de k do algoritmo *ITSimple*. Contudo, ρ aumentou significativamente a velocidade de convergência dos algoritmos baseados no Baricentro Iterativo, o que levou à diminuição dos seus tempos de execução [53]. Por outro lado, concluiu-se também que as versões aleatória e politópica do algoritmo do Baricentro Iterativo não conseguiram melhorar as estimativas das suas versões não iterativas. De facto, os algoritmos *ITRandom* e *ITPoly* não melhoraram os resultados dos algoritmos *RandMZOFT* e *PolymZOFT*. Assim sendo, não há vantagem em utilizar estas versões do Baricentro Iterativo para refinar estimativas de outros algoritmos.

Uma vez que a formulação do algoritmo do Baricentro Iterativo Modificado (*ITModified*) foi baseada na filosofia dos estimadores bayesianos, esperava-se que este algoritmo produzisse melhores resultados. Como essa melhoria de resultados não se verificou, é necessário um estudo mais aprofundado em torno do comportamento do algoritmo, não só para se averiguar o que falhou em concreto, mas também para se conseguir melhorar o seu desempenho.

Neste estudo de caso, o erro de previsão $e(k)$ é ruído branco gaussiano. Segundo estas condições, os algoritmos *PEM MOLI* e *MOLIZOE* são Estimadores de Máxima Verosimilhança e, por esse motivo, esperava-se que gerassem os melhores resultados. Porém, nas três simulações MC, as melhores estimativas foram produzidas pelo algoritmo *PolymZOFT*. É na Tabela 7.3 que estão resumidos os resultados das simulações onde o ruído é mais elevado (SNR = 0dB). Nesta tabela

verifica-se uma diferença mais significativa na melhoria das estimativas geradas pelo algoritmo *PolyMZOFT*. Também é notório que o algoritmo *ARX MOLI*, equivalente ao comando "*arx*" do MATLAB, tem os piores resultados, visto que o ruído de equação do sistema não é ruído branco.

Dado que o algoritmo *MOLIZOE* e o algoritmo *PEM MOLI* minimizam a mesma função custo, espera-se que estes produzam precisamente os mesmos resultados. De facto, os dois algoritmos apresentaram estimativas equivalentes em todos os parâmetros identificados, o que veio, por seu turno, mostrar que ambos conseguiram sempre chegar ao mínimo dessa função custo. Porém, embora o *MOLIZOE* necessite de cerca do dobro das iterações para convergir, os tempos de execução dos dois algoritmos foram equivalentes. Na verdade, as iterações do *MOLIZOE* são muito menos pesadas em termos computacionais do que as do *PEM MOLI*, o que se traduziu num menor tempo de execução do algoritmo. O *MOLIZOE* é também ainda um algoritmo mais adaptável a novas situações, pois, em cada iteração, este algoritmo não necessita de calcular nenhuma derivada, tendo apenas de determinar o valor da função custo de um número limitado de "curiosidades".

Concluiu-se, então, que a implementação simples do Baricentro Iterativo gerou resultados aceitáveis. No entanto, a sua velocidade de convergência é bastante baixa, sendo necessárias cerca de 3000 iterações para que o algoritmo atinja o valor ótimo. No que se prende com o desempenho deste algoritmo, inicializado aleatoriamente ou sob a forma de um politopo, constatou-se que as estimativas que se geraram não foram melhores que as estimativas da sua versão não iterativa, *Barycenter MOLI-ZOFT*. Assim, entende-se que é essencial um estudo mais aprofundado do método de otimização do Baricentro Discreto, ajustando-se e determinando-se melhores valores para os parâmetros de *design* deste método. É também fundamental compreender-se quais as limitações de convergência do Baricentro, uma vez que este método está na base de todos os algoritmos que foram propostos.

Tabela 7.1: Resultados: SNR = 20 dB.

True Values		α_0	α_1	b_1	b_2	k_1	k_2
Algorithm	ρ	0.7000	-1.5000	-0.5000	1.0000	-0.6000	1.0000
ItSimple	1	0.7040 (0.0070)	-1.5066 (0.0093)	-0.5091 (0.0132)	1.0011 (0.0067)	-0.3958 (0.1815)	0.5486 (0.2823)
	0.99	0.7030 (0.0076)	-1.5048 (0.0098)	-0.5062 (0.0126)	1.0009 (0.0057)	-0.4250 (0.1756)	0.6460 (0.2815)
	0.9	0.7033 (0.0074)	-1.5053 (0.0095)	-0.5067 (0.0127)	1.0003 (0.0060)	-0.4102 (0.1966)	0.6178 (0.2973)
ItRandom	1	0.7034 (0.0070)	-1.5057 (0.0089)	-0.5070 (0.0109)	1.0008 (0.0050)	-0.5274 (0.0626)	0.7014 (0.1388)
	0.99	0.7033 (0.0069)	-1.5054 (0.0088)	-0.5064 (0.0106)	1.0006 (0.0051)	-0.5142 (0.0693)	0.7085 (0.1341)
	0.9	0.7035 (0.0071)	-1.5058 (0.0092)	-0.5073 (0.0117)	1.0006 (0.0054)	-0.5160 (0.0920)	0.6944 (0.1592)
ItPoly	1	0.6994 (0.0070)	-1.4991 (0.0087)	-0.4984 (0.0103)	0.9999 (0.0050)	-0.5121 (0.0565)	0.9628 (0.0584)
	0.99	0.6994 (0.0068)	-1.4991 (0.0084)	-0.4983 (0.0100)	0.9999 (0.0051)	-0.5211 (0.0511)	0.9699 (0.0619)
	0.9	0.6994 (0.0069)	-1.4992 (0.0086)	-0.4985 (0.0103)	0.9999 (0.0103)	-0.5180 (0.0591)	0.9713 (0.0683)
ItModified		0.6879 (0.0082)	-1.4849 (0.0103)	-0.4844 (0.0119)	0.9991 (0.0055)	-0.6971 (0.1139)	1.4041 (0.1411)
RandMZOFT		0.7037 (0.0071)	-1.5065 (0.0093)	-0.5082 (0.0114)	1.0004 (0.0049)	0.5509 (0.0620)	0.6949 (0.1695)
PolyMZOFT		0.6994 (0.0069)	-1.4992 (0.0085)	-0.4984 (0.0101)	0.9999 (0.0051)	-0.5185 (0.0400)	0.9693 (0.0488)
ARXMOLI		0.6848 (0.0073)	-1.4809 (0.0089)	-0.4805 (0.0108)	0.9991 (0.0057)	-0.6848 (0.0073)	1.4809 (0.0089)
OutputMOLI		0.7066 (0.0074)	-1.5112 (0.0101)	-0.5190 (0.0174)	1.0024 (0.0103)	0 (0)	0 (0)
PemMOLI		0.7000 (0.0069)	-1.5003 (0.0085)	-0.4997 (0.0100)	0.9999 (0.0049)	-0.6159 (0.0542)	0.9984 (0.0564)
MOLIZOE		0.7000 (0.0069)	-1.5003 (0.0085)	-0.4997 (0.0100)	0.9999 (0.0049)	-0.6159 (0.0542)	0.9984 (0.0564)

Tabela 7.2: Resultados: SNR = 10 dB.

True Values		α_0	α_1	b_1	b_2	k_1	k_2
		0.7000	-1.5000	-0.5000	1.0000	-0.6000	1.0000
Algorithm	ρ						
ItSimple	1	0.7257 (0.0222)	-1.5415 (0.0343)	-0.5619 (0.0565)	1.1984 (0.0235)	-0.4528 (0.1416)	0.6275 (0.2347)
	0.99	0.7202 (0.0256)	-1.5314 (0.0371)	-0.5459 (0.0559)	1.0050 (0.0210)	-0.4645 (0.1540)	0.6886 (0.2449)
	0.9	0.7239 (0.0258)	-1.5381 (0.0380)	-0.5574 (0.0595)	1.0070 (0.0217)	-0.4494 (0.1676)	0.6424 (0.2426)
ItRandom	1	0.7266 (0.0187)	-1.5432 (0.0268)	-0.5609 (0.0429)	1.0067 (0.0205)	-0.5286 (0.0613)	0.6904 (0.1127)
	0.99	0.7223 (0.0204)	-1.5353 (0.0279)	-0.5488 (0.0418)	1.0048 (0.0192)	-0.5428 (0.0716)	0.7440 (0.1287)
	0.9	0.7231 (0.0203)	-1.5372 (0.0302)	-0.5524 (0.0445)	1.0052 (0.0197)	-0.5260 (0.0957)	0.7196 (0.1483)
ItPoly	1	0.6997 (0.0209)	-1.4994 (0.0277)	-0.5012 (0.0376)	0.9991 (0.0178)	-0.5223 (0.0517)	0.9328 (0.0528)
	0.99	0.7003 (0.0201)	-1.5005 (0.0265)	-0.5023 (0.0366)	0.9991 (0.0178)	-0.5312 (0.0563)	0.9334 (0.0544)
	0.9	0.7000 (0.0210)	-1.5000 (0.0280)	-0.5017 (0.0383)	0.9993 (0.0181)	-0.5320 (0.0596)	0.9342 (0.0530)
ItModified		0.6785 (0.0397)	-1.4736 (0.0523)	-0.4750 (0.0592)	0.9964 (0.0189)	-0.5516 (0.1764)	1.0424 (0.1570)
RandMZOFT		0.7278 (0.0185)	-1.5480 (0.0279)	-0.5705 (0.0458)	1.0084 (0.0221)	-0.5536 (0.0605)	0.6886 (0.1416)
PolyMZOFT		0.7015 (0.0188)	-1.5016 (0.0249)	-0.5031 (0.0350)	0.9992 (0.0177)	0.5178 (0.0359)	0.9187 (0.0385)
ARXMOLI		0.5943 (0.0263)	-1.3682 (0.0336)	-0.3702 (0.0422)	0.9951 (0.0169)	-0.5943 (0.0263)	1.3682 (0.0336)
OutputMOLI		0.7483 (0.0161)	-1.5793 (0.0220)	-0.6249 (0.0569)	1.0001 (0.0369)	0 (0)	0 (0)
PemMOLI		0.6992 (0.0210)	-1.5002 (0.0284)	-0.5024 (0.0380)	0.9993 (0.0176)	-0.6014 (0.0669)	0.9806 (0.0627)
MOLIZOE		0.6992 (0.0210)	-1.5002 (0.0284)	-0.5024 (0.0380)	0.9993 (0.0176)	-0.6014 (0.0669)	0.9806 (0.0627)

Tabela 7.3: Resultados: SNR = 0 dB.

True Values		α_0	α_1	b_1	b_2	k_1	k_2
Algorithm	ρ	-0.7000	-1.5000	-0.5000	1.0000	-0.6000	1.0000
ItSimple	1	0.7791 (0.0578)	-1.6199 (0.0804)	-0.6751 (0.1309)	1.0209 (0.0587)	-0.5483 (0.1465)	0.7288 (0.2175)
	0.99	0.7726 (0.0563)	-1.6040 (0.0756)	-0.6474 (0.1153)	1.0164 (0.0609)	-0.5481 (0.1319)	0.7573 (0.1972)
	0.9	0.7769 (0.0556)	-1.6189 (0.0758)	-0.6771 (0.1363)	1.0199 (0.0647)	-0.5639 (0.1609)	0.7445 (0.1921)
ItRandom	1	0.7816 (0.0370)	-1.6232 (0.0584)	-0.6751 (0.0953)	1.0236 (0.0595)	-0.5763 (0.0670)	0.7592 (0.1325)
	0.99	0.7719 (0.0436)	-1.6044 (0.0604)	-0.6473 (0.0958)	1.0207 (0.0579)	-0.5752 (0.0725)	0.7879 (0.1310)
	0.9	0.7672 (0.0505)	-1.5992 (0.0710)	-0.6411 (0.1223)	1.0185 (0.0601)	-0.5715 (0.0956)	0.7921 (0.1353)
ItPoly	1	0.7198 (0.0326)	-1.5262 (0.0407)	-0.5435 (0.0826)	1.0162 (0.0553)	-0.5889 (0.0600)	0.9299 (0.0422)
	0.99	0.7190 (0.0344)	-1.5255 (0.0434)	-0.5425 (0.0918)	1.0156 (0.0555)	-0.5890 (0.0617)	0.9321 (0.0451)
	0.9	0.7180 (0.0342)	-1.5247 (0.0441)	-0.5423 (0.0816)	1.0164 (0.0554)	-0.5901 (0.0672)	0.9346 (0.0512)
ItModified		0.6670 (0.0775)	-1.4594 (0.1060)	-0.4703 (0.1375)	1.0148 (0.0573)	-0.5619 (0.1438)	0.9930 (0.0788)
RandMZOFT		0.7883 (0.0410)	-1.6379 (0.0628)	-0.7016 (0.1108)	1.0263 (0.0593)	-0.5901 (0.0509)	0.7438 (0.1198)
PolyMZOFT		0.7367 (0.0241)	-1.5410 (0.0249)	-0.5555 (0.0728)	1.0142 (0.0564)	-0.5309 (0.0360)	0.8665 (0.0419)
ARXMOLI		0.4065 (0.0574)	-1.1511 (0.0622)	-0.1563 (0.0866)	1.0193 (0.0573)	-0.4065 (0.0574)	1.1511 (0.0622)
OutputMOLI		0.9302 (0.0437)	-1.8044 (0.0492)	-0.4361 (0.2275)	0.5128 (0.2652)	0 (0)	0 (0)
PemMOLI		0.6937 (0.0477)	-1.4929 (0.0644)	-0.5054 (0.1077)	1.0159 (0.0564)	-0.5882 (0.0892)	0.9754 (0.0596)
MOLIZOE		0.6937 (0.0477)	-1.4929 (0.0644)	-0.5053 (0.1077)	1.0159 (0.0564)	-0.5882 (0.0892)	0.9754 (0.0596)

Capítulo 8

Conclusões

8.1 Contribuições

Nesta dissertação, propuseram-se novos algoritmos de identificação para sistemas MISO LTI baseados na parametrização MOLI, em que o processo de otimização dos parâmetros é desempenhado por um método sem derivadas.

O algoritmo do Baricentro Iterativo, formulado no capítulo 4, consiste numa abordagem recursiva do método do Baricentro, tendo-se proposto três inicializações para este algoritmo que diferem na forma como o baricentro inicial das "curiosidades" é gerado: inicialização simples, inicialização aleatória e inicialização politópica. A esta proposta acrescentou-se ainda uma abordagem bayesiana do Baricentro, o que surge também, nesta dissertação, como uma das contribuições mais significativas. Demonstrou-se que em certas condições o cálculo de um baricentro é equivalente a um Estimador de Bayes. Este estimador tem em conta a informação *à priori* do sistema, já que é um estimador de média condicionada. Consequentemente, formularam-se novos algoritmos de identificação que assentam nesta interpretação bayesiana. Surgiu, por um lado, o algoritmo do Baricentro Iterativo Modificado que encara cada iteração como uma estimativa de Bayes, a qual vai tendo cada vez mais informação sobre o sistema a estimar. Formulou-se ainda o algoritmo das Elipses, *MOLIZOE*, que, em cada iteração, determina o baricentro de um conjunto de "curiosidades", dispostas simetricamente na superfície de uma elipsoide. O algoritmo aproxima-se, assim, do baricentro ótimo. Contudo, apenas consegue determina-lo se este for coincidente com o máximo da função de densidade de probabilidade *à posteriori*. Como esta não é uma função simétrica em torno do seu máximo, a convergência para o baricentro não é assegurada. Procura-se, então, que, em vez do baricentro, o algoritmo determine o máximo dessa função. Comprime-se, então, a função em torno desse ponto máximo à medida que as elipses se vão estreitando e aproximando do valor ótimo. De facto, se a largura da função em torno do máximo tender para zero, o baricentro tende para o máximo e o algoritmo consegue convergir. Ao maximizar a função de densidade de probabilidade *à posteriori*, também se minimiza a soma dos quadrados dos erros de previsão, o que corresponde a uma abordagem de Mínimos Quadrados Separáveis. O *MOLIZOE* é, portanto, um método equivalente aos métodos PEM, diferindo apenas na metodologia de otimização da função

custo. Enquanto que o *MOLIZOE* utiliza um método sem derivadas para minimizar essa função custo, os métodos PEM utilizam métodos de gradiente, tais como o de Gauss-Newton ou o de Levenberg–Marquardt.

No estudo de caso apresentado no capítulo 7, que consistiu em três simulações MC de experiências de identificação de um sistema de ordem 2, com SNR = 20dB, SNR = 10dB e SNR = 0dB, foram testados e comparados algoritmos de identificação com outros já existentes. Concluiu-se que a implementação simples do Baricentro Iterativo gerou resultados aceitáveis. No entanto, a sua velocidade de convergência foi bastante baixa, tendo sido necessárias cerca de 3000 iterações para que o algoritmo atingisse o valor ótimo. Por sua vez, no que se prende com o desempenho deste algoritmo inicializado aleatoriamente ou sob a forma de um polítopo, constatou-se que as estimativas que se geraram não foram melhores que as estimativas das versões não iterativas do algoritmo *Barycenter MOLI-ZOFT*, descritas no capítulo 3. Ainda no contexto dos algoritmos de baricentro iterativo, a utilização de um Fator de Esquecimento não veio melhorar a precisão das estimativas. Contudo, esse Fator de Esquecimento diminuiu significativamente o número de iterações necessárias para a convergência dos algoritmos, o que constitui uma solução interessante no sentido da diminuição do tempo de execução dos algoritmos. Concluiu-se também que relativamente ao Baricentro Iterativo Modificado, este não produziu melhores estimativas do que as do Baricentro Iterativo Simples, não sendo seguramente este o resultado esperado.

Como era expectável, os resultados do *MOLIZOE* foram exatamente iguais aos do *PEM MOLI*, uma vez que, como já se referiu anteriormente, estes dois algoritmos minimizam a mesma função custo, utilizando cada um uma abordagem de Mínimos Quadrados Separáveis. Contudo, para se determinar o mínimo da função custo, enquanto que o *PEM MOLI* utiliza um método de gradiente, mais concretamente o método de Gauss-Newton, o *MOLIZOE* utiliza um método sem gradiente: o método do Baricentro. Apesar do *MOLIZOE* necessitar de mais iterações para chegar ao valor ótimo, os tempos de execução dos dois algoritmos foram equivalentes, uma vez que as iterações do *MOLIZOE* são muito mais leves em termos computacionais do que as do *PEM MOLI*. Nas condições da simulação MC, verifica-se que ambos os algoritmos se comportam como Estimadores de Máxima Verosimilhança, o que confirma as expectativas de que estes algoritmos produziram as estimativas mais exatas. No entanto, o *MOLIZOE* e o *PEM MOLI* foram suplantados pelo *Polytopic Barycenter MOLI-ZOFT*, um algoritmo de baricentro não iterativo que utiliza um reduzido número de "curiosidades". Estas "curiosidades" estão nos vértices de um polítopo e correspondem a polinómios característicos com raízes reais e complexas.

8.2 Trabalho Futuro

Como já foi referido, um estudo mais aprofundado do método do Baricentro traria uma melhor compreensão do funcionamento deste algoritmo de otimização, quando aplicado no contexto da Identificação de Sistemas. Essa compreensão poderia levar a perceber-se melhor o porquê do algoritmo *Barycenter MOLI-ZOFT* com inicialização politópica ter gerado as estimativas mais exatas no estudo de caso apresentado. Por exemplo, poder-se-ia compreender se a qualidade

dessas estimativas foi influenciada pelo valor da função peso ou pelo processo de geração das "curiosidades" iniciais. Poder-se-ia também, assim, questionar qual a melhor forma de gerar essas "curiosidades".

No que se prende com a interpretação bayesiana do Baricentro, esta ajudou a compreender melhor o funcionamento do método. Porém, a interpretação que se fez não se traduziu ainda num estimador bayesiano ótimo. No entanto, deu origem ao algoritmo das elipses que, não sendo um estimador bayesiano, é um estimador ótimo da família dos estimadores PEM, tendo a vantagem de não utilizar derivadas. Por seu turno, o algoritmo das elipses, o *MOLIZOE*, é ainda muito parecido com o UKF, um algoritmo utilizado para estimar o estado de sistemas não lineares [54]. Uma vez que já existem muitos resultados sobre o UKF, seria muito interessante ver se estes podem ser aplicados ao *MOLIZOE*.

No contexto desta dissertação, além do estudo de caso, testaram-se ainda os algoritmos de identificação num sistema mais complexo, não tendo nenhum algoritmo apresentado resultados satisfatórios. Esses resultados insatisfatórios devem-se ao facto de a largura das elipses ter convergido muito rapidamente para valores muito pequenos, o que levou a que o algoritmo encalhe numa zona não convexa da função custo. No entanto, uma inicialização adequada conseguiu conduzir o algoritmo para o mínimo. Seria, então, importante definir qual a largura ótima das elipses, permitindo que estas não só se contraíam, mas também se expandam, de acordo com critérios objetivos. Também os problemas de inicialização do baricentro e da largura inicial das elipses devem ser objeto de estudo, no sentido de se compreender se há alguma forma de garantir a convergência para o mínimo.

No que diz respeito ao ruído inerente ao sistema, e no contexto do previsor formulado para estimação dos parâmetros do modelo, um estimador ótimo deve garantir que esse erro do previsor MOLI é ruído branco. Pode-se, assim, pensar em determinar um estimador ótimo através do branqueamento dos resíduos do previsor, onde as técnicas de subespaço de estados, que resolvem o problema da realização estocástica, podem dar o seu contributo.

Em suma, os algoritmos que foram desenvolvidos nesta dissertação identificam sistemas MISO LTI. Estes algoritmos poderão também ser utilizados em sistemas com múltiplas entradas e múltiplas saídas, isto é, em sistemas MIMO. Por sua vez, e com base nos resultados obtidos no estudo de caso, seria igualmente interessante utilizar estes algoritmos de identificação em sistemas Lineares de Parâmetros Variantes (*Linear Parameter-Varying* - LPV) MISO e MIMO.

Referências

- [1] Rodrigo Alvite Romano, Felipe Pait, e Rafael Corsi Ferrão. Matchable-observable linear models for multivariable identification: Structure selection and experimental results. Em Decision and Control (CDC), 2015 IEEE 54th Annual Conference on, páginas 3391–3396. IEEE, 2015.
- [2] A Stephen Morse e Felipe M Pait. MIMO design models and internal regulators for cyclicly switched parameter-adaptive control systems. IEEE Transactions on Automatic Control, 39(9):1809–1818, 1994.
- [3] Rodrigo A Romano, Paulo Lopes dos Santos, Felipe Pait, TP Perdicoúlis, e José A Ramos. Machine learning barycenter approach to identifying lpv state-space models. Em American Control Conference (ACC), 2016, páginas 6351–6356. IEEE, 2016.
- [4] P Lopes dos Santos, R Romano, TP Azevedo-Perdicoúlis, Daniel E Rivera, e JA Ramos. Lpv system identification using the matchable observable linear identification approach. Em Decision and Control (CDC), 2017 IEEE 56th Annual Conference on, páginas 4626–4631. IEEE, 2017.
- [5] P. Lopes dos Santos, M.T. Freigoun, D.E. Rivera, E.B. Hekler, C.A. Martín, R. Romano, T-P Perdicoúlis, e J.A. Ramos. A molizoft system identification approach of the just walk data. IFAC-PapersOnLine, 50(1):12508 – 12513, 2017. URL: <http://www.sciencedirect.com/science/article/pii/S2405896317327027>, doi:<https://doi.org/10.1016/j.ifacol.2017.08.2060>.
- [6] Mohammad T Freigoun, César A Martín, Alicia B Magann, Daniel E Rivera, Sayali S Phatak, Elizabeth V Korinek, e Eric B Hekler. System identification of just walk: A behavioral mhealth intervention for promoting physical activity. Em American Control Conference (ACC), 2017, páginas 116–121. IEEE, 2017.
- [7] Carl Friedrich Gauss. Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss. sumtibus Frid. Perthes et IH Besser, 1809.
- [8] R.A. Fisher. On an absolute criterion for fitting frequency curves. Messenger of Mathematics, 41:155–160, 1912.
- [9] Effective construction of linear state-variable models from input/output functions. at-Automatisierungstechnik, 14(1-12):545–548, 1966.
- [10] Karl-Johan Åström e Bohlin Torsten. Numerical identification of linear dynamic systems from normal operating records. IFAC Proceedings Volumes, 2(2):96–111, 1965.
- [11] L.S. de Jong. Numerical aspects of realization algorithms in linear systems theory. Tese de doutoramento, LS de Jong, 1975.

- [12] Lieuwe Setse L.S. de Jong. Numerical aspects of recursive realization algorithms. SIAM Journal on Control and optimization, 16(4):646–659, 1978.
- [13] Hirotugu Akaike. Stochastic theory of minimal realization. IEEE Transactions on Automatic Control, 19(6):667–674, 1974.
- [14] Pierre Faurre. Realisations markoviennes de processus stationnaires (markov realizations of stationary processes). Relatório técnico, INSTITUT DE RECHERCHE D’INFORMATIQUE ET D’AUTOMATIQUE LE CHESNAY (FRANCE), 1973.
- [15] P Faurre. Stochastic realization algorithms. system identification: Advances and case studies. eds. mehra r. and lainiotis d, 1976.
- [16] Peter Van Overschee e Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, 30(1):75–93, 1994.
- [17] Peter Van Overschee e BL De Moor. Subspace identification for linear systems: Theory—Implementation—Applications. Springer Science & Business Media, 2012.
- [18] Wallace E Larimore. Canonical variate analysis in identification, filtering, and adaptive control. Em Decision and Control, 1990., Proceedings of the 29th IEEE Conference on, páginas 596–604. IEEE, 1990.
- [19] Michel Verhaegen. Identification of the deterministic part of mimo state space models given in innovations form from input-output data. Automatica, 30(1):61–74, 1994.
- [20] Tohru Katayama. Subspace methods for system identification. Springer Science & Business Media, 2006.
- [21] Guillaume Mercère e Laurent Bako. Parameterization and identification of multivariable state-space systems: A canonical approach. Automatica, 47(8):1547–1555, 2011.
- [22] Petre Stoica e Magnus Jansson. MIMO system identification: State-space and subspace approximations versus transfer function and instrumental variables. IEEE Transactions on Signal Processing, 48(11):3087–3099, 2000.
- [23] Tomas McKelvey. Identification of state-space models from time and frequency data. Department of Electrical Engineering, Linköping University, 1995.
- [24] Brett Ninness, Adrian Wills, e Adam Mills. Unit: A freely available system identification toolbox. Control Engineering Practice, 21(5):631–644, 2013.
- [25] Lennart Ljung. Prediction error estimation methods. Circuits, Systems and Signal Processing, 21(1):11–21, 2002.
- [26] Konstantinos G Derpanis. Cramer-rao bound. 2006.
- [27] Lennart Ljung. Convergence analysis of parametric identification methods. IEEE transactions on automatic control, 23(5):770–783, 1978.
- [28] Lennart Ljung et al. Theory for the user. Prentice Hall, 1987.
- [29] Giorgio Picci. Some numerical aspects of multivariable systems identification. Em Algorithms and Theory in Filtering and Control, páginas 76–101. Springer, 1982.

- [30] Wallace E Larimore. System identification, reduced-order filtering and modeling via canonical variate analysis. Em American Control Conference, 1983, páginas 445–451. IEEE, 1983.
- [31] Wallace E Larimore. The adaptx software for automated and real-time multivariable system identification. IFAC Proceedings Volumes, 36(16):1453–1458, 2003.
- [32] Niek H Bergboer, Vincent Verdult, e Michel HG Verhaegen. An efficient implementation of maximum likelihood identification of lti state-space models by local gradient search. Em Decision and Control, 2002, Proceedings of the 41st IEEE Conference on, volume 1, páginas 616–621. IEEE, 2002.
- [33] Adrian Wills e Brett Ninness. On gradient-based search for multivariable system estimates. IEEE Transactions on Automatic Control, 53(1):298–306, 2008.
- [34] Karl Johan Åström e Peter Eykhoff. System identification—a survey. Automatica, 7(2):123–162, 1971.
- [35] Hirotugu Akaike. Some problems in the application of the cross-spectral method. Spectral analysis of time series, 1967.
- [36] Torsten Söderström, Lennart Ljung, e Ivar Gustavsson. Identifiability conditions for linear multivariable systems operating under feedback. IEEE Transactions on Automatic Control, 21(6):837–840, 1976.
- [37] Tung-Sang Ng, Graham C Goodwin, e Brian DO Anderson. Identifiability of mimo linear dynamic systems operating in closed loop. Automatica, 13(5):477–485, 1977.
- [38] Ivar Gustavsson, Lennart Ljung, e Torsten Söderström. Identification of processes in closed loop—identifiability and accuracy aspects. Automatica, 13(1):59–75, 1977.
- [39] K Sin e G Goodwin. Checkable conditions for identifiability of linear systems operating in closed loop. IEEE Transactions on Automatic Control, 25(4):722–729, 1980.
- [40] Brian DO Anderson e MR Gevers. Identifiability of linear stochastic systems operating under linear feedback. Automatica, 18(2):195–213, 1982.
- [41] Torsten Söderström e Petre Stoica. System identification. 1989.
- [42] Paul Van den Hof. Closed-loop issues in system identification. IFAC Proceedings Volumes, 30(11):1547–1560, 1997.
- [43] Lennart Ljung. System Identification Toolbox 7: Getting Started Guide. The MathWorks, 2008.
- [44] Gene H Golub e Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM Journal on numerical analysis, 10(2):413–432, 1973.
- [45] Linda Kaufman. A variable projection method for solving separable nonlinear least squares problems. BIT Numerical Mathematics, 15(1):49–57, 1975.
- [46] Rodrigo Alvite Romano e Felipe Pait. Matchable-observable linear models and direct filter tuning: An approach to multivariable identification. IEEE Transactions on Automatic Control, 62(5):2180–2193, 2017.

- [47] Patrícia Gomes Saraiva, Paulo Lopes dos Santos, Felipe Pait, Rodrigo Alvite Romano, e T-P Perdicoúlis. An iterative moli-zoft approach for the identification of miso lti systems. Em CONTROLO2018 - 13th APCA International Conference on Automatic Control and Soft Computing, páginas 346–351. APCA, 2018.
- [48] Guilherme Scabin Vicinansa, Diego Colón, e Felipe Miguel Pait. Discrete barycenter method for direct optimization. 2014.
- [49] Steven M Kay. Fundamentals of statistical signal processing, volume I: estimation theory. Prentice Hall, 1993.
- [50] Don Johnson. Minimum mean squared error estimators. Connexions, 2004.
- [51] Simon J Julier e Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. Proceedings of the IEEE, 92(3):401–422, 2004.
- [52] Yoan D Landau. Adaptive control: The model reference approach. IEEE Transactions on Systems, Man, and Cybernetics, (1):169–170, 1984.
- [53] F. M Pait. The Barycenter Method for Direct Optimization. ArXiv e-prints, Janeiro 2018. URL: <https://arxiv.org/abs/1801.10533>, arXiv:1801.10533.
- [54] Eric A Wan e Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. Em Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000, páginas 153–158. Ieee, 2000.