

# Algoritmo *Array* Rápido para Filtragem de Sistemas Lineares Sujeitos a Saltos Markovianos com Variação Estruturada dos Parâmetros no Tempo

Gildson Q. de Jesus<sup>1</sup>, Guilherme P. Andrade<sup>2</sup>

Departamento de Ciências Exatas e Tecnológicas, UESC, Ilhéus, BA.

**Resumo.** Neste artigo foi desenvolvido o algoritmo *array* rápido do filtro para sistemas lineares sujeitos a saltos Markovianos com variação estruturada dos parâmetros. Este tipo de algoritmo, que foi originalmente desenvolvido para filtros de sistemas no espaço de estado, além de conservar as boas propriedades numéricas do algoritmo *array* raiz quadrada, possui a vantagem de poder ser aplicado a sistemas cujos parâmetros variam no tempo de forma estruturada.

**Palavras-chave.** Filtragem, *Array* Rápido, Markoviano, Tempo Discreto.

## 1 Introdução

Filtragem para sistemas lineares sujeitos a saltos Markovianos (SLSM) tem sido um tema largamente estudado na literatura. Em [1] foi desenvolvido o estimador linear mínimo médio quadrático (LMMSE, sigla em inglês) para SLSM, baseado na equação algébrica de Riccati. Apesar deste tipo de filtro ser muito útil em aplicações online, foram detectados alguns problemas nos cálculos da equação de Riccati como, por exemplo, erros de arredondamentos que tornam a recursão da equação de Riccati instável numericamente. Para resolver estes problemas, foram propostos algoritmos alternativos para os cálculos dos filtros, a saber: os algoritmos *array* raiz quadrada [4] e os algoritmos *array* rápidos [5]. Estes métodos têm sido aplicados na literatura como um recurso para aliviar problemas computacionais, associados à equação de Riccati usadas nos filtros.

Os algoritmos *array* rápidos apresentam vantagens computacionais se comparados com a implementação da equação de Riccati, como a redução da faixa dinâmica dos números calculados em implementação por aritmética de pontos fixos e cálculos mais seguros e rápidos da matriz de covariância do erro de estimativa. Estes foram originalmente aplicados para sistemas no espaço estado, em situações onde os parâmetros do sistema são invariantes no tempo [2], ou quando o sistema possui parâmetros com certa variação no tempo estruturada [3]. Para esta última classe de sistemas, os algoritmos *array* rápidos são

---

<sup>1</sup>gildsonj@gmail.com

<sup>2</sup>guilhermepa8@hotmail.com

denominados recursão de Chandrasekhar estendida. Os algoritmos *array* rápidos para filtragem de SLSM com variação estruturada dos parâmetros ainda não foram desenvolvidos na literatura. Este artigo pretende preencher esta lacuna.

## 2 Preliminares

O algoritmo *array* rápido estendido desenvolvido neste trabalho está baseado sobre o seguinte SLSM discreto no tempo

$$\begin{aligned} x_{i+1} &= F_{i,\Theta_i}x_i + G_{i,\Theta_i}u_i, \\ y_i &= H_{i,\Theta_i}x_i + D_{i,\Theta_i}v_i \quad \text{para } i \geq 0, \end{aligned} \quad (1)$$

sendo  $x_i \in \mathbb{R}^n$  a variável de estado,  $u_i \in \mathbb{R}^p$  a variável de distúrbio do estado,  $y_i \in \mathbb{R}^m$  a variável de saída, e  $v_i \in \mathbb{R}^q$  a variável de distúrbio da saída;  $\{\Theta_i\}$  é a cadeia de Markov discreta no tempo com espaço de estados finito  $\{1, \dots, N\}$  e a matriz de transição de probabilidade  $P = [p_{jk}]$ . Seja  $\pi_{i,j} := P(\Theta_i = j)$ ;  $F_{i,k} \in \mathbb{R}^{n \times n}$ ,  $G_{i,k} \in \mathbb{R}^{n \times q_1}$ ,  $H_{i,k} \in \mathbb{R}^{m \times n}$ , e  $D_{i,k} \in \mathbb{R}^{m \times q_2}$  ( $k = 1, \dots, N$ ) as matrizes de parâmetros variantes no tempo de dimensões apropriadas. Os distúrbios aleatórios  $\{u_i\}$  and  $\{v_i\}$  são assumidos como tendo média nula, são estacionários e mutuamente independentes com matrizes de covariância iguais a  $U_i$  e  $V_i$  respectivamente.  $x_0 1_{\{\Theta_0=k\}}$ ,  $k = 1, \dots, N$ , são vetores aleatórios com  $\mathcal{E} \{x_0 1_{\{\Theta_0=k\}}\} = \mu_k$  (sendo que  $1_{\{\cdot\}}$  denota a medida de Dirac) e  $\mathcal{E} \{x_0 x_0^T 1_{\{\Theta_0=k\}}\} = V_k$ ;  $x_0$ ,  $\{\Theta_i\}$  são independentes de  $\{u_i\}$  e  $\{v_i\}$ .

As estimativas são desenvolvidas considerando a seguinte variável de estado aumentado

$$\begin{aligned} z_i &:= [z_{i,1}^T \ \dots \ z_{i,N}^T]^T \in \mathbb{R}^{Nn}, \\ z_{i,k} &:= x_i 1_{\{\Theta_i=k\}} \in \mathbb{R}^n. \end{aligned} \quad (2)$$

De acordo com [1], o modelo aumentado de (1) é escrito como

$$\begin{aligned} z_{i+1} &= \mathcal{F}_i z_i + \psi_i, \\ y_i &= \mathcal{H}_i z_i + \varphi_i, \quad \text{para } i \geq 0, \end{aligned} \quad (3)$$

sendo as matrizes de parâmetros dadas por

$$\mathcal{F}_i := \begin{bmatrix} p_{11}F_{i,1} & \dots & p_{N1}F_{i,N} \\ \vdots & \ddots & \vdots \\ p_{1N}F_{i,1} & \dots & p_{NN}F_{i,N} \end{bmatrix} \in \mathbb{R}^{Nn \times Nn}, \quad \mathcal{H}_i := [H_{i,1} \ \dots \ H_{i,N}] \in \mathbb{R}^{m \times Nn}, \quad (4)$$

as variáveis de distúrbio de saída e estado  $\varphi_i$  e  $\psi_i$  são dadas, respectivamente, por

$$\varphi_i := D_{i,\Theta_i}w_i, \quad \psi_i := \mathcal{M}_{i+1}z_i + \vartheta_i. \quad (5)$$

sendo

$$\begin{aligned} \mathcal{M}_{i+1} &:= \begin{bmatrix} \mathcal{M}_{i+1,1} \\ \vdots \\ \mathcal{M}_{i+1,N} \end{bmatrix}, \mathcal{M}_{i+1,k} := \begin{bmatrix} (1_{\{\Theta_{i+1}=k\}} - p_{1k})F_{i,1} \\ \vdots \\ (1_{\{\Theta_{i+1}=k\}} - p_{Nk})F_{i,N} \end{bmatrix}^T, \\ \vartheta_i &:= \begin{bmatrix} 1_{\{\Theta_{i+1}=1\}}G_{i,\Theta_i}u_i \\ \vdots \\ 1_{\{\Theta_{i+1}=N\}}G_{i,\Theta_i}u_i \end{bmatrix}. \end{aligned} \quad (6)$$

As matrizes de segundo momento associada às variáveis de distúrbio  $\varphi_i$  e  $\psi_i$  são

$$\begin{aligned} R_i &:= \mathcal{E}\{\varphi_i\varphi_i^T\} = \mathcal{D}_i\mathcal{D}_i^T, \\ W_i &:= \mathcal{E}\{\psi_i\psi_i^T\} = \text{diag}[Z_{i+1,k}] - \mathcal{F}_i(\text{diag}[Z_{i,k}])\mathcal{F}_i^T, \end{aligned} \quad (7)$$

sendo

$$\mathcal{D}_i := [D_{i,1}\pi_{i,1}W_i^{1/2} \quad \dots \quad D_{i,N}\pi_{i,N}W_i^{1/2}], \quad \pi_{i,k} := \mathcal{P}(\theta_i = k), \quad (8)$$

$$Z_i := \text{diag}[Z_{i,k}], \quad Z_{i,k} := \mathcal{E}\{z_{i,k}z_{i,k}^T\}, \quad (9)$$

e  $Z_{i,k} \geq 0, k = 1, \dots, N$  são dados pela equação recursiva

$$Z_{0,k} := \bar{V}_k, \quad Z_{i+1,k} := \sum_{j=1}^N p_{jk}F_{i,j}Z_{i,j}F_{i,j}^T + \sum_{j=1}^N \pi_{i,j}p_{jk}G_{i,j}G_{i,j}^T,$$

O filtro LMMSE desenvolvido em [1] é dado por

$$\hat{x}_{i|i} := \sum_{j=1}^N \hat{z}_{i,j|i}, \quad (10)$$

$$\hat{z}_{i|i} := \hat{z}_{i|i-1} + Z_{i|i-1}\mathcal{H}_i^T(\mathcal{H}_iZ_{i|i-1}\mathcal{H}_i^T + R_i)^{-1}(y_i - \mathcal{H}_i\hat{z}_{i|i-1}), \quad (11)$$

$$\hat{z}_{0|-1} := \mathcal{E}(z_0) = [\mu_1^T \quad \dots \quad \mu_N^T]^T, \quad \hat{z}_{i|i-1} := \mathcal{F}_i\hat{z}_{i-1|i-1}, \quad (12)$$

$$\tilde{Z}_{0|-1} := \Pi_0, \quad (13)$$

$$\tilde{Z}_{i+1|i} := \mathcal{F}_i\tilde{Z}_{i|i-1}\mathcal{F}_i^T - \mathcal{F}_i\tilde{Z}_{i|i-1}\mathcal{H}_i^T(\mathcal{H}_i\tilde{Z}_{i|i-1}\mathcal{H}_i^T + R_i)^{-1}\mathcal{H}_i\tilde{Z}_{i|i-1}\mathcal{F}_i^T + W_i. \quad (14)$$

### 3 Algoritmo *Array* Rápido para SLSM

O algoritmo *array* rápido, deduzido neste trabalho, é baseado no seguinte resultado fundamental.

**Lema 3.1.** [2] *Sejam  $A$  e  $B$  matrizes  $n \times m$  (com  $n \leq m$ ), e seja  $J = \text{diag}(I_p, -I_q)$  uma matriz assinatura com  $p + q = m$ . Se  $AJA^T = BJB^T$  é não-nula, então existe uma matriz  $J$ -unitária  $\Xi$  tal que  $A = B\Xi$ .*

O algoritmo *array* rápido do filtro LMMSE foi desenvolvido considerando os parâmetros  $F_{i,k}$ ,  $G_{i,k}$  e  $H_{i,k}$  do SLSM (1) variantes no tempo de forma estruturada, esta variação é feita utilizando uma matriz  $\bar{\Psi}_{i,j} \in \mathfrak{R}^{n \times n}$  conveniente. Os parâmetros  $\mathcal{F}_i$  e  $\mathcal{H}_i$  do modelo aumentado (3) variam utilizando as matrizes  $\Psi_i \in \mathfrak{R}^{Nn \times Nn}$ , obtida por

$$\Psi_i := \text{diag}[\bar{\Psi}_i] = \begin{bmatrix} \bar{\Psi}_i & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \bar{\Psi}_i \end{bmatrix}. \quad (15)$$

A variação dos parâmetros é definida de acordo com as seguintes regras

$$F_{i+1,j} \bar{\Psi}_i := \bar{\Psi}_{i+1} F_{i,j}, \quad \bar{\Psi}_{i+1} G_{i,j} := G_{i+1,j}, \quad H_{i,j} := H_{i+1,j} \bar{\Psi}_i. \quad (16)$$

O parâmetro  $R_i$ , definido em (7), varia pela equação de diferença

$$\delta R_i := R_{i+1} - R_i = V_i N_i V_i^T. \quad (17)$$

Os parâmetros  $\mathcal{F}_i$  e  $\mathcal{H}_i$  do modelo aumentado variam de acordo com

$$\mathcal{F}_{i+1} \Psi_i := \Psi_{i+1} \mathcal{F}_i, \quad \mathcal{H}_i := \mathcal{H}_{i+1} \Psi_i. \quad (18)$$

No próximo resultado, o algoritmo *array* rápido para o cálculo da equação de Riccati em (14), como segue:

---

*Algoritmo Array Rápido Estendido para LMMSE*

---

**Passo 0: Condições Iniciais**

$$\begin{aligned} \tilde{Z}_{0|-1} &:= \Pi_0, \\ \tilde{Z}_{1|0} &:= \mathcal{F}_0 \tilde{Z}_{1|0} \mathcal{F}_0^T - \mathcal{F}_0 \tilde{Z}_{1|0} \mathcal{H}_0^T (\mathcal{H}_0 \tilde{Z}_{1|0} \mathcal{H}_0^T + R_0)^{-1} \mathcal{H}_0 \tilde{Z}_{1|0} \mathcal{F}_0^T + \text{diag}[Z_{1,k}] - \mathcal{F}_0 (\text{diag}[Z_{0,k}]) \mathcal{F}_0^T, \\ L_0 S_0 L_0^T &:= \tilde{Z}_{1|0} - \Psi_0 \tilde{Z}_{0|-1} \Psi_0^T, \quad V_0 M_0 V_0^T := R_1 - R_0, \\ Q_0 E_0 Q_0 &= \text{diag}[K_{1,k} \Lambda_{1,k} K_{1,k}^T] - \mathcal{F}_1 (\text{diag}[K_{0,k} \Lambda_{0,k} K_{0,k}^T]) \mathcal{F}_1^T, \\ K_{p,0} &:= \mathcal{F}_0 \tilde{Z}_{1|0} \mathcal{H}_0^T R_{e,0}^{-T/2}, \quad R_{e,0} := \mathcal{H}_0 \tilde{Z}_{1|0} \mathcal{H}_0^T + R_0, \end{aligned} \quad (19)$$

**Passo 1:** Calcule  $L_{i+1}$  usando uma matriz  $(I \oplus S_i \oplus M_i \oplus E_i)$ -unitária  $\Gamma_i$  de dimensões apropriadas

$$\begin{bmatrix} R_{e,i}^{1/2} & \mathcal{H}_{i+1} L_i & V_i & 0 \\ \Psi_{i+1} K_{p,i} & \mathcal{F}_{i+1} L_i & 0 & Q_i \end{bmatrix} \Gamma_i := \begin{bmatrix} R_{e,i+1}^{1/2} & 0 & 0 & 0 \\ K_{p,i+1} & L_{i+1} & 0 & 0 \end{bmatrix} \quad (20)$$

sendo

$$\begin{aligned} L_i S_i L_i^T &:= \tilde{Z}_{i+1|i} - \Psi_i \tilde{Z}_{i|i-1} \Psi_i^T, \quad V_i M_i V_i^T := R_{i+1} - R_i, \\ Q_i E_i Q_i &:= \text{diag}[K_{i+1,k} \Lambda_{i+1,k} K_{i+1,k}^T] - \mathcal{F}_{i+1} (\text{diag}[K_{i,k} \Lambda_{i,k} K_{i,k}^T]) \mathcal{F}_{i+1}^T, \\ K_{p,i} &:= \mathcal{F}_i \tilde{Z}_{i|i-1} \mathcal{H}_i^T R_{e,i}^{-T/2}, \quad R_{e,i} := \mathcal{H}_i \tilde{Z}_{i|i-1} \mathcal{H}_i^T + R_i. \end{aligned} \quad (21)$$

**Passo 2:** Se desejado,  $\tilde{Z}_{i+2|i+1}$  pode ser calculado como

$$\tilde{Z}_{i+2|i+1} := \Psi_{i+1} \tilde{Z}_{i+1|i} \Psi_{i+1}^T + L_{i+1} S_{i+1} L_{i+1}^T. \quad (22)$$

**Prova:** Considere a equação de Riccati em (14), dada por

$$\begin{aligned} \tilde{Z}_{i+2|i+1} &= \mathcal{F}_{i+1} \tilde{Z}_{i+1|i} \mathcal{F}_{i+1}^T - \mathcal{F}_{i+1} \tilde{Z}_{i+1|i} \mathcal{H}_{i+1}^T (\mathcal{H}_{i+1} \tilde{Z}_{i+1|i} \mathcal{H}_{i+1}^T + R_{i+1})^{-1} \\ &\times \mathcal{H}_{i+1} \tilde{Z}_{i+1|i} \mathcal{F}_{i+1}^T + \text{diag} [Z_{i+2,k}] - \mathcal{F}_{i+1} (\text{diag} [Z_{i+1,k}]) \mathcal{F}_{i+1}^T, \end{aligned} \quad (23)$$

sendo,  $R_{i+1} = \mathcal{D}_{i+1} \mathcal{D}_{i+1}^T$ .

Adicionando o termo  $-\Psi_{i+1} \tilde{Z}_{i+1|i} \Psi_{i+1}^T$  em ambos os lados de (23), obtém-se

$$\begin{aligned} \tilde{Z}_{i+2|i+1} - \Psi_{i+1} \tilde{Z}_{i+1|i} \Psi_{i+1}^T &= -\Psi_{i+1} \tilde{Z}_{i+1|i} \Psi_{i+1}^T + \mathcal{F}_{i+1} \tilde{Z}_{i+1|i} \mathcal{F}_{i+1}^T - \mathcal{F}_{i+1} \tilde{Z}_{i+1|i} \mathcal{H}_{i+1}^T \\ &\times (\mathcal{H}_{i+1} \tilde{Z}_{i+1|i} \mathcal{H}_{i+1}^T + R_{i+1})^{-1} \mathcal{H}_{i+1} \tilde{Z}_{i+1|i} \mathcal{F}_{i+1}^T + \text{diag} [Z_{i+2,k}] - \mathcal{F}_{i+1} (\text{diag} [Z_{i+1,k}]) \mathcal{F}_{i+1}^T. \end{aligned} \quad (24)$$

O lado direito da equação (24) é o complemento de Schur do bloco  $-(1, 1)$

$$\begin{bmatrix} \mathcal{H}_{i+1} \tilde{Z}_{i+1|i} \mathcal{H}_{i+1}^T + \mathcal{D}_{i+1} \mathcal{D}_{i+1} & \mathcal{H}_{i+1} \tilde{Z}_{i+1|i} \mathcal{F}_{i+1}^T \\ \mathcal{F}_{i+1} \tilde{Z}_{i+1|i} \mathcal{H}_{i+1}^T & \mathcal{W} \end{bmatrix} \quad (25)$$

$$\mathcal{W} = -\Psi_{i+1} \tilde{Z}_{i+1|i} \Psi_{i+1}^T + \mathcal{F}_{i+1} \tilde{Z}_{i+1|i} \mathcal{F}_{i+1}^T + \text{diag} [Z_{i+2,k}] - \mathcal{F}_{i+1} (\text{diag} [Z_{i+1,k}]) \mathcal{F}_{i+1}^T \quad (26)$$

Considere agora, as definições de (21), os parâmetros variando no tempo conforme (18) e a diferença (17). A matriz (25) pode ser fatorada como

$$\begin{bmatrix} R_{e,i}^{1/2} & \mathcal{H}_{i+1} L_i & V_i & 0 \\ \Psi_{i+1} K_{p,i} & \mathcal{F}_{i+1} L_i & 0 & Q_i \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & S_i & 0 & 0 \\ 0 & 0 & M_i & 0 \\ 0 & 0 & 0 & E_i \end{bmatrix} \begin{bmatrix} R_{e,i}^{T/2} & K_{p,i}^T \Psi_{i+1}^T \\ L_i^T \mathcal{H}_{i+1}^T & L_i^T \mathcal{F}_{i+1}^T \\ V_i^T & 0 \\ 0 & Q_i^T \end{bmatrix}. \quad (27)$$

Consequentemente, o pré-array é dado por

$$\begin{bmatrix} R_{e,i}^{1/2} & \mathcal{H}_{i+1} L_i & V_i & 0 \\ \Psi_{i+1} K_{p,i} & \mathcal{F}_{i+1} L_i & 0 & Q_i \end{bmatrix}. \quad (28)$$

Do Lema (3.1), existe uma matriz  $J$ -unitária  $\Gamma_i$ , com  $J = \text{diag}(I, S_i, M_i, E_i)$ , que triangulariza  $A_i$

$$\begin{bmatrix} R_{e,i}^{1/2} & \mathcal{H}_{i+1} L_i & V_i & 0 \\ \Psi_{i+1} K_{p,i} & \mathcal{F}_{i+1} L_i & 0 & Q_i \end{bmatrix} \Gamma_i = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}, \quad (29)$$

para obter os elementos  $X, Y, Z$ , pode-se elevar o quadrado de ambos os lados da equação (29) e comparar os respectivos elementos.

Consequentemente, o pós-array pode ser escrito como

$$\begin{bmatrix} R_{e,i+1}^{1/2} & 0 \\ K_{p,i+1} & L_{i+1} \end{bmatrix}. \quad (30)$$

## 4 Exemplo Numérico

Nesta seção, será apresentado um exemplo numérico do algoritmo *array* rápido para filtragem de SLSM. Os dados para simulação são os seguintes

$$F_1 = \begin{bmatrix} 0.7 & 0 \\ 0.1 & 0.1 \end{bmatrix}, F_2 = \begin{bmatrix} 0.6 & 0 \\ 0.1 & 0.2 \end{bmatrix}, G_1 = \begin{bmatrix} 0.87 & 0 \\ 0 & 1.2 \end{bmatrix}, G_2 = \begin{bmatrix} 0.87 & 0 \\ 0 & 1.2 \end{bmatrix},$$

$$H_1 = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}^T, H_2 = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}^T, \pi = \begin{bmatrix} 0.05 \\ 0.95 \end{bmatrix}^T, P = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix}, D_1 = D_2 = 0.0008.$$

Foram calculados os valores de " $\tilde{Z}_{i|i-1}$ " para três diferentes implementações: ponto flutuante, ponto fixo para equação de Riccati explícita e para o algoritmo *array* rápido. O cálculo feito com ponto flutuante foi utilizado como referência. Os desempenhos, tanto da equação de Riccati quanto do algoritmo *array* rápido, são idênticos com ponto flutuante. Para o cálculo com ponto fixo, foi usada uma arquitetura em 16-bits que pode representar números no intervalo de  $-65.543$  a  $65.543$ . Essas implementações foram feitas via MatLab através do *fix-point Simulink toolbox*. Pode-se notar na Figura 1, a vantagem do algoritmo *array* rápido em comparação com a implementação por equação de Riccati explícita. Com o algoritmo *array* rápido, o resultado do cálculo em ponto fixo foi equivalente ao resultado obtido no cálculo em ponto flutuante. Na Figura 2, podemos ver que os valores máximo e mínimo singulares de " $L_i$ " tendem a zero logo nas primeiras iterações, demonstrando a eficiência na velocidade do algoritmo.

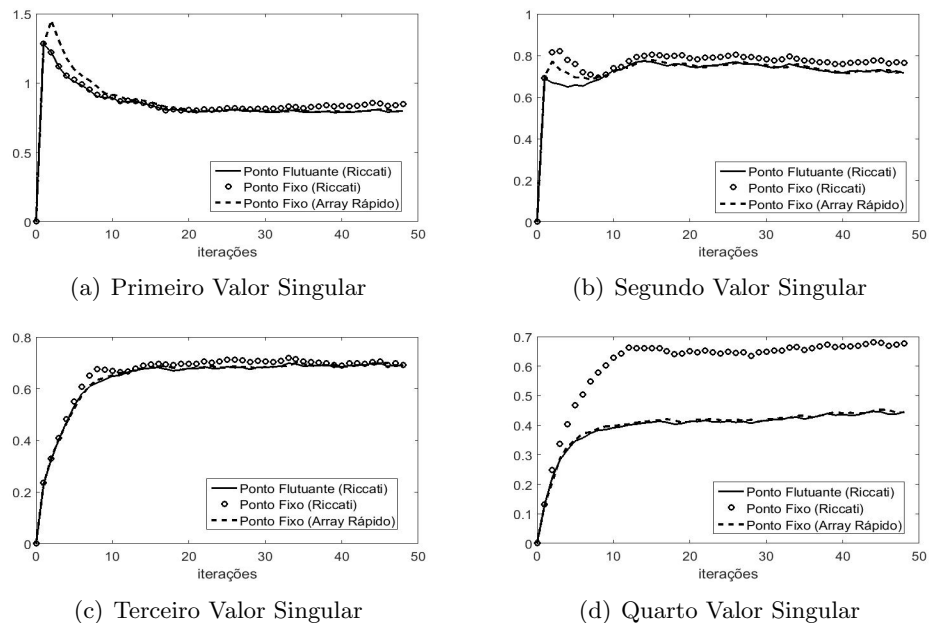


Figura 1: Valores Singulares de  $\tilde{Z}_{i|i-1}^{-1}$ .

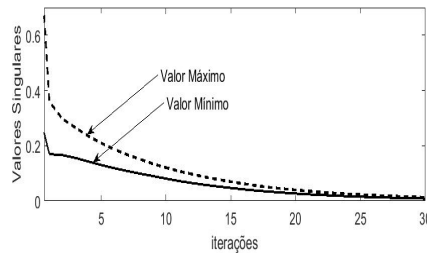


Figura 2: Valores Singulares de  $L_i$

## 5 Conclusões

Neste artigo, foi desenvolvido o algoritmo *array* rápido com variação estruturada dos parâmetros do filtro recursivo LMMSE para SLSM. No exemplo numérico apresentado foi verificado que este algoritmo é numericamente mais estável, além disso, possui um esforço computacional menor.

## Agradecimentos

À FAPESB e UESC pelo suporte na realização deste trabalho.

## Referências

- [1] O. L. V. Costa and S. Guerra. Stationary filter for linear minimum mean square error estimation for discrete-time Markovian jump linear systems. *IEEE Transactions on Automatic Control*, volume 47, no. 8, pages 1351-1356, 2002.
- [2] M. Morf, G. S. Sidhu and T. Kailath. Some new algorithms for recursive estimation in constant, linear, discrete-time systems. *IEEE Transactions on Automatic Control*, volume AC-19, n0. 4, pages 315-323, 1974.
- [3] A. H. Sayed and T. Kailath. Extend Chandrasekhar recursions. *IEEE Transactions on Automatic Control*, volume 39, no. 3, pages 619-623, 1994.
- [4] M. H. Terra, J. Y. Ishihara and G. Jesus. Information Filtering and *Array* Algorithms for Discrete-Time Markovian Jump Linear Systems. *IEEE Transactions on Automatic Control*, volume 54, no. 1, pages 158-162, 2009.
- [5] M. H. Terra, J. Y. Ishihara and G. Jesus. Fast *array* algorithm for filtering of markovian jump linear systems. *Int. J. Adapt. Control Signal Process.*, volume 26, no. 1, pages 84-94, 2012.