

Algoritmo *Array* Rápido para Filtragem Robusta de Sistemas Singulares com Variação Estruturada dos Parâmetros no Tempo

Guilherme P. Andrade¹, Gildson Q. de Jesus²

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

Resumo. Neste artigo foi desenvolvido o algoritmo *array* rápido para o filtro robustos na forma de informação para sistemas singulares com variação estruturado nos parâmetros. Este tipo de algoritmo, originalmente desenvolvido para sistemas no espaço de estado, apresenta vantagens se comparado com o filtro convencional calculado através da equação de Riccati, pois aumenta a eficiência e a estabilidade numérica devido ao uso de transformações ortogonais nos cálculos, reduz a faixa dinâmica dos valores calculados por aritmética de ponto fixo, e reduz o esforço computacional. Um exemplo numérico baseado em implementação de ponto fixo é apresentado para demonstrar as vantagens deste algoritmo.

Palavras-chave. Algoritmos *Array* Rápidos, Tempo Discreto, Variação Estruturada do Tempo, Filtragem de Sistema Singular.

1 Introdução

A abordagem para a filtragem de sistemas desenvolvidas em [5], sintetizada através do filtro de Kalman, tem sido aplicada em vários problemas práticos de engenharia. Apesar das intensas aplicações e das indiscutíveis vantagens desse tipo de abordagem, o filtro de Kalman padrão calculado através da equação de Riccati apresentou limitações numéricas que têm sido bastante estudadas nos últimos anos, dentre estas vale ressaltar as divergências devido a falta de fidelidade dos algoritmos numéricos ou modelagens não apropriadas dos sistemas a serem estimados [3]. Para contornar esses problemas têm sido desenvolvidos novos algoritmos para diferentes implementações do filtro. Neste artigo será deduzido o algoritmo *array* rápido para filtragem robusta na forma de informação de sistemas singulares com variação estruturada dos parâmetros no tempo.

O filtro de Kalman calculado via algoritmo *array* rápido apresenta algumas vantagens se comparado ao cálculo tradicional, tendo um aumento da eficiência e da estabilidade numéricas, devido ao uso de transformações ortogonais nos cálculos, além de uma redução da faixa dinâmica dos valores calculados em implementações por aritmética de ponto fixo, veja [1] e as referências contidas nela.

¹guilhermepa8@hotmail.com

²gildsonj@gmail.com

De acordo com o conhecimento dos autores, o algoritmo *array* rápido ainda não foi usado para calcular filtros robustos de sistemas singulares com variação estruturada dos parâmetros. O objetivo deste trabalho é preencher essa lacuna.

2 Preliminares

O algoritmo *array* rápido que será apresentado foi desenvolvido baseado no seguinte sistema singular sujeito a incertezas paramétricas

$$\begin{aligned} (E_{i+1} + \delta E_{i+1})x_{i+1} &= (F_i + \delta F_i)x_i + w_i, \\ y_i &= (H_i + \delta H_i)x_i + v_i, \quad \text{para } i \geq 0, \end{aligned} \quad (1)$$

sendo $x_i \in \mathfrak{R}^n$ a variável singular, $y_i \in \mathfrak{R}^p$ a medida de saída, $w_i \in \mathfrak{R}^m$ e $v_i \in \mathfrak{R}^p$ ruídos de processo e medida, $E_{i+1} \in \mathfrak{R}^{m \times n}$, $F_i \in \mathfrak{R}^{m \times n}$ e $H_i \in \mathfrak{R}^{p \times n}$ matrizes conhecidas do sistema nominal. δE_{i+1} , δF_i e δH_i são perturbações variantes no tempo para as matrizes do sistema nominal definidas como

$$\delta F_i = M_{f,i} \Delta_i N_{f,i}, \quad \delta E_{i+1} = M_{f,i} \Delta_i N_{e,i+1}, \quad \delta H_i = M_{h,i} \Delta_i N_{h,i}, \quad \|\Delta\| \leq 1, \quad (2)$$

sendo $M_{f,i}$, $M_{h,i}$, $N_{e,i+1}$, $N_{f,i}$, $N_{h,i}$ matrizes conhecidas e Δ_i uma matriz arbitrária limitada. A condição inicial e os ruídos de processo e de medida, $\{x_0, w_i, v_i\}$, são assumidos como sendo variáveis aleatórias de média zero não correlacionadas com estatísticas de segunda ordem

$$\mathcal{E} \left(\begin{bmatrix} x_0 \\ w_i \\ v_i \end{bmatrix} \begin{bmatrix} x_0 \\ w_j \\ v_j \end{bmatrix}^T \right) = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q_i \delta_{ij} & 0 \\ 0 & 0 & R_i \delta_{ij} \end{bmatrix} > 0, \quad (3)$$

sendo $\delta_{ij} = 1$ se $i = j$ e $\delta_{ij} = 0$ caso contrário.

O filtro robusto na forma de informação para sistemas singulares é apresentado a seguir. A estimativa robusta filtrada foi desenvolvida em [10]

$$\begin{aligned} P_{0|0}^{-1} \hat{x}_{0|0} &= H_0^T R_0^{-1} y_0, \\ P_{i+1|i+1}^{-1} \hat{x}_{i+1|i+1} &= E_i^T \hat{Q}_i^{-1} (I - F_i (P_{i|i}^{-1} + \bar{N}_{f,i}^T \bar{N}_{f,i})^{-1} F_i^T \hat{Q}_i^{-1})^{-1} \\ &\quad \times F_i (P_{i|i}^{-1} + \bar{N}_{f,i}^T \bar{N}_{f,i})^{-1} P_{i|i}^{-1} \hat{x}_{i|i} + H_i^T \hat{R}_i^{-1} y_{i+1}, \end{aligned} \quad (4)$$

$$\begin{aligned} P_{0|0}^{-1} &= \Pi_0, \\ P_{i+1|i+1}^{-1} &= E_i^T \hat{Q}_i^{-1} E_i - E_i^T \hat{Q}_i^{-1} F_i (P_{i|i}^{-1} + \bar{N}_{f,i}^T \bar{N}_{f,i} + F_i^T \hat{Q}_i^{-1} F_i)^{-1} F_i^T \hat{Q}_i^{-1} E_i \\ &\quad + H_i^T \hat{R}_i^{-1} H_i + \bar{N}_{h,i}^T \bar{N}_{h,i} + \bar{N}_{e,i}^T \bar{N}_{e,i}, \end{aligned} \quad (5)$$

sendo \hat{Q}_i^{-1} e \hat{R}_i^{-1} matrizes de parâmetro corrigidas dadas por

$$\begin{aligned} \hat{Q}_i^{-1} &:= Q_i^{-1} + Q_i^{-1} M_{f,i} (\hat{\lambda} I - M_{f,i}^T Q_i^{-1} M_{f,i})^{-1} M_{f,i}^T Q_i^{-1}, \\ \hat{R}_i^{-1} &:= R_i^{-1} + R_i^{-1} M_{h,i} (\hat{\lambda} I - M_{h,i}^T R_i^{-1} M_{h,i})^{-1} M_{h,i}^T R_i^{-1}, \end{aligned} \quad (6)$$

e

$$\bar{N}_{e,i} = \hat{\lambda}^{1/2} N_{e,i}, \quad \bar{N}_{f,i} = \hat{\lambda}^{1/2} N_{f,i}, \quad \bar{N}_{h,i} = \hat{\lambda}^{1/2} N_{h,i}. \quad (7)$$

Observação 2.1. *Uma condição suficiente para a existência desse filtro de informação robusto é que $\begin{bmatrix} E_i \\ H_i \end{bmatrix}$ tenha posto coluna completo.*

Observação 2.2. *O ajuste ótimo desse filtro não é possível ser feito online. Para este tipo de aplicação filtros sub-ótimos podem ser ajustados com $\lambda_i = (1 + \alpha)\lambda_{i,i}$ sendo que para qualquer $\alpha > 0$ há garantia de estabilidade desse filtro.*

3 Algoritmos *Array* Rápidos para Filtragem Robusta de Sistemas Singulares

A seguir, é apresentado um resultado auxiliar utilizado na dedução do algoritmo *array* rápido desta seção.

Lema 3.1. [1] *Seja A e B matrizes $n \times m$ (com $n \leq m$), e seja $J = (I_p \oplus -I_q)$ uma matriz assinatura com $p + q = m$. Se $AJA^T = BJB^T$ tem posto completo, então existe uma matriz J -unitária Θ tal que $A = B\Theta$.*

Considere $\Psi_i \in \mathbb{R}^{n \times n}$, uma matriz conveniente. Os parâmetros $E, F, H, \bar{N}_{e,i}, \bar{N}_{f,i}, \bar{N}_{h,i}$ variam de forma estruturada no tempo da seguinte forma

$$\begin{aligned} E_i \Psi_{i+1}^T &:= E_{i+1}, & F_i \Psi_i^T &:= F_{i+1}, & H_i \Psi_{i+1}^T &:= H_{i+1}, \\ \bar{N}_{e,i} \Psi_{i+1}^T &:= \bar{N}_{e,i+1}, & \bar{N}_{f,i} \Psi_i^T &:= \bar{N}_{f,i+1}, & \bar{N}_{h,i} \Psi_{i+1}^T &:= \bar{N}_{h,i+1}. \end{aligned} \quad (8)$$

A variação dos parâmetros \hat{Q}^{-1} e \hat{R}^{-1} são feitas utilizando a forma de diferença, demonstrada em [6]:

$$\delta \hat{Q}_i^{-1} := \hat{Q}_{i+1}^{-1} - \hat{Q}_i^{-1} = U_i N_i U_i^T, \quad \delta \hat{R}_i^{-1} := \hat{R}_{i+1}^{-1} - \hat{R}_i^{-1} = V_i M_i V_i^T, \quad (9)$$

onde M_i e N_i são matrizes assinatura.

Utilizando estas definições, o algoritmo *array* rápido que calcula a equação de Riccati (5) de maneira alternativa, utilizada na estimativa filtrada, pode ser calculado como segue.

Algoritmo *Array* Rápido

Passo 0: Calcular as condições iniciais

$$\begin{aligned} P_{0|0}^{-1} &= \Pi_0, \\ P_{1|1}^{-1} &= E_1^T \hat{Q}_1^{-1} E_1 - E_1^T \hat{Q}_1^{-1} F_1 (\Pi_0 + \bar{N}_{f,1}^T \bar{N}_{f,1} + F_1^T \hat{Q}_1^{-1} F_1)^{-1} F_1^T \hat{Q}_1^{-1} E_1 \\ &+ H_1^T \hat{R}_1^{-1} H_1 + \bar{N}_{h,1}^T \bar{N}_{h,1} + \bar{N}_{e,1}^T \bar{N}_{e,1}. \end{aligned} \quad (10)$$

Passo 1: Calcular M_{i+1} utilizando uma matriz unitária $\Theta_{2,i}$ de dimensões apropriadas

$$\begin{bmatrix} \Psi_i R_{\epsilon,i}^{1/2} & L_i & 0 & F_{i+1}^T U_i \\ \Psi_{i+1} K_{f,i} & 0 & H_{i+1}^T V_i & E_{i+1}^T U_i \end{bmatrix} \Theta_{2,i} = \begin{bmatrix} R_{\epsilon,i+1}^{1/2} & 0 \\ K_{f,i+1} & L_{i+1} \end{bmatrix}, \quad (11)$$

sendo

$$\begin{aligned} L_i S_i L_i^T &= P_{i+1|i+1}^{-1} - \Psi_i P_{i|i}^{-1} \Psi_i^T, \\ K_{f,i} &= E_i^T \hat{Q}_i^{-1} F_i R_{\epsilon,i}^{-T/2}, \quad R_{\epsilon,i} = P_{i|i}^{-1} + \bar{N}_{f,i}^T \bar{N}_{f,i} + F_i^T \hat{Q}_i^{-1} F_i, \\ V_i M_i V_i^T &= \hat{R}_{i+1}^{-1} - \hat{R}_i^{-1}, \quad U_i N_i U_i^T = \hat{Q}_{i+1}^{-1} - \hat{Q}_i^{-1}. \end{aligned} \quad (12)$$

Passo 2: Portanto, $P_{i+1|i}$ pode ser calculado através da seguinte equação de diferença

$$P_{i+1|i+1}^{-1} = \Psi_i P_{i|i}^{-1} \Psi_i^T + L_i S_i L_i^T, \quad (13)$$

sendo $S_i \in \mathbb{R}^{\alpha \times \alpha}$ uma matriz assinatura.

Prova: A demonstração deste algoritmo é feita da seguinte maneira. Considere a equação

$$P_{i+2|i+2}^{-1} - \Psi_{i+1} P_{i+1|i+1}^{-1} \Psi_{i+1}^T = L_{i+1} S_{i+1} L_{i+1}^T. \quad (14)$$

Da equação (5), (14) pode ser reescrita como

$$\begin{aligned} P_{i+2|i+2}^{-1} - \Psi_{i+1} P_{i+1|i+1}^{-1} \Psi_{i+1}^T &= -\Psi_{i+1} P_{i+1|i+1}^{-1} \Psi_{i+1}^T + E_{i+1}^T \hat{Q}_{i+1}^{-1} E_{i+1} - K_{f,i+1} K_{f,i+1}^T \\ &\quad + H_{i+1}^T \hat{R}_{i+1}^{-1} H_{i+1} + \bar{N}_{h,i+1}^T \bar{N}_{h,i+1} + \bar{N}_{e,i+1}^T \bar{N}_{e,i+1}, \\ K_{f,i+1} &= E_{i+1}^T \hat{Q}_{i+1}^{-1} F_{i+1} R_{\epsilon,i+1}^{-T/2}, \\ R_{\epsilon,i+1} &= P_{i+1|i+1}^{-1} + \bar{N}_{f,i+1}^T \bar{N}_{f,i+1} + F_{i+1}^T \hat{Q}_{i+1}^{-1} F_{i+1}. \end{aligned} \quad (15)$$

O complemento de Schur de, $P_{i+2|i+2}^{-1} - \Psi_{i+1} P_{i+1|i+1}^{-1} \Psi_{i+1}^T$, é dado por

$$\begin{bmatrix} R_{\epsilon,i+1} & F_{i+1}^T \hat{Q}_{i+1}^{-1} E_{i+1} \\ E_{i+1}^T \hat{Q}_{i+1}^{-1} F_{i+1} & W \end{bmatrix}, \quad (16)$$

sendo

$$W = -\Psi_{i+1} P_{i+1|i+1}^{-1} \Psi_{i+1}^T + E_{i+1}^T \hat{Q}_{i+1}^{-1} E_{i+1} + H_{i+1}^T \hat{R}_{i+1}^{-1} H_{i+1} + \bar{N}_{h,i+1}^T \bar{N}_{h,i+1} + \bar{N}_{e,i+1}^T \bar{N}_{e,i+1}. \quad (17)$$

A matriz (16) pode ser fatorada da seguinte forma

$$\begin{bmatrix} \Psi_i R_{\epsilon,i}^{1/2} & L_i & 0 & F_{i+1}^T U_i \\ \Psi_{i+1} K_{f,i} & 0 & H_{i+1}^T V_i & E_{i+1}^T U_i \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & S_i & 0 & 0 \\ 0 & 0 & M_i & 0 \\ 0 & 0 & 0 & N_i \end{bmatrix} \begin{bmatrix} R_{\epsilon,i}^{T/2} \Psi_i^T & K_{f,i}^T \Psi_{i+1}^T \\ L_i^T & 0 \\ 0 & V_i^T H_{i+1} \\ U_i^T F_{i+1} & U_i^T E_{i+1} \end{bmatrix}. \quad (18)$$

Da fatoração (18), surge a matriz

$$\begin{bmatrix} \Psi_i R_{\epsilon,i}^{1/2} & L_i & 0 & F_{i+1}^T U_i \\ \Psi_{i+1} K_{f,i} & 0 & H_{i+1}^T V_i & E_{i+1}^T U_i \end{bmatrix}. \quad (19)$$

A matriz (19) é denominada *pré-array*. Pode-se definir o *pós-array*, fatorando (16) através da decomposição

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & \Delta A \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix}, \quad (20)$$

sendo ΔA o complemento de Schur de A em $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$. Usando esta propriedade têm-se

$$\begin{bmatrix} R_{\epsilon,i+1}^{1/2} & 0 \\ K_{f,i+1} & L_{i+1} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & S_{i+1} \end{bmatrix} \begin{bmatrix} R_{\epsilon,i+1}^{T/2} & K_{f,i+1}^T \\ 0 & L_{i+1}^T \end{bmatrix}. \quad (21)$$

Portanto, de (21) surge a matriz

$$\begin{bmatrix} R_{\epsilon,i+1}^{1/2} & 0 \\ K_{f,i+1} & L_{i+1} \end{bmatrix}, \quad (22)$$

denominada *pós-array*.

De acordo com o Lema (3.1), existe uma matriz $(I \oplus S_i \oplus M_i \oplus N_i)$ -unitária Θ_i , tal que, igualando (19) e (22), que resulta na recursão dita de Chandrasekhar (11).

4 Exemplos Numéricos

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

$$E = \begin{bmatrix} 1.14 & 0 & 0 \\ 0 & 1.17 & 0 \\ 0 & 0 & 0 \end{bmatrix}, F = \begin{bmatrix} 0.97 & 0 & 0 \\ 0.27 & -0.78 & 0 \\ 0.12 & 0.12 & 0.68 \end{bmatrix}, H = \begin{bmatrix} 0.11 & 0 & 0 \\ 0 & 0.56 & 0 \\ 0 & 0 & 0.32 \end{bmatrix},$$

$$Q = \begin{bmatrix} 0.04 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0.05 \end{bmatrix}, R = \begin{bmatrix} 0.09 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.03 \end{bmatrix}, N_h = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0.01 \end{bmatrix},$$

$$N_e = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, N_f = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}, M_f = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}, M_h = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.8 \end{bmatrix},$$

e $\hat{\lambda} = 40$. Foram calculados os valores singulares de " $P_{i|i}^{-1}$ " para três diferentes implementações: ponto flutuante, utilizado como referência, ponto fixo para equação de Riccati explícita e para o algoritmo *array* rápido. Os desempenhos tanto da equação de Riccati quanto do algoritmo *array* 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. Na implementação

em ponto fixo da equação de Riccati ocorreram erros numéricos. 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, pode-se notar que os valores singulares de " L_i ", tanto máximo quanto mínimo, tendem a zero logo nas primeiras iterações do algoritmo, isto demonstra a velocidade do cálculo do algoritmo.

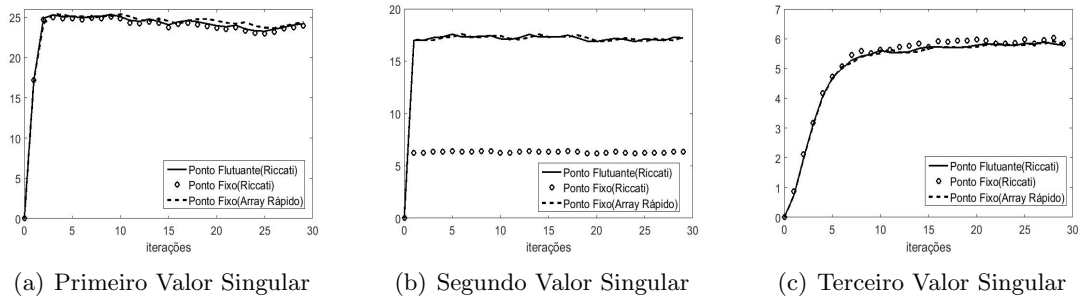


Figura 1: Valores Singulares de P_i^{-1} .

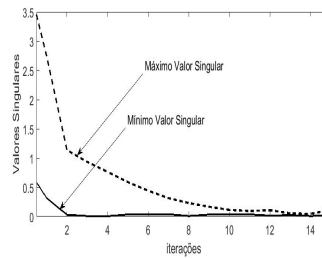


Figura 2: Valores Singulares de L_i

5 Conclusões

O objetivo deste trabalho foi desenvolver o algoritmo *array* rápido para filtragem robusta de sistemas singulares com variação estruturada dos parâmetros no tempo. Tal algoritmo foi proposto como alternativa para o cálculo da equação de Riccati usada no filtro convencional [10]. As implementações dos algoritmos, em ponto fixo e em ponto flutuante, demonstraram a robustez numérica em comparação à equação de Riccati.

Agradecimentos

À Fapesb e UESC pelo suporte na realização deste trabalho.

Referências

- [1] B. Hassibi, T. Kailath and A. H. Sayed. Array algorithms for H-infinity estimation. *IEEE Transactions on Automatic Control*, volume 45, no.4, pages 702-706, 2000.
- [2] J. Y. Ishihara, M. H. Terra and J. C. T. Campos. Robust Kalman filter for descriptor systems. *Proceedings of American Control Conference*, Boston, MA, June/July 2004.
- [3] A. H. Jazwinski. *Stochastic processes and Filtering Theory*. Academic Press, New York, NY, 1970.
- [4] G. Jesus, M. H. Terra, e J. Y. Ishihara. Algoritmos array rápidos para filtragem robusta de sistemas singulares. *Anais da XVII Congresso Brasileiro de Automática*. Juiz de Fora, Minas Gerais, Brasil, 2008.
- [5] R. E. Kalman. A new approach to linear filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, volume 82D, pages 35-45, 1960.
- [6] M. Morf and T. Kailath. Square root algorithms for least squares estimation, *IEEE Trans. Automat. Contr.*, volume AC-20, no. 4, pages 487-497, Aug. 1975.
- [7] A. H. Sayed and T. Kailath. Extend chandrasekhar recursions. *IEEE Transactions on Automatic Control*, volume 39, no. 3, pages 619-623, 1994.
- [8] A. H. Sayed. A framework for state-space estimation with uncertain models. *IEEE Transactions on Automatic Control*, volume AC-46, no. 7, pages 998-1013, 2001.
- [9] S. P. Singh and R. W. Liu. Existence of state equation representation if linear largescale dynamic systems, *IEEE Transactions on Circuit Theory*, volume 20, no. 3, pages 239-246, 1973.
- [10] M. H. Terra, J. Y. Ishihara and A. C. J. Padoan. Information filtering and array algorithms for descriptor system subject to parameter uncertainties, *IEEE Transactions on Signal Processing*, volume 5, no. 1, pages 1-9, 2007.