ESTABILIDADE NA MÉDIA QUADRÁTICA DE SISTEMAS LINEARES COM SALTOS MARKOVIANOS EM TEMPO CONTÍNUO VIA MÉTODO DE PLANOS DE CORTE E ALGORITMO GENÉTICO

VALESKA M. DE SOUZA[1], ANTONIO A. M. RAPOSO[1].

1. *Departamento de Matemática, Universidade Federal do Maranhão*
Av. dos Portugueses, 1966 - Bacanga, 65085-580, São Luís, MA, BRASIL
E-mails: valeska@ufma.br, eng_mat_antonioamraposo@hotmail.com

Abstract— This paper proposes an linear optimization approach to mean square stability analysis of Markovian Jumps Linear Systems in continuous time. Specifically, the results extend to the Lyapunov-Metzler equation, obtaining convex optimization problems in terms of a set of LMIs. An algorithm based on the combination of the Method of Cutting Plans and Genetic Algorithm is proposed for the numerical solution of the resulting problem using the C++ programming language.

Keywords— Linear Optimization, Mean Square Stability, Method of Cutting Plans, Genetic Algorithm.

Resumo— Este trabalho propõe uma abordagem de otimização linear para análise de estabilidade na média quadrática de Sistemas Lineares com Saltos Markovianos em tempo contínuo. Mais especificamente, estendem-se os resultados da equação de Lyapunov-Metzler, obtendo problemas de otimização convexa, em termos de um conjunto de LMIs. Um algoritmo baseado na combinação do Método de Planos de Corte e Algoritmo Genético é proposto para a resolução numérica do problema resultante utilizando linguagem de programação C++.

Palavras-chave— Otimização Linear, Estabilidade na Média Quadrática, Método de Planos de Corte, Algoritmo Genético.

1 Introdução

Sistemas Lineares com Saltos Markovianos (SLSM) representam uma classe de sistemas estocásticos cuja dinâmica muda de forma repentina em certos instantes e comportam-se como sistemas lineares nos demais instantes. Os saltos entre os modos de operação do sistema podem ser modelados por uma cadeia de Markov com espaço de estados finito. Estes sistemas são exemplos de sistemas híbridos, pois combinam uma parte do estado que toma valores contínuos e outra parte que toma valores discretos.

Um SLSM definido no espaço de probabilidade $(\Omega, \mathrm{F}, \mathrm{P})$, onde Ω é o espaço amostral; F é uma σ-álgebra e P é uma medida de probabilidade; pode ser representado por meio da seguinte equação estocástica

$$\begin{aligned} \dot{x}(t) &= A(\sigma_t)x(t), \quad t \in R^+ \\ x(0) &= x_0, \ \sigma_0 = \sigma \end{aligned} \tag{1}$$

onde $x \in R^n$ são os estados do processo, $x_0 \in R^n$ é o estado inicial, $\sigma_t \in \overline{N} = \{1, 2, \cdots, N\}$ representa uma cadeia de Markov homogênea a tempo contínuo com espaço de estados discreto e σ é uma variável aleatória. Se $\sigma_t = i$ então $A(\sigma_t) = A_i, \forall \sigma_t \in \overline{N}$. Além disso, a probabilidade de transição é dada por

$$\begin{aligned} p_{ij}(\Delta) &= \mathrm{P}(\sigma_{t+\Delta} = j \mid \sigma_t = i) \\ &= \begin{cases} \lambda_{ij}\Delta + o(\Delta) & \text{se} \quad i \neq j \\ 1 + \lambda_{ii}\Delta + o(\Delta) & \text{se} \quad i = j \end{cases} \end{aligned} \tag{2}$$

onde λ_{ij} é definido como a taxa de transição entre os estados i e j, e $\lambda_{ij} \geq 0, i \neq j$, $\lambda_{ii} = -\sum_{i \neq j} \lambda_{ij}$. A matriz de taxa de transição é denotada por $\Lambda = \{\lambda_{ij}\}$.

Um problema fundamental no estudo dos SLSM é a noção de estabilidade. Na literatura há diferentes definições de estabilidade, dentre as mais significativas temos a estabilidade na média quadrática (MS-estabilidade, do inglês *Mean Square*), estabilidade δ-momento e a estabilidade quase certa; vide (Costa and Marques, 2006; de Souza and Colaneri, 2007; Fang and Loparo, 2002). Este artigo apresenta uma metodologia para análise da MS-estabilidade de SLSM baseada em algoritmo desenvolvido utilizando linguagem de programação C++. O enfoque principal é a síntese de um algoritmo que tivesse uma linguagem mais acessível para vários profissionais de diversas áreas do conhecimento podendo ser aplicado em diversas situações.

2 Propósito

Para análise da MS-estabilidade de SLSM em tempo contínuo, apresentamos um algoritmo baseado em inequações matriciais lineares (LMIs, do inglês *Linear Matrix Inequalities*); vide (Boyd et al, 1994). Denominaremos este algoritmo de MPCAG, pois utiliza o Método de Planos de Corte (MPC), combinado com o Algoritmo Genético (AG) para testar problemas de MS-estabilidade.

Como os métodos MPC geram hiperplanos que aproximam gradativamente o conjunto de soluções factíveis, determinar a MS-estabilidade de um SLSM

resume-se em resolver um Problema de Programação Linear (PPL). O AG é proposto para resolver este PPL sob penalidade das restrições com o uso do Método da Barreira Logarítmica (MBL). Optou-se por utilizar o AG para fins de comparação com outros métodos de resolução de PPL.

3 Métodos

Definição 1. (**MS-estabilidade**) O sistema (1) é dito MS-estável se para quaisquer $x(0)$ e distribuição inicial de σ_t:

$$\lim_{t\to\infty} E\left\{\left\|x(t,x_0)\right\|^2\right\}=0.$$

O próximo teorema caracteriza a MS-estabilidade de sistemas SLSM usando LMIs.

Teorema 1 (El Ghaoui and Rami, 1996). *O sistema SLSM (1) é MS-estável se, e somente se, existem matrizes* $P_i=P_i^T>0$, *que satisfazem as inequações*

$$A_i^T P_i + P_i A_i + \sum_{j=1}^{N}\lambda_{ij}P_j < 0, i=1,2,\ldots,N \quad (3)$$

Equivalentemente, podemos reescrever o Teorema 1 substituindo as LMIs (3) pelas equações de Lyapunov-Metzler

$$A_i^T P_i + P_i A_i + \sum_{j=1}^{N}\lambda_{ij}P_j = -Q_i. \quad (4)$$

Sendo $Q_i=Q_i^T>0$ matrizes arbitrárias, podemos escolher $Q_i=Q$ e obtemos

Teorema 2. *O sistema SLSM (1) é MS-estável se, e somente se, para qualquer matriz* $Q=Q^T>0$, *existem matrizes* $P_i=P_i^T>0$, *que satisfazem as inequações de Lyapunov-Metzler*

$$A_i^T P_i + P_i A_i + \sum_{j=1}^{N}\lambda_{ij}P_j + Q < 0,\ i=1,2,\ldots,N \quad (5)$$

Corolário 1. *Se, para qualquer matriz* $Q=Q^T>0$, *existe uma única matriz* $P=P^T>0$, *tal que*

$$A_i^T P + PA_i + Q < 0 \quad (6)$$

então o sistema (1) é MS-estável.

Como as inequações de Lyapunov (3) definem N conjuntos convexos C_i e a interseção de um número finito de conjuntos convexos é um conjunto convexo, define-se o domínio convexo como

$$D=\left\{\bigcap_{i=1}^{N}C_i\right\},$$

afim de resolvê-las via análise convexa.

Utiliza-se um método de plano de corte proposto por Ferreira (1994). Métodos de planos de corte geram hiperplanos que aproximam gradativamente o conjunto de soluções factíveis do problema. Deste modo, o próximo teorema nos permite encontrar, caso exista, uma solução para as LMIs (5).

Teorema 3. *Dada uma matriz* $Q=Q^T>0$. *Considere o seguinte problema de otimização:*

$$min:\sum_{i=1}^{N}Tr(P_i)$$

sujeito a:

$$P_i=P_i^T>0$$
$$A_i^T P_i + P_i A_i + \sum_{j=1}^{N}\lambda_{ij}P_j + Q \le 0, i=1,2,\ldots,N \quad (7)$$

As seguintes afirmações são verdadeiras:

1. *O problema (7) é convexo;*
2. *Para* $N=1$ *sua solução é igual a equação de Lyapunov-Metzler;*
3. *Para* P_0 *não factível, existe sempre um hiperplano que separa* P_0 *do conjunto de soluções factíveis.*

Com base em (Ferreira, 1994), temos os seguintes hiperplanos suporte para as restrições do Teorema 3:

- Para as restrições $P_i=P_i^T>0$ tem-se os seguintes hiperplanos suporte:

$$g(P_i)=x^T(-P_i)x\le-\varepsilon \quad (8)$$

- Para as restrições

$$A_i^T P_i + P_i A_i + \sum_{j=1}^{N}\lambda_{ij}P_j + Q \le 0$$

tem-se os seguintes hiperplanos suporte:

$$h(P_i)=x^T\left(A_i^T P_i + P_i A_i + \sum_{j=1}^{N}\lambda_{ij}P_j + Q\right)x\le-(1+\varepsilon) \quad (9)$$

onde $x\in R^n$ e ε é uma constante próxima de zero utilizada com critério de parada do algoritmo MPCAG.

Para solucionar o problema convexo (7) utilizou-se uma abordagem heurística com o AG. Escolheu-se o AG devido a sua capacidade de trabalhar com uma população de soluções simultaneamente, e porque é facilmente hibridizado com outras técnicas, no caso com o MPC.

Entretanto, como (7) é um problema de otimização linear com restrições, aplicar o AG para solucioná-lo exigiu a penalização das restrições com uma função barreira logarítmica, (Camponogara, 2006) e (Engelbrecht, 2007). Deste modo, o AG resolve o seguinte problema de otimização:

$$min:\sum_{i=1}^{N}Tr(P_i)-\mu\left(\sum_{i=1}^{N}ln(P_i)+\sum_{i=1}^{N}R_i+\sum_{i=1}^{N}S_i\right) \quad (10)$$

onde $\mu\ge0$ é o parâmetro da barreira, e

$R_i=\ln\{-[g(P_i)+\varepsilon]\}$ e $S_i=\ln\{-[h(P_i)+\varepsilon+1]\}$.

Com estas considerações, a Figura 1 apresenta a estrutura do algoritmo MPCAG.

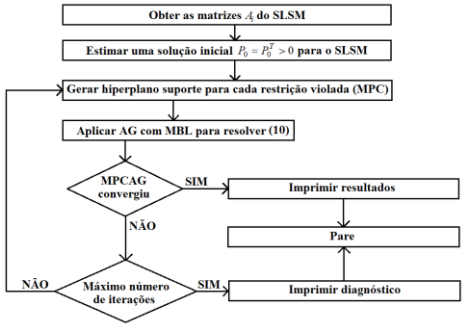

Figura 1. Estrutura do algoritmo MPCAG

A próxima seção apresenta resultados obtidos com o algoritmo MPCAG.

4 Resultados

Antes de ilustramos os resultados, as constantes utilizadas no MPC e no MPCAG são dadas na Tabela 1.

MPC	
ε	10^{-6}
Máximo número de iterações	100
MBL	
μ	10^{-12}
AG	
Tamanho da população	8
Número de gerações	100
Tamanho do cromossomo	22
Taxa de crossover	0,6
Taxa de mutação	0,01

Tabela 1. Parâmetros do MPC e MPCAG

Exemplo 1. Considere o SLSM com dois modos de operação

$$A_1 = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix} \text{ e } A_2 = \begin{pmatrix} 0 & 1 \\ -2 & -1 \end{pmatrix}, \text{ com}$$

matriz de transição correspondente

$$\Lambda = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \text{ e } Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Para testar a MS-estabilidade pelo Teorema 1, resolvemos as LMIs (3) utilizando o LMI Control Toolbox do MATLAB. Como as soluções

$$P_1 = \begin{pmatrix} 66{,}2129 & 18{,}5573 \\ 18{,}5573 & 42{,}7632 \end{pmatrix} \text{ e}$$

$$P_2 = \begin{pmatrix} 67{,}1042 & 9{,}8615 \\ 9{,}8615 & 37{,}1508 \end{pmatrix}$$

são matrizes definidas positivas o sistema é MS-estável. Agora, usando o Teorema 2, obtemos

$$P_1 = \begin{pmatrix} 60{,}5176 & 16{,}9978 \\ 16{,}9978 & 39{,}1968 \end{pmatrix} \text{ e}$$

$$P_2 = \begin{pmatrix} 61{,}3231 & 9{,}0570 \\ 9{,}0570 & 34{,}0353 \end{pmatrix}.$$

Com a utilização do Corolário 1 obtemos

$$P = \begin{pmatrix} 53{,}3923 & 10{,}7207 \\ 10{,}7207 & 29{,}9222 \end{pmatrix}.$$

Por fim, para testar a MS-estabilidade pelo Teorema 3, resolvemos o problema de otimização convexa (7) usando a função *mincx* do LMI Control Toolbox do MATLAB. Com a seguinte solução para uma única matriz P:

$$P = \begin{pmatrix} 2{,}5754 & 0{,}6391 \\ 0{,}6391 & 1{,}4927 \end{pmatrix}, \text{ após 5 iterações.}$$

E com a seguinte solução para P_1 e P_2:

$$P_1 = \begin{pmatrix} 1{,}6621 & 0{,}4919 \\ 0{,}4919 & 0{,}9328 \end{pmatrix} \text{ e } P_2 = \begin{pmatrix} 1{,}6452 & 0{,}2545 \\ 0{,}2545 & 0{,}8141 \end{pmatrix},$$

após 12 iterações.

Os resultados que seguem ilustram a utilização do algoritmo MPCAG para testar a MS-estabilidade segundo o Corolário 1.

As Figuras 2 e 3 ilustram a evolução do algoritmo de planos de corte do Exemplo 1 com o uso do algoritmo Simplex.

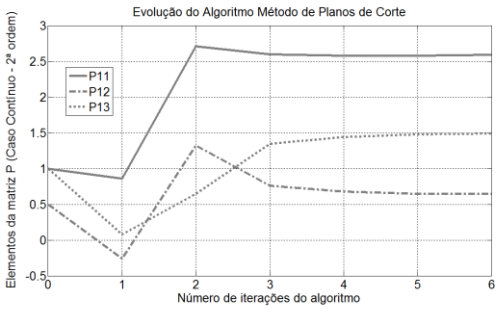

Figura 2. Evolução do algoritmo MPC para o SLSM do Exemplo 1 considerando uma única matriz P

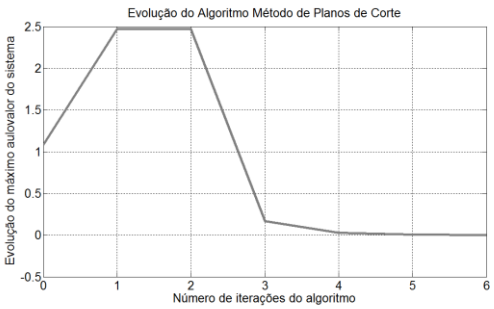

Figura 3. Convergência do algoritmo MPC para o SLSM do Exemplo 1 considerando uma única matriz P

Assim, após 6 iterações, o algoritmo MPC obtém, $P^{MPC} = \begin{pmatrix} 2{,}6 & 0{,}6 \\ 0{,}6 & 1{,}5 \end{pmatrix}$.

As Figura 4 e 5 ilustram a evolução do algoritmo MPCAG.

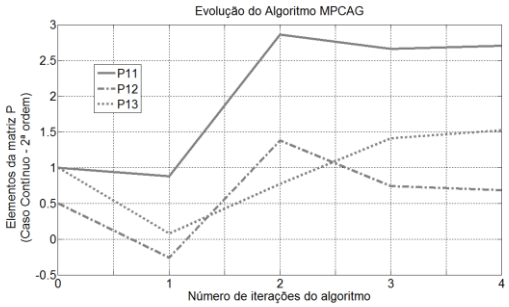

Figura 4. Evolução do algoritmo MPCAG para o Exemplo 1 considerando uma única matriz P

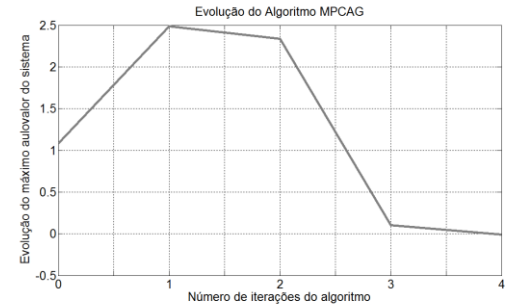

Figura 5. Convergência do algoritmo MPCAG para o Exemplo 1 considerando uma única matriz P

Após 4 iterações, o algoritmo proposto obtém $P^{MPCAG} = \begin{pmatrix} 2{,}703122 & 0{,}686982 \\ 0{,}686982 & 1{,}524846 \end{pmatrix}$.

Exemplo 2. Consideremos o SLSM do Exemplo 1.

Este exemplo ilustra a aplicação do algoritmo MPCAG utilizando o Teorema 3.

As Figuras 6 e 7 ilustram a evolução do algoritmo MPC com o uso do algoritmo Simplex.

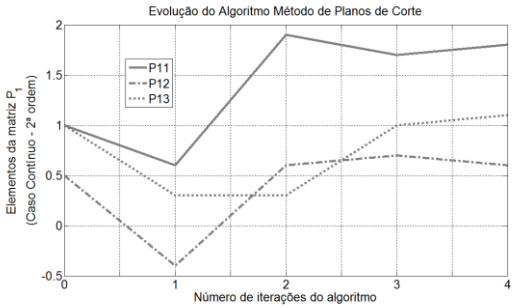

Figura 6. Evolução do algoritmo MPC para os elementos da matriz P_1 do sistema do Exemplo 2

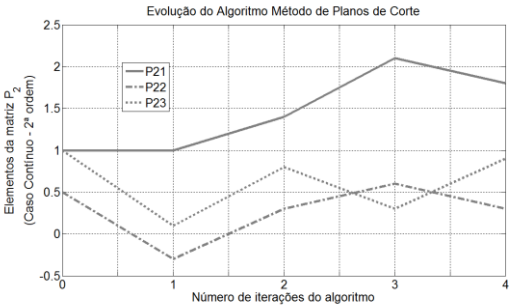

Figura 7. Evolução do algoritmo MPC para os elementos da matriz P_2 do sistema do Exemplo 2

Após 4 iterações, o algoritmo MPC obtém,

$$P_1^{MPC} = \begin{pmatrix} 1{,}8 & 0{,}6 \\ 0{,}6 & 1{,}1 \end{pmatrix} \text{ e } P_2^{MPC} = \begin{pmatrix} 1{,}8 & 0{,}3 \\ 0{,}3 & 0{,}9 \end{pmatrix}.$$

As Figuras 8 e 9 ilustram a evolução do sistema SLSM, do Exemplo 2, por meio do algoritmo MPCAG, em termos das matrizes P_i.

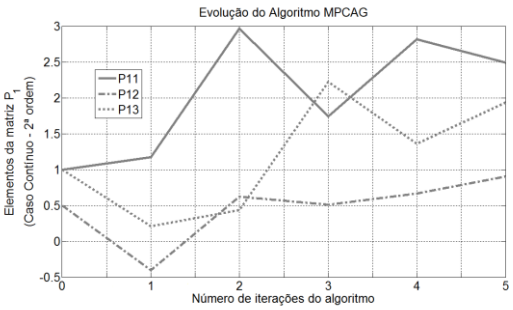

Figura 8. Evolução do algoritmo MPCAG para os elementos da matriz P_1 do SLSM do Exemplo 2

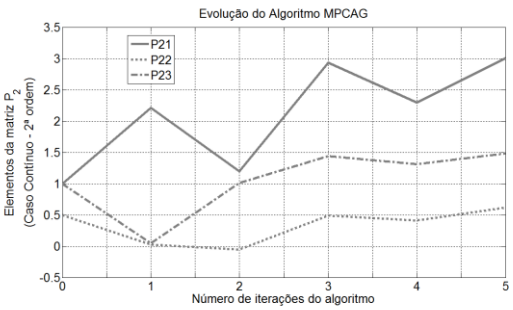

Figura 9. Evolução do algoritmo MPCAG para os elementos da matriz P_2 do SLSM do Exemplo 2

Após 5 iterações, o algoritmo MPCAG obtém,

$$P_1^{MPCAG} = \begin{pmatrix} 2{,}489800 & 0{,}905002 \\ 0{,}905002 & 1{,}934790 \end{pmatrix} \text{ e}$$

$$P_2^{MPCAG} = \begin{pmatrix} 3{,}005470 & 0{,}617145 \\ 0{,}617145 & 1{,}479320 \end{pmatrix}.$$

As Figuras 10 e 11 demonstram a convergência dos algoritmos MPC e MPCAG para o SLSM.

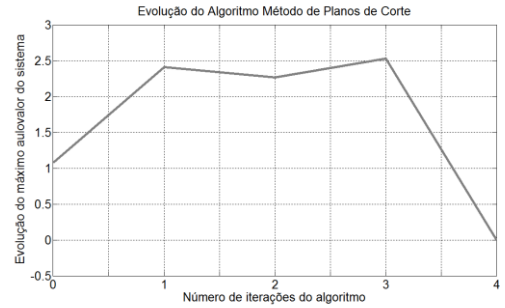

Figura 10. Convergência do MPC para o SLSM do Exemplo 2

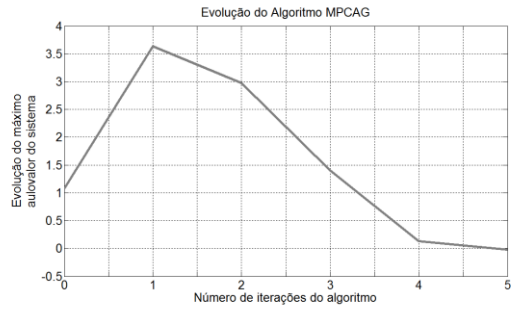

Figura 11. Convergência do MPCAG para o SLSM do Exemplo 2

5 Discussões

A partir dos resultados ilustrados nas Figuras de 2 a 11, percebemos que a análise da MS-estabilidade para SLSM via MPC combinado com AG – MPCAG, gerou resultados satisfatórios quando comparado, em termos de iterações e tempo computacional, à análise do mesmo problema via Programação Linear (PL). O MPCAG aplicado ao SLSM (5), quando comparado ao MPC, mostrou-se mais eficiente, em termos de matriz P. Resultados mais satisfatórios podem ser atingidos com a manipulação dos parâmetros do AG, como tamanho da população, taxa de crossover e taxa de mutação. Vale ressaltar que a introdução de uma única matriz $Q = Q^T > 0$ no Teorema 3 reduz o custo computacional quando comparado à resolução do mesmo problema com o uso das inequações de Lyapunov-Metzler ($Q_i = Q_i^T > 0$). As figuras com a evolução de cada algoritmo foram geradas através do software MATLAB.

6 Conclusão

Neste trabalho, propomos um algoritmo híbrido: uma combinação do MPC com um AG, para análise da MS-estabilidade de sistemas lineares a saltos markovianos em tempo contínuo. Uma pesquisa bibliográfica foi realizada e não se constatou trabalhos desenvolvidos acerca do tema. Desse modo, para compararmos a eficiência do MPCAG primeiramente implementou-se o MPC via Simplex.

O MPCAG mostrou-se eficiente à análise de estabilidade de vários exemplos de SLSM em tempo contínuo de segunda e terceira ordens.

Atualmente o MPCAG está sendo empregado para a análise de MSS de SLSM em tempo discreto.

Agradecimentos

Os autores agradecem ao apoio financeiro concedido pela Fundação de Amparo à Pesquisa e Desenvolvimento Científico do Maranhão (FAPEMA).

Referências Bibliográficas

Boyd, S.P. El Ghaoui, L. Feron, E and Balakrishnan, V. (1994). Linear Matrix Inequalities in System and Control Theory, SIAM, Philadelphia, PA.

Camponogara, E. (2006). Métodos de otimização: Teoria e prática. Departamento de Automação de Sistemas – UFSC.

Costa, O.L.V. and Marques, R. (2004). Comments on stochastic stability of jump linear systems, IEEE Trans. Automat. Control 49: 1414-1416.

El Ghaoui, L. and Rami, M. (1996). Robust state-feedback stabilization of jump linear systems via LMIs. International Journal of Robust Nonlinear Control, pp.1015-1022.

Engelbrecht, A. P. (2007). Computational intelligence: an introduction - 2nd ed., John Wiley & Sons.

de Souza, V. M. (2007). Markovian Jump Linear Systems Via Convex Optimization. Doctoral Dissertation, Dipartimento di Elettronica e Informazione – Politecnico di Milano.

de Souza, V. M. and Colaneri, P. (2007). Some recent results on almost sure stavility of continuous-time markov jump linear sytems. Proc. Of the 6th Congress of Logic Applied to Technology, LAPTEC, Santos, Brasil.

Fang, Y. and Loparo, K. (2002). Stochastic stability of jump linear systems. IEEE Trans. Automat. Control 47:1204-1208.

Feng, X. Loparo, Y. and Chizeck, H. (1992). Stochastic stability properties of jump linear sysrems, IEEE Transactions on Automatic Control 37(1):38-53.

Ferreira, J. D. (1994). Análise e Controle de Sistemas Lineares via Desigualdades Matriciais Lineares. Dissertação de Mestrado, Departamento de Telemática – Unicamp.