

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Análise e melhoramento do método variacional para controle ótimo de sistemas lineares com saltos markovianos sem observação da variável de salto

Junior Rodrigues Ribeiro

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C²MC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Junior Rodrigues Ribeiro

**Análise e melhoramento do método variacional para
controle ótimo de sistemas lineares com saltos markovianos
sem observação da variável de salto**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Eduardo Fontoura Costa

**USP – São Carlos
Julho de 2019**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R484a Ribeiro, Junior Rodrigues
 Análise e melhoramento do método variacional
 para controle ótimo de sistemas lineares com saltos
 markovianos sem observação da variável de salto /
 Junior Rodrigues Ribeiro; orientador Eduardo
 Fontoura Costa. -- São Carlos, 2019.
 98 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2019.

 1. Controle ótimo. 2. Sistemas dinâmicos lineares.
 3. Cadeias de Markov. 4. Parâmetros com salto. 5.
 Condicionamento numérico. I. Costa, Eduardo
 Fontoura, orient. II. Título.

Junior Rodrigues Ribeiro

**Analysis and improvement of the variational method for
control of Markov jump linear systems with no jump
observation**

Master dissertation submitted to the Institute of
Mathematics and Computer Sciences – ICMC-USP,
in partial fulfillment of the requirements for the
degree of the Master Program in Computer Science
and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Prof. Dr. Eduardo Fontoura Costa

USP – São Carlos
July 2019

*Dedico aos meus pais, irmãos, minha avó e em especial à minha irmã caçula, Nayana, quem
muito estimo e desejo sucesso!*

AGRADECIMENTOS

O primeiro agradecimento, nada mais justo, é ao Senhor Deus, que me deu saúde e me proporcionou esta oportunidade.

Menciono minha gratidão à Irmã Nena, que mesmo com sua idade anciã, concedeu apoio nas horas solitárias, com os assuntos vagos para rirmos e nos divertirmos, além dos muitos almoços e cafés que tomamos juntos e da moradia à sua vizinhança concedida.

À professora Marina Andretta, que me ajudou com algumas dicas de otimização não-linear, as quais não foram diretamente empregadas no presente trabalho, mas foram de fundamental importância para que pudéssemos compreender mais a fundo o problema aqui tratado.

Ao meu orientador, professor Eduardo, que tem direcionado o meu caminho até o final desta pesquisa, fazendo-me crescer como pessoa e como pesquisador.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Essa bolsa faz parte do projeto PICME da OBMEP, na qual fui medalhista de bronze em sua 7ª Edição (2011).

*“Disse o néscio no seu coração: Não há Deus.
Têm-se corrompido, fazem-se abomináveis em suas obras,
não há ninguém que faça o bem.”
(Bíblia Sagrada. Salmos 14:1.)*

RESUMO

RIBEIRO, J. R. **Análise e melhoramento do método variacional para controle ótimo de sistemas lineares com saltos markovianos sem observação da variável de salto**. 2019. 98 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Sistemas Lineares com Saltos Markovianos (SLSMs) são estudados desde a década de 1960 e vêm ganhando visibilidade desde então, com diversas aplicações dentre as quais Finanças, Robótica e Engenharias diversas. Um problema de regulação trata de controlar o SLSM buscando fazer sua trajetória se aproximar de zero. Quando os saltos markovianos são observados, o problema é simples e bem resolvido, muito diferente de quando não se observam os saltos. Neste trabalho é estudado um algoritmo da literatura utilizado para resolver o problema de regulação sem observação dos saltos, chamado Método Variacional (MV). Sendo um dos melhores métodos para o dado problema, enfrenta dificuldades de cunho numérico. Neste trabalho se procura analisar e melhorar o condicionamento dos subproblemas envolvidos, de forma a favorecer a convergência do método. São testadas abordagens diferentes usando preconditionadores e comparados os resultados, permitindo concluir que três das cinco abordagens é que trouxeram os melhores resultados. Por se tratar de sistemas lineares do tipo $Ax = b$, as abordagens de condicionamento podem ser adaptadas para outros problemas semelhantes.

Palavras-chave: Controle ótimo, Sistemas dinâmicos lineares, Cadeias de Markov, Parâmetros com salto, Condicionamento numérico.

ABSTRACT

RIBEIRO, J. R. **Analysis and improvement of the variational method for control of Markov jump linear systems with no jump observation.** 2019. 98 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Markov Jump Linear Systems (MJLSs) have been studied since the decade of 1960 and they are gaining visibility ever since, due to a wide range of applications, such as Finance, Robotics, several Engineerings among others. The so called regulation problem is to control the MJLS seeking to make its trajectory to approach zero. When markovian jumps are observed, the problem is simple and the solution given as closed formulas, which is quite different from the situation when jumps are not observed. We study an algorithm available in literature called Variational Method (VM). Even though it is one of the best methods to solve the problem, it has some numerical difficulties. We analyse its performance and propose some ideas aiming at the ill-conditioning of the subproblems involved, in order to improve the convergence of the method. Different approaches are tested using preconditioners and the results are compared, indicating that three approaches of the five tested ones are promising for convergence improvement. Because the subproblems are linear systems of type $Ax = b$, these approaches can be adapted to similar problems.

Keywords: Optimal control, Dynamic linear systems, Markov chains, Jump parameters, Numerical conditioning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração: aplicações de sistemas lineares com saltos markovianos	27
Figura 2 – Convergência do Método Variacional.	27
Figura 3 – Condicionamento dos sistemas \mathbb{S}_k do MV em instâncias e iterações diversas.	28
Figura 4 – Esquema de uma cM com $N = 2$	32
Figura 5 – Duas realizações da cM $\{\mathcal{N}, \mathbb{P}, \pi_0\}$	33
Figura 6 – Grafo representando a cM.	33
Figura 7 – Duas realizações da cM $\{\mathcal{N}, \mathbb{P}, \pi_0\}$	34
Figura 8 – Grafo representando a cM.	34
Figura 9 – Esquema do SLSM: θ_k é aleatório conforme uma cM.	35
Figura 10 – Uma realização da cM e a respectiva trajetória de x_k	38
Figura 11 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\ x_k\ ^2 \mathbf{1}_{\{\theta_k\}})$	39
Figura 12 – Uma realização da cM e a respectiva trajetória de x_k	39
Figura 13 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\ x_k\ ^2 \mathbf{1}_{\{\theta_k\}})$	40
Figura 14 – Uma realização da cM e a respectiva trajetória de x_k	40
Figura 15 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\ x_k\ ^2 \mathbf{1}_{\{\theta_k\}})$	40
Figura 16 – Uma realização da cM e a respectiva trajetória de x_k	41
Figura 17 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\ x_k\ ^2 \mathbf{1}_{\{\theta_k\}})$	41
Figura 18 – A transformação linear deforma o espaço quase em uma reta (exemplo em \mathbb{R}^2).	52
Figura 19 – Comparação dos custos com as máscaras e sem elas, para um total de 1000 instâncias.	66
Figura 20 – Comparação dos custos com aproximação quadrática e sem ela.	66
Figura 21 – Comparação dos custos com uso do P1 e sem ele.	67
Figura 22 – Comparação dos custos com uso do P2 e sem ele.	68
Figura 23 – Comparação dos custos com uso do P5 e sem ele.	68
Figura 24 – Perfis de desempenho dos Precondicionadores.	69
Figura 25 – Comparação do uso da estratégia de varredura dos valores de α	70
Figura 26 – Perfis de desempenho dos métodos de solução de sistemas lineares com P2.	71
Figura 27 – Perfis de desempenho dos métodos de solução de sistemas lineares com P3.	72
Figura 28 – Perfis de desempenho dos métodos de solução de sistemas lineares com P4.	73

Figura 29 – Comparação dos custos com as máscaras e sem elas, com P1.	91
Figura 30 – Comparação dos custos com as máscaras e sem elas, com P2.	92
Figura 31 – Comparação dos custos com as máscaras e sem elas, com P3.	92
Figura 32 – Comparação dos custos com as máscaras e sem elas, com P4.	93
Figura 33 – Comparação dos custos com aproximação quadrática e sem ela, com P1. . .	93
Figura 34 – Comparação dos custos com aproximação quadrática e sem ela, com P2. . .	94
Figura 35 – Comparação dos custos com aproximação quadrática e sem ela, com P3. . .	94
Figura 36 – Comparação dos custos com aproximação quadrática e sem ela, com P4. . .	95
Figura 37 – Comparação dos custos com uso do P3 e sem ele.	95
Figura 38 – Comparação dos custos com uso do P4 e sem ele.	96
Figura 39 – Comparação dos custos com uso de P1 e P2.	96
Figura 40 – Comparação dos custos com uso de P2 e P3.	97
Figura 41 – Comparação dos custos com uso de P2 e P4.	97
Figura 42 – Comparação dos custos com uso de P3 e P4.	98

LISTA DE ALGORITMOS

Algoritmo 1 – Método Variacional	46
Algoritmo 2 – Operador $\mathcal{T}(\cdot)$ (OPERT)	48
Algoritmo 3 – Cálculo do custo \mathbf{J} (CUSTO)	48
Algoritmo 4 – A-Ortogonalização de Gram-Schmidt	90

LISTA DE TABELAS

Tabela 1 – Exemplo de Perfil de Desempenho.	62
Tabela 2 – Descrição das etiquetas dos condicionadores utilizadas no texto.	67
Tabela 3 – Descrição das etiquetas dos métodos utilizadas no texto.	70
Tabela 4 – Comparação dos métodos na otimalidade (em 10^0), com P2.	71
Tabela 5 – Comparação dos métodos na otimalidade (em 10^0), com P3.	72
Tabela 6 – Comparação dos métodos na otimalidade (em 10^0), com P4.	73

LISTA DE SÍMBOLOS

cM — Cadeia de Markov

$SLSM$ — Sistema linear com salto markoviano (em inglês MJLS - Markov Jump Linear System)

FO — Função Objetivo

J — Custo da FO

\mathbb{S} — Sistema linear, do tipo $Ax = b$

k — Tempo discreto

η — Iteração do Método Variacional

MV — Método Variacional

\mathcal{N} — Espaço de estados markoviano (conjunto de índices $i = 1, \dots, N$)

N — Dimensão do espaço \mathcal{N}

\mathbb{P} — Matriz de probabilidade de transição da cM

θ_k — Estado da cM no instante k , também chamado estado discreto do sistema dinâmico ou markoviano

$'$ — Transposição de matriz

Φ_θ — Conjunto de parâmetros do SLSM, indexados em \mathcal{N}

x_k — Estado contínuo do sistema dinâmico no instante k

RLQ — Regulador Linear Quadrático

\mathcal{H} — Horizonte de planejamento (conjunto de índices $k = 0, \dots, T$)

$\mathbb{V}_{m \times n}^{\mathcal{N}}$ — Espaço vetorial de coleções de matrizes reais $m \times n$.

T — Tamanho do horizonte \mathcal{H}

n — Dimensão do estado contínuo x

m — Dimensão do controle u

p — Dimensão do ruído ω

π_k — Vetor distribuição de probabilidade de θ_k

$\|\cdot\|$ — Norma euclidiana

G — Matriz de ganho

$\mathbf{E}(\cdot)$ — Operador esperança

X^k — Matriz de segundo momento de x_k

$\mathbf{1}_{\{\mathcal{C}\}}$ — Função indicadora de \mathcal{C}

$\bar{\mathbf{A}}, \bar{\mathbf{C}}$ — Matrizes de malha fechada

$\mathcal{T}(\cdot), \mathcal{E}(\cdot), \mathcal{L}(\cdot)$ — Operadores sobre coleções de matrizes

Σ^k — Matriz de covariância do ruído ω ponderada por π_k

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação e abordagens	27
1.2	Organização da dissertação	29
2	SOBRE CADEIAS DE MARKOV E SISTEMAS LINEARES COM SALTOS MARKOVIANOS	31
2.1	Cadeias de Markov	31
2.2	Sistemas Lineares com Saltos Markovianos	34
2.3	Exemplos numéricos	38
3	MODELO DETERMINÍSTICO E MÉTODO VARIACIONAL	43
3.1	O custo determinístico	43
3.2	Método variacional	45
4	IMPLEMENTAÇÕES	51
4.1	Mudança de base para os ganhos G_k	54
4.2	Mudança de base para os ganhos G_k : segunda formulação	55
4.3	Máscaras	58
4.4	Aproximação quadrática	62
4.5	Metodologia de avaliação	62
5	RESULTADOS E DISCUSSÕES	65
5.1	Máscaras	65
5.2	Aproximação quadrática multivariada	66
5.3	Precondicionadores	67
5.4	Varredura	69
5.5	Métodos	70
6	CONSIDERAÇÕES FINAIS	75
	REFERÊNCIAS	77
	APÊNDICE A CUSTO COMO FUNÇÃO QUADRÁTICA EM G_k	79

APÊNDICE B	APROXIMAÇÃO QUADRÁTICA MULTIVARIADA POR MÍNIMOS QUADRADOS	85
APÊNDICE C	MÉTODO DE DIREÇÕES CONJUGADAS	89
C.1	Ortogonalização de Gram-Schmidt	90
APÊNDICE D	MAIS GRÁFICOS	91
D.1	Máscaras	91
D.2	Aproximação quadrática	93
D.3	Precondicionadores	95

INTRODUÇÃO

Com o aumento exponencial de tecnologias novas na atualidade, especialmente as eletrônicas, torna-se importante o desenvolvimento de controles para certos objetos, como por exemplo, carros, motores elétricos, aviões, robôs, dentre muitos outros.

No intuito de corrigir a rota de um avião, a trajetória de um robô, a velocidade de trabalho de um motor, dentre outros exemplos, de forma automática, surge a ideia de previsão do sistema e correção de falhas.

Uma classe de sistemas dinâmicos lineares estocásticos que começou a ser estudada desde a década de 1960 com [Dynkin \(1965\)](#) são os sistemas lineares com saltos markovianos (SLSMs). Esses permitem modelar alterações repentinas e bruscas, como em uma pane, falha, ou qualquer outra mudança inesperada dos parâmetros no sistema, o que é comum no mundo real. No momento em que ocorre uma alteração dessas, diz-se ter ocorrido um *salto* (troca) entre parâmetros do SLSM; markoviano porque o modelo para esses saltos é uma cadeia de Markov (ou cM).

Alguns problemas de controle de SLSMs, como *Rastreamento com Alvos Dinâmicos* e o *Regulador Linear Quadrático*, são modelados como problema de otimização, em que algumas restrições são penalizadas, atribuindo-se-lhe um certo custo \mathbf{J} a ser minimizado, dado pela Função Objetivo (FO). Essa penalização e alguns outros fatores tornam o problema não linear, podendo ser resolvido por métodos de descida clássicos da literatura, como os encontrados, por exemplo, em [Ribeiro e Karas \(2014\)](#) e [Nocedal e Wright \(2006\)](#). Entretanto, o problema geralmente é de alta dimensão se tratado com algoritmos de descida, o que induz a utilização de um princípio variacional para quebrar o problema em uma sequência de problemas menores.

Neste trabalho é abordado o problema de regulação de um sistema dinâmico linear com custo quadrático sujeito a saltos markovianos não observados em seus parâmetros. É assumido que apenas o chamado estado contínuo¹ é observado, problema mais complexo do que

¹ Diz-se estado contínuo por tomar valores no espaço \mathbb{R}^n , enquanto que a outra variável de estado

aquele em que o estado é completamente observado (ambos os estados contínuo e markoviano, problema cuja solução é dada por uma fórmula fechada, chamada Equações de Riccati Acopladas (ABOU-KANDIL *et al.*, 2003; ABOU-KANDIL; FREILING; JANK, 1995)). Para resolver esses problemas de controle com observação parcial, Val e Başar (1999) desenvolvem um algoritmo iterativo baseado no princípio variacional, em que se resolve um sistema linear \mathbb{S}_k do tipo $Ax = b$ para cada instante de tempo discreto k , o **subproblema** \mathbb{S}_k , formando uma sequência de sistemas lineares $\{\mathbb{S}_k\}^\eta$ para cada iteração η do método.

Algumas variantes desse Método Variacional (MV) foram desenvolvidas, bem como o emprego de algoritmos clássicos de descida, tais como em Vargas (2004) com uma abordagem de horizonte retrocedente aplicada nos problemas de Rastreamento e Regulação com ruído aditivo, Bortolin (2012) trabalhando com o método clássico de Newton Modificado, usando um esquema de observação parcial da cM para o problema de Regulação e desenvolvendo um gerador de instâncias, Silva (2012) com algoritmos evolutivos tipo Genético para o problema de Regulação, e Oliveira (2014) com Gradiente e Newton em uma abordagem de atualização de múltiplos ganhos para o problema de Regulação.

Dentre esses algoritmos propostos na literatura, o Método Variacional dispõe das melhores soluções e tempo computacional de uma forma geral, mas enfrenta alguns obstáculos. Por ser baseado principalmente na resolução de uma sequência de sistemas lineares \mathbb{S}_k , volta-se a atenção ao condicionamento das matrizes desses sistemas \mathbb{S}_k e para os métodos de solução dos mesmos. Nenhum estudo da literatura dá atenção a esse aspecto, o que levanta algumas questões: o condicionamento é realmente um problema, ou as matrizes são em geral bem condicionadas? O método de solução de \mathbb{S}_k é eficiente ou deixa um grande resíduo? Se empregado um método iterativo, ele converge ou precisa de uma modificação? Dado um erro na solução de um desses \mathbb{S}_k , o que acontece com o valor \mathbf{J} da FO?

Algumas destas questões são respondidas já nos testes computacionais preliminares, confirmando a problemática e voltando a atenção às abordagens para resolvê-la. Assim, busca-se melhorar o desempenho do Método Variacional, bem como tentar responder essas questões.

Os SLSMs podem se aplicados em diversas áreas, como por exemplo em Finanças (CAJUEIRO, 2002; VAL; BAŞAR, 1999), Robótica (VARGAS; COSTA; VAL, 2013), Engenharia Aeroespacial (STOICA; YAESH, 2002), Sistemas de Comunicação (ABDOLLAHI, 2008; ABDOLLAHI; KHORASANI, 2011), dentre várias outras.

(variável de salto) toma valores em \mathbb{N} , sendo chamada de estado discreto ou markoviano.

Figura 1 – Ilustração: aplicações de sistemas lineares com saltos markovianos



(a) Finanças

Fonte: [González \(2017\)](#).

(b) Robótica

Fonte: [Europe \(2017\)](#).

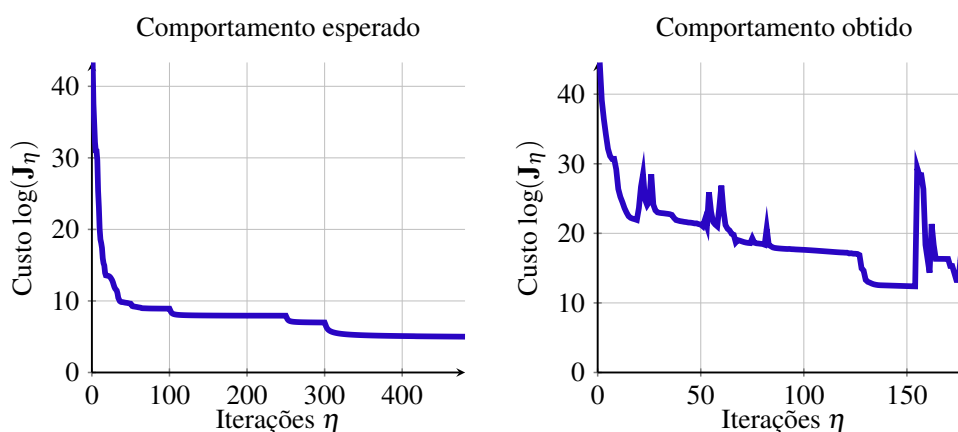
(c) Engenharia aeroespacial

Fonte: [NASA \(2016\)](#).

1.1 Motivação e abordagens

O próximo gráfico ilustra o comportamento geral do valor da FO (custo J) *versus* o número de iterações do MV. *Nem sempre* se consegue o comportamento esperado do decaimento do custo ao longo das iterações, conforme provado matematicamente em [Val e Başar \(1999\)](#), por vezes ferindo a monotonicidade, como se pode ver na [Figura 2](#).

Figura 2 – Convergência do Método Variacional.



Observação 1. Como visto no gráfico da convergência ([Figura 2](#)), por não se ter um resultado matemático que garanta otimalidade da solução, não há um critério de parada para o MV que seja completamente satisfatório. Assim, se o leitor for implementar o controle ótimo em um dispositivo físico (Arduino por exemplo), é importante deixar o MV executando por um bom tempo, pois mesmo supondo que o algoritmo convirja conforme o gráfico do comportamento

[Figura 1a](#): GONZÁLEZ, B. N. Palacio de la Bolsa de Madrid. 2017. Licença © CC BY-SA 4.0.

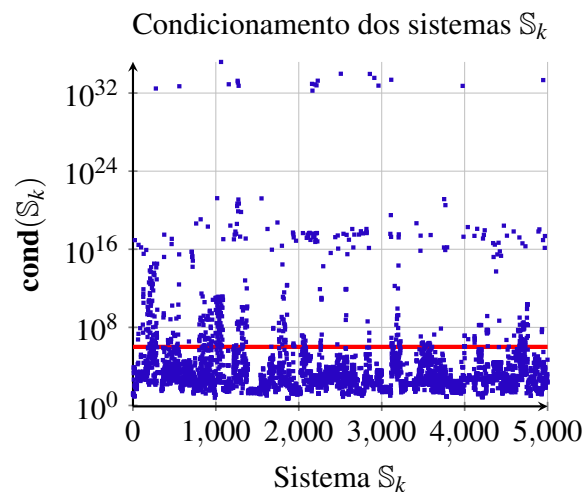
[Figura 1b](#): EUROPE, S. R. NAO Robot. 2017. Licença © CC BY-SA 4.0.

[Figura 1c](#): NASA. NASA Goddard Space Flight Center. 2016. Licença © CC BY 2.0.

esperado (vide a referida figura), pode produzir alguns “falsos positivos”, isto é, parece que o algoritmo convergiu e pára de diminuir substancialmente o custo. No entanto, repentinamente o custo volta a decrescer bastante, e isso se repete por diversas vezes (atente-se para as iterações $\eta = 100$, $\eta = 250$ e $\eta = 300$ do gráfico mencionado). Por esse motivo, o critério de parada por custo relativo é válido, mas não muito eficiente. Ainda assim, não é fácil estabelecer um critério de parada satisfatório, pois vai depender de cada caso.

Procurando investigar se o número de condição dos sistemas lineares \mathbb{S}_k são um problema a ser tratado, verificou-se que os mesmos são razoavelmente bem condicionados na maioria dos casos, mas que ainda há uma grande quantidade de sistemas lineares mal condicionados, conforme se vê no próximo gráfico. Nele foram colocados os números de condição dos sistemas lineares \mathbb{S}_k de diversas instâncias escolhidas ao acaso, em iterações também diversificadas, contabilizando um total de 5000 sistemas lineares analisados. Na [Figura 3](#) pode-se ver que muitas vezes o condicionamento numérico do sistema \mathbb{S}_k é muito grande, e com isso as soluções obtidas para esses subproblemas não são boas, prejudicando a convergência. Em alguns testes particulares realizados, o condicionamento numérico de até 10^6 é razoavelmente aceitável para o Matlab (pacote utilizado para implementação do algoritmo), com uma precisão de aproximadamente 10 dígitos corretos. Na figura, a linha vermelha indica 10^6 .

Figura 3 – Condicionamento dos sistemas \mathbb{S}_k do MV em instâncias e iterações diversas.



Algumas abordagens para tratar o condicionamento dos sistemas \mathbb{S}_k foram testadas, sendo a mais proveitosa delas o uso de preconditionadores para os \mathbb{S}_k , baseados na decomposição de valores singulares ou SVD (*Singular Value Decomposition*), por meio de uma mudança de base. Ainda que teoricamente equivalentes, algumas variações dessa implementação dos preconditionadores tiveram resultados bem diferentes, apenas trocando a prioridade de operadores lineares. Isso instigou a procura de formulações distintas para a mudança de base envolvida, levando aos resultados da [Seção 4.1](#) e da [Seção 4.2](#). Uma modificação foi feita com a adição de um parâmetro α permitindo a criação de diferentes bases para o preconditionamento.

Outra metodologia aplicada foi a modelagem dos parâmetros que constituem \mathbb{S}_k por meio de Aproximação Quadrática Multivariada por Mínimos Quadrados, (cuja formulação foi escrita no [Apêndice B](#)). Mais uma abordagem foi a utilização de um tipo de máscara em um dos parâmetros do modelo com a função de minimizar a perda numérica de uma certa estrutura dos espaços vetoriais ao longo das iterações.

1.2 Organização da dissertação

No [Capítulo 2](#) é apresentada uma introdução às cM's com dois exemplos numéricos, que pode ser omitida, caso o leitor já tenha familiaridade, e em seguida o modelo do SLSM para o problema de Regulação com horizonte finito, objeto deste trabalho. Quatro exemplos numéricos são dados para ilustrar o comportamento dos SLSMs sem controle e com controle, sem ruído aditivo e com ruído aditivo.

No [Capítulo 3](#) é apresentado o modelo determinístico e algumas definições importantes, para então apresentar o Método Variacional e um esquema de seu funcionamento. No [Capítulo 4](#) fazem-se as discussões das implementações e por fim, no [Capítulo 5](#) os resultados obtidos. Alguns resultados foram colocados nos apêndices, por serem extensos e servirem mais como demonstração ou apresentação dos fatos utilizados. Ao final são apresentadas algumas considerações e futuros trabalhos no [Capítulo 6](#).

SOBRE CADEIAS DE MARKOV E SISTEMAS LINEARES COM SALTOS MARKOVIANOS

No intuito de contextualizar ao leitor a respeito do que são cadeias de Markov e a seguir sobre Sistemas Lineares com Saltos Markovianos (SLSMs), faz-se uma introdução ao assunto neste capítulo, com uma seção de exemplos em seguida.

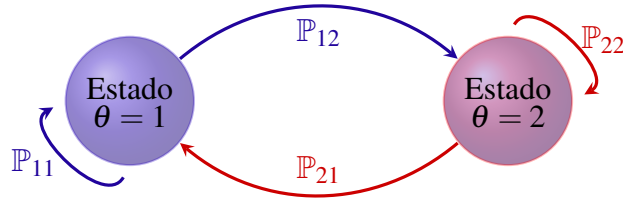
2.1 Cadeias de Markov

Uma cadeia de Markov (cM) em tempo discreto é uma sequência de variáveis aleatórias $\{\theta_k\}_{k \in \mathbb{N}}$, que toma valores em um conjunto de índices chamado de espaço de estados $\mathcal{N} = \{1, 2, \dots, N\}$, com uma matriz estocástica contendo as probabilidades da transição entre esses estados $\mathbb{P} \in [0, 1]^{N \times N}$. Também pode ser atribuída uma distribuição de probabilidade inicial $\pi_0 \in [0, 1]^{N \times 1}$ para o primeiro “sorteio” da cM (valor que o primeiro estado da cadeia θ_0 assume). Assim, essa cadeia pode ser representada por $\{\mathcal{N}, \mathbb{P}, \pi_0\}$.

Em cada instante de tempo k , o estado da cadeia assume aleatoriamente um valor do espaço de estados \mathcal{N} com alguma probabilidade dependente apenas do estado imediatamente anterior $k - 1$. Na matriz de transição se situam as probabilidades condicionais de mudança de estado. Veja o exemplo na [Figura 4](#), o esquema de uma cM com dois estados, ou seja, $N = 2$.

Considere $\mathbf{Prob}(\cdot)$ uma medida de probabilidade no espaço fundamental. Na [Figura 4](#), a probabilidade condicional de que $\theta_k = 2$ dado que $\theta_{k-1} = 1$ é dada por \mathbb{P}_{12} , ou seja, $\mathbf{Prob}(\theta_k = 2 | \theta_{k-1} = 1) = \mathbb{P}_{12}$. Da mesma forma, a probabilidade condicional de que $\theta_k = 1$ dado que $\theta_{k-1} = 1$ é dada por \mathbb{P}_{11} , ou seja, $\mathbf{Prob}(\theta_k = 1 | \theta_{k-1} = 1) = \mathbb{P}_{11}$ e assim por diante. De uma forma geral, tem-se a matriz de probabilidade de transição definida como sendo:

$$\mathbb{P} = [\mathbb{P}_{ij}] = \mathbf{Prob}(\theta_k = j | \theta_{k-1} = i). \quad (2.1)$$

Figura 4 – Esquema de uma cM com $N = 2$.

Fonte: Elaborada pelo autor.

Um estado da cM é dito **transiente** ou **temporário** quando após uma finita quantidade de vezes que este estado é “sorteado”, ele não será mais sorteado. Formalmente, se a partir de um estado j existe uma probabilidade estritamente positiva de que a cadeia não visite nunca mais esse estado, então ele é dito transiente. Um estado da cM que não é transiente é dito **recorrente**, ou seja, se a cadeia passar por esse estado alguma vez, então *ele será revisitado uma quantidade infinita de vezes*.

Uma propriedade muito útil advinda do teorema da probabilidade total é que, dada uma distribuição de probabilidade π_k dos estados da cM, para conhecer a distribuição π_{k+1} , basta multiplicá-lo pela matriz \mathbb{P}' , em que \mathbb{P}' indica a matriz transposta de \mathbb{P} , formando a recursão

$$\pi_{k+1} = \mathbb{P}' \pi_k. \quad (2.2)$$

Exemplo 1. Considerando o espaço de estados markoviano $\mathcal{N} = \{1, 2, 3\}$, uma distribuição de probabilidade inicial $\pi_0 = (1/2, 1/16, 7/16)'$ e uma matriz de probabilidade de transição \mathbb{P} definida abaixo, pode-se obter várias de suas propriedades, calcular as distribuições π_k para qualquer instante k e também simular a cadeia $\{\mathcal{N}, \mathbb{P}, \pi_0\}$.

$$\mathbb{P} = \begin{array}{c} \text{Estados} \\ \theta_k = 1 \\ \theta_k = 2 \\ \theta_k = 3 \end{array} \begin{array}{ccc} \theta_{k+1} = 1 & \theta_{k+1} = 2 & \theta_{k+1} = 3 \\ \left[\begin{array}{ccc} 0.9 & 0 & 0.1 \\ 0 & 0 & 1 \\ 0 & 0.9 & 0.1 \end{array} \right] \end{array}$$

Tem-se pela [Equação 2.2](#) que $\pi_1 \approx (0.4500, 0.3937, 0.1562)'$, $\pi_2 \approx (0.4050, 0.1406, 0.4543)'$ e $\pi_\infty \approx \pi_{10^4} \approx (0, 0.4736, 0.5263)'$.

$$\mathbb{P}^{10} \approx \begin{bmatrix} 0.3486 & 0.3085 & 0.3428 \\ 0 & 0.6571 & 0.3428 \\ 0 & 0.3085 & 0.6914 \end{bmatrix} \quad \mathbb{P}^\infty \approx \mathbb{P}^{10^4} \approx \begin{bmatrix} 0 & 0.4736 & 0.5263 \\ 0 & 0.4736 & 0.5263 \\ 0 & 0.4736 & 0.5263 \end{bmatrix}$$

Pela matriz \mathbb{P} , pode-se ver que a probabilidade de que o estado se mantenha em 1 é de 90%, em 2 é zero (se $\theta_k = 2$, então $\theta_{k+1} \neq 2$, qualquer que seja k) e em 3 é 10%. Ao observar a matriz \mathbb{P}^∞ , nota-se que o estado 1 é transiente e os outros são recorrentes. Ademais, tem-se que, no limite (vide π_∞), a probabilidade de que a cadeia sorteie o estado 2 é $\mathbf{Prob}(\theta_k = 2 | \theta_{k-1}) \approx 0.4736$, enquanto que a probabilidade de sortear o estado 3 é $\mathbf{Prob}(\theta_k = 3 | \theta_{k-1}) \approx 0.5263$.

No gráfico abaixo constam os sorteios realizados da cadeia de $k = 1$ até $k = 200$.

Figura 5 – Duas realizações da cM $\{\mathcal{N}, \mathbb{P}, \pi_0\}$.

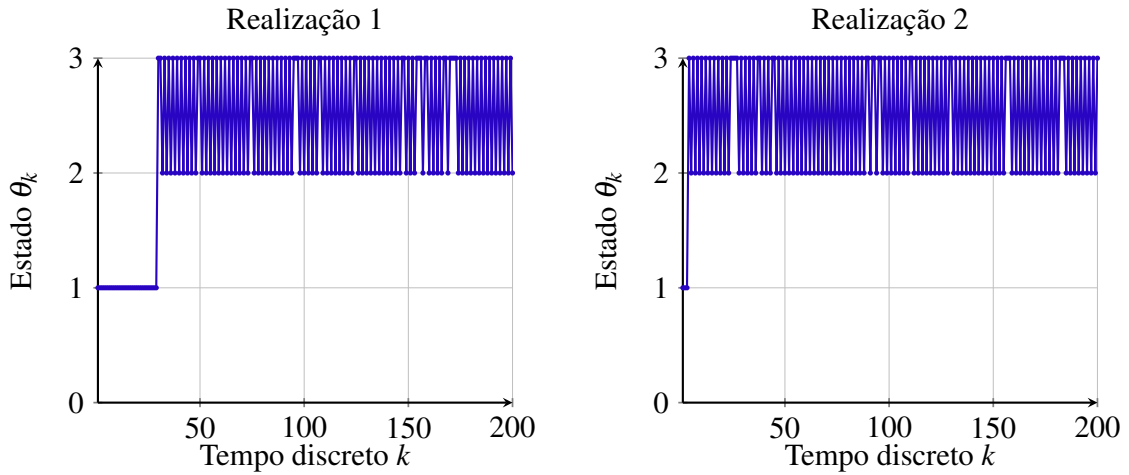
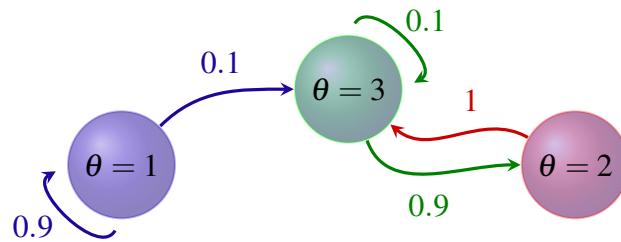


Figura 6 – Grafo representando a cM.



Fonte: Elaborada pelo autor.

Exemplo 2. Considerando o espaço de estados markoviano $\mathcal{N} = \{1, 2\}$, uma distribuição de probabilidade inicial $\pi_0 = (0.98, 0.02)'$ e uma matriz de probabilidade de transição \mathbb{P} definida abaixo, pode-se obter várias de suas propriedades, calcular as distribuições π_k para qualquer instante k e também simular a cadeia $\{\mathcal{N}, \mathbb{P}, \pi_0\}$.

$$\mathbb{P} = \begin{matrix} & \text{Estados} & \theta_{k+1} = 1 & \theta_{k+1} = 2 \\ \begin{matrix} \theta_k = 1 \\ \theta_k = 2 \end{matrix} & & \begin{bmatrix} 0.92 & 0.08 \\ 0.01 & 0.99 \end{bmatrix} \end{matrix}$$

Tem-se pela [Equação 2.2](#) que $\pi_1 = (0.9018, 0.0982)'$, $\pi_2 \approx (0.8306, 0.1693)'$ e $\pi_\infty \approx \pi_{10^4} \approx (0.1111, 0.8888)'$.

$$\mathbb{P}^{10} \approx \begin{bmatrix} 0.4572 & 0.5427 \\ 0.0678 & 0.9321 \end{bmatrix} \quad \mathbb{P}^\infty \approx \mathbb{P}^{10^4} \approx \begin{bmatrix} 0.1111 & 0.8888 \\ 0.1111 & 0.8888 \end{bmatrix}$$

Nota-se que ambos os estados são recorrentes, veja \mathbb{P}^∞ . Ademais, tem-se que, no limite (vide π_∞), há uma grande probabilidade de que a cadeia sorteie o estado 2, pois, como se vê, $\mathbf{Prob}(\theta_k = 2 | \theta_{k-1}) \approx 8/9$, enquanto que a probabilidade de sortear o estado 1 é $\mathbf{Prob}(\theta_k = 1 | \theta_{k-1}) \approx 1/9$.

No gráfico abaixo constam os sorteios realizados da cadeia de $k = 1$ até $k = 200$.

Figura 7 – Duas realizações da cM $\{\mathcal{N}, \mathbb{P}, \pi_0\}$.

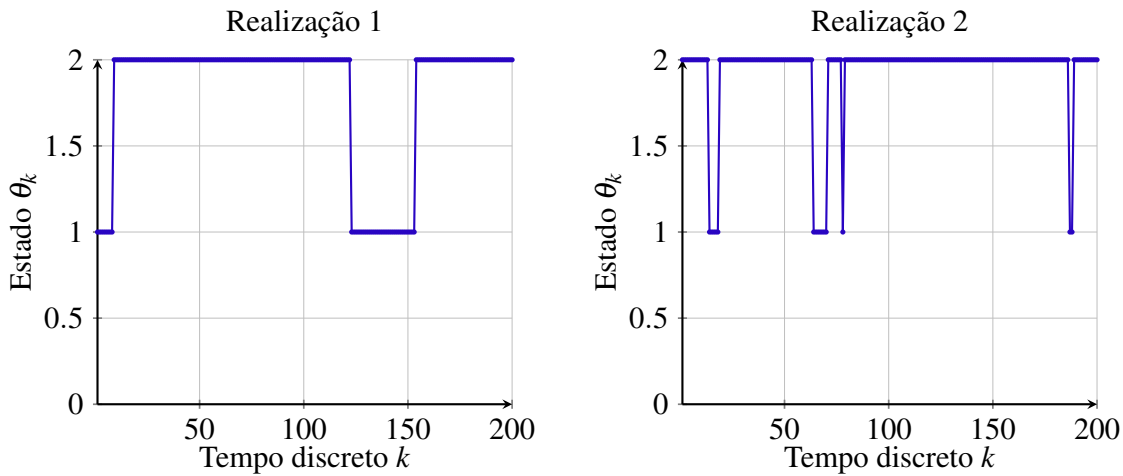
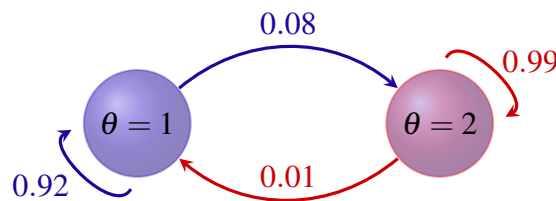


Figura 8 – Grafo representando a cM.



Fonte: Elaborada pelo autor.

Uma propriedade importante das cM's, chamada de Propriedade de Markov, é que elas possuem dependência apenas na vizinhança, ou seja, as cadeias não dependem de todo o histórico, mas apenas do instante anterior:

$$\mathbf{Prob}(\theta_k | \theta_{k-1}, \theta_{k-2}, \dots, \theta_0) = \mathbf{Prob}(\theta_k | \theta_{k-1}).$$

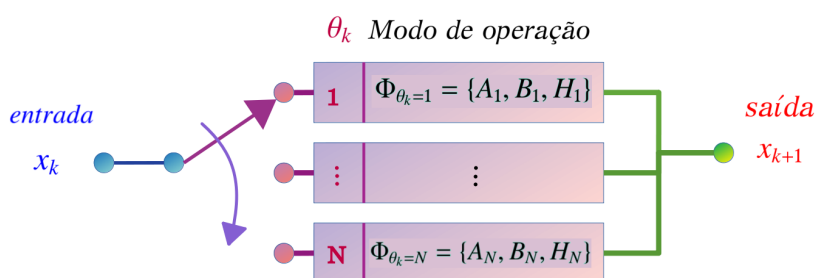
2.2 Sistemas Lineares com Saltos Markovianos

Os chamados Sistemas Lineares com Saltos Markovianos (SLSMs) foram introduzidos por [Dynkin \(1965\)](#) e têm sido usado para modelar sistemas dinâmicos lineares que mudam de comportamento bruscamente (salto). Uma cM é utilizada para modelar os saltos nos parâmetros, de modo que, num instante discreto k em que a cadeia esteja no estado θ_k , o SLSM utiliza um conjunto de parâmetros Φ_{θ_k} . Os exemplos numéricos que virão mais adiante ilustram a ideia de um SLSM.

O estado de um SLSM possui dois componentes: um *estado* contínuo em \mathbb{R}^n que representa a variável do sistema dinâmico (denotado pelo vetor x_k) e um *estado* discreto em

\mathcal{N} representando a cM (denotado por θ_k), de modo que a cada instante de tempo discreto k , o estado completo do sistema é (x_k, θ_k) . O vetor x pode representar, por exemplo, grandezas físicas do sistema real como posição, velocidade, aceleração, corrente elétrica, etc., os quais variam no tempo de acordo com um modelo linear. Os θ_k representam o modo de operação do sistema dinâmico, como por exemplo, modo “normal”, “falha”, “fácil”, “difícil”, “compra”, “venda”, etc., para os quais os parâmetros Φ_θ precisam refletir essas nomenclaturas. Veja o esquema da Figura 9.

Figura 9 – Esquema do SLSM: θ_k é aleatório conforme uma cM.



Fonte: Elaborada pelo autor.

Em se tratando dos SLSMs, existem dois problemas de controle importantes: o problema de **Rastreamento de Alvos Dinâmicos**, cujo objetivo é controlar o sistema de modo que o estado contínuo x_k se aproxime de uma trajetória $\{\bar{x}_k\}_{k \in \mathcal{H}}$ previamente definida. Outro problema, chamado de **Problema de Regulação** ou Regulador Linear Quadrático (RLQ), é um caso particular do primeiro, em que a trajetória desejada é a origem ($\bar{x}_k = 0 \forall k \in \mathcal{H}$). Vargas (2004) e Val e Başar (1999) estudam os dois casos.

Quando o valor de x_k é lido do sistema do mundo real através de sensores, bem como cada um dos θ_k , diz-se que o SLSM tem o **estado completamente observado**, (x_k, θ_k) . Se isso acontece, o controle ótimo para esse SLSM é bem conhecido na literatura, tem solução direta por meio de **Equações de Riccati Acopladas**. Entretanto, quando não se tem acesso à cM (θ_k é totalmente desconhecido, exceto pela sua distribuição inicial π_0), diz-se que o SLSM tem o **estado parcialmente observado**, (x_k) , e a abordagem anterior já não funciona. Nesse cenário, o método iterativo baseado em procedimento variacional proposto por Val e Başar (1999) e incrementado por Vargas (2004) vem sendo empregado.

Nem sempre é possível e às vezes é pouco viável mensurar o valor de θ em muitas aplicações reais, de modo que seria interessante obter uma forma de calcular o controle para esse cenário com estado parcialmente observado (θ desconhecido). Já existem na literatura trabalhos e métodos para resolver esse tipo de problema com algumas variações de cenário ou abordagem, como se pode ver, por exemplo, em Val e Başar (1999), Vargas (2004), Bortolin (2012), Silva (2012), Oliveira (2014). Dentre todos estes, o Método Variacional vem se mostrando um dos melhores em termos de tempo e qualidade de solução (custo).

Definição 1 (Espaço vetorial $\mathbb{V}_{m \times n}^{\mathcal{N}}$). Defina o seguinte espaço vetorial de coleções de matrizes: $\mathbb{V}_{m \times n}^{\mathcal{N}} = \{U_i; U_i \in \mathbb{R}^{m \times n} \forall i \in \mathcal{N} = \{1, \dots, N\}\}$. Caso as matrizes sejam quadradas, faça $m = n$. Considere também o operador traço de matriz $\text{tr}(\cdot)$. Será interessante definir uma multiplicação entre elementos de espaços distintos: $\bullet : (\mathbb{V}_{n \times m}^{\mathcal{N}}, \mathbb{R}^{m \times n}) \mapsto \mathbb{V}_{n \times n}^{\mathcal{N}}$ ou $\bullet : (\mathbb{R}^{m \times n}, \mathbb{V}_{n \times m}^{\mathcal{N}}) \mapsto \mathbb{V}_{m \times m}^{\mathcal{N}}$. Assim, defina a soma $X + Y$, produto $X \bullet Y$ e produto interno $\langle X, Y \rangle$ entre pontos desse espaço vetorial conforme

$$\left\{ \begin{array}{l} [X + Y]_i = X_i + Y_i, \quad \forall i \in \mathcal{N}, X, Y \in \mathbb{V}_{m \times n}^{\mathcal{N}} \text{ (soma)}, \\ [XY]_i = X_i Y_i, \quad \forall i \in \mathcal{N}, X \in \mathbb{V}_{m \times n}^{\mathcal{N}}, Y \in \mathbb{V}_{n \times m}^{\mathcal{N}} \text{ (produto)}, \\ [B \bullet G]_i = B_i G, \quad \forall i \in \mathcal{N}, B \in \mathbb{V}_{n \times m}^{\mathcal{N}}, G \in \mathbb{R}^{m \times n} \text{ (produto)}, \\ \langle X, Y \rangle = \sum_{i \in \mathcal{N}} \text{tr}(X_i' Y_i), \quad X, Y \in \mathbb{V}_{m \times n}^{\mathcal{N}} \text{ (produto interno)}. \end{array} \right. \quad (2.3)$$

Simplificando, as operações são feitas matriz por matriz, cada uma com seu índice correspondente $i \in \mathcal{N}$. No caso de \bullet , é como se fossem duas coleções sendo multiplicadas matriz a matriz, mas que uma dessas coleções tem todas as matrizes repetidas; no exemplo $B \bullet G$, G é como se fosse uma coleção de matrizes repetidas. Assim, ainda que essa notação não seja usual, é intuitiva, bastando observar os espaços das variáveis.

Observação 2 (Abusos de notação sobre coleções de matrizes). No intuito de esclarecer a notação utilizada a seguir, fazem-se algumas explicações dos abusos de notação. Inicialmente, defina o conjunto $\mathcal{N} = \{1, 2\}$ para exemplificar. Tenha o leitor em mente de que serão utilizadas coleções de matrizes, como por exemplo $A \in \mathbb{V}_{n \times n}^{\mathcal{N}}$, de modo que, quando se tratar de uma matriz específica dessa coleção, indica-se um índice como em A_i . Exemplo: $A = \{A_1, A_2\}$, mas quando se tratar de uma dessas matrizes especificamente, atribui-se-lhe um índice $i \in \mathcal{N}$, de modo que A representa toda a coleção e A_2 representa a segunda matriz da coleção. A notação será análoga para outras coleções de matrizes, como B, H, C e D . Essa notação com apenas um índice é usada para coleções invariantes no tempo, sendo pontos de $\mathbb{V}_{m \times n}^{\mathcal{N}}$ para m e n adequados.

Existem algumas coleções de matrizes no modelo que formam uma sequência/trajetória no tempo, e por isso insere-se mais um índice $k \in \mathcal{H} = \{0, 1, 2, 3\}$ (por exemplo) para especificar qual o instante de tempo a que se refere, além do índice markoviano $i \in \mathcal{N}$, como por exemplo $X = \{X_i^k\}_{i \in \mathcal{N}}^{k \in \mathcal{H}} = \{\{X_i^0\}, \{X_i^1\}, \{X_i^2\}, \{X_i^3\}\}_{i \in \mathcal{N}}$, em que X é toda a trajetória de coleções, $X^k \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ é a coleção no instante k , $X_i^k \in \mathbb{R}^{n \times n}$ é a matriz i da coleção X^k . Sendo assim, X^2 denota a coleção de matrizes $X^2 = \{X_1^2, X_2^2\}$. A notação será análoga para outras matrizes, como $L, \Sigma, \sigma, \bar{A}, \bar{C}$ e π . Essa notação com os índices $k \in \mathcal{H}$ será então usada para coleções dinâmicas no tempo.

Pensando em um algoritmo iterativo, que é o caso do MV, ainda pode-se adicionar mais um índice η indicando a iteração do método, de modo que, por exemplo, X seja um *array* pentadimensional, pois tem dimensão $n \times n \times N \times T \times \text{maxIt}$, em que maxIt é o número

De modo que A é um *array* tridimensional e A_i é um *array* bidimensional.

De modo que X é um *array* quadridimensional, X^k é um *array* tridimensional e X_i^k é bidimensional.

máximo de iterações do MV. Por outro lado, algumas matrizes do modelo não têm componente markoviana, que é o caso das matrizes de ganho G , ainda que sejam dinâmicas no tempo. Assim, os índices de G são apenas o tempo k e a iteração η .

É dada abaixo a definição de um modelo discreto para um RLQ com horizonte de planejamento finito $\mathcal{H} = \{0, 1, \dots, T\}$, objeto de estudo desta dissertação.

Definição 2 (Modelo RLQ). Considere $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ e $\omega_k \in \mathbb{R}^p$, em que ω_k são vetores aleatórios i.i.d. independentes das demais variáveis do sistema dinâmico, de média zero e matriz de covariância igual à identidade. Considere ainda a cadeia de Markov $\{\mathcal{N}, \mathbb{P}, \pi_0\}$ que modela os saltos dos parâmetros Φ_θ , com $\mathcal{N} = \{1, \dots, N\}$. Além disso, sejam as matrizes $A, C \in \mathbb{V}_{n \times n}^{\mathcal{N}}$, $B \in \mathbb{V}_{n \times m}^{\mathcal{N}}$, $H \in \mathbb{V}_{n \times p}^{\mathcal{N}}$ e $D \in \mathbb{V}_{m \times m}^{\mathcal{N}}$, de modo a formar os conjuntos de parâmetros $\Phi_\theta = \{A_\theta, B_\theta, H_\theta, C_\theta, D_\theta\}$ em que $C_i \geq 0$ (positiva semidefinida) e $D_i > 0$ (positiva definida) para cada $i \in \mathcal{N}$. Assim, para cada $k \in \mathcal{H}$, o modelo do RLQ é

$$\mathbb{H}_0 \begin{cases} x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k + H_{\theta_k} \omega_k, & (2.4a) \\ z_k = x_k' C_{\theta_k} x_k + u_k' D_{\theta_k} u_k, & (2.4b) \\ x_0 \text{ dado, } \theta_0 \sim \pi_0, \mathbb{P}, \theta_k \in \mathcal{N}, k \in \mathcal{H}, \end{cases}$$

em que u_k representa o controle aplicado ao sistema dinâmico. A saída z_k é um custo associado à penalização da norma de x_k e da respectiva ação de controle u_k e as matrizes simétricas $C_i = C_i'$ e $D_i = D_i'$ são os pesos da penalização, exigindo-se portanto a positividade de C_i e D_i para todo $i \in \mathcal{N}$. Exige-se $C \geq 0$ para que o custo de $\|x_k\|$ não seja negativo, e $D > 0$ para que, além de não permitir custo negativo para $\|u_k\|$, não permita que $\|u_k\| \rightarrow$ com custo zero (controle infinito sem custo). A matriz de probabilidade de transição da cadeia de Markov é dada pela matriz $\mathbb{P} \in [0, 1]^{N \times N}$ conforme descrito na [Equação 2.1](#), e contém probabilidades condicionais com relação a θ_k . O vetor $\pi_0 \in [0, 1]^N$ é a distribuição de probabilidade de θ_0 . Essa distribuição π_k é facilmente calculada para qualquer instante de tempo pela [Equação 2.2](#). Desse modelo, os parâmetros $\{\mathcal{N}, \mathbb{P}, \pi_0\}$ da cM e Φ_θ para todo $\theta \in \mathcal{N}$ do SLSM são todos conhecidos a priori.

Uma vez que não se tem acesso à informação de θ_k , o controle deve se basear apenas no conhecimento de x_k e considerar a distribuição de probabilidade de θ_k , que é facilmente calculada pela [Equação 2.2](#). Para o modelo \mathbb{H}_0 , a equação de realimentação do controle pode ser escrita como uma função linear do estado contínuo, o que simplifica a implementação. Se o modelo \mathbb{H}_0 fosse outro, a equação de realimentação poderia não ser linear, como no caso em que o modelo considera entradas exógenas, para o qual a realimentação é uma função afim. No entanto, para \mathbb{H}_0 o controle é linear e escreve-se

$$u_k = G_k x_k, \quad (2.5)$$

em que as matrizes $G_k \in \mathbb{R}^{m \times n}$, $\forall k \in \mathcal{H} \setminus \{T\}$ são chamadas de **ganhos** do controle e não dependem de θ .

Independentes e identicamente distribuídos.

Caso não seja a matriz identidade, mais adiante, a seu tempo, será indicada a alteração necessária.

Quando se tem informação completa do estado (x_k, θ_k) , usa-se um controle linear dependente de θ da forma $u_k = G_k^{\theta_k} x_k$, encontrado nas Equações de Riccati Acopladas.

Como pode-se notar até agora, o modelo é aleatório e pouco tratável. Por esse motivo, opta-se por reescrevê-lo de forma determinística. Esse modelo determinístico para cálculo do custo, que será tratado no **Capítulo 3**, é baseado no valor esperado da matriz de segundo momento de x dado θ , como em $\mathbf{E}(x_k x_k' \mathbf{1}_{\{\theta_k=i\}})$ (valor esperado de $x_k x_k'$ dado que $\theta_k = i$). Essa matriz de segundo momento de x_k dado $\theta_k = i$ será denotada por X_i^k . Ela possibilita o cálculo do valor esperado de $\|x_k\|^2$, bastando tomar o traço: $\mathbf{E}(\|x_k\|^2) = \sum_{i \in \mathcal{N}} \mathbf{E}(\|x_k \mathbf{1}_{\{\theta_k=i\}}\|^2) = \sum_{i \in \mathcal{N}} \mathbf{tr}(X_i^k)$.

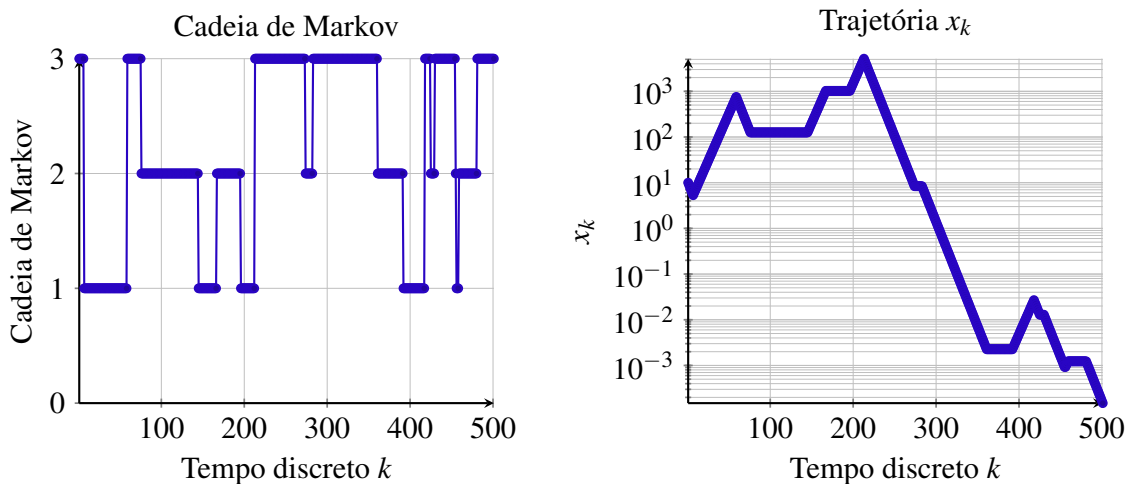
2.3 Exemplos numéricos

Nesta seção constam quatro exemplos numéricos unidimensionais para ilustrar o comportamento dos SLSMs: um exemplo sem controle, nem ruído aditivo; outro exemplo sem controle, mas com ruído aditivo; outro exemplo com um controle estabilizante escolhido sem nenhum critério específico, apenas para exemplificar, sem adição de ruído; e por fim um exemplo com ruído aditivo e controle. Nestes exemplos *não* foi utilizado o MV para cálculo da solução. Portanto, parâmetros C e D do modelo \mathbb{H}_0 não serão apresentados.

Exemplo 3. (Sem controle e sem ruído) Considere $n = m = p = 1$, ou seja, $x, u, \omega \in \mathbb{R}^1$, e a cadeia de Markov $\{\mathcal{N}, \mathbb{P}, \pi_0\}$, com $\mathcal{N} = \{1, 2, 3\}$, em que $\pi_0 = (0.3, 0.3, 0.4)'$. Os parâmetros Φ_θ são $A = \{1.1, 1, 0.9\}$, $B = \{0, 0, 0\}$ e $H = \{0, 0, 0\}$. É dado $x_0 = 10$.

$$\mathbb{P} = \begin{bmatrix} 0.96 & 0.02 & 0.02 \\ 0.02 & 0.96 & 0.02 \\ 0.02 & 0.02 & 0.96 \end{bmatrix} \quad \begin{array}{ll} A_1 = 1.1 & \text{(modo ampliar)} \\ A_2 = 1 & \text{(modo conservar)} \\ A_3 = 0.9 & \text{(modo reduzir)} \end{array} \quad \begin{array}{ll} B_1 = 0 & H_1 = 0 \\ B_2 = 0 & H_2 = 0 \\ B_3 = 0 & H_3 = 0 \end{array}$$

Figura 10 – Uma realização da cM e a respectiva trajetória de x_k .



Perceba na **Figura 10** o comportamento aleatório da cM e o respectivo efeito na trajetória do sistema dinâmico, o qual caminha “sem rumo” e se afasta bastante de zero.

Uma realização de x_k consiste em um sorteio da cM e do ruído ω_k quando houver, além de fazer a evolução do SLSM pelo modelo \mathbb{H}_0 .

Figura 11 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\|x_k\|^2 \mathbf{1}_{\{\theta_k\}})$.

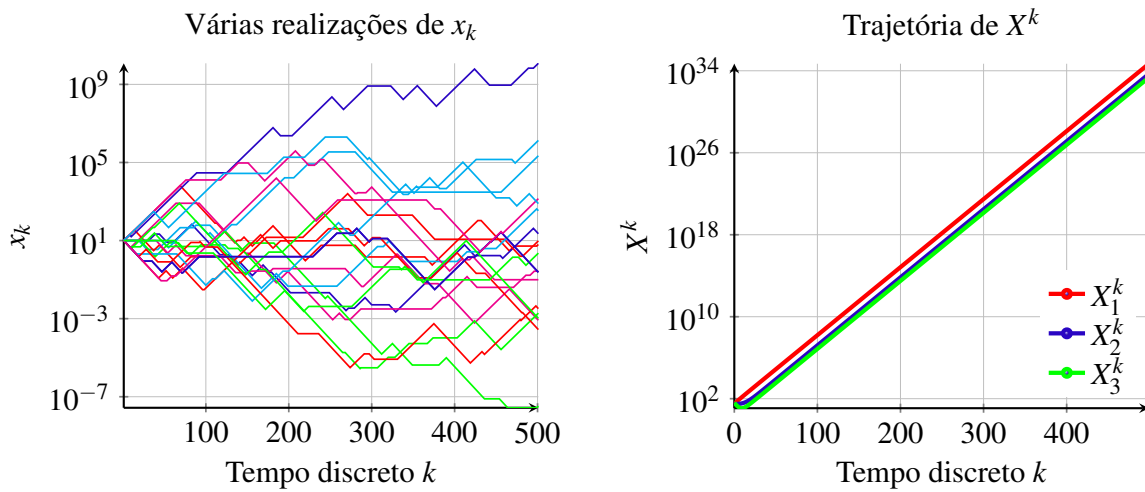
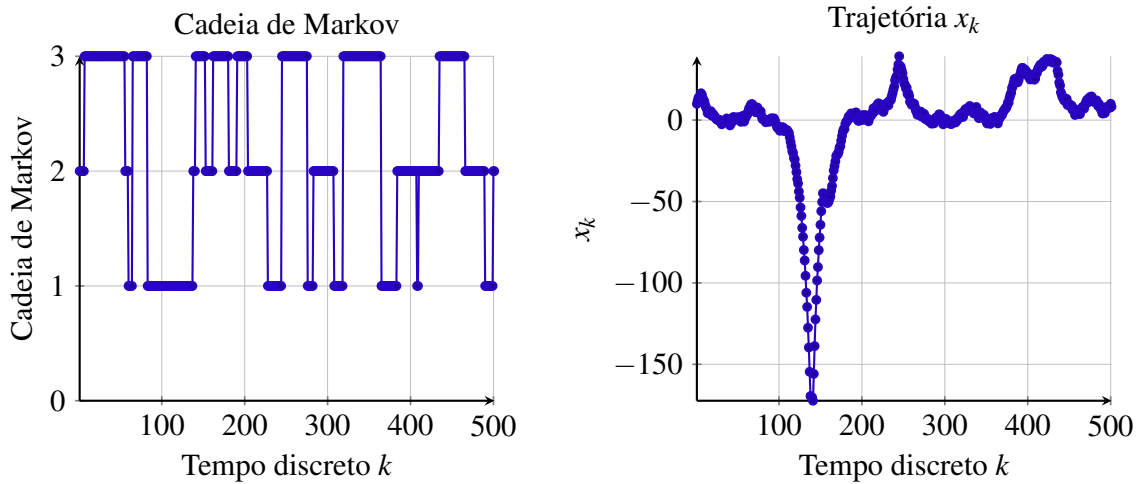


Figura 12 – Uma realização da cM e a respectiva trajetória de x_k .

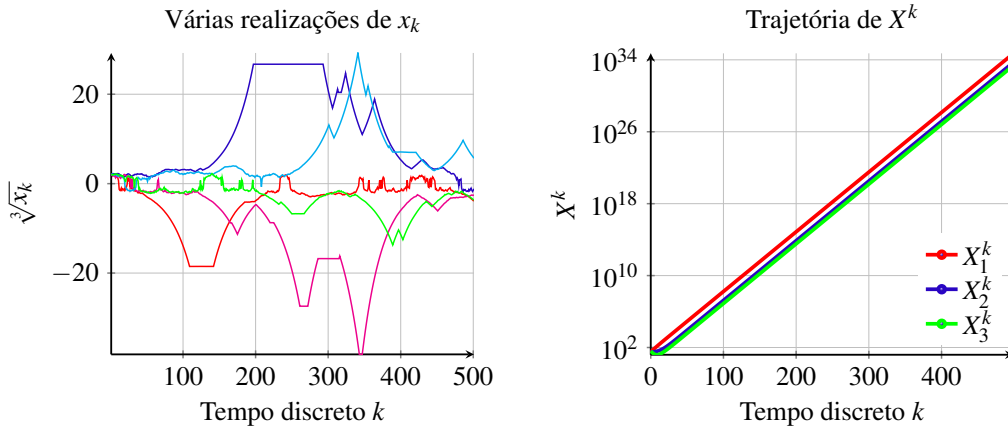


Já na [Figura 11](#) é possível perceber mais nitidamente esse comportamento aleatório do SLSM com as diversas realizações do sistema dinâmico, fazendo parecer uma ramificação a partir do ponto de origem. Note ainda que x_k às vezes pode ir para zero e às vezes pode tender ao infinito, pois o sistema não está sendo controlado. Por fim, veja o gráfico do valor esperado da norma quadrática de x_k para cada um dos estados markovianos. Note que para todo θ , X_θ^k tende ao infinito, pois a trajetória não está sendo corrigida (controlada).

Exemplo 4. (Sem controle e com ruído) Considere o mesmo problema do [Exemplo 3](#). Mas agora com $H = \{0.1, 0.3, 0.5\}$ (Φ é outro conjunto de parâmetros). É dado $x_0 = 10$. Além da aleatoriedade da cM, o SLSM representado na [Figura 12](#) tem um ruído de média zero e variância 1. Com essa nova incerteza, o sistema que passava apenas por números positivos agora pode passar aos negativos, como se observa no gráfico. Note o afastamento do zero a partir de $k = 100$, uma vez que o sistema não está sendo controlado.

O valor esperado da norma quadrática de x_k também tende ao infinito neste exemplo, por não haver controle de sua trajetória, como se vê na [Figura 13](#).

Figura 13 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\|x_k\|^2 \mathbf{1}_{\{\theta_k\}})$.



Exemplo 5. (Com controle e sem ruído) Considere ainda o mesmo problema do [Exemplo 3](#), mas com os parâmetros $B = \{1, 3, 2\}$ e $H = \{0, 0, 0\}$. É dado $x_0 = 10$ e um ganho constante no tempo $G_k = -0.3 \forall k$ (não é o ganho ótimo):

Figura 14 – Uma realização da cM e a respectiva trajetória de x_k .

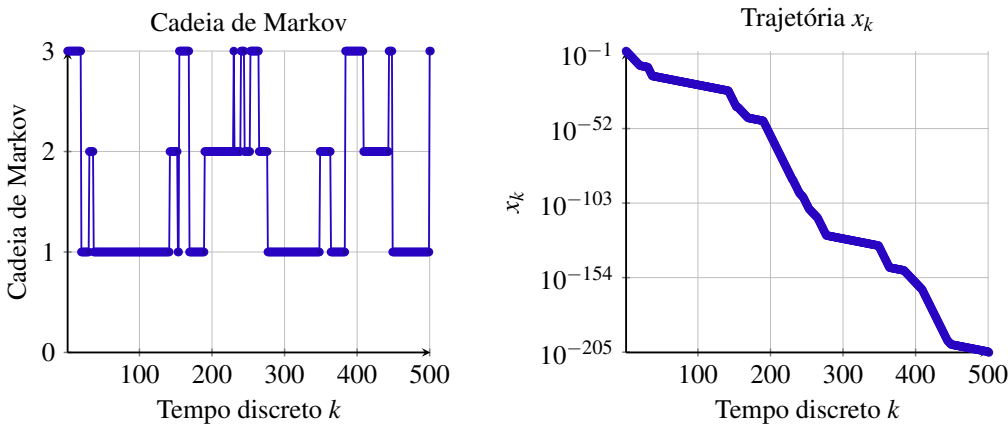
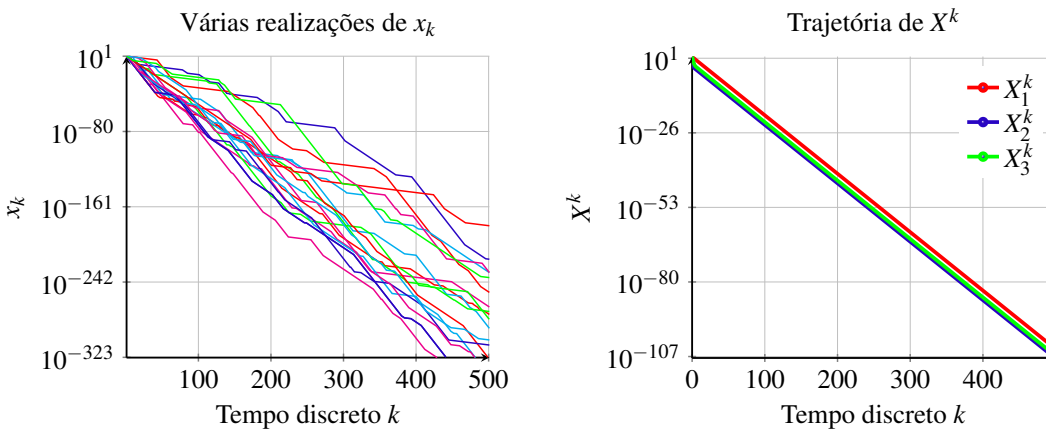


Figura 15 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\|x_k\|^2 \mathbf{1}_{\{\theta_k\}})$.



Neste exemplo, é notória a convergência do sistema para zero, uma vez que o ganho G_k utilizado estabiliza o sistema. Como não há ruído, o controle aplicado faz a trajetória convergir para a origem, como se identifica na [Figura 14](#). O fato é mais evidente olhando as diversas realizações da cM na [Figura 15](#), bem como X^k .

Exemplo 6. (Com controle e ruído) Considere o problema do [Exemplo 3](#). Os parâmetros Φ_θ são $A = \{1.1, 1, 0.9\}$, $B = \{1, 3, 2\}$ e $H = \{0.1, 0.3, 0.5\}$. É dado $x_0 = 10$ e um ganho constante no tempo $G_k = -0.3 \forall k$ (não é o ganho ótimo):

Figura 16 – Uma realização da cM e a respectiva trajetória de x_k .

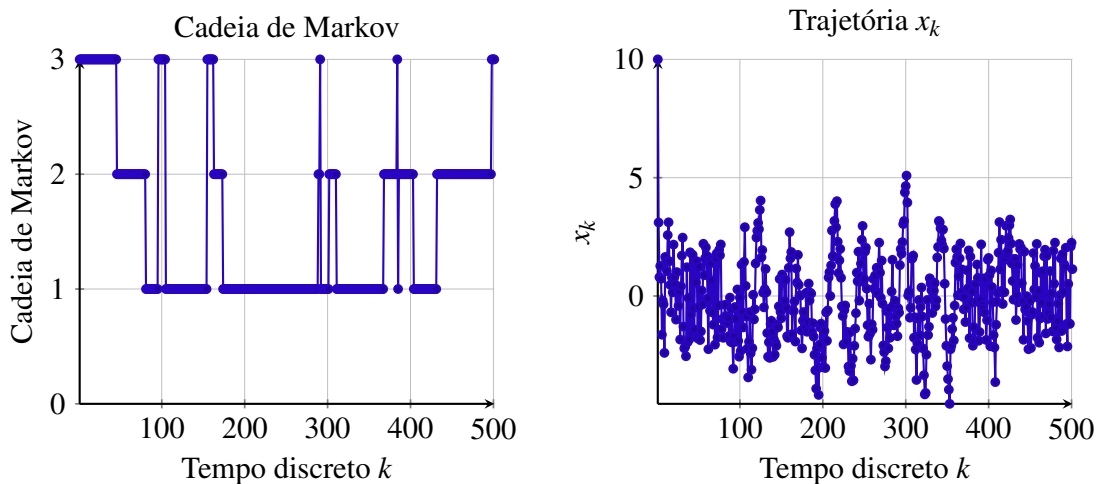
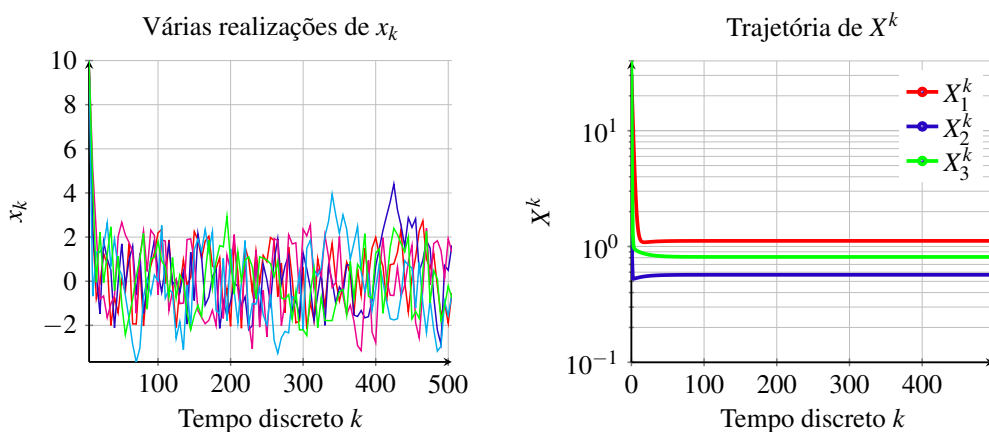


Figura 17 – A trajetória do SLSM para várias simulações da cM e o valor esperado $\mathbf{E}(\|x_k\|^2 \mathbf{1}_{\{\theta_k\}})$.



Neste último cenário, em que há ruído adicionado ao sistema, percebe-se que o ganho G_k escolhido estabiliza-o. Veja na [Figura 16](#) que o sistema controlado permanece orbitando a origem, pois o controle corrige a trajetória toda vez que o sistema tenta se afastar dela.

Na [Figura 17](#) pode-se verificar várias trajetórias diferentes para o sistema controlado, todas com comportamento semelhante. Note que a norma quadrática de x_k dessa vez não tende a zero, mas percebe que é limitada, diferentemente dos casos sem controle, em que tendia ao infinito.

Veja que nos dois exemplos com controle, este não necessitou da informação de θ , pois foi mantido constante ao longo do horizonte, não importando o valor da cM, de forma que $G_k = -0.3$ é um ganho estabilizante para o cenário sem observação dos saltos markovianos. O ganho que costuma-se chamar de ótimo é aquele que estabiliza o sistema com o mínimo de energia possível. Outros ganhos que estabilizam o sistema são considerados subótimos.

Observação 3 (Estabilidade). Perceba nos exemplos anteriores que quando a média $X_{\theta_k}^k = \mathbf{E}(x_k x_k' \mathbb{1}_{\{\theta_k\}})$ permanece limitada ao longo do horizonte $k \in \mathcal{H}$, então o SLSM permanece limitado (estável), como é o caso do [Exemplo 5](#) e do [Exemplo 6](#), fato que é o objetivo final do controle RLQ. Quando isso não acontece, o sistema é instável, possibilitando trajetórias divergentes, como é o caso do [Exemplo 3](#) e do [Exemplo 4](#).

MODELO DETERMINÍSTICO E MÉTODO VARIACIONAL

Neste capítulo é tratado o modelo determinístico baseado no valor esperado da matriz de segundo momento de x , bem como aponta uma condição necessária de otimalidade já bastante explorados na literatura, de modo que serão apresentados em formulários. O método variacional é descrito e explicado, bem como apresentado um algoritmo ilustrando-o.

3.1 O custo determinístico

É interessante notar que, ao se considerar a [Equação 2.5](#) no modelo RLQ da [Definição 2](#), pode-se trocar a variável u_k por G_k . Assim, dado um ganho G_k e definindo as coleções auxiliares

$$\bar{\mathbf{A}}^k = A + B \bullet G_k \quad \text{e} \quad \bar{\mathbf{C}}^k = C + G_k' \bullet D \bullet G_k, \quad (3.1)$$

(chamadas **matrizes de malha fechada** ou em inglês *closed loop*), o sistema \mathbb{H}_0 da [Equação 2.4a](#) – [2.4b](#) passa a ser escrito como

$$\mathbb{H}_1 \begin{cases} x_{k+1} = \bar{\mathbf{A}}_{\theta_k}^k x_k + H_{\theta_k} \omega_k, & (3.2a) \\ z_k = x_k' \bar{\mathbf{C}}_{\theta_k}^k x_k, & (3.2b) \end{cases}$$

em que a ação de controle é dada por G_k e já está “embutida” em $\bar{\mathbf{A}}$ e $\bar{\mathbf{C}}$.

Definição 1. Definem-se alguns operadores que serão utilizados no modelo mais adiante. Considere as coleções de matrizes $U \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ e $V \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ arbitrárias. Assim, definem-se

$$\mathcal{T}_V^i(U) = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} V_j U_j V_j', \quad (3.3a)$$

$$\mathcal{E}^i(U) = \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} U_j, \quad (3.3b)$$

$$\mathcal{L}_V^i(U) = V_i' \mathcal{E}^i(U) V_i. \quad (3.3c)$$

O resultado a seguir será deixado como um formulário, pois na literatura já tem várias menções do mesmo com notação diversa.

Definição 2 (Política de controle e variável X). Dada uma sequência de ganhos $\{G|_0^{T-1}\} = \{G_0, G_1, \dots, G_{T-1}\}$ chamada de **política de controle**, para se obter o valor esperado $\mathbf{E}(x_k x_k' \mathbf{1}_{\{\theta_k=i\}}) = X_i^k$ ao longo de $k \in \mathcal{H}$, usando-se a [Equação 3.3a](#), tem-se para cada $i \in \mathcal{N}$:

$$\begin{cases} X_i^0 = x_0 x_0' \pi_0^i, & (3.4a) \\ X_i^{k+1} = \mathcal{T}_{\tilde{\mathbf{A}}^k}^i(X^k) + \Sigma_i^k, & (3.4b) \\ \Sigma_i^k = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_k^j H_j H_j', & (3.4c) \end{cases}$$

de forma que as matrizes Σ_i^k representam a covariância do ruído aditivo ω_k ponderada pela distribuição de θ_k . Note a matriz de malha fechada $\tilde{\mathbf{A}}$ na [Equação 3.4b](#).

Observação 4. Algumas propriedades imediatas dessa definição são que X_i^k e Σ_i^k são matrizes simétricas e positivas semidefinidas para todo $i \in \mathcal{N}$ e para todo $k \in \mathcal{H}$ (ou seja, $X = X' \geq 0$ e $\Sigma = \Sigma' \geq 0$). Além disso, X^0 é uma coleção de matrizes de posto 1, e portanto singular.

Definição 3 (Custo total determinístico). Dada uma política de controle $\{G|_0^{T-1}\}$ arbitrária, considerando a [Equação 3.1](#) e a [Equação 2.3](#), calcula-se o custo total \mathbf{J} do horizonte de planejamento \mathcal{H} dessa política de forma determinística pela seguinte equação:

$$\mathbf{J}(\{G|_0^{T-1}\}) = \sum_{k \in \mathcal{H} \setminus \{T\}} J_k = \sum_{k \in \mathcal{H} \setminus \{T\}} \langle X^k, \bar{\mathbf{C}}^k \rangle. \quad (3.5)$$

em que o valor esperado de z_k da [Equação 3.2b](#) é dado por

$$J_k = \mathbf{E}(z_k) = \mathbf{E}(x_k' \bar{\mathbf{C}}_{\theta_k}^k x_k) = \sum_{i \in \mathcal{N}} \text{tr}(x_k x_k' \bar{\mathbf{C}}_i^k \mathbf{1}_{\{\theta_k=i\}}) = \langle X^k, \bar{\mathbf{C}}^k \rangle.$$

Para resolver o problema de controle RLQ, cuja dimensão é de $m \times n \times T$, quebra-se-lhe em subproblemas de dimensão $m \times n$. Para tanto, introduzem-se mais duas variáveis L e σ para calcular o custo total \mathbf{J} .

Definição 3 (Custo total). Dada uma política de controle $\{G|_\ell^{T-1}\}$, considerando a soma dos custos de J_ℓ a J_{T-1} (chamada **custo de continuação a partir de $k = \ell$**) e as equações (3.1) e (2.2), tem-se

$$J|_\ell^{T-1} = \sum_{k=\ell}^{T-1} J_k = \langle X^\ell, L^\ell \rangle + \pi_\ell' \sigma^\ell, \quad (3.6)$$

Veja por exemplo [Bortolin \(2012, Proposição 2.1\)](#) ou [Oliveira \(2014, Proposição 2.1\)](#).

Caso a covariância do ruído seja W em vez da matriz identidade (veja a [Definição 2](#)), modifica-se o modelo acima para $\Sigma_i^k = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_k^j H_j W H_j'$.

demonstrado na [Equação A.5](#), considerando-se o [Definição 1](#), em que $L^k \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ e $\sigma^k \in \mathbb{V}_{1 \times 1}^{\mathcal{N}}$ para todo $k \in \mathcal{H}$, e:

$$\begin{cases} L_i^T = 0_{n \times n}, & \sigma_i^T = 0, & (3.7a) \\ L_i^k = \bar{C}_i^k + \mathcal{L}_{\bar{A}^k}^i(L^{k+1}), & & (3.7b) \\ \sigma_i^k = \mathbf{tr}(\mathcal{E}^i(L^{k+1})H_iH_i') + \mathcal{E}^i(\sigma^{k+1}). & & (3.7c) \end{cases}$$

Nota-se então, que o custo da política $\{G|_0^{T-1}\}$ definido pelo somatório em [\(3.5\)](#) é dado pelo produto interno das coleções X , L , π e σ , conforme

$$\mathbf{J}(\{G|_0^{T-1}\}) = \langle X^0, L^0 \rangle + \pi'_0 \sigma^0. \quad (3.8)$$

Observação 5. Uma propriedade imediata dessa definição é que L_i^k é matriz simétrica positiva semidefinida para todo $i \in \mathcal{N}$ e para todo $k \in \mathcal{H}$ (ou seja, $L = L' \geq 0$). Diferentemente de X que evolui no tempo usando o operador $\mathcal{T}_{(\cdot)}(\cdot)$, note a regressão no tempo para L usando o operador $\mathcal{L}_{(\cdot)}(\cdot)$ e σ , ou seja, inicia-se com $k = T$ e volta-se no tempo em direção a $k = 0$.

3.2 Método variacional

Definidos todos os elementos necessários para entender o problema e a formulação determinística para o cálculo do custo, apresentam-se os principais resultados deste capítulo e o algoritmo variacional.

Definição 4 (Problema variacional). O problema variacional consiste em minimizar o custo de continuação dado pela [Equação 3.6](#), para cada $k = T - 1, T - 2, \dots, 1, 0$:

$$\begin{cases} \min_{G_k \in \mathbb{R}^{m \times n}} J|_k^{T-1} \\ \text{sujeito às equações da Definição 2 e da Definição 3.} \end{cases} \quad (3.9)$$

Uma condição necessária de otimalidade da forma “derivadas parciais iguais a zero” aplicada ao custo de continuação permite escrever o próximo resultado.

Proposição 1 (Otimalidade). Seja $\Lambda_i^{k+1} = \mathcal{E}^i(L^{k+1})$. O ganho G_k que resolve o problema variacional dado na [Equação 3.9](#) é solução do sistema linear:

$$\sum_{i \in \mathcal{N}} [(D_i + B_i' \Lambda_i^{k+1} B_i) G_k + B_i' \Lambda_i^{k+1} A_i] X_i^k = 0. \quad (3.10)$$

A demonstração pode ser vista na [Equação A.8](#).

Para resolver o sistema linear acima, usa-se sua forma vetorizada:

$$\sum_{i \in \mathcal{N}} [X_i^k \otimes (D_i + B_i' \Lambda_i^{k+1} B_i)] \mathbf{vec}(G_k) = - \sum_{i \in \mathcal{N}} \mathbf{vec}(B_i' \Lambda_i^{k+1} A_i X_i^k), \quad (3.11)$$

Esse é o sistema linear \mathbb{S}_k^η referido na introdução.

em que o operador $\text{vec}(\cdot)$ empilha as colunas de uma matriz, formando um vetor, e \otimes representa o produto de *Kronecker*. Veja o exemplo de \otimes e $\text{vec}(\cdot)$ abaixo:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \text{e} \quad B = \begin{bmatrix} b_{11} & b_{12} \end{bmatrix},$$

$$A \otimes B = \begin{bmatrix} [a_{11}B] & [a_{12}B] \\ [a_{21}B] & [a_{22}B] \end{bmatrix} = \begin{bmatrix} [a_{11}b_{11} & a_{11}b_{12}] & [a_{12}b_{11} & a_{12}b_{12}] \\ [a_{21}b_{11} & a_{21}b_{12}] & [a_{22}b_{11} & a_{22}b_{12}] \end{bmatrix},$$

$$\text{vec}(A) = \text{vec}\left(\begin{bmatrix} [a_{11}] & [a_{12}] \\ [a_{21}] & [a_{22}] \end{bmatrix}\right) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix}, \quad \text{vec}(B) = \text{vec}\left(\begin{bmatrix} [b_{11}] & [b_{12}] \end{bmatrix}\right) = \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix}.$$

Definição 5 (Vetorização e sua inversa). Defina $\text{vec}(\cdot) : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{mn \times 1}$, $\text{vec}(X) = x$ o operador que empilha as colunas X_r da matriz X , $r = 1, \dots, n$, formando um vetor $x \in \mathbb{R}^{mn \times 1}$. Defina $\text{vec}_{m,n}^{-1}(\cdot) : \mathbb{R}^{mn \times 1} \rightarrow \mathbb{R}^{m \times n}$, $\text{vec}_{m,n}^{-1}(x) = X$, o operador inverso de $\text{vec}(X)$, que recupera a matriz $X \in \mathbb{R}^{m \times n}$ a partir do vetor $x \in \mathbb{R}^{mn \times 1}$ (*reshape*).

Observação 6. O cálculo dos ganhos G_k não precisa de σ^k , pois $\partial \sigma^k / \partial G_k = 0$. Como a variável X precisará ser atualizada de $k = 0$ até $k = T$ a cada iteração η do MV, é mais fácil calcular o custo total \mathbf{J} somando as parcelas $\langle X^k, \bar{C}^k \rangle$ conforme a [Equação 3.5](#), em vez de usar a [Equação 3.8](#).

Observação 7. Das Observações 4 e 5, note que a matriz $X_i^k \otimes (D_i + B_i' \Lambda_i^{k+1} B_i)$ do sistema é simétrica, pois os fatores o são, e positiva semidefinida para todo $i \in \mathcal{N}$, pois, de acordo com [Zhou, Doyle e Glover \(1996, p. 25\)](#), os autovalores de $P \otimes Q$ são os respectivos produtos dos autovalores de P pelos autovalores de Q , que, no caso, são todos não negativos.

Observação 8. Os algoritmos [Algoritmo 1](#), [Algoritmo 2](#) e [Algoritmo 3](#) a seguir consideram indexação a partir de 1 e não a partir de zero. Elementos de dimensões iguais estão alocados multiplicando por zero um deles que já está na memória.

Algoritmo 1 – Método Variacional

- 1: **procedimento** VARIACIONAL($A, B, H, C, D, \mathbb{P}, x_1, \pi_1, N, T, \{G_1^{T-1}\}_{\eta=1}, n, m, \text{maxit}$)
 - 2: **global** A, B, C, D, \mathbb{P} ▷ Agilizar o acesso pelas outras funções
 - 3: $\eta \leftarrow 1$ ▷ Iteração do método
 - 4: **para** $i \leftarrow 1$ **até** N **faça** ▷ Calculando o primeiro X^k
 - 5: $X_i^{1,\eta} \leftarrow x_1 * x_1' * \pi_1^i$
 - 6: **fim para**
 - 7: **para** $k \leftarrow 1$ **até** $T - 1$ **faça** ▷ Calculando a covariância do ruído Σ^k
 - 8: $\pi_{k+1} \leftarrow \mathbb{P}' * \pi_k$ ▷ Calculando a distribuição π de θ
 - 9: **para** $i \leftarrow 1$ **até** N **faça**
 - 10: $\text{somaSigma} \leftarrow H_1 * H_1' * 0$
-

$$f(G) = f(\vec{a}) = \vec{a}' \mathbf{V}' \left[\sum_{i \in \mathcal{N}} X_i \otimes Z_i \right] \mathbf{V} \vec{a} + \vec{a}' \mathbf{V}' \mathbf{vec} \left(\sum_{i \in \mathcal{N}} 2M_i' X_i \right) + c.$$

Dessa forma, derivando a expressão acima em relação a \vec{a} e igualando a zero, tem-se o seguinte sistema linear:

$$\mathbf{V}' \left[\sum_{i \in \mathcal{N}} X_i \otimes Z_i \right] \mathbf{V} \vec{a} = \mathbf{V}' \mathbf{vec} \left(- \sum_{i \in \mathcal{N}} M_i' X_i \right), \quad (4.7)$$

onde a solução desejada para o ganho é

$$\mathbf{vec}(G) \stackrel{(4.6)}{=} \mathbf{V} \vec{a}.$$

4.2 Mudança de base para os ganhos G_k : segunda formulação

Sendo $x \in \mathbb{R}^n$ e $u \in \mathbb{R}^m$, e portanto $G \in \mathbb{R}^{m \times n}$, da [Equação A.6](#), tem-se

$$\mathbf{J} = f(G) = c + \underbrace{\sum_{i \in \mathcal{N}} \mathbf{tr} (X_i G' Z_i G)}_{\phi} + \underbrace{\sum_{i \in \mathcal{N}} 2 \mathbf{tr} (X_i M_i' G)}_{\psi}.$$

Para simplificar notação, define-se o conjunto $\mathcal{C} = \{1, \dots, mn\}$. Da [Equação 4.4](#), escrevendo G em uma nova base $\mathbf{K} = \{K_i | K_i \in \mathbb{R}^{m \times n} \forall i \in \mathcal{C}\}$, em que as matrizes K_i são linearmente independentes, vem

$$G = a_1 K_1 + a_2 K_2 + \dots + a_{mn} K_{mn} = \sum_{k \in \mathcal{C}} a_k K_k, \quad a_k \in \mathbb{R}.$$

Com essa nova representação do ganho G serão feitas contas com $m = 1$ e $n = 3$, mas o processo será generalizado para quaisquer m e n nas linhas seguintes.

Inicialmente, analisa-se a parcela ϕ :

$$\begin{aligned} \phi &= \sum_{i \in \mathcal{N}} \mathbf{tr} \left[X_i \left(\sum_{k=1}^3 a_k K_k \right)' Z_i \left(\sum_{k=1}^3 a_k K_k \right) \right] \\ &= \sum_{i \in \mathcal{N}} \mathbf{tr} \left[X_i (a_1 K_1 + a_2 K_2 + a_3 K_3)' Z_i (a_1 K_1 + a_2 K_2 + a_3 K_3) \right] \\ &= \sum_{i \in \mathcal{N}} \mathbf{tr} \left[X_i (a_1 a_1 K_1' Z_i K_1 + a_1 a_2 K_1' Z_i K_2 + a_1 a_3 K_1' Z_i K_3 + \right. \\ &\quad \left. + a_2 a_1 K_2' Z_i K_1 + a_2 a_2 K_2' Z_i K_2 + a_2 a_3 K_2' Z_i K_3 + \right. \\ &\quad \left. + a_3 a_1 K_3' Z_i K_1 + a_3 a_2 K_3' Z_i K_2 + a_3 a_3 K_3' Z_i K_3) \right] \end{aligned} \quad (4.8)$$

Exemplo: se $m = 3$, $n = 2$, então $\mathcal{C} = \{1, 2, 3, 4, 5, 6\}$.

Escolhendo um índice $p \in \mathcal{C}$ (por exemplo $p = 1$, como ilustrado em vermelho em (??)), destacando a parte de ϕ que depende de a_p , deixando as demais parcelas como constante c_{ip} :

$$\phi = \phi(a_p) = \sum_{i \in \mathcal{N}} \left[c_{ip} + a_p^2 \mathbf{tr}(X_i K'_p Z_i K_p) + 2a_p \sum_{\substack{q \in \mathcal{C} \\ q \neq p}} a_q \mathbf{tr}(X_i K'_p Z_i K_q) \right],$$

de modo que sua derivada em relação a a_p é

$$\begin{aligned} \frac{d\phi(a_p)}{da_p} &= \sum_{i \in \mathcal{N}} \left[2a_p \mathbf{tr}(X_i K'_p Z_i K_p) + 2 \sum_{\substack{q \in \mathcal{C} \\ q \neq p}} a_q \mathbf{tr}(X_i K'_p Z_i K_q) \right] \\ &= \sum_{i \in \mathcal{N}} \sum_{q \in \mathcal{C}} 2a_q \mathbf{tr}(X_i K'_p Z_i K_q). \end{aligned} \quad (4.9)$$

Agora fazendo as contas para a parcela ψ , vem:

$$\psi = \psi(a_p) = \sum_{i \in \mathcal{N}} 2 \mathbf{tr} \left[X_i M_i (a_p K_p + \sum_{\substack{q \in \mathcal{C} \\ q \neq p}} a_q K_q) \right] = \sum_{i \in \mathcal{N}} 2a_p \mathbf{tr}(X_i M_i K_p) + \sum_{\substack{q \in \mathcal{C} \\ q \neq p}} 2a_q \mathbf{tr}(X_i M_i K_q).$$

Escolhido um índice $p \in \mathcal{C}$, derivando ψ em relação a a_p , tem-se

$$\frac{d\psi(a_p)}{da_p} = \sum_{i \in \mathcal{N}} 2 \mathbf{tr}(X_i M_i K_p). \quad (4.10)$$

Voltando à expressão de $f(G)$,

$$\frac{df}{d\vec{a}} = \frac{d\phi}{d\vec{a}} + \frac{d\psi}{d\vec{a}},$$

em que $\vec{a} = (a_1, \dots, a_{mn})'$, de forma que, para cada $p \in \mathcal{C}$ se tem

$$\frac{\partial f}{\partial a_p} = \frac{\partial \phi}{\partial a_p} + \frac{\partial \psi}{\partial a_p}.$$

Fazendo o gradiente de f igual a zero, fica

$$\frac{\partial \phi}{\partial a_p} = -\frac{\partial \psi}{\partial a_p}, \quad \forall p \in \mathcal{C},$$

$$(4.9)(4.10) \implies 2 \sum_{i \in \mathcal{N}} \sum_{q \in \mathcal{C}} a_q \mathbf{tr}(X_i K'_p Z_i K_q) = - \sum_{i \in \mathcal{N}} 2 \mathbf{tr}(X_i M_i K_p), \quad \forall p \in \mathcal{C},$$

e isso dá o seguinte sistema linear:

$$\sum_{i \in \mathcal{N}} \begin{bmatrix} \mathbf{tr}(X_i K'_1 Z_i K_1) & \mathbf{tr}(X_i K'_1 Z_i K_2) & \dots & \mathbf{tr}(X_i K'_1 Z_i K_{mn}) \\ \mathbf{tr}(X_i K'_2 Z_i K_1) & \mathbf{tr}(X_i K'_2 Z_i K_2) & \dots & \mathbf{tr}(X_i K'_2 Z_i K_{mn}) \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{tr}(X_i K'_{mn} Z_i K_1) & \mathbf{tr}(X_i K'_{mn} Z_i K_2) & \dots & \mathbf{tr}(X_i K'_{mn} Z_i K_{mn}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{mn} \end{bmatrix} = - \sum_{i \in \mathcal{N}} \begin{bmatrix} \mathbf{tr}(X_i M_i K_1) \\ \mathbf{tr}(X_i M_i K_2) \\ \vdots \\ \mathbf{tr}(X_i M_i K_{mn}) \end{bmatrix}. \quad (4.11)$$

Se for definido $W_i = X_i \otimes Z_i$ e $\vec{K}_p = \mathbf{vec}(K_p)$, o sistema acima pode ser reescrito como

$$\sum_{i \in \mathcal{N}} \begin{bmatrix} \vec{K}'_1 W_i \vec{K}_1 & \vec{K}'_1 W_i \vec{K}_2 & \dots & \vec{K}'_1 W_i \vec{K}_{mn} \\ \vec{K}'_2 W_i \vec{K}_1 & \vec{K}'_2 W_i \vec{K}_2 & \dots & \vec{K}'_2 W_i \vec{K}_{mn} \\ \vdots & \vdots & \dots & \vdots \\ \vec{K}'_{mn} W_i \vec{K}_1 & \vec{K}'_{mn} W_i \vec{K}_2 & \dots & \vec{K}'_{mn} W_i \vec{K}_{mn} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{mn} \end{bmatrix} = - \sum_{i \in \mathcal{N}} \begin{bmatrix} \mathbf{tr}(X_i M_i K_1) \\ \mathbf{tr}(X_i M_i K_2) \\ \vdots \\ \mathbf{tr}(X_i M_i K_{mn}) \end{bmatrix}. \quad (4.12)$$

Ou ainda, vetorizando o segundo membro ao se definir $Y_i = M'_i X_i$ e $\vec{Y}_i = \mathbf{vec}(Y_i)$, o sistema finalmente chega a

$$\sum_{i \in \mathcal{N}} \begin{bmatrix} \vec{K}'_1 W_i \vec{K}_1 & \vec{K}'_1 W_i \vec{K}_2 & \dots & \vec{K}'_1 W_i \vec{K}_{mn} \\ \vec{K}'_2 W_i \vec{K}_1 & \vec{K}'_2 W_i \vec{K}_2 & \dots & \vec{K}'_2 W_i \vec{K}_{mn} \\ \vdots & \vdots & \dots & \vdots \\ \vec{K}'_{mn} W_i \vec{K}_1 & \vec{K}'_{mn} W_i \vec{K}_2 & \dots & \vec{K}'_{mn} W_i \vec{K}_{mn} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{mn} \end{bmatrix} = - \sum_{i \in \mathcal{N}} \begin{bmatrix} \vec{Y}'_i \vec{K}_1 \\ \vec{Y}'_i \vec{K}_2 \\ \vdots \\ \vec{Y}'_i \vec{K}_{mn} \end{bmatrix}. \quad (4.13)$$

Conclui-se assim essa segunda formulação da mudança de base

$$G = \sum_{k \in \mathcal{C}} a_k K_k.$$

Um ponto importante, que justifica as estratégias de condicionamento utilizadas neste trabalho, é que a ordem de precedência entre os operandos é numericamente importante, ainda que matematicamente sejam equivalentes. Trata-se do somatório na [Equação 4.7](#). Se o somatório for executado antes de se fazer a mudança de base, a solução é numericamente pior em relação àquela obtida fazendo-se primeiro a mudança de base. Esse fenômeno incentivou a busca por formulações alternativas para a mudança de base, como por exemplo na [Equação 4.12](#) e na [Equação 4.13](#). Com essas premissas, fazem-se as seguintes definições das técnicas de condicionamento.

Definição 7 (Auxiliares). No intuito de não repetir definições, além de esclarecer a notação

Veja como foi feito para que $\mathbf{tr}(X_i G' Z_i G) = \mathbf{vec}(G)' [X_i \otimes Z_i] \mathbf{vec}(G)$ na [Equação A.6](#) e [Equação A.7](#), bem como $\mathbf{tr}(X_i M_i G) = \mathbf{vec}(M'_i X_i)' \mathbf{vec}(G)$.

utilizada, defina

$$\left\{ \begin{array}{l} W_i = X_i \otimes Z_i \in \mathbb{R}^{mn \times mn}, \quad \forall i \in \mathcal{N} \quad (4.14a) \\ \sum_{i \in \mathcal{N}} W_i \stackrel{SVD}{=} U \Sigma V' \quad (4.14b) \\ \vec{K}_k = \mathbf{vec}(K_k), \quad K_k \in \mathbb{R}^{m \times n} \quad \forall k = 1, \dots, mn \quad (4.14c) \\ \mathbf{V}(\alpha) \stackrel{(4.2)}{=} U[\Sigma_+]^{-\frac{1}{\alpha}} = [\vec{K}_1 \mid \vec{K}_2 \mid \dots \mid \vec{K}_{mn}] = (\text{matriz}) \quad (4.14d) \\ \mathbf{vec}(G) = \mathbf{V}(\alpha) \vec{a} = (\text{solução}) \quad (4.14e) \\ G = \sum_{k=1}^{mn} K_k a_k = (\text{solução}) \quad (4.14f) \\ \vec{a} = (a_1, \dots, a_{mn})' \quad (4.14g) \\ \mathbf{K}(\alpha) = \left\{ K_k = \mathbf{vec}_{m,n}^{-1} \left(\mathbf{V}(\alpha)_{\text{coluna}=k} \right) \right\} = (\text{coleção de matrizes}) \quad (4.14h) \\ \vec{\mathbf{K}}(\alpha) = \left\{ \vec{K}_k = \mathbf{V}(\alpha)_{\text{coluna}=k} \right\} = (\text{coleção de vetores}) \quad (4.14i) \end{array} \right.$$

O sistema linear \mathbb{S}_k dado na [Equação 3.11](#) será reescrito aplicando-se preconditionadores. O novo sistema linear $\tilde{\mathbb{S}}_k$ possibilitará obter as coordenadas \vec{a} da solução G na nova base \mathbf{V} , \mathbf{K} ou $\vec{\mathbf{K}}$. Para recuperar a solução G , basta utilizar a [Equação 4.14e](#) e recuperar a matriz por meio de $G = \mathbf{vec}_{m,n}^{-1}(\mathbf{vec}(G))$ ou então diretamente somando as matrizes pela [Equação 4.14f](#). Assim, o objetivo agora é obter as coordenadas de G na nova base, e não mais G .

Definição 8 (Precondicionador 1). A mudança de base em $\mathbf{V}(\alpha)$ da [Equação 4.7](#) tem como preferência o operador somatório, usando a [Equação 4.14d](#):

$$\left[\mathbf{V}(\alpha)' \left[\sum_{i \in \mathcal{N}} W_i \right] \mathbf{V}(\alpha) \right] \vec{a} = \mathbf{V}(\alpha)' \mathbf{vec} \left(- \sum_{i \in \mathcal{N}} M_i' X_i \right). \quad (4.15)$$

Definição 9 (Precondicionador 2). A mudança de base em $\mathbf{V}(\alpha)$ (base de vetores) da [Equação 4.7](#) tem a preferência diante do operador somatório, com a [Equação 4.14d](#).

$$\left[\sum_{i \in \mathcal{N}} \mathbf{V}(\alpha)' W_i \mathbf{V}(\alpha) \right] \vec{a} = \mathbf{V}(\alpha)' \mathbf{vec} \left(- \sum_{i \in \mathcal{N}} M_i' X_i \right). \quad (4.16)$$

Definição 10 (Precondicionador 3). A mudança de base em $\mathbf{K}(\alpha)$ (base de matrizes) usando o operador traço explicitamente, conforme a [Equação 4.11](#), usando a [Equação 4.14h](#).

$$\sum_{i \in \mathcal{N}} \begin{bmatrix} \mathbf{tr}(X_i K_1' Z_i K_1) & \dots & \mathbf{tr}(X_i K_1' Z_i K_{mn}) \\ \vdots & \dots & \vdots \\ \mathbf{tr}(X_i K_{mn}' Z_i K_1) & \dots & \mathbf{tr}(X_i K_{mn}' Z_i K_{mn}) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_{mn} \end{bmatrix} = - \sum_{i \in \mathcal{N}} \begin{bmatrix} \mathbf{tr}(X_i M_i' K_1) \\ \vdots \\ \mathbf{tr}(X_i M_i' K_{mn}) \end{bmatrix}. \quad (4.17)$$

Definição 11 (Precondicionador 4). A mudança de base em $\vec{\mathbf{K}}(\alpha)$ e $\mathbf{K}(\alpha)$ usando o operador traço implicitamente, conforme a [Equação 4.12](#), com as equações [Equação 4.14h](#) e [Equação 4.14i](#).

$$\sum_{i \in \mathcal{N}} \begin{bmatrix} \vec{K}'_1 W_i \vec{K}_1 & \dots & \vec{K}'_1 W_i \vec{K}_{mn} \\ \vdots & \dots & \vdots \\ \vec{K}'_{mn} W_i \vec{K}_1 & \dots & \vec{K}'_{mn} W_i \vec{K}_{mn} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_{mn} \end{bmatrix} = - \sum_{i \in \mathcal{N}} \begin{bmatrix} \text{tr}(X_i M_i K_1) \\ \vdots \\ \text{tr}(X_i M_i K_{mn}) \end{bmatrix}. \quad (4.18)$$

Definição 12 (Precondicionador 5). A mudança de base não é fundamentada na SVD, mas sim no Escalonamento de Jacobi, usando em seguida a ideia da [Definição 9](#).

$$\mathbf{V}(\alpha) = \left| \mathbf{diag} \left(\sum_{i \in \mathcal{N}} W_i \right) \right|^{-\frac{1}{\alpha}}. \quad (4.19)$$

$$\left[\sum_{i \in \mathcal{N}} \mathbf{V}(\alpha)' W_i \mathbf{V}(\alpha) \right] \vec{a} = \mathbf{V}(\alpha)' \mathbf{vec} \left(- \sum_{i \in \mathcal{N}} M'_i X_i \right). \quad (4.20)$$

Esses preconditionadores serão chamados simplesmente de P1, P2, P3, P4 e P5 respectivamente. Em contraste, quando não houver preconditionador, será referido por P0.

4.3 Máscaras

A pseudoinversa A^+ de uma dada matriz A , bastante conhecida na literatura de álgebra linear aplicada, como em [Trefethen e Bau III \(1997, Lecture 11\)](#) por exemplo, é dada pela seguinte expressão:

$$A^+ = (A'A)^{-1}A'.$$

Ela tem a propriedade de “resolver” o sistema linear $Ax = b$ minimizando o resíduo $\|r\| = \|Ax - b\|$ (problema de mínimos quadrados). Se A não é singular, então $A^+ = A^{-1}$. Uma propriedade especial da pseudoinversa é que o espaço nulo ([Petersen \(2012, Proposition 4.9.2\)](#))

$$\mathbf{núcleo}(A) = \mathbf{núcleo}(A^+) = \mathbf{núcleo}(A^+A) \quad (4.21)$$

Definição 13. Dadas duas matrizes arbitrárias de mesma dimensão, definem-se as máscaras

$$\mathbf{mask}(P, Q) = (QQ^+)'P(QQ^+). \quad (4.22)$$

Definição 14 (Espaço Nulo). Define-se o espaço nulo de um operador linear $L \in \mathbb{V}_{n \times n}^{\mathcal{N}} \rightarrow \mathbb{V}_{n \times n}^{\mathcal{N}}$ com relação ao produto interno $\langle \cdot, \cdot \rangle$, denotado por $\mathbf{núcleo}^*(L)$, como sendo a reunião de todos os pontos $X \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ tais que

$$\langle X, L \rangle = 0.$$

Como o produto interno é comutativo, isso significa que L também está no espaço nulo de X .

Lema 1. Sejam $X, L \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ simétricos e positivos semidefinidos. Assim, $\text{tr}(X_i L_i) = 0$ para algum $i \in \mathcal{N}$ se, e somente se $X_i L_i = 0$. Dessa forma, $\langle X, L \rangle = 0$ se, e somente se $XL = 0$.

Demonstração. (\Rightarrow) Inicialmente, como X_i, L_i são simétricos e positivos semidefinidos, existem suas decomposições espectrais (vide [Nocedal e Wright \(2006, p. 597\)](#))

$$X_i = \sum_{r=1}^p \alpha_r x_r x_r' \quad \text{e} \quad L_i = \sum_{s=1}^p \beta_s \ell_s \ell_s', \quad \text{com } x_r, \ell_s \in \mathbb{R}^n \quad \forall r, s \in \{1, \dots, n\}, \alpha_r, \beta_s \in \mathbb{R}, i \in \mathcal{N},$$

em que $\alpha_r, \beta_s \geq 0 \quad \forall r, s \in \{1, \dots, n\}$ são os autovalores de X_i e L_i respectivamente. Assim,

$$\begin{aligned} 0 = \text{tr}(X_i L_i) &= \text{tr}\left(\sum_{r=1}^n \alpha_r x_r x_r' \sum_{s=1}^n \beta_s \ell_s \ell_s'\right) = \sum_{r=1}^n \sum_{s=1}^n \alpha_r \beta_s \text{tr}(x_r x_r' \ell_s \ell_s') \\ &= \sum_{r=1}^n \sum_{s=1}^n \alpha_r \beta_s \text{tr}(\ell_s' \ell_s x_r' x_r) = \sum_{r=1}^n \sum_{s=1}^n \alpha_r \beta_s (x_r' \ell_s)^2. \end{aligned}$$

Isso mostra que, para cada par (r, s) pelo menos uma das seguintes coisas acontece: ou $\alpha_r = 0$, ou $\beta_s = 0$ ou $x_r' \ell_s = 0$. Como isso acontece para todos os pares (r, s) , segue o resultado

$$X_i L_i = \sum_{r=1}^n \sum_{s=1}^n \alpha_r \beta_s x_r x_r' \ell_s \ell_s' = 0.$$

(\Leftarrow) Imediatamente se observa que, se $X_i L_i = 0$ então $\text{tr}(X_i L_i) = 0$. A segunda parte é imediata, pois se a soma de coisas não negativas é zero, então cada uma delas é zero. \square

Proposição 2. Sejam $X, L \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ dados pelo [Definição 2](#) e pelo [Definição 3](#) no modelo sem ruído, isto é, $H = 0$. Denote por $\mathbf{L}_0^k = L^k|_{\{G_r=0, k \leq r < T\}}$. Assim, se para um dado $k \in \mathcal{H}$ acontecer $\langle X^k, L^k \rangle = 0$, onde é utilizado um ganho G_k , então:

1. $G_k X^k = 0; C X^k = 0;$
2. $X^{k+1}|_{G_k \neq 0} = X^{k+1}|_{\{G_k=0\}};$
3. $\langle X^{k+t}, L^{k+t} \rangle = 0; 0 < t < T$ se não houver ruído aditivo;

Demonstração. (1) Pelas hipóteses, tem-se que $L^k = C + G_k' \bullet D \bullet G_k + \mathcal{L}_{\bar{A}^k}(L^{k+1})$, em que $C = C' \geq 0$ e $D = D' > 0$. Assim,

$$0 = \langle X^k, L^k \rangle \geq \langle X^k, G_k' \bullet D \bullet G_k \rangle + \langle X^k, C \rangle \geq 0.$$

Pelo [Lema 1](#), tem-se que $G_k X_i = 0 \quad \forall i \in \mathcal{N}$, pois $\langle X^k, G_k' \bullet D \bullet G_k \rangle = \langle G_k', D \bullet G_k X^k \rangle = 0$ e $D > 0$. Também $C X^k = 0$ pelo mesmo lema.

(2) Pela definição de X no [Definição 2](#),

$$X_i^{k+1} = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji}(A_j + B_j G_k) X_j^k (A_j + B_j G_k)'$$

Por (1), isso se reduz à fórmula sem o ganho G_k

$$X_i^{k+1} = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji}(A_j) X_j^k (A_j)'$$

(3) Pela definição dos custos de continuação (caso sem ruído):

$$0 = J|_k^{T-1} = \langle X^k, L^k \rangle \geq \langle X^{k+t}, L^{k+t} \rangle = J|_{k+t}^{T-1} \geq 0.$$

□

Lema 2. Denote por $\mathbf{L}_0^k = L^k|_{\{G_r=0, k \leq r < T\}}$, conforme o Definição 3. Então, para qualquer L^k dado pelas mesmas equações,

$$\text{núcleo}^*(L^k) \subseteq \text{núcleo}^*(\mathbf{L}_0^k).$$

Demonstração. A prova será por indução. Inicialmente, note que ambos L e \mathbf{L}_0 iniciam no mesmo ponto $L^T = 0$. Agora, suponha que, para qualquer ganho G_k dado seja válido

$$\begin{aligned} L_i^k &= C_i + G'_k D_i G_k + \mathcal{L}_{(A+BG_k)}^i(L^{k+1}) = C_i + G'_k D_i G_k + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij}(A_i + B_i G_k)' L_j^{k+1}(A_i + B_i G_k) \\ &= C_i + G'_k D_i G_k + \mathcal{L}_A^i(\mathbf{L}_0^{k+1}) + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \left(G'_k B'_i L_j^{k+1} A_i + A'_i L_j^{k+1} B_i G_k + G'_k B'_i L_j^{k+1} B_i G_k \right) \\ &= \mathbf{L}_{0i}^k + G'_k D_i G_k + \underbrace{\sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \left(G'_k B'_i L_j^{k+1} A_i + A'_i L_j^{k+1} B_i G_k + G'_k B'_i L_j^{k+1} B_i G_k \right)}_{F_i^k} \\ &= \mathbf{L}_{0i}^k + F_i^k, \end{aligned}$$

em que F_i^k é simétrica positiva semidefinida. Então

$$\begin{aligned} L_i^{k-1} &= C_i + G'_{k-1} D_i G_{k-1} + \mathcal{L}_{(A+BG_{k-1})}^i(L^k) \\ &= C_i + G'_{k-1} D_i G_{k-1} + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij}(A_i + B_i G_{k-1})' L_j^k(A_i + B_i G_{k-1}) \\ &= C_i + G'_{k-1} D_i G_{k-1} + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij}(A_i + B_i G_{k-1})' \left[\mathbf{L}_{0j}^k + F_j^k \right] (A_i + B_i G_{k-1}) \\ &= C_i + \mathcal{L}_A^i(\mathbf{L}_0^k) + G'_{k-1} D_i G_{k-1} + \underbrace{\mathcal{L}_{(A+BG_{k-1})}^i(F^k)}_{F_i^{k-1} \dots} + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \left(G'_{k-1} B'_i \mathbf{L}_{0j}^k A_i + A'_i \mathbf{L}_{0j}^k B_i G_{k-1} + \dots \right. \\ &\quad \left. \dots + \underbrace{G'_{k-1} B'_i \mathbf{L}_{0j}^k B_i G_{k-1}}_{\dots F_i^{k-1}} \right) \\ &= \mathbf{L}_{0i}^{k-1} + F_i^{k-1}. \end{aligned}$$

Ao se fazer mais um passo indutivo análogo, consegue-se a demonstração. Dessa forma, para qualquer $X^k \in \mathbb{V}_{n \times n}^{\mathcal{N}}$ simétrico e positivo semidefinido, se tem

$$0 = \langle X^k, L^k \rangle \geq \langle X^k, \mathbf{L}_0^k \rangle \geq 0,$$

e portanto $\text{núcleo}^*(L^k) \subseteq \text{núcleo}^*(\mathbf{L}_0^k)$ para todo $k \in \mathcal{H}$.

□

A propriedade do [Lema 2](#) às vezes pode ser perdida por erros numéricos ao longo das iterações do MV. Uma forma de melhorar isso é aplicando as máscaras definidas na [Equação 4.22](#), conforme o lema a seguir.

Lema 3 (Espaço nulo). Suponha que erros numéricos acabem por contaminar L , dada pelo [Definição 3](#) para algum G , passando a ser $\widehat{L} = L + \delta L$, com $\delta L > 0$ simétrico positivo definido. Considere $\mathbf{L}_0 = L^k|_{\{G_r=0, k \leq r < T\}} \in \mathbb{V}_{n \times n}^{\mathcal{N}}$. Assim,

$$\mathbf{núcleo}^*(\mathbf{mask}(\widehat{L}, \mathbf{L}_0)) = \mathbf{núcleo}^*(\mathbf{L}_0). \quad (4.23)$$

Demonstração. Considere $L + \delta L$ agora simétrica e positiva definida. Então

$$\begin{aligned} 0 &= \langle X, (\mathbf{L}_0^+ \mathbf{L}_0)' (L + \delta L) (\mathbf{L}_0^+ \mathbf{L}_0) \rangle \\ &= \langle (\mathbf{L}_0^+ \mathbf{L}_0) X (\mathbf{L}_0^+ \mathbf{L}_0)', L \rangle + \langle (\mathbf{L}_0^+ \mathbf{L}_0) X (\mathbf{L}_0^+ \mathbf{L}_0)', \delta L \rangle. \end{aligned}$$

Como $\delta L > 0$, então, pelo [Lema 1](#), $\mathbf{núcleo}^*(\delta L) = 0$ e então $X \in \mathbf{núcleo}^*(\mathbf{L}_0^+ \mathbf{L}_0)$. Mas pela [Equação 4.21](#), $X \in \mathbf{núcleo}^*(\mathbf{L}_0)$. Assim, $\mathbf{núcleo}^*(\mathbf{mask}(\widehat{L}, \mathbf{L}_0)) \subseteq \mathbf{núcleo}^*(\mathbf{L}_0)$. Por outro lado, se $X \in \mathbf{núcleo}^*(\mathbf{L}_0)$, então imediatamente se tem pelo [Lema 1](#) que

$$\langle X, (\mathbf{L}_0^+ \mathbf{L}_0)' (L + \delta L) (\mathbf{L}_0^+ \mathbf{L}_0) \rangle = \langle (\mathbf{L}_0^+ \mathbf{L}_0)', (L + \delta L) (\mathbf{L}_0^+ \mathbf{L}_0) X \rangle = 0.$$

Com isso, $\mathbf{núcleo}^*(\mathbf{L}_0) \subseteq \mathbf{núcleo}^*(\mathbf{mask}(\widehat{L}, \mathbf{L}_0))$. □

Dessa forma, corrige-se a cada $k \in \mathcal{H}$ e cada iteração η do MV a estrutura do espaço $\mathbb{V}_{n \times n}^{\mathcal{N}}$ fazendo

$$L^k \leftarrow \mathbf{mask}(L^k, \mathbf{L}_0^k).$$

4.4 Aproximação quadrática

Para um certo instante de tempo $k \in \mathcal{H}$, dado um conjunto de ganhos G_p em torno de uma estimativa \widehat{G} e avaliados os custos de continuação a partir de k , isto é, $J|_k^{T-1}$ em função de G_p para cada um deles, conforme dado na equação [Equação 3.6](#), pode-se fazer um ajuste quadrático via mínimos quadrados para determinar as matrizes A , b e c de modo que

$$f(G) = \mathbf{vec}(G)' A \mathbf{vec}(G) + b' \mathbf{vec}(G) + c,$$

conforme escrito no [Apêndice B](#).

Com isso, estima-se o sistema linear (matriz e termo independente) baseado nas observações dos pontos G_p que é dada pela formulação do custo de continuação. Essa ideia foi cogitada porque algumas vezes as curvas de nível do custo (curvas de nível para uma quadrática) não eram muito suaves, possivelmente devido a erros de arredondamento. A ideia seria suavizar as curvas para achar seu mínimo. Ainda que válida, essa ideia não rendeu frutos, conforme será visto na seção de resultados mais adiante.

4.5 Metodologia de avaliação

Os perfis de desempenho (ou em inglês *Performance Profile*, vide [Gould e Scott \(2016\)](#)) são uma ferramenta para comparar o desempenho de métodos ao se minimizar alguma estatística, como por exemplo, menor tempo de execução, menor número de iterações, menor valor de função objetivo, etc.

Trata-se de um gráfico que apresenta o percentual de “vitórias” de um determinado método sobre os demais. Um parâmetro de relaxamento/subotimalidade f é também adicionado, de forma que: se $f = 1$, então o perfil de desempenho é calculado em relação ao “valor ótimo”; se $f = 2$, então o perfil contabiliza vitórias para todos os casos em que a estatística é menor ou igual ao dobro do ótimo; e assim por diante. De uma forma geral, quanto maior o valor do perfil de desempenho, melhor o método. Veja o exemplo abaixo.

Tabela 1 – Exemplo de Perfil de Desempenho.

instâncias »	a	b	c	d	e	Perfil com $f = 1$	Perfil com $f = 1.5$
Método A	120	53	57	35	17	$(1+0+0+1+0)/5 = 40\%$	$(1+0+0+1+1)/5 = 60\%$
Método B	123	20	30	35	12	$(0+1+1+1+1)/5 = 80\%$	$(1+1+1+1+1)/5 = 100\%$
$1 \times$ “Ótimo”	120	20	30	35	12	—	—
$1.5 \times$ “Ótimo”	180	30	45	52.5	18	—	—

Como pode-se ver na [Tabela 1](#), os melhores valores obtidos em cada uma das instâncias foram 120, 20, 30, 35 e 12. Esses números são os “ótimos”. Com isso, o perfil do método A para $f = 1$ é o percentual de vezes que ele teve desempenho melhor (valor menor) ou igual a esses dados, sendo 2 vitórias e 3 derrotas, resultando em um perfil de 40% na “otimalidade” ($f = 1$). Por outro lado, se fizer um relaxamento, buscando por “subotimalidade”, em que é aceito um valor até no máximo de 1.5 vezes o valor “ótimo”, então o perfil do método A tem 3 vitórias e portanto, 60%.

Como não há bancos de testes para este tipo de problema, serão feitos experimentos com instâncias geradas aleatoriamente, como também fizeram [Bortolin \(2012\)](#), [Oliveira \(2014\)](#) e [Silva \(2012\)](#), usando o gerador de SLSMs por eles implementado.

O problema abordado é o *Regulador Linear Quadrático*, com horizonte finito, sem observação do estado markoviano. As implementações são feitas em Matlab2017a, executadas em uma máquina Windows 8 com 64GB de memória RAM e processador Intel Core i7-4930K base x64 e CPU 3.4GHz.

O ótimo neste contexto se refere ao menor valor obtido, não o menor valor possível.

RESULTADOS E DISCUSSÕES

Com as estratégias das máscaras e a aproximação quadrática, de condicionamento e varredura abordados no [Capítulo 4](#), foram realizados testes computacionais em 1000 instâncias geradas aleatoriamente, sendo resolvidas e comparados os seus custos “ótimos”. Uma observação bastante importante é que o termo “ótimo” a ser empregado daqui em diante se refere ao *menor valor obtido* na execução da instância, dados os seus critérios de parada, e não ao menor valor possível. Dessa forma o termo indicará uma solução ótima ou subótima. O tamanho do horizonte foi sorteado $T \sim U[10, 100]$, bem como o número máximo de iterações $maxit \sim U[10, 100]$. Os *critérios de parada* do algoritmo foram o número de iterações e o tempo máximo de cinco minutos.

As cadeias de Markov geradas têm de 2 a 5 estados, podendo ou não serem periódicas. Não foi dada atenção à periodicidade das cMs, fugindo do enfoque numérico. A dimensão de x e u para as instâncias geradas também se compreenderam entre 2 e 5. Aplicações realísticas podem ter mais estados e dimensões maiores.

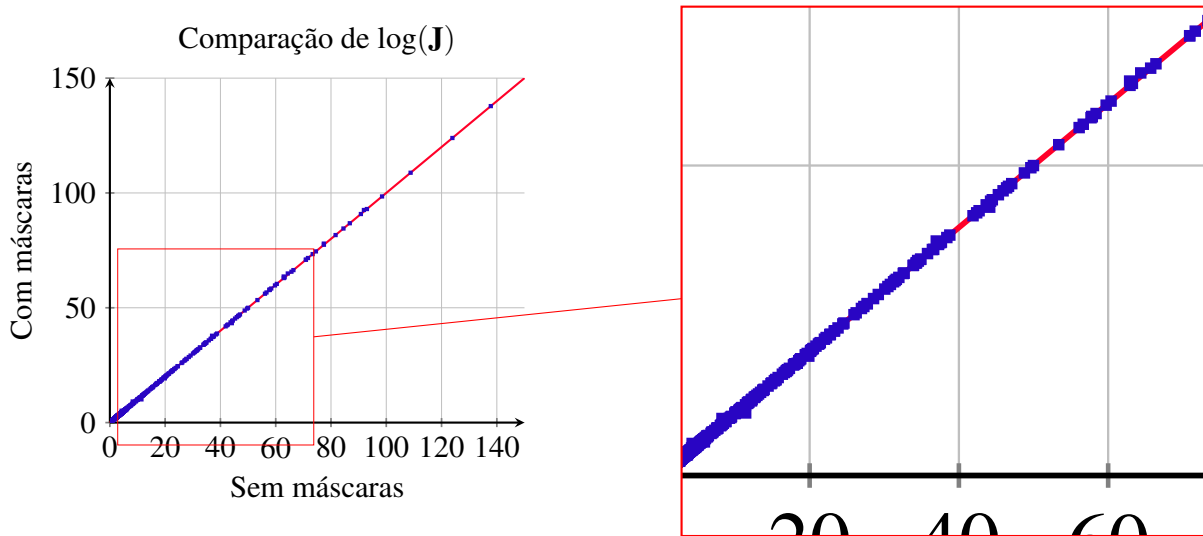
Primeiro serão discutidas as máscaras, a aproximação quadrática, os condicionadores com e sem a varredura, todos com um método sorteado aleatoriamente dentre os oito propostos. Por fim, depois de discutidas as estratégias, compara-se os métodos utilizando as melhores delas, bem como os resultados obtidos. Outros gráficos tidos como menos importantes serão deixados no [Apêndice D](#).

5.1 Máscaras

Nesta seção é apresentado o gráfico comparando os custos utilizando as máscaras *versus* sem elas.

Como se pode ver claramente no gráfico da [Figura 19](#), o uso das máscaras na variável L não melhorou os custos/convergência do MV significativamente. Isso se deve talvez ao fato de

Figura 19 – Comparação dos custos com as máscaras e sem elas, para um total de 1000 instâncias.

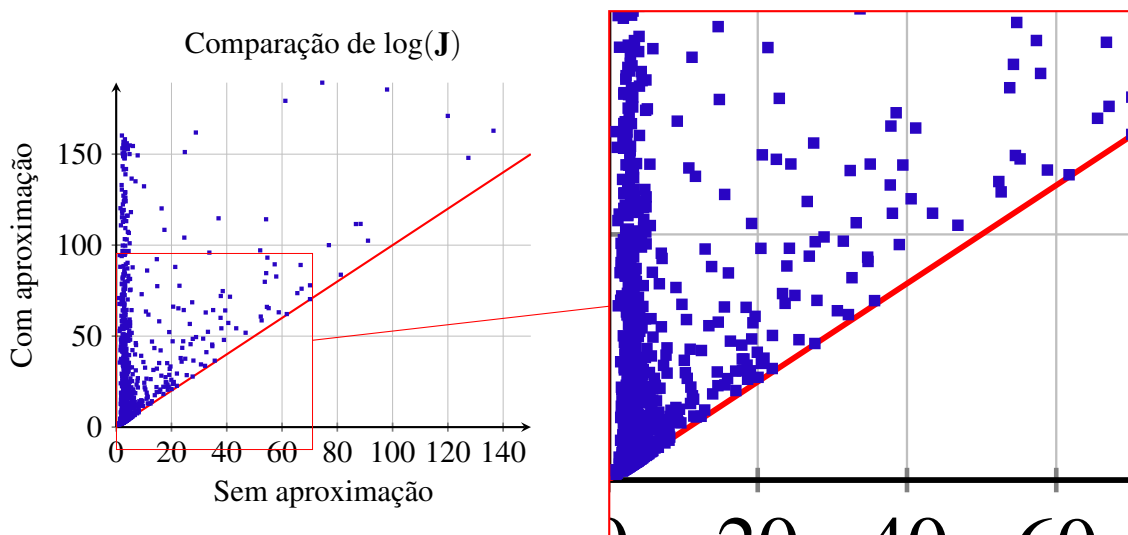


o horizonte escolhido ser pequeno, em comparação com aplicações mais realísticas, contexto em que elas podem ter um efeito mais pronunciado. Outro motivo talvez seja porque a grande maioria das matrizes L tem posto completo nas instâncias testadas, como foi detectado em simulações, de forma que as máscaras são nada mais do que matrizes identidade e assim não têm nenhum efeito. Neste último caso, se L tem posto baixo, então naturalmente as máscaras terão algum impacto, pois serão diferentes da matriz identidade. Outros testes utilizando as máscaras com os preconditionadores propostos não obtiveram nenhuma melhora.

5.2 Aproximação quadrática multivariada

Nesta seção é apresentado o gráfico comparando os custos utilizando a estratégia de aproximação quadrática para o cálculo dos ganhos.

Figura 20 – Comparação dos custos com aproximação quadrática e sem ela.



Como se pode ver claramente no gráfico da [Figura 20](#), o uso da aproximação quadrática na busca do ganho ótimo G piorou muito os custos. Talvez seja porque o domínio no qual os ganhos G_p foram sorteados não foi adequado. Ainda assim, parece que essa abordagem não é muito frutífera.

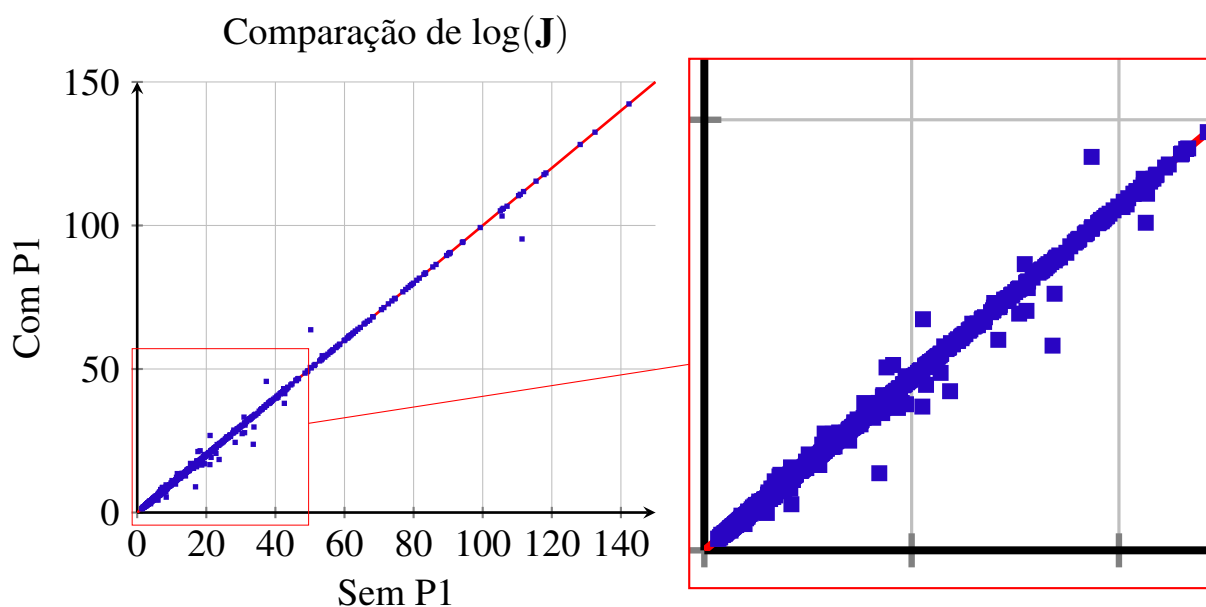
5.3 Precondicionadores

O desempenho dos Precondicionadores 1,2,3,4,5, desde a [Definição 8](#) até a [Definição 12](#) serão comparados nesta seção. Para facilitar a leitura serão dadas etiquetas para os preconditionadores, conforme a [Tabela 2](#).

Etiqueta	Descrição
P0	Nenhum preconditionador adicionado;
P1	Precondicionador da Definição 8 ;
P2	Precondicionador da Definição 9 ;
P3	Precondicionador da Definição 10 ;
P4	Precondicionador da Definição 11 ;
P5	Precondicionador da Definição 12 ;

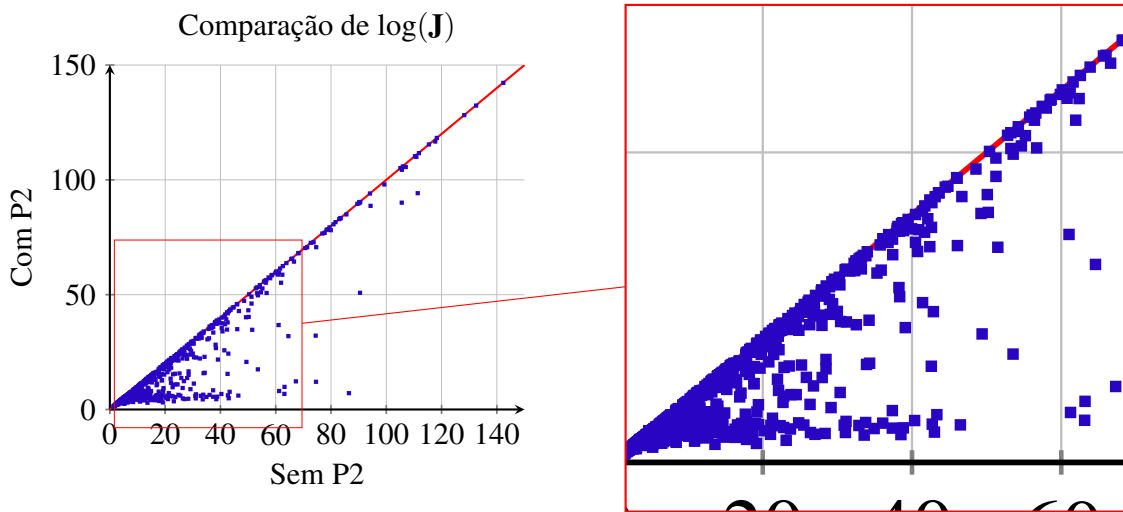
Tabela 2 – Descrição das etiquetas dos preconditionadores utilizadas no texto.

Figura 21 – Comparação dos custos com uso do P1 e sem ele.



Como se pode ver claramente no gráfico da [Figura 21](#), o uso do P1 no sistema linear \mathbb{S}_k melhorou os custos de várias instâncias, mas também piorou o custo em várias outras, de forma que, no geral, não obteve melhora significativa.

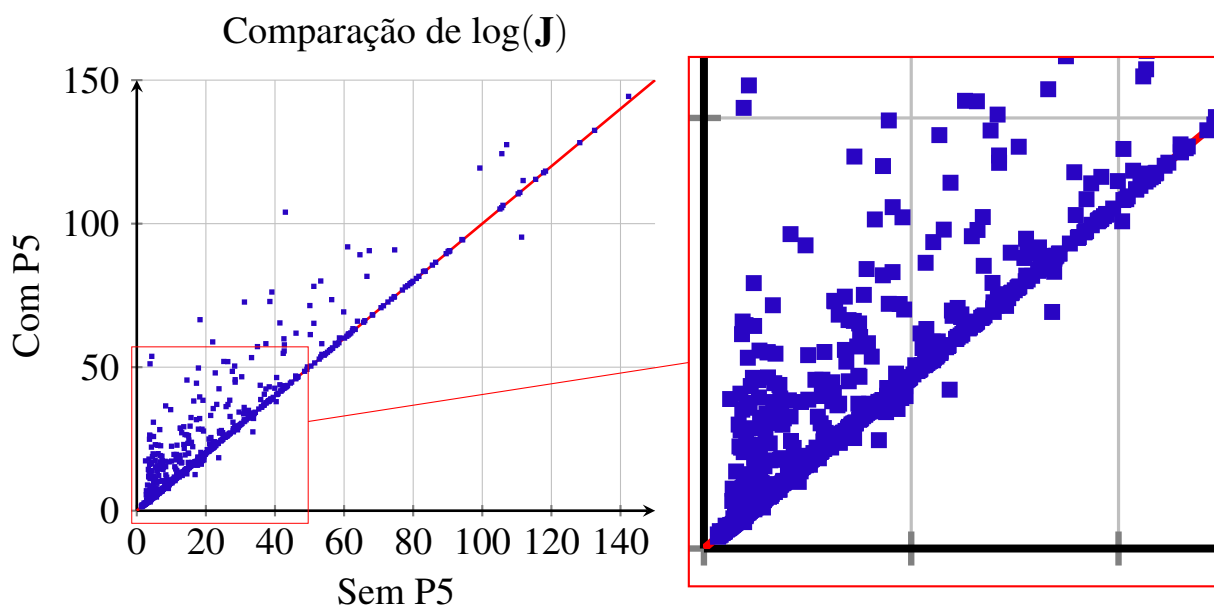
Figura 22 – Comparação dos custos com uso do P2 e sem ele.



Como se pode ver claramente no gráfico da [Figura 22](#), o uso do P2 no sistema linear S_k melhorou bastante os custos de várias instâncias, o que justifica os diferentes modelos para mudança de base criados: comparando-se P1 e P2, a única diferença entre eles é a ordem de precedência entre a soma e a multiplicação, que é sabido ser indiferente, mas que numericamente teve um efeito muito mais positivo quando se deu prioridade à multiplicação das matrizes para depois somá-las, em vez de somá-las e depois multiplicá-las. Detectado inicialmente este fenômeno, buscou-se outras formulações, no intuito de encontrar outras que fossem ainda melhores, o que resultou em P3 e P4. Para diversificar as abordagens, foi testada outra ideia, resultando no P5.

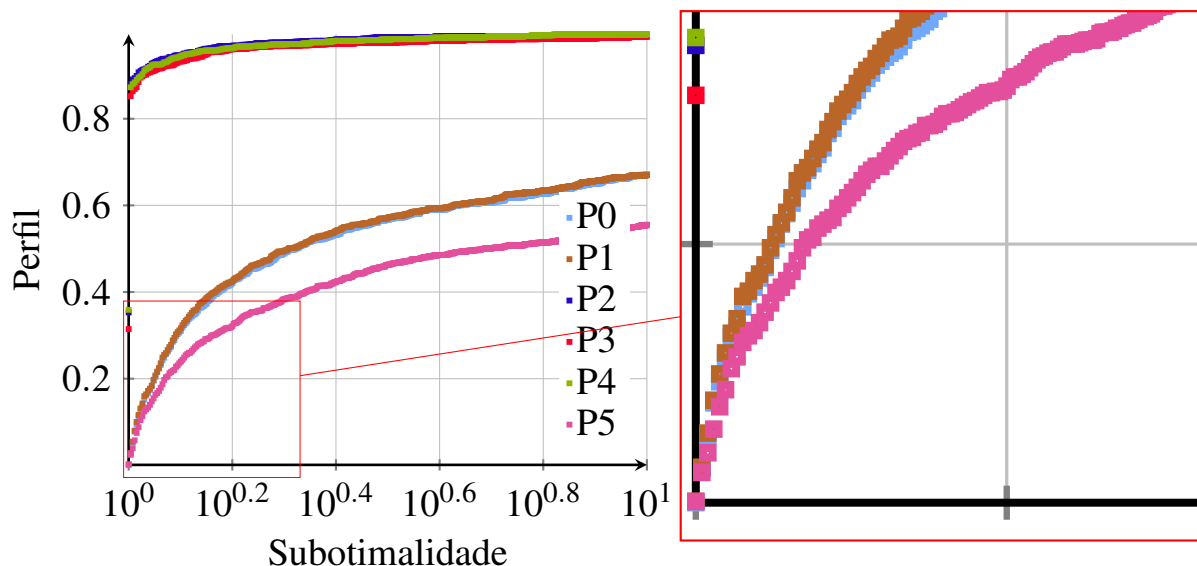
Os gráficos relativos aos P3 e P4 são semelhantes ao do P2, e por isso foram dispostos no [Apêndice D](#).

Figura 23 – Comparação dos custos com uso do P5 e sem ele.



Claramente se observa que o P5 não é um bom candidato, pois piorou grandemente o desempenho do MV.

Figura 24 – Perfis de desempenho dos Precondicionadores.

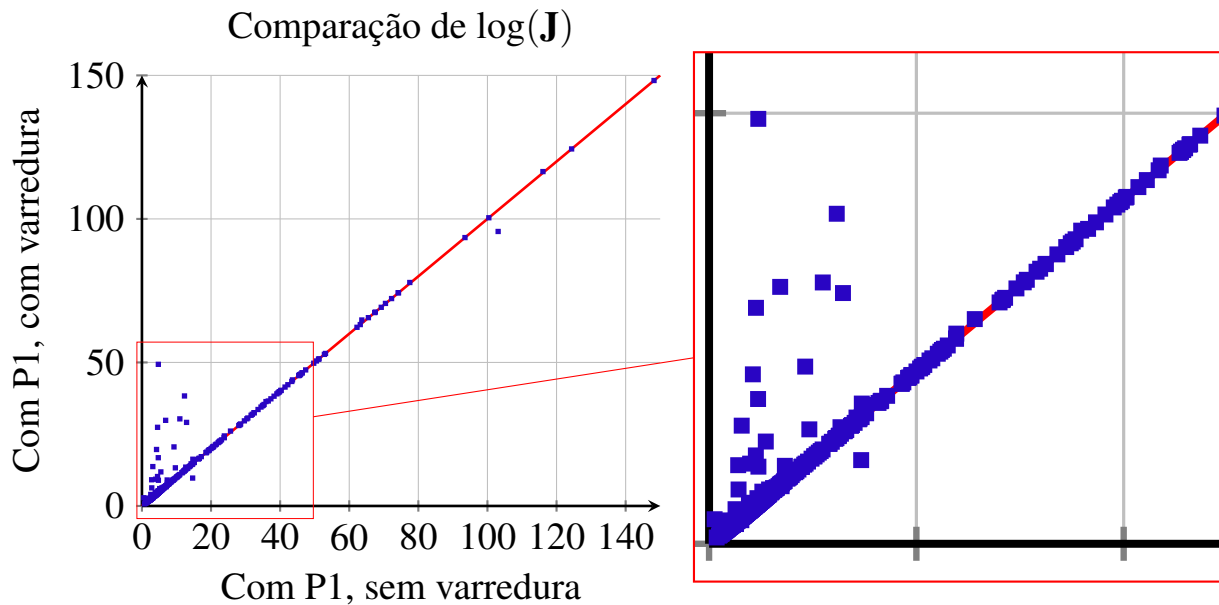


Pode-se observar no gráfico da [Figura 24](#) que P2, P3 e P4 (gráficos em azul, vermelho e verde respectivamente) obtiveram os melhores desempenhos na otimalidade (representado por 10^0) e na subotimalidade, até 10^1 vezes o custo ótimo. O desempenho do P1 (em marrom) é ligeiramente melhor que aquele em que não foi usado nenhum precondicionador (em ciano), indicado por P0. O P5 (em magenta) obteve o pior desempenho.

5.4 Varredura

Buscando saber se o parâmetro α indicado nas definições dos precondicionadores melhora ou não o desempenho dos mesmos, foram realizados alguns testes. Como visto, $\alpha = 2$ é o parâmetro que faz com que o número de condição do sistema linear seja 1. Aqui foram testados valores no conjunto $\{1 + 0.2k, k \in \{0, \dots, 10\}\}$ e escolhido aquele que deixou o menor resíduo.

Como visto na [Figura 25](#), a estratégia de variar o parâmetro α com esses valores dados acima não melhoraram muito os resultados, tendo algum impacto positivo, mas não muito. Talvez outros valores para α deem melhores resultados.

Figura 25 – Comparação do uso da estratégia de varredura dos valores de α .

5.5 Métodos

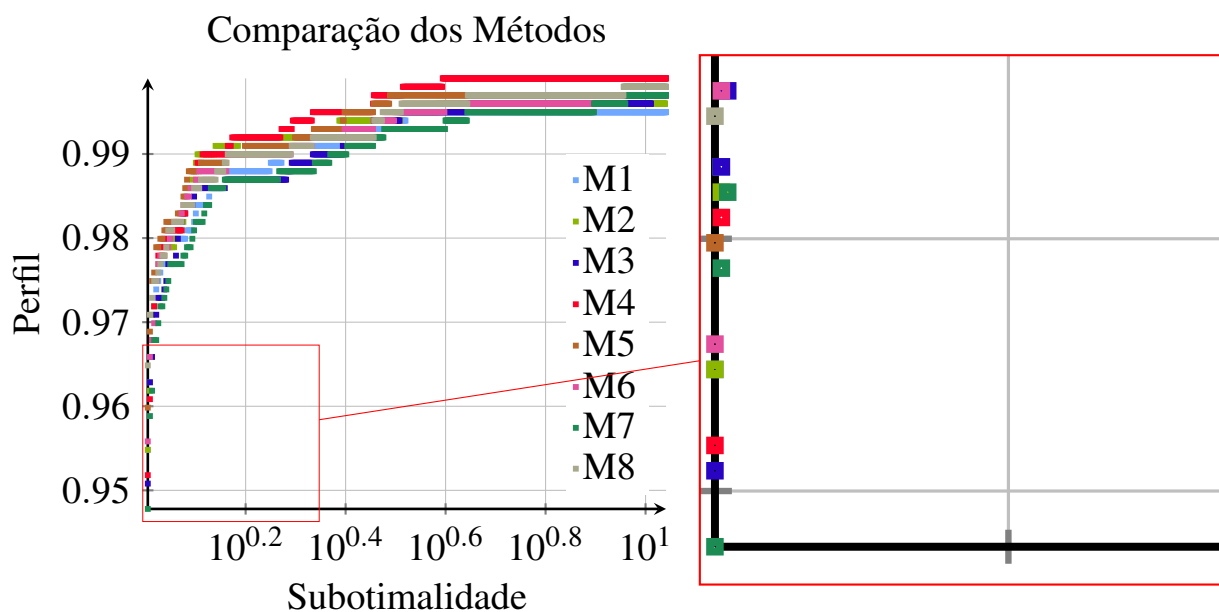
Em se tratando dos métodos de resolução de sistemas lineares, foram testados oito, sendo cinco deles nativos do pacote Matlab e outros três implementados. Eles serão referenciados pelas etiquetas M1 até M8, conforme a descrição a seguir.

Etiqueta	Descrição
M1	Fatoração do tipo LDL (resolução de sistemas triangulares) – Matlab;
M2	Pseudoinversa – Matlab;
M3	Gradientes conjugados – implementado;
M4	Fatoração do tipo LU (resolução de sistemas triangulares) – Matlab;
M5	Gradientes biconjugados – Matlab;
M6	Direções conjugadas – implementado;
M7	Jacobi-Richardson com rotações de Givens – implementado;
M8	Inversa – Matlab;

Tabela 3 – Descrição das etiquetas dos métodos utilizadas no texto.

Nota – O M7 usa rotações de Givens apenas para que sua convergência seja garantida.

Figura 26 – Perfis de desempenho dos métodos de solução de sistemas lineares com P2.

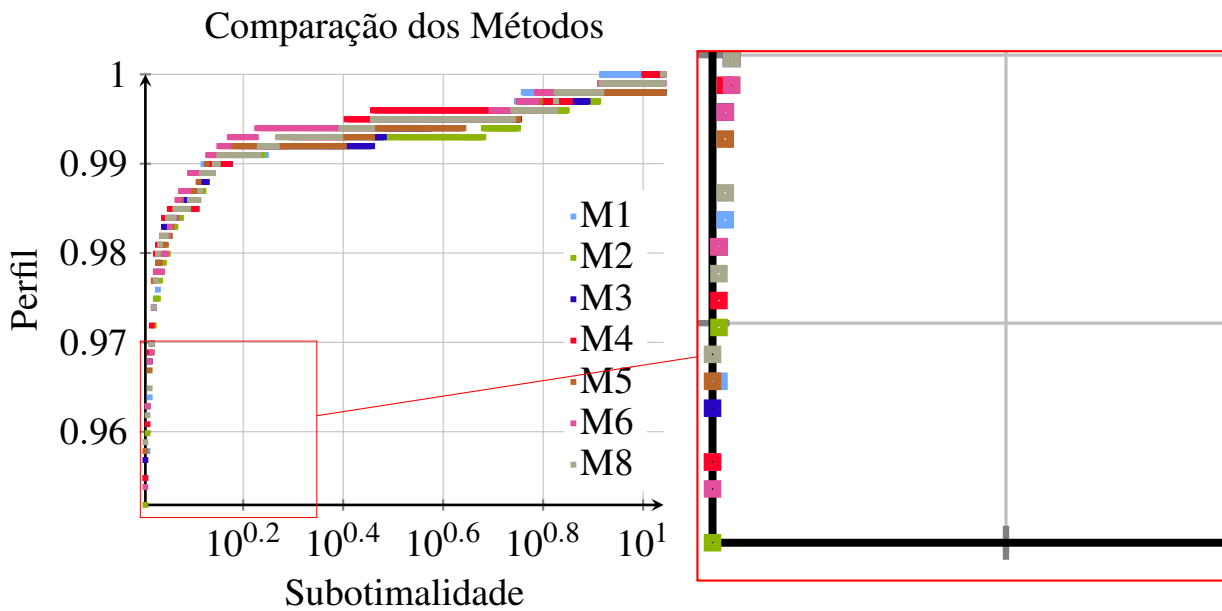


	M1	M2	M3	M4	M5	M6	M7	M8
Desempenho	15.96%	14.55%	16.56%	13.15%	17.46%	15.16%	11.34%	17.47%

Tabela 4 – Comparação dos métodos na otimalidade (em 10^0), com P2.

Observa-se nesses resultados que, de uma forma geral, todos os métodos obtiveram um desempenho semelhante, em torno de 11% a 17% na otimalidade. Particularmente, os métodos M4 e M7 não tiveram resultados tão bons quanto os demais na otimalidade, com o P2, ao passo que M3, M5 e M8 tiveram os melhores resultados, segundo a Tabela 4. Os M4 (vermelho), M5 (marrom) e M8 (cinza) foram os mais robustos na subotimalidade, como indicado na Figura 26.

Figura 27 – Perfis de desempenho dos métodos de solução de sistemas lineares com P3.

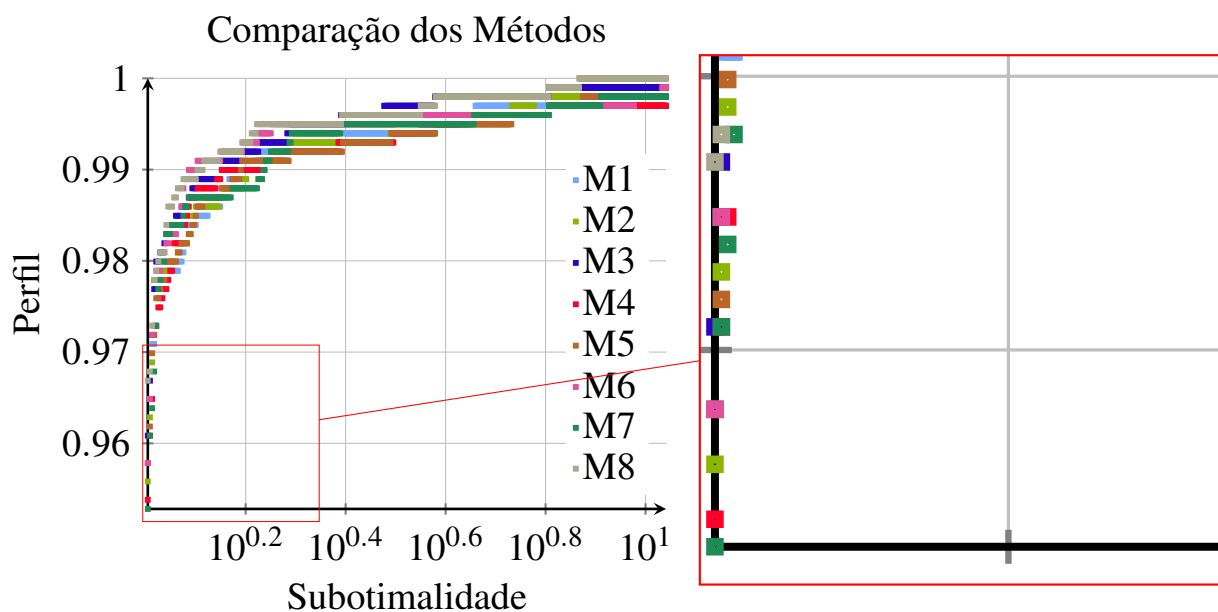


	M1	M2	M3	M4	M5	M6	M7	M8
Desempenho	18.17%	16.96%	17.67%	15.06%	16.66%	14.65%	02.51%	18.47%

Tabela 5 – Comparação dos métodos na otimalidade (em 10^0), com P3.

Com o P3 o M7 teve um desempenho muito ruim, de forma que foi excluído do gráfico por dificultar a visualização dos demais. Observa-se nesses resultados que, de uma forma geral, todos os métodos obtiveram um desempenho semelhante, em torno de 14% a 18% na otimalidade. Particularmente, os métodos M4, M6 e M7 não tiveram resultados tão bons quanto os demais na otimalidade, com o P3, enquanto que M1, M3 e M8 tiveram os melhores resultados, segundo a [Tabela 5](#). Os M4 (vermelho), M6 (magenta) e M8 (cinza) foram os mais robustos na subotimalidade, conforme visto na [Figura 27](#).

Figura 28 – Perfis de desempenho dos métodos de solução de sistemas lineares com P4.



	M1	M2	M3	M4	M5	M6	M7	M8
Desempenho	14.65%	13.75%	17.77%	14.65%	15.56%	15.06%	13.15%	16.76%

Tabela 6 – Comparação dos métodos na otimalidade (em 10^0), com P4.

Observa-se nesses resultados que, de uma forma geral, todos os métodos obtiveram um desempenho semelhante, em torno de 13% a 17% na otimalidade. Particularmente, os métodos M2 e M7 não tiveram resultados tão bons quanto os demais na otimalidade, com o P4, enquanto que M3 e M8 foram os melhores, segundo a [Tabela 6](#). Os M3 (azul), M6 (magenta) e M8 (cinza) foram os mais robustos na subotimalidade, conforme visto na [Figura 28](#).

CONSIDERAÇÕES FINAIS

O problema de controle do Regulador Linear Quadrático foi estudado com um horizonte finito em um sistema dinâmico discreto linear sujeito a saltos markovianos não observados em seus parâmetros, com um ruído aditivo.

O Método Variacional que vinha tendo bons índices de acordo com a literatura foi implementado e procuradas suas fragilidades. Percebeu-se que o método tinha uma convergência débil em alguns casos, provocada por erros numéricos em subproblemas mal condicionados. Detectada essa fraqueza, buscou-se formular técnicas de condicionamento para seus subproblemas, resultando em quatro diferentes formulações baseadas na Decomposição em Valores Singulares (SVD) e outra baseada no Escalonamento de Jacobi.

Apesar de os quatro condicionadores serem equivalentes matematicamente, foi percebida uma enorme diferença de *performance* quando se inverteu a ordem entre as operações de soma e multiplicação, sendo que os rotulados P2, P3 e P4 se saíram muito melhores diante dos P0 (sem condicionador) e P1. O P5 (Escalonamento de Jacobi) foi o que obteve o pior desempenho, de modo que foi descartado dos testes finais, bem como P0 e P1.

Como um fenômeno tão curioso foi observado com relação aos erros numéricos, uma abordagem que consistiu de variar um certo expoente para o cálculo da base para os condicionadores foi testada, com pouco sucesso, de forma que o parâmetro $\alpha = 2$ foi de fato o melhor entre eles, como indicado matematicamente.

Outra abordagem consistiu em tomar os parâmetros do modelo e aproximá-los por um modelo quadrático em mínimos quadrados, na busca de encontrar soluções melhores. No entanto, essa abordagem também não rendeu frutos.

Buscando evitar a perda da estrutura dos espaços nulos do parâmetro L foram utilizadas máscaras de forma a reduzir os erros numéricos, conforme discutido na seção de implementação das mesmas. Essa abordagem não trouxe resultados significantes, talvez por causa das instâncias

testadas.

Oito métodos de resolução de sistemas lineares do tipo $Ax = b$ foram testados utilizando P2, P3 e P4, sendo que, de uma forma geral os métodos tiveram desempenho semelhante. Particularmente na otimalidade os métodos que tiveram melhor desempenho foram M3, M5 e M8 com P2; M1, M3 e M8 com P3; e M3 e M8 com P4. Na subotimalidade, os métodos que tiveram melhor desempenho foram M4, M5 e M8 com P2; M4, M6 e M8 com P3; e M3, M6 e M8 com P4.

Sugestões de trabalhos futuros

- Testar outros preconditionadores além do SVD e do Jacobi;
- Testar valores diferentes para α constante das [Definição 8](#)–[Definição 12](#), buscando determinar o melhor parâmetro, se $\alpha = 2$ ou se outro;
- Testar uma forma melhor de sortear os pontos G_p na aproximação quadrática;
- Analisar outros modelos sem observação do estado contínuo, com e sem ruído;
- Ampliar o estudo para o problema de rastreamento;
- Testar outras amostras com propriedades variadas da teoria de controle, como observabilidade, detetabilidade, estabilizabilidade, dentre outras, utilizando-se das abordagens desenvolvidas.

REFERÊNCIAS

ABDOLLAHI, F. **An H-infinity Dynamic Routing Control of Networked Multi-agent Systems**. Tese (Doutorado), Montreal, P.Q., Canada, 2008. AAINR45644. Citado na página 26.

ABDOLLAHI, F.; KHORASANI, K. A decentralized markovian jump h_∞ control routing strategy for mobile multi-agent networked systems. **IEEE Transactions on Control Systems Technology**, v. 19, n. 2, p. 269–283, 2011. Disponível em: <<https://doi.org/10.1109/TCST.2010.2046418>>. Citado na página 26.

ABOU-KANDIL, H.; FREILING, G.; IONESCU, V.; JANK, G. **Matrix Riccati Equations in Control and Systems Theory**. Birkhauser Basel, 2003. Disponível em: <<https://doi.org/10.1007/978-3-0348-8081-7>>. Citado na página 26.

ABOU-KANDIL, H.; FREILING, G.; JANK, G. On the solution of discrete-time markovian jump linear quadratic control problems. **Automatica**, v. 31, n. 5, p. 765 – 768, 1995. ISSN 0005-1098. Disponível em: <<http://www.sciencedirect.com/science/article/pii/000510989400164E>>. Citado na página 26.

BORTOLIN, D. C. **Métodos numéricos para o controle linear quadrático com saltos e observação parcial de estado**. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, 2012. Disponível em: <<https://doi.org/10.11606/D.55.2012.tde-28032012-151127>>. Citado nas páginas 26, 35, 44 e 63.

CAJUEIRO, D. O. **Stochastic optimal control of jumping Markov parameter processes with applications to finance**. Tese (Doutorado) — Instituto Tecnológico de Aeronáutica–ITA, Brazil, 2002. Disponível em: <<http://www.bd.bibl.ita.br/tesesdigitais/000490363.pdf>>. Citado na página 26.

CUNHA, M. C. C. **Método Numéricos**. Campinas, SP: Editora Unicamp, 2000. Citado na página 51.

DYNKIN, E. B. Markov processes. **Academic Press**, v. 150, n. 3696, p. 1099–1131, 1965. Disponível em: <<https://doi.org/10.1126/science.150.3696.598>>. Citado nas páginas 25 e 34.

EUROPE, S. R. **NAO Robot**. 2017. Licença © CC BY-SA 4.0. Disponível em: <https://commons.wikimedia.org/wiki/File:NAO_Robot_.jpg>. Acesso em: 01/11/2018. Citado na página 27.

FRANCO, N. M. B. **Cálculo Numérico**. São Paulo, SP: Pearson Prentice Hall, 2013. Citado na página 51.

GONZÁLEZ, B. N. **Palacio de la Bolsa de Madrid**. 2017. Licença © CC BY-SA 4.0. Disponível em: <https://commons.wikimedia.org/wiki/File:Valores,_Palacio_de_la_Bolsa,_Madrid,_Espana%C3%B1a,_2017.gif>. Acesso em: 01/11/2018. Citado na página 27.

- GOULD, N.; SCOTT, J. A note on performance profiles for benchmarking software. **ACM Transactions on Mathematical Software**, v. 42, n. 2, p. 15:1–15:5, 2016. Disponível em: <<http://dx.doi.org/10.1145/2950048>>. Citado na página 62.
- NASA. **NASA Goddard Space Flight Center**. 2016. Licença © CC BY 2.0. Disponível em: <[https://commons.wikimedia.org/wiki/File:Antares_Test_Flight_Scheduled_for_April_17_\(8632520692\).jpg](https://commons.wikimedia.org/wiki/File:Antares_Test_Flight_Scheduled_for_April_17_(8632520692).jpg)>. Acesso em: 01/11/2018. Citado na página 27.
- NOCEDAL, J.; WRIGHT, S. J. **Numerical optimization**. 2. ed. New York, NY: Springer, 2006. Citado nas páginas 25, 51, 52 e 59.
- OLIVEIRA, L. T. de. **Método variacional com atualização múltipla de ganhos para controle de sistemas lineares com parâmetros sujeitos a saltos Markovianos não observados**. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, 2014. Disponível em: <<https://doi.org/10.11606/D.55.2014.tde-24092014-145046>>. Citado nas páginas 26, 35, 44 e 63.
- PETERSEN, P. **Linear Algebra**. New York: Springer, 2012. Disponível em: <<https://doi.org/10.1007/978-1-4614-3612-6>>. Citado na página 59.
- RIBEIRO, A. A.; KARAS, E. W. **Otimização contínua: aspectos teóricos e computacionais**. Curitiba, PR: Cengage Learning, 2014. Citado na página 25.
- SILVA, C. A. **Algoritmos para o custo médio a longo prazo de sistemas com saltos markovianos parcialmente observados**. Tese (Doutorado) — Universidade de São Paulo, São Carlos, 2012. Disponível em: <<https://doi.org/10.11606/T.55.2012.tde-25012013-095943>>. Citado nas páginas 26, 35 e 63.
- STOICA, A.; YAESH, I. Jump markovian-based control of wing deployment for an uncrewed air vehicle. **Journal of Guidance, Control, and Dynamics**, v. 25, n. 2, p. 407–411, 2002. Disponível em: <<https://doi.org/10.2514/2.4896>>. Citado na página 26.
- TREFETHEN, L. N.; Bau III, D. **Numerical Linear Algebra**. Philadelphia, PA: SIAM, 1997. Citado nas páginas 52, 53 e 58.
- VAL, J. B. R. do; BAŞAR, T. Receding horizon control of jump linear systems and a macroeconomic policy problem. **Journal of Economics Dynamics and Control**, v. 23, n. 8, p. 1099–1131, 1999. Disponível em: <[https://doi.org/10.1016/S0165-1889\(98\)00058-X](https://doi.org/10.1016/S0165-1889(98)00058-X)>. Citado nas páginas 26, 27 e 35.
- VARGAS, A. do N. **Controle por horizonte retrocedente de sistemas lineares com saltos markovianos e ruído aditivo**. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 2004. Disponível em: <<http://repositorio.unicamp.br/jspui/handle/REPOSIP/260118>>. Citado nas páginas 26 e 35.
- VARGAS, A. N.; COSTA, E. F.; VAL, J. B. do. On the control of markov jump linear systems with no mode observation: application to a dc motor device. **International Journal of Robust and Nonlinear Control**, v. 23, n. 10, p. 1136–1150, 2013. Disponível em: <<https://doi.org/10.1002/rnc.2911>>. Citado na página 26.
- ZHOU, K.; DOYLE, J. C.; GLOVER, K. **Robust and optimal control**. Upper Saddle River, NJ: Prentice Hall, 1996. Citado nas páginas 46, 79 e 85.

CUSTO COMO UMA FUNÇÃO QUADRÁTICA EM G_k

Primeiramente, apresentam-se as propriedades básicas dos operadores traço $\mathbf{tr}(\cdot)$ e vetorização $\mathbf{vec}(\cdot)$, que podem ser vistas em [Zhou, Doyle e Glover \(1996\)](#).

$$\left\{ \begin{array}{ll} \mathbf{tr}(ABC) = \mathbf{tr}(BCA) = \mathbf{tr}(CAB) & \text{(produto é cíclico)} \\ \|v\|^2 = v'v = \mathbf{tr}(v'v) = \mathbf{tr}(vv') & \\ \mathbf{tr}(aB + C) = a\mathbf{tr}(B) + \mathbf{tr}(C) & \text{(linearidade)} \\ \mathbf{tr}(AB) = \mathbf{vec}(A)'\mathbf{vec}(B) = \sum_i \sum_j a_{ij}b_{ij} & \text{(vetorizado do traço de um produto)} \\ \mathbf{vec}(ABC) = [C' \otimes A]\mathbf{vec}(B) & \text{(vetorizado do produto de 3 matrizes)} \end{array} \right. \quad (\text{A.1})$$

em que \otimes representa o produto de *Kronecker*.

A linearidade dos operadores traço $\mathbf{tr}(\cdot)$ e somatório $\sum(\cdot)$ serão amplamente utilizadas neste e nos demais apêndices, bem como a simetria de matrizes. Portanto o leitor deve estar atento a essas propriedades.

Relembrando alguns aspectos do problema, cujas definições serão copiadas aqui, vem:

$$\left\{ \begin{array}{ll} \bar{\mathbf{A}}_i^k = A_i + B_i G_k & \mathcal{L}_{\bar{\mathbf{A}}}^i(L) = \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \bar{\mathbf{A}}_i' L_j \bar{\mathbf{A}}_i \\ \bar{\mathbf{C}}_i^k = C_i + G_k' D_i G_k & L_i^T = 0_{n \times n}, \quad \sigma_i^T = 0 \quad \forall i \in \mathcal{N} \\ \mathcal{E}^i(U) = \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} U_j & L_i^k = \bar{\mathbf{C}}_i^k + \mathcal{L}_{\bar{\mathbf{A}}}^i(L^{k+1}) \\ X_i^{k+1} = \mathcal{F}_{\bar{\mathbf{A}}}^i(X^k) + \Sigma_i^k & \sigma_i^k = \mathbf{tr}(\mathcal{E}^i(L^{k+1}) H_i H_i') + \mathcal{E}^i(\sigma^{k+1}) \\ \mathcal{F}_{\bar{\mathbf{A}}}^i(X) = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \bar{\mathbf{A}}_i X_j \bar{\mathbf{A}}_i' & \langle U, V \rangle = \sum_{i \in \mathcal{N}} \mathbf{tr}(U_i' V_i) \\ \Sigma_i^k = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_k^j H_j H_j' & i \in \mathcal{N} = \{1, \dots, N\}, \quad k \in \mathcal{H} = \{0, \dots, T\} \end{array} \right. \quad (\text{A.2})$$

em que $X_i^k = (X_i^k)'$, $C_i = C_i'$, $\bar{\mathbf{C}}_i^k = (\bar{\mathbf{C}}_i^k)'$, $\Sigma_i^k = (\Sigma_i^k)'$, $D_i = D_i'$ e $L_i^k = (L_i^k)'$ são matrizes simétricas para todo $k \in \mathcal{H}$ e todo $i \in \mathcal{N}$. Também, o produto interno $\langle \cdot, \cdot \rangle$ definido em (A.2) é linear

(verifique), propriedade que será utilizada a seguir. O custo total \mathbf{J} é dado pelo somatório dos custos parciais J_k :

$$\begin{aligned} \mathbf{J} &= J_0 + J_1 + \dots + J_{T-1} = \langle X^0, C + G'_0 D G_0 \rangle + \langle X^1, C + G'_1 D G_1 \rangle + \dots \\ &\quad \dots + \langle X^{T-1}, C + G'_{T-1} D G_{T-1} \rangle = \sum_{k \in \mathcal{H} \setminus \{T\}} \langle X^k, \bar{\mathbf{C}}^k \rangle. \end{aligned}$$

Dados dois ganhos arbitrários G_{T-2} e G_{T-1} (com isso se tem $\bar{\mathbf{C}}^{T-1}$ e $\bar{\mathbf{C}}^{T-1}$) e somando os custos J_{T-2} e J_{T-1} do final do horizonte, tem-se:

$$\begin{aligned} J_{T-2} + J_{T-1} &= \langle X^{T-2}, \bar{\mathbf{C}}^{T-2} \rangle + \langle X^{T-1}, \bar{\mathbf{C}}^{T-1} \rangle \\ &= \langle X^{T-2}, \bar{\mathbf{C}}^{T-2} \rangle + \langle \mathcal{F}_{\bar{\mathbf{A}}^{T-2}}(X^{T-2}) + \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle \\ &= \langle X^{T-2}, \bar{\mathbf{C}}^{T-2} \rangle + \langle \mathcal{F}_{\bar{\mathbf{A}}^{T-2}}(X^{T-2}), \bar{\mathbf{C}}^{T-1} \rangle + \langle \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle \\ &= \langle X^{T-2}, \bar{\mathbf{C}}^{T-2} \rangle + \langle X^{T-2}, \mathcal{L}_{\bar{\mathbf{A}}^{T-2}}(\bar{\mathbf{C}}^{T-1}) \rangle + \langle \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle \\ &= \langle X^{T-2}, \bar{\mathbf{C}}^{T-2} + \mathcal{L}_{\bar{\mathbf{A}}^{T-2}}(\bar{\mathbf{C}}^{T-1}) \rangle + \langle \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle \\ &= \langle X^{T-2}, L^{T-2} \rangle + \underbrace{\langle \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle}_{\pi'_{T-2} \sigma^{T-2}} \\ &= \langle X^{T-2}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2}. \end{aligned}$$

Na terceira e quinta igualdades, a linearidade de $\langle \cdot, \cdot \rangle$ permite separar (agrupar) o termo em duas (uma) parcelas. Na quarta igualdade, o operador $\mathcal{L}_{(\cdot)}(\cdot)$ é o autoadjunto de $\mathcal{F}_{(\cdot)}(\cdot)$. Para provar isso, basta usar as definições de \mathcal{L} , \mathcal{F} e $\langle \cdot, \cdot \rangle$, trocando-se a ordem dos somatórios e usando a propriedade 1 de (A.1), além das definições (A.2):

$$\begin{aligned} \langle \mathcal{F}_{\bar{\mathbf{A}}}(X), \bar{\mathbf{C}} \rangle &= \sum_{i \in \mathcal{N}} \text{tr}(\mathcal{F}_{\bar{\mathbf{A}}}^i(X) \bar{\mathbf{C}}_i) \stackrel{(A.2)}{=} \sum_{i \in \mathcal{N}} \text{tr} \left(\left(\sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \bar{\mathbf{A}}_i X_j \bar{\mathbf{A}}_i' \right) \bar{\mathbf{C}}_i \right) \\ &\stackrel{(A.1)}{=} \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{N}} \mathbb{P}_{ji} \text{tr} \left(X_j \bar{\mathbf{A}}_i' \bar{\mathbf{C}}_i \bar{\mathbf{A}}_i \right) = \sum_{j \in \mathcal{N}} \text{tr} \left(X_j \sum_{i \in \mathcal{N}} \mathbb{P}_{ji} \bar{\mathbf{A}}_i' \bar{\mathbf{C}}_i \bar{\mathbf{A}}_i \right) \\ &\stackrel{(A.2)}{=} \sum_{j \in \mathcal{N}} \text{tr} \left(X_j \mathcal{L}_{\bar{\mathbf{A}}}^j(\bar{\mathbf{C}}) \right) = \langle X, \mathcal{L}_{\bar{\mathbf{A}}}(\bar{\mathbf{C}}) \rangle. \quad \square \end{aligned}$$

Para provar que $\langle \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle = \pi'_{T-2} \sigma^{T-2}$, em que π_{T-2} representa a distribuição de probabilidade de θ_{T-2} , iniciam-se os parâmetros $L_i^T = 0_{n \times n}$, $\sigma_i^T = 0$ para todo $i \in \mathcal{N}$.

Por (A.2), tem-se que

$$L_i^{T-1} = \bar{\mathbf{C}}_i^{T-1}, \quad \sigma_i^{T-1} = 0, \quad \forall i \in \mathcal{N}.$$

Como

$$\Sigma_i^{T-2} = \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_{T-2}^j H_j H_j', \quad \mathcal{E}^j(L^{T-1}) = \sum_{i \in \mathcal{N}} \mathbb{P}_{ji} \bar{\mathbf{C}}_i^{T-1},$$

então

$$\begin{aligned}
\langle \Sigma^{T-2}, L^{T-1} \rangle &= \langle \Sigma^{T-2}, \bar{\mathbf{C}}^{T-1} \rangle = \sum_{i \in \mathcal{N}} \text{tr}(\Sigma_i^{T-2} \bar{\mathbf{C}}_i^{T-1}) \stackrel{(A.1)}{=} \sum_{i \in \mathcal{N}} \text{tr}(\bar{\mathbf{C}}_i^{T-1} \Sigma_i^{T-2}) \\
&= \sum_{i \in \mathcal{N}} \text{tr} \left(\bar{\mathbf{C}}_i^{T-1} \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_{T-2}^j H_j H_j' \right) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{T-2}^j \text{tr} \left(\mathbb{P}_{ji} \bar{\mathbf{C}}_i^{T-1} H_j H_j' \right) \\
&= \sum_{j \in \mathcal{N}} \pi_{T-2}^j \text{tr} \left(\sum_{i \in \mathcal{N}} \mathbb{P}_{ji} \bar{\mathbf{C}}_i^{T-1} H_j H_j' \right) = \sum_{j \in \mathcal{N}} \pi_{T-2}^j \underbrace{\text{tr} \left(\mathcal{E}^j(L^{T-1}) H_j H_j' \right)}_{s_j^{T-2}} \\
&= \sum_{j \in \mathcal{N}} \pi_{T-2}^j s_j^{T-2} = \pi'_{T-2} \sigma^{T-2},
\end{aligned}$$

onde $\sigma^{T-2} = s^{T-2}$ nessa ocasião.

Fazendo algumas contas análogas, mas para $k \in \{T-3, T-2, T-1\}$, e dado um ganho arbitrário G_{T-3} , vem:

$$\begin{aligned}
\sum_{k=T-3}^{T-1} J_k &= \langle X^{T-3}, \bar{\mathbf{C}}^{T-3} \rangle + \langle X^{T-2}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2} \\
&= \langle X^{T-3}, \bar{\mathbf{C}}^{T-3} \rangle + \langle \mathcal{F}_{\bar{\mathbf{A}}^{T-3}}(X^{T-3}) + \Sigma^{T-3}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2} \\
&= \langle X^{T-3}, \bar{\mathbf{C}}^{T-3} \rangle + \langle \mathcal{F}_{\bar{\mathbf{A}}^{T-3}}(X^{T-3}), L^{T-2} \rangle + \langle \Sigma^{T-3}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2} \\
&= \langle X^{T-3}, \bar{\mathbf{C}}^{T-3} \rangle + \langle X^{T-3}, \mathcal{L}_{\bar{\mathbf{A}}^{T-3}}(L^{T-2}) \rangle + \langle \Sigma^{T-3}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2} \\
&= \langle X^{T-3}, \bar{\mathbf{C}}^{T-3} + \mathcal{L}_{\bar{\mathbf{A}}^{T-3}}(L^{T-2}) \rangle + \underbrace{\langle \Sigma^{T-3}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2}}_{\pi'_{T-3} \sigma^{T-3}} \\
&= \langle X^{T-3}, L^{T-3} \rangle + \pi'_{T-3} \sigma^{T-3}.
\end{aligned} \tag{A.3}$$

Para provar que $\langle \Sigma^{T-3}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2} = \pi'_{T-3} \sigma^{T-3}$, procede-se de forma análoga à anterior, mas com alguns ajustes. Primeiramente, verifique que

$$\pi_{T-2}^i \stackrel{(2.2)}{=} \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_{T-3}^j. \tag{A.4}$$

$$\begin{aligned}
\langle \Sigma^{T-3}, L^{T-2} \rangle &= \sum_{i \in \mathcal{N}} \text{tr}(L_i^{T-2} \Sigma_i^{T-3}) = \sum_{i \in \mathcal{N}} \text{tr} \left(L_i^{T-2} \sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_{T-3}^j H_j H_j' \right) \\
&= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{T-3}^j \text{tr} \left(\mathbb{P}_{ji} L_i^{T-2} H_j H_j' \right) = \sum_{j \in \mathcal{N}} \pi_{T-3}^j \text{tr} \left(\sum_{i \in \mathcal{N}} \mathbb{P}_{ji} L_i^{T-2} H_j H_j' \right) \\
&= \sum_{j \in \mathcal{N}} \pi_{T-3}^j \underbrace{\text{tr} \left(\mathcal{E}^j(L^{T-2}) H_j H_j' \right)}_{s_j^{T-3}} = \sum_{j \in \mathcal{N}} \pi_{T-3}^j s_j^{T-3}
\end{aligned}$$

Para terminar, vem:

$$\begin{aligned}
\langle \Sigma^{T-3}, L^{T-2} \rangle + \pi'_{T-2} \sigma^{T-2} &= \sum_{j \in \mathcal{N}} \pi_{T-3}^j s_j^{T-3} + \sum_{i \in \mathcal{N}} \pi_{T-2}^i \sigma_i^{T-2} \\
&\stackrel{(A.4)}{=} \sum_{j \in \mathcal{N}} \pi_{T-3}^j s_j^{T-3} + \sum_{i \in \mathcal{N}} \left(\sum_{j \in \mathcal{N}} \mathbb{P}_{ji} \pi_{T-3}^j \right) \sigma_i^{T-2} \\
&= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{T-3}^j s_j^{T-3} + \mathbb{P}_{ji} \pi_{T-3}^j \sigma_i^{T-2} \\
&= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \pi_{T-3}^j (s_j^{T-3} + \mathbb{P}_{ji} \sigma_i^{T-2}) \\
&= \sum_{j \in \mathcal{N}} \pi_{T-3}^j \left(s_j^{T-3} + \sum_{i \in \mathcal{N}} \mathbb{P}_{ji} \sigma_i^{T-2} \right) \\
&= \sum_{j \in \mathcal{N}} \pi_{T-3}^j \underbrace{\left(\text{tr}(\mathcal{E}^j(L^{T-2})H_j H_j') + \mathcal{E}^j(\sigma^{T-2}) \right)}_{\sigma_j^{T-3}} \\
&\stackrel{(A.2)}{=} \sum_{j \in \mathcal{N}} \pi_{T-3}^j \sigma_j^{T-3} = \pi'_{T-3} \sigma^{T-3},
\end{aligned}$$

o que conclui a demonstração. \square

O argumento indutivo utilizado acima demonstra para todo $\ell \in \mathcal{H} \setminus \{T\}$ a seguinte expressão:

$$J|_{\ell}^{T-1} = \sum_{k=\ell}^{T-1} J_k = \langle X^{\ell}, L^{\ell} \rangle + \pi'_{\ell} \sigma^{\ell}, \quad (A.5)$$

o que é chamado de *custo de continuação a partir do estágio ℓ* , de forma que, se $\ell = 0$, então

$$\mathbf{J} = \langle X^0, L^0 \rangle + \pi'_0 \sigma^0.$$

Para escrever o custo \mathbf{J} em função de um determinado ganho G_k , mais algumas contas são necessárias. Tomando um índice $k = \ell \in \mathcal{H} \setminus \{T\}$, o custo total pode ser escrito como

$$\mathbf{J} = \sum_{k=0}^{\ell-1} \langle X^k, \bar{\mathbf{C}}^k \rangle + J|_{\ell}^{T-1} = \sum_{k=0}^{\ell-1} \langle X^k, \bar{\mathbf{C}}^k \rangle + \langle X^{\ell}, L^{\ell} \rangle + \pi'_{\ell} \sigma^{\ell}$$

que, conforme a [Equação A.2](#), fica

$$\mathbf{J} = \underbrace{\sum_{k=0}^{\ell-1} \langle X^k, \bar{\mathbf{C}}^k \rangle}_{\phi} + \underbrace{\langle X^{\ell}, \bar{\mathbf{C}}^{\ell} + \mathcal{L}_{\bar{\mathbf{A}}^{\ell}}(L^{\ell+1}) \rangle}_{\psi}.$$

Aqui, o custo indicado por ϕ contabiliza de $k = 0$ até $k = \ell - 1$. O custo indicado por ψ contabiliza de $k = \ell$ até $k = T - 1$ (custo de continuação a partir de ℓ). E no entanto, apenas a segunda parcela de ψ depende do ganho G_{ℓ} , conforme (A.2): veja que $\bar{\mathbf{C}}^k$ e $\bar{\mathbf{A}}^k$ (e portanto L^k) dependem de G_k , X^k depende de G_{k-1} e σ^k depende de L^{k+1} , que por sua vez depende de G_{k+1} .

Dessa forma, os termos que não dependem de G_ℓ são escritos como uma constante c_1 . Então, tem-se a seguinte expressão que será desenvolvida:

$$\mathbf{J} = c_1 + \left\langle X^\ell, \bar{\mathbf{C}}^\ell + \mathcal{L}_{\mathbf{A}^\ell}(L^{\ell+1}) \right\rangle$$

Para simplificar a notação, escreve-se $X^\ell = X$, $G_\ell = G$ e $L^{\ell+1} = L$. Agora, as contas serão abertas e alguns termos constantes em relação a G serão separados. As equações em (A.2) serão utilizadas, bem como a linearidade de $\Sigma(\cdot)$ e de $\text{tr}(\cdot)$.

$$\begin{aligned} \mathbf{J} &= c_1 + \sum_{i \in \mathcal{N}} \text{tr} \left[X_i \left((\mathbf{C}_i + G' D_i G) + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} (A_i + B_i G)' L_j (A_i + B_i G) \right) \right] \\ &= c_1 + \sum_{i \in \mathcal{N}} \left[\text{tr}(X_i \mathbf{C}_i) + \text{tr}(X_i G' D_i G) + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \text{tr}(X_i (A_i + B_i G)' L_j (A_i + B_i G)) \right] \\ &= c_2 + \sum_{i \in \mathcal{N}} \left[\text{tr}(X_i G' D_i G) + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \text{tr} \left(X_i (A_i' L_j A_i + A_i' L_j B_i G + G' B_i' L_j A_i + G' B_i' L_j B_i G) \right) \right] \\ &= c + \sum_{i \in \mathcal{N}} \left[\text{tr}(X_i G' D_i G) + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \text{tr} \left(X_i (A_i' L_j B_i G + G' B_i' L_j A_i + G' B_i' L_j B_i G) \right) \right] \\ &= c + \sum_{i \in \mathcal{N}} \left[\text{tr} \left(X_i \left[G' D_i G + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} G' B_i' L_j B_i G \right] \right) + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \text{tr} \left(X_i \left[A_i' L_j B_i G + G' B_i' L_j A_i \right] \right) \right] \\ &= c + \sum_{i \in \mathcal{N}} \left[\text{tr} \left(X_i G' \left[D_i + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} B_i' L_j B_i \right] G \right) + 2 \text{tr} \left(X_i \left[\sum_{j \in \mathcal{N}} \mathbb{P}_{ij} A_i' L_j B_i \right] G \right) \right] \\ &= c + \sum_{i \in \mathcal{N}} \text{tr} \left(X_i G' Z_i G \right) + 2 \text{tr} \left(X_i M_i G \right), \end{aligned}$$

em que

$$\begin{aligned} c_2 &= c_1 + \sum_{i \in \mathcal{N}} \text{tr}(X_i \mathbf{C}_i), & c &= c_2 + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} \text{tr}(X_i A_i' L_j A_i), \\ Z_i &= D_i + \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} B_i' L_j B_i, & M_i &= \sum_{j \in \mathcal{N}} \mathbb{P}_{ij} A_i' L_j B_i, \quad \forall i \in \mathcal{N}. \end{aligned}$$

Com isso, obtém-se

$$\mathbf{J} = f(G) = c + \sum_{i \in \mathcal{N}} \text{tr} \left(X_i G' Z_i G \right) + 2 \text{tr} \left(X_i M_i G \right). \quad (\text{A.6})$$

Se for manipulado mais um pouco, usando as propriedades 4 e 5 em (A.1), considerando a simetria de X e Z , vem:

$$\text{tr}(X_i M_i G) = \text{vec}(M_i' X_i)' \text{vec}(G),$$

$$\text{tr}(X_i G' Z_i G) = \text{vec}(Z_i G X_i)' \text{vec}(G) = \text{vec}(G)' \text{vec}(Z_i G X_i) = \text{vec}(G)' [X_i \otimes Z_i] \text{vec}(G).$$

Assim, pode-se ainda, escrever a Equação A.6 como

$$\mathbf{J} = f(G) = c + \text{vec}(G)' \left[\sum_{i \in \mathcal{N}} X_i \otimes Z_i \right] \text{vec}(G) + 2 \text{vec} \left(\sum_{i \in \mathcal{N}} M_i' X_i \right)' \text{vec}(G). \quad (\text{A.7})$$

Dessa forma, pode-se ver que o custo \mathbf{J} é quadrático em $\text{vec}(G)$.

A função custo $f(G)$ pode, portanto, ser calculada usando as matrizes de ganho G como na [Equação A.6](#) ou usando os vetorizados dessas matrizes, $\text{vec}(G)$, como na [Equação A.7](#).

Derivando a [Equação A.7](#) em relação a $\text{vec}(G)$ e avaliando em zero, tem-se o seguinte sistema linear (condição necessária de otimalidade):

$$\sum_{i \in \mathcal{N}} [X_i \otimes (D_i + B_i' \Lambda_i B_i)] \text{vec}(G) = - \sum_{i \in \mathcal{N}} \text{vec}(B_i' \Lambda_i A_i X_i), \quad (\text{A.8})$$

em que $\Lambda_i = \mathcal{E}^i(L)$.

APROXIMAÇÃO QUADRÁTICA MULTIVARIADA POR MÍNIMOS QUADRADOS

Seja $x, b \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$ e $c \in \mathbb{R}$ e considere a seguinte função quadrática $f : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$f(x) = x'Ax + b'x + c.$$

Suponha que se tenha um conjunto de pontos $x_p \in \mathbb{R}^n$, com $p \in \mathcal{P} = \{1, \dots, P\}$, e necessita-se calcular qual é a função quadrática que melhor se ajusta a eles. Para isso, precisa-se calibrar os parâmetros A , b e c de modo a representar os pontos o mais preciso possível. Esse problema pode ser escrito em termos de mínimos quadrados (minimizar o erro quadrático):

$$\min_{A,b,c} M(A,b,c) = \sum_{p \in \mathcal{P}} [f(x_p) - (x_p'Ax_p + b'x_p + c)]^2.$$

De [Zhou, Doyle e Glover \(1996\)](#), a fórmula

$$\frac{d \operatorname{tr}(PXQ)}{dX} = P'Q'$$

pode ser empregada para calcular a seguinte expressão, que será usada para calcular a derivada de $M(A, b, c)$ em relação a A na terceira equação do sistema abaixo (regra da cadeia):

$$\frac{d(x'Ax)}{dA} = \frac{d \operatorname{tr}(x'Ax)}{dA} = xx'.$$

Assim, calculando as derivadas e avaliando em zero, obtém-se o sistema:

$$\begin{cases} \left. \frac{\partial M(A, b, c)}{\partial c} \right|_0 = 2 \sum_{p \in \mathcal{P}} \left(f(x_p) - x_p' A x_p - b' x_p - c \right) (-1) = 0 \\ \left. \frac{\partial M(A, b, c)}{\partial b} \right|_0 = 2 \sum_{p \in \mathcal{P}} \left(f(x_p) - x_p' A x_p - x_p' b - c \right) (-x_p) = 0 \\ \left. \frac{\partial M(A, b, c)}{\partial A} \right|_0 = 2 \sum_{p \in \mathcal{P}} \left(f(x_p) - x_p' A x_p - b' x_p - c \right) (-x_p x_p') = 0. \end{cases}$$

Algumas das propriedades em (A.1) serão utilizadas:

$$\begin{cases} \mathbf{tr}(ABC) = \mathbf{tr}(BCA) = \mathbf{tr}(CAB) & (i) \\ \mathbf{tr}(AB) = \mathbf{vec}(A)' \mathbf{vec}(B) & (ii) \\ \mathbf{vec}(ABC) = [C' \otimes A] \mathbf{vec}(B) & (iii). \end{cases}$$

O sistema passa a ser então:

$$\begin{cases} \sum_{p \in \mathcal{P}} x_p' A x_p + b' x_p + c = \sum_{p \in \mathcal{P}} f(x_p) \\ \sum_{p \in \mathcal{P}} (x_p' A x_p) x_p + (b' x_p) x_p + c x_p = \sum_{p \in \mathcal{P}} f(x_p) x_p \\ \sum_{p \in \mathcal{P}} (x_p' A x_p) x_p x_p' + (b' x_p) x_p x_p' + c x_p x_p' = \sum_{p \in \mathcal{P}} f(x_p) x_p x_p'. \end{cases}$$

Por $(x_p' A x_p)$ ser escalar, na terceira equação, pode-se fazer $(x_p' A x_p) x_p x_p' = x_p (x_p' A x_p) x_p'$. O mesmo se aplica à parcela vizinha, $(b' x_p) x_p x_p' = x_p (b' x_p) x_p'$. E ainda, na segunda equação, pode-se fazer $(b' x_p) x_p = x_p (b' x_p) = x_p (x_p' b)$ utilizando a mesma ideia, bem como inverter a ordem dos fatores nas parcelas vizinhas, o que dá:

$$\begin{cases} \sum_{p \in \mathcal{P}} c + b' x_p + x_p' A x_p = \sum_{p \in \mathcal{P}} f(x_p) \\ \sum_{p \in \mathcal{P}} x_p c + x_p (x_p' b) + x_p (x_p' A x_p) = \sum_{p \in \mathcal{P}} f(x_p) x_p \\ \sum_{p \in \mathcal{P}} x_p x_p' c + x_p (b' x_p) x_p' + x_p (x_p' A x_p) x_p' = \sum_{p \in \mathcal{P}} f(x_p) x_p x_p'. \end{cases}$$

Na primeira equação,

$$x_p' A x_p = \mathbf{tr}(x_p' A x_p) \stackrel{(i)}{=} \mathbf{tr}(x_p x_p' A) \stackrel{(ii)}{=} \mathbf{vec}(x_p x_p')' \mathbf{vec}(A).$$

Na segunda equação, utilizando esta última expressão, fica

$$x_p (x_p' A x_p) = x_p \mathbf{vec}(x_p x_p')' \mathbf{vec}(A).$$

Na terceira equação, vetorizando as duas últimas parcelas do primeiro membro, vem

$$\mathbf{vec}(x_p (b' x_p) x_p') \stackrel{(iii)}{=} [x_p x_p' \otimes x_p] \mathbf{vec}(b') = [x_p x_p' \otimes x_p] b,$$

$$\mathbf{vec}(x_p(x'_p A x_p)x'_p) \stackrel{(iii)}{=} [x_p x'_p \otimes x_p x'_p] \mathbf{vec}(A).$$

Com essas últimas alterações, vetorizando a terceira equação, o sistema passa a ser

$$\begin{cases} \sum_{p \in \mathcal{P}} c + x'_p b + \mathbf{vec}(x_p x'_p)' \mathbf{vec}(A) = \sum_{p \in \mathcal{P}} f(x_p) \\ \sum_{p \in \mathcal{P}} x_p c + (x_p x'_p) b + x_p \mathbf{vec}(x_p x'_p)' \mathbf{vec}(A) = \sum_{p \in \mathcal{P}} f(x_p) x_p \\ \sum_{p \in \mathcal{P}} \mathbf{vec}(x_p x'_p) c + [x_p x'_p \otimes x_p] b + [x_p x'_p \otimes x_p x'_p] \mathbf{vec}(A) = \sum_{p \in \mathcal{P}} f(x_p) \mathbf{vec}(x_p x'_p). \end{cases}$$

Assim, finalmente, o sistema linear formado pode ser escrito como

$$\sum_{p \in \mathcal{P}} \begin{bmatrix} 1 & x'_p & \mathbf{vec}(x_p x'_p)' \\ x_p & x_p x'_p & x_p \mathbf{vec}(x_p x'_p)' \\ \mathbf{vec}(x_p x'_p) & [x_p x'_p \otimes x_p] & [x_p x'_p \otimes x_p x'_p] \end{bmatrix} \begin{bmatrix} c \\ b \\ \mathbf{vec}(A) \end{bmatrix} = \sum_{p \in \mathcal{P}} \begin{bmatrix} f(x_p) \\ f(x_p) x_p \\ f(x_p) \mathbf{vec}(x_p x'_p) \end{bmatrix}. \quad (\text{B.1})$$

Apenas uma observação final, como $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$, são necessários pelo menos $n^2 + n + 1$ pontos x_p para que o problema fique bem definido (o menor valor de P).

MÉTODO DE DIREÇÕES CONJUGADAS

Considere $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função quadrática:

$$f(x) = \frac{1}{2}x'Ax + b'x + c,$$

$$\nabla f(x) = Ax + b.$$

Sabe-se que, dependendo da caracterização de A , um ponto estacionário (minimizador ou maximizador) global x^* é tal que

$$Ax^* = -b. \quad (\text{C.1})$$

Considerando que A é positiva definida (portanto A tem posto completo, é uma base que gera \mathbb{R}^n e x^* é o minimizador global da quadrática), pode-se conseguir uma base $\mathbf{D} = [d_1, \dots, d_n] \not\equiv 0$ para \mathbb{R}^n que seja A -conjugada/ortogonal, ou seja, $d_i'Ad_j = 0$ para todo $i \neq j$. Obtida uma base \mathbf{D} para \mathbb{R}^n que seja A -ortogonal, qualquer ponto pode ser escrito como combinação linear dos elementos dessa base, e portanto,

$$x^* = \alpha_1 d_1 + \alpha_2 d_2 + \dots + \alpha_n d_n = \sum_{k=1}^n \alpha_k d_k. \quad (\text{C.2})$$

Pré-multiplicando a equação acima por $d_i'A$, tem-se

$$\begin{aligned} d_i'Ax^* &= \alpha_1 d_i'Ad_1 + \alpha_2 d_i'Ad_2 + \dots + \alpha_n d_i'Ad_n \\ &= \alpha_i d_i'Ad_i, \end{aligned}$$

e então

$$\alpha_i = \frac{d_i'Ax^*}{d_i'Ad_i} \stackrel{(\text{C.1})}{=} -\frac{d_i'b}{d_i'Ad_i}. \quad (\text{C.3})$$

Assim, a solução ótima será

$$x^* \stackrel{(\text{C.2}), (\text{C.3})}{=} \sum_{k=1}^n -\left(\frac{d_k'b}{d_k'Ad_k}\right) d_k.$$

C.1 Ortogonalização de Gram-Schmidt

Dada uma matriz A simétrica positiva definida, objetivando obter uma base \mathbf{B} que seja A -conjugada para o espaço \mathbb{R}^n a partir de uma base \mathbf{D} arbitrária (que pode ser a base canônica por exemplo), pode-se usar o processo de Gram-Schmidt.

Dada $\mathbf{D} = [d_1, \dots, d_n]$ uma base de \mathbb{R}^n , ou seja, um conjunto l.i., deseja-se obter uma base $\mathbf{B} = [b_1, \dots, b_n]$ para \mathbb{R}^n que seja A -conjugada, isto é, uma base em que $b_i'Ab_j = 0$ para todo $i \neq j$. Assim, definem-se:

$$\begin{aligned} b_1 &= d_1 \\ b_2 &= d_2 - \frac{d_2'Ab_1}{b_1'Ab_1}b_1 \\ b_3 &= d_3 - \frac{d_3'Ab_1}{b_1'Ab_1}b_1 - \frac{d_3'Ab_2}{b_2'Ab_2}b_2 \\ &\vdots \\ b_n &= d_n - \frac{d_n'Ab_1}{b_1'Ab_1}b_1 - \frac{d_n'Ab_2}{b_2'Ab_2}b_2 - \dots - \frac{d_n'Ab_{n-1}}{b_{n-1}'Ab_{n-1}}b_{n-1}. \end{aligned}$$

Assim, \mathbf{B} é uma base A -conjugada, e então pode-se aplicar o método de Direções Conjugadas, conforme descrito no [Apêndice C](#).

Este processo pode ser feito por meio do [Algoritmo 4](#).

Algoritmo 4 – A -Ortogonalização de Gram-Schmidt

- 1: **procedimento** GRAMSCHMIDT(A, D) ▷ Cria uma base B que seja A -conjugada.
 - 2: $B \leftarrow D$ ▷ Os vetores das bases B e D são vetores-coluna.
 - 3: **para** $i \leftarrow 2$ **até** n **faça** ▷ Assim, B e D são matrizes, representadas por suas colunas.
 - 4: **para** $k \leftarrow i$ **até** n **faça**
 - 5: $B(:, k) \leftarrow B(:, k) - \frac{D(:, k)'AB(:, i-1)}{B(:, i-1)'AB(:, i-1)}B(:, i-1)$ ▷ $X(:, n)$ representa a
 - 6: **fim para** ▷ n -ésima coluna de X .
 - 7: **fim para**
 - 8: **retorna** B
 - 9: **fim procedimento**
-

MAIS GRÁFICOS

Neste apêndice constam dados extras dos resultados, sob a forma de gráficos. Em todos os gráficos, o número de instâncias testadas é o mesmo, de 1000.

D.1 Máscaras

Figura 29 – Comparação dos custos com as máscaras e sem elas, com P1.

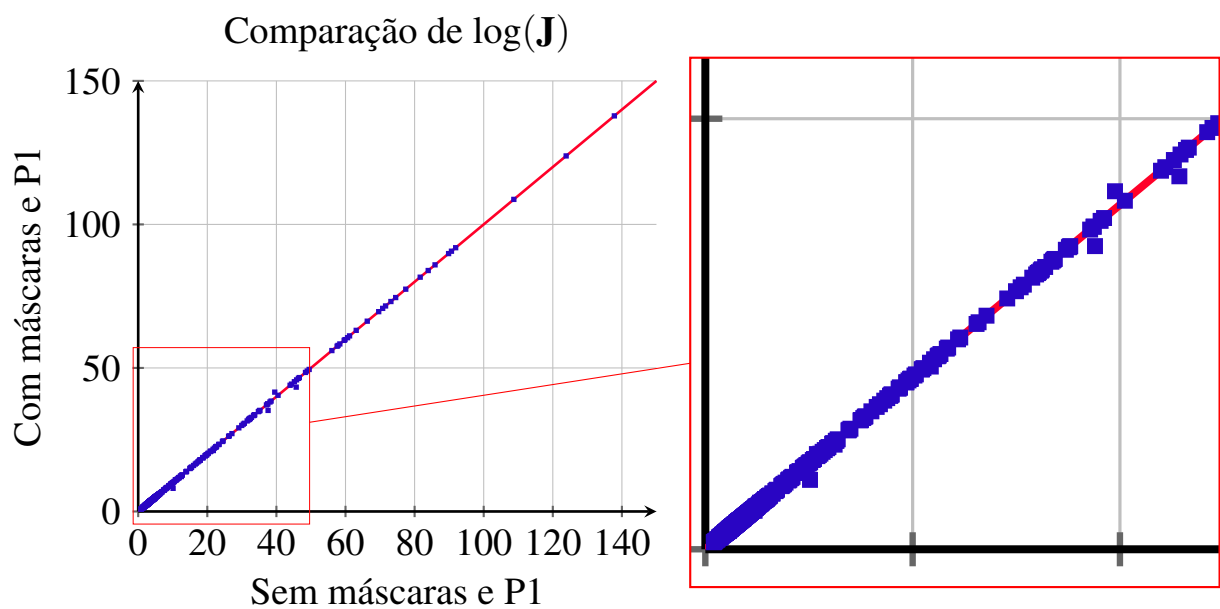


Figura 30 – Comparação dos custos com as máscaras e sem elas, com P2.

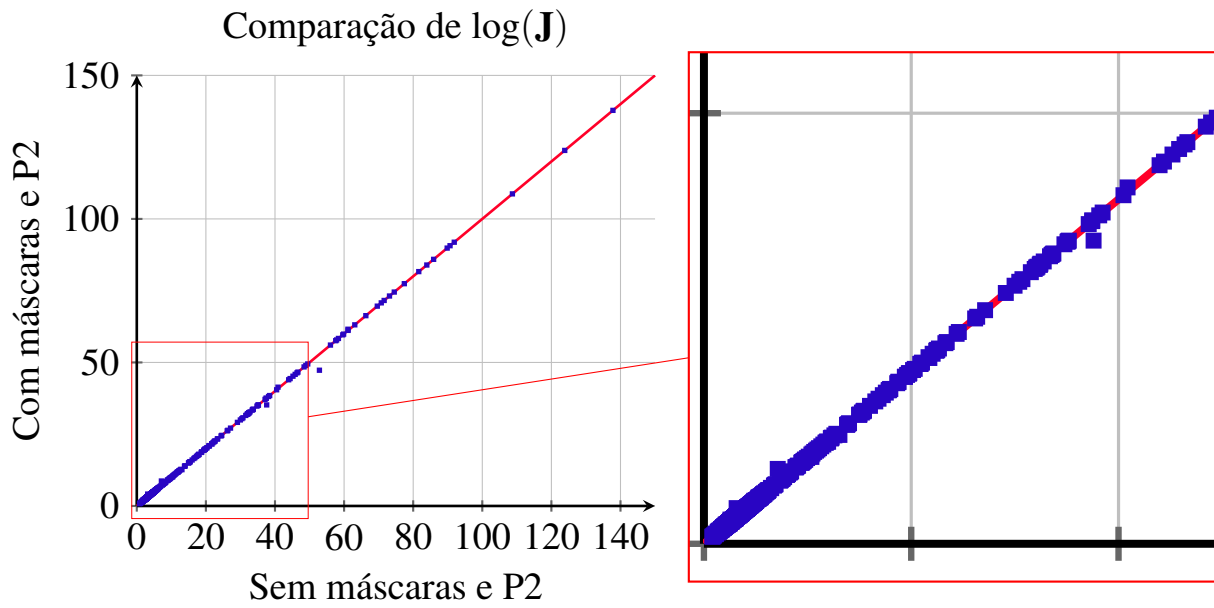


Figura 31 – Comparação dos custos com as máscaras e sem elas, com P3.

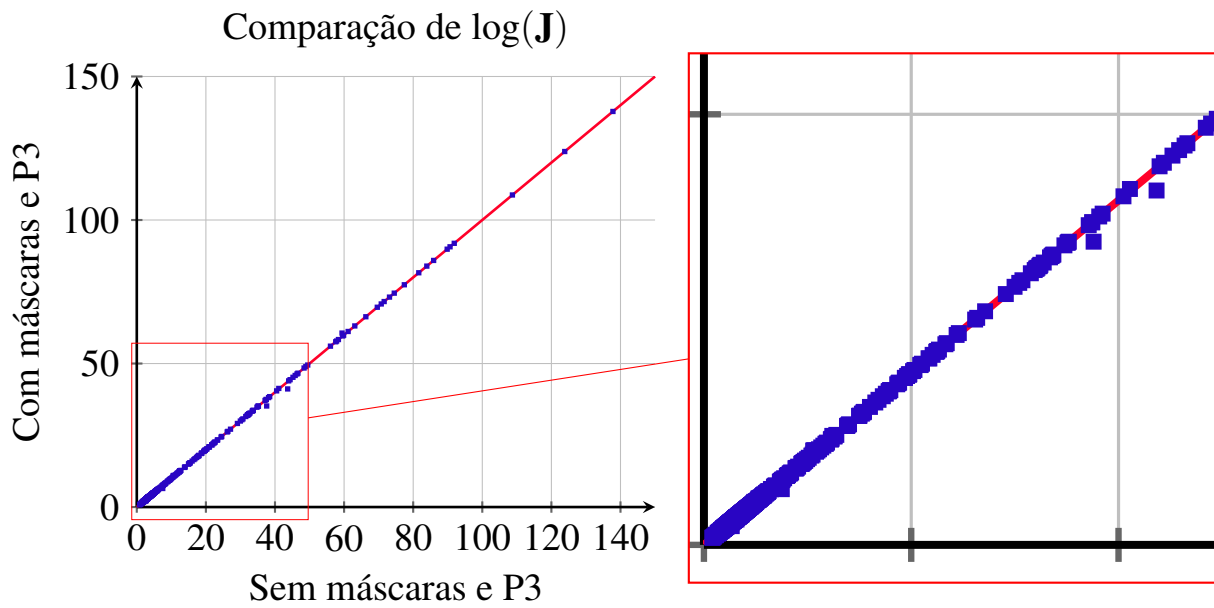
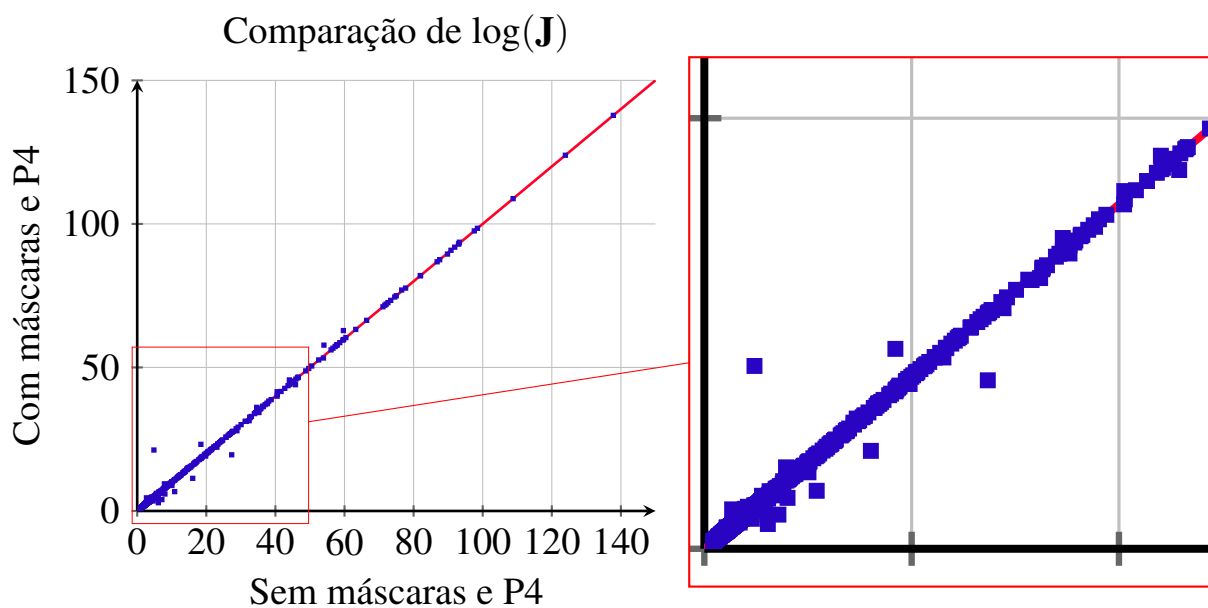


Figura 32 – Comparação dos custos com as máscaras e sem elas, com P4.



D.2 Aproximação quadrática

Figura 33 – Comparação dos custos com aproximação quadrática e sem ela, com P1.

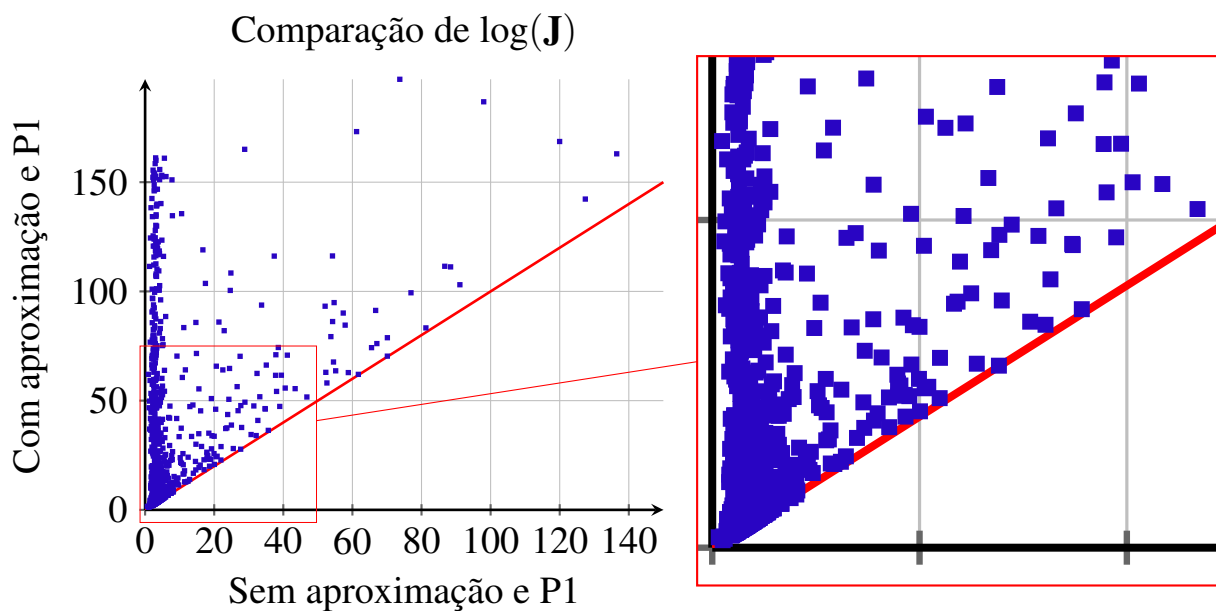


Figura 34 – Comparação dos custos com aproximação quadrática e sem ela, com P2.

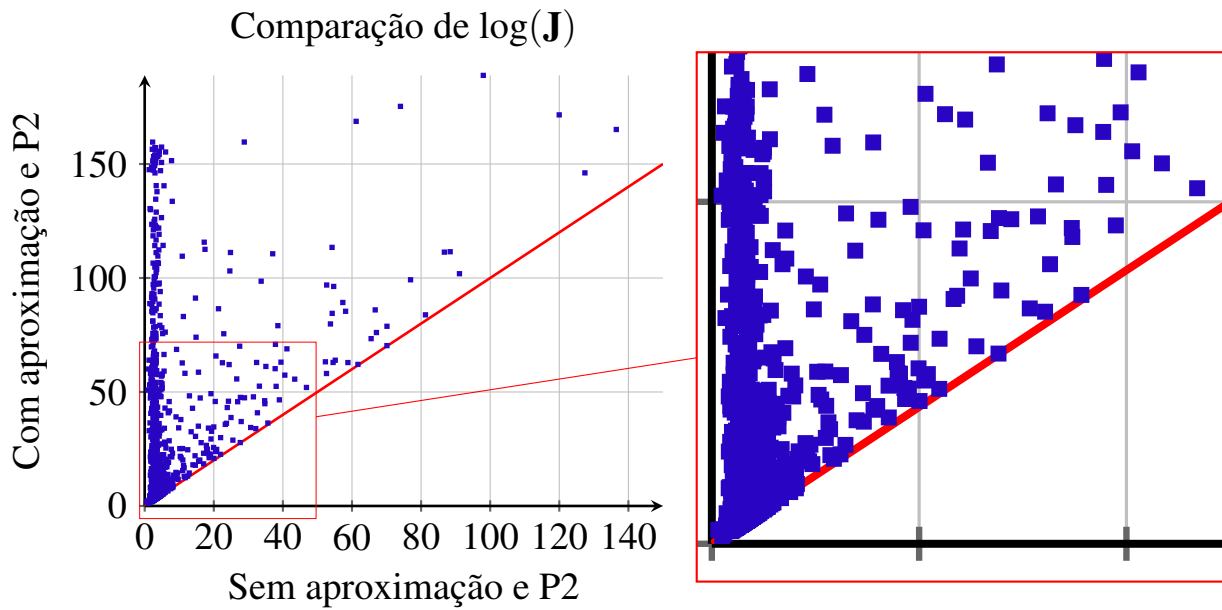


Figura 35 – Comparação dos custos com aproximação quadrática e sem ela, com P3.

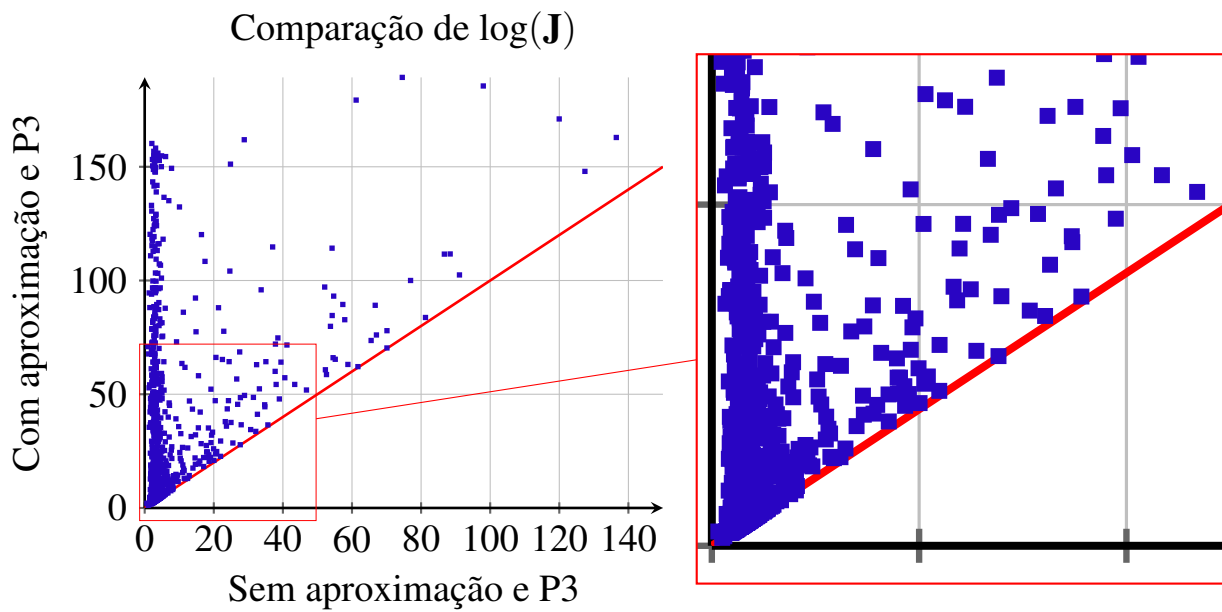
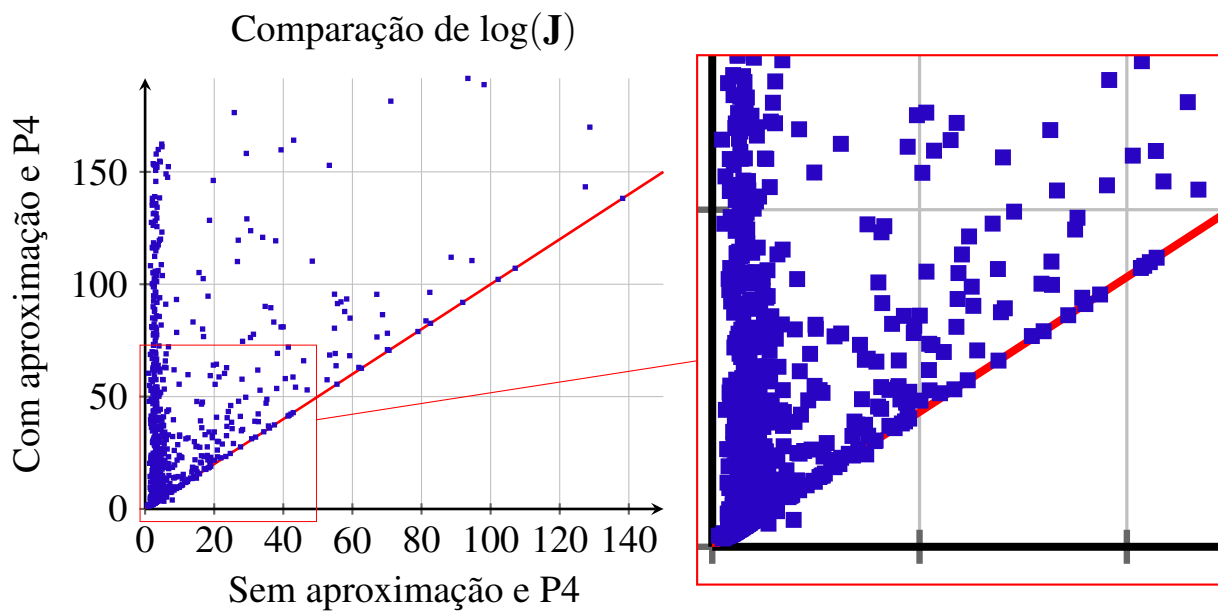


Figura 36 – Comparação dos custos com aproximação quadrática e sem ela, com P4.



D.3 Precondicionadores

Figura 37 – Comparação dos custos com uso do P3 e sem ele.

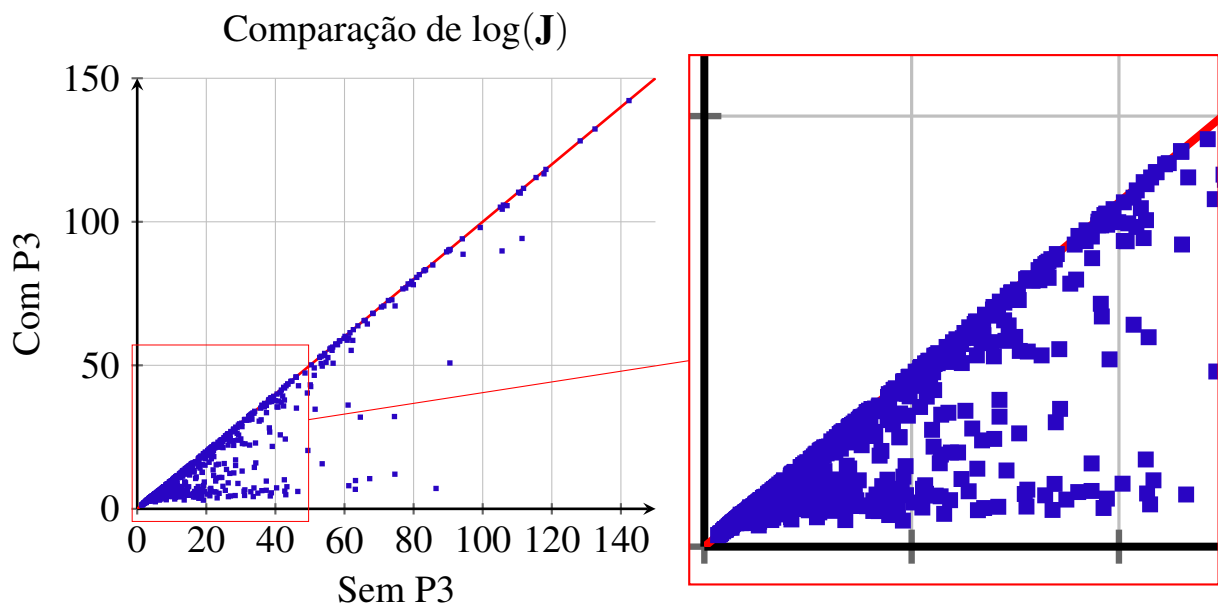


Figura 38 – Comparação dos custos com uso do P4 e sem ele.

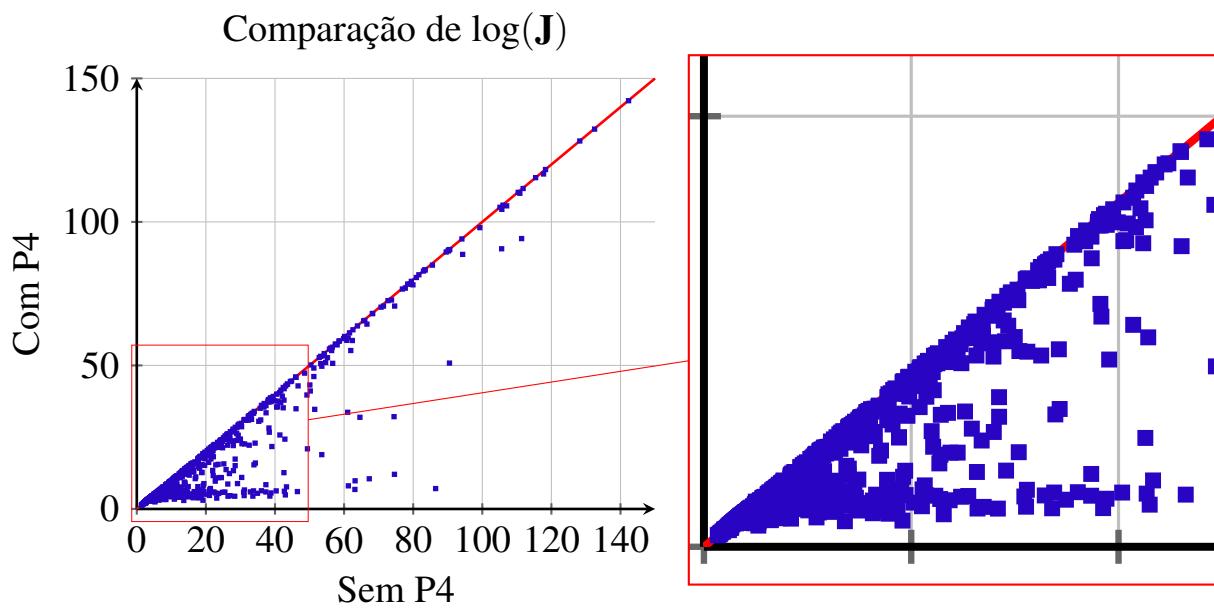


Figura 39 – Comparação dos custos com uso de P1 e P2.

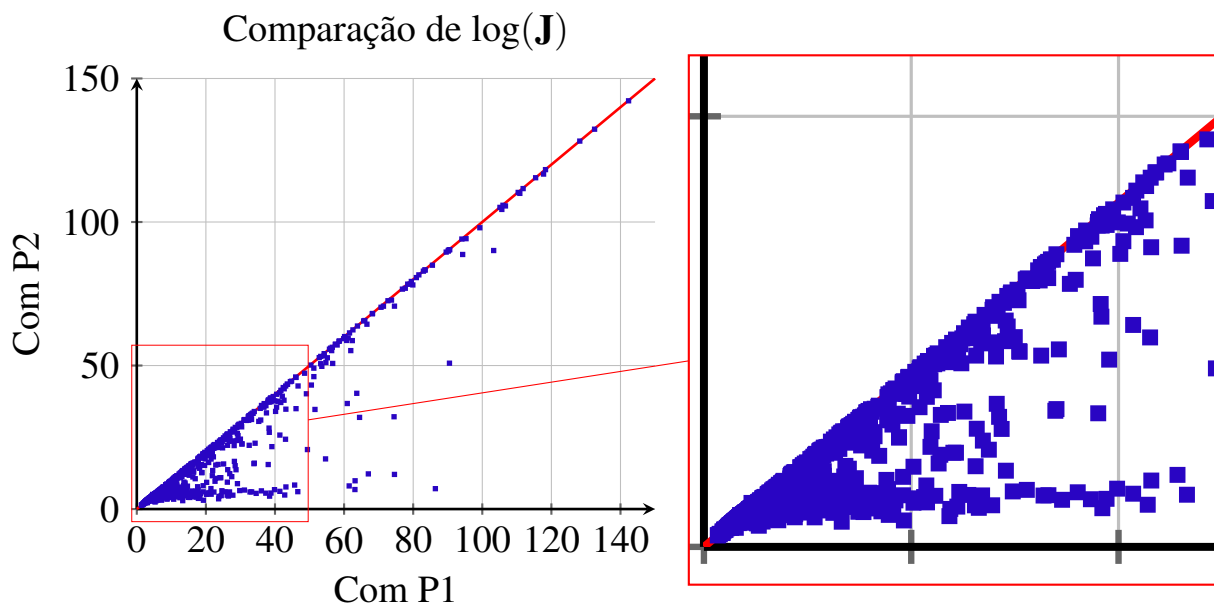


Figura 40 – Comparação dos custos com uso de P2 e P3.

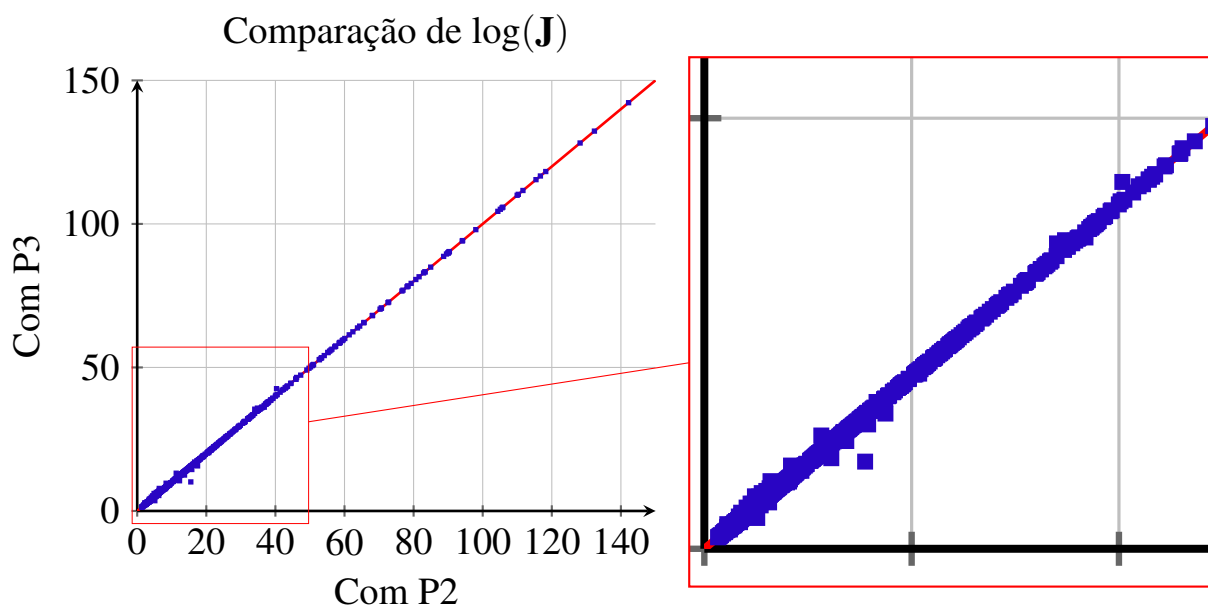


Figura 41 – Comparação dos custos com uso de P2 e P4.

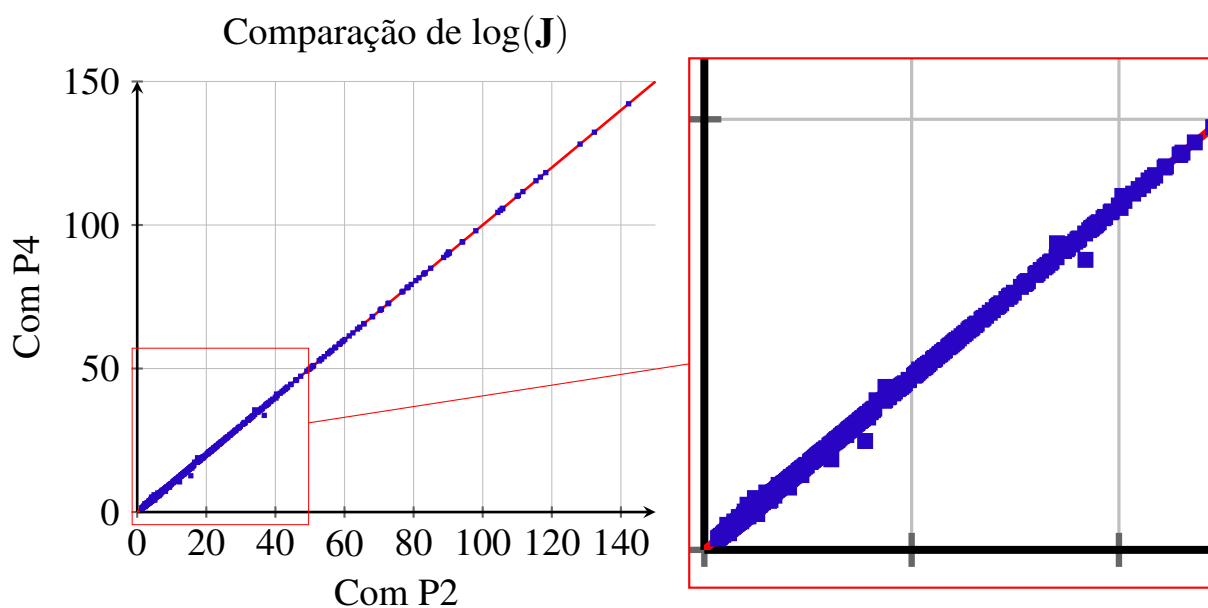


Figura 42 – Comparação dos custos com uso de P3 e P4.

