WATSON ROBERT MACÊDO SANTOS

Métodos para Solução da Equação HJB-Riccati via Família de Estimadores Paramétricos RLS Simplificados e Dependentes de Modelo

São Luís - MA
2014

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnológicas
Programa de Pós Graduação em Engenharia de Eletricidade

WATSON ROBERT MACÊDO SANTOS

Métodos para Solução da Equação HJB-Riccati via Família de Estimadores Paramétricos RLS Simplificados e Dependentes de Modelo

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Eletricidade da UFMA como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia de Eletricidade. Área de concentração: Automação e Controle

Orientador: Prof. Ewaldo Eder Carvalho Santana, Dr.
Co-orientador: Prof. João Viana da Fonseca Neto, Dr.

São Luis - MA
2014

Santos, Watson Robert Macêdo.

Métodos para solução da equação HJB – Riccati via família de estimadores paramétricos RLS simplificados e dependentes de modelo/ Watson Robert Macêdo Santos. – São Luís, 2014.

112 f.

Impresso por computador (fotocópia).

Orientador: Ewaldo Eder Carvalho Santana.

Co-orientador: João Viana da Fonseca Neto.

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade, 2014.

1. Regulador linear ótimo. 2. Teoria do controle ótimo. 3. Modelos multivariáveis. I. Título.

CDU 621.3:004

MÉTODOS PARA SOLUÇÃO DA EQUAÇÃO HJB-RICCATI VIA FAMÍLIA DE ESTIMADORES PARAMÉTRICOS RLS SIMPLIFICADOS E DEPENDENTES DE MODELO

Watson Robert Macedo Santos

Dissertação aprovada em 21 de agosto de 2014.

Prof. Ewaldo Eder Carvalho Santana, Dr.
(Orientador)

Prof. João Viana Fonseca Neto, Dr.
(Co-orientador)

Prof. Vandilberto Pereira Pinto, Dr.
(Membro da Banca Examinadora)

Prof. Allan Kardec Duailibe Barros Filho, Ph.D.
(Membro da Banca Examinadora)

Aos meus pais, aos meus irmãos, aos meus familiares e aos amigos.

Agradecimentos

Deixo aqui registrado meus sinceros agradecimentos a todos que contribuiram de alguma forma para a realização deste trabalho, em especial:

A DEUS pela minha existência, e por permitir o trilhar deste caminho.

Aos meus pais, José Ribamar Santos e Alair Macêdo Santos, pelo incentivo, apoio e compreensão ao longo destes dois anos.

Aos orientadores, Ewaldo Eder Carvalho Santana e João Viana da Fonseca Neto pela amizade, incentivo, motivação e dedicação, fundamentais para o desenvolvimento deste trabalho de pesquisa.

Aos meus irmãos, Welyson José, Marcelo Matreyonne e Carlos Dhreonny pelo apoio nesta caminhada.

À Jacilane Pires, por sua compreensão nos momentos que me ausentei, pelo companheirismo e pelo apoio.

Aos meus familiares pelo encorajamento.

Aos colegas de trabalho mais próximos: Denis Fabrício, Moisés Santos, Aleksandro Nogueira, Marcio Gonçalves, Madson Cruz, Mailson Silva, Luisa Azevedo, Sarah Mesquita e Gustavo Andrade pela convivência descontraída, as trocas de experiências e apoio durante esta jornada.

À professora Patrícia Helena, pelas trocas de experiências e valorosas contribuições para o desenvolvimento deste trabalho.

Aos membros da banca examinadora pelos comentários, sugestões e contribuições, que ajudaram a melhorar a qualidade e a redação final do manuscrito.

Ao Professor José Mairton, pelo suporte e os conhecimentos transmitidos principalmente no primeiro ano desta caminhada.

À PPGEE/UFMA pela estrutura que oferece aos estudantes/pesquisadores, e ao Alcides sempre disponível e disposto em colaborar.

"O importante não é ver o que ninguém viu, mas sim, pensar o que ninguém pensou sobre algo que todo mundo vê."

Arthur Schopenhauer

Resumo

Devido a demanda por equipamentos de alto desempenho e o custo crescente da energia, o setor industrial desenvolve equipamentos que atendem a minimização dos seus custos operacionais. A implantação destas exigências geram uma demanda por projetos e implementações de sistemas de controle de alto desempenho. A teoria de controle ótimo é uma alternativa para solucionar este problema, porque considera no seu projeto as especificações normativas de projeto do sistema, como também as relativas aos seus custos operacionais. Motivado por estas perspectivas, apresenta-se o estudo de métodos e o desenvolvimento de algoritmos para solução aproximada da Equação Hamilton-Jacobi-Bellman, do tipo Equação Discreta de Riccati, livre e dependente de modelo do sistema dinâmico. As soluções propostas são desenvolvidas no contexto de programação dinâmica adaptativa (ADP) que baseiam-se nos métodos para o projeto on-line de Controladores Ótimos, do tipo Regulador Linear Quadrático Discreto. A abordagem proposta é avaliada em modelos de sistemas dinâmicos multivariáveis, tendo em vista a implementação on-line de leis de controle ótimo.

Palavras-chave: Teoria de Controle Ótimo, Regulador Linear Quadrático Discreto, Equação Hamilton-Jacobi-Bellman, Programação Dinâmica Adaptativa, Modelos Multivariáveis.

Abstract

Due to the demand for high-performance equipments and the rising cost of energy, the industrial sector is developing equipments to attend minimization of the theirs operational costs. The implementation of these requirements generate a demand for projects and implementations of high-performance control systems. The optimal control theory is an alternative to solve this problem, because in its design considers the normative specifications of the system design, as well as those that are related to the operational costs. Motivated by these perspectives, it is presented the study of methods and the development of algorithms to the approximated solution of the Equation Hamilton-Jacobi-Bellman, in the form of discrete Riccati equation, model free and dependent of the dynamic system. The proposed solutions are developed in the context of adaptive dynamic programming that are based on the methods for online design of optimal control systems, Discrete Linear Quadratic Regulator type. The proposed approach is evaluated in multivariable models of the dynamic systems to evaluate the perspectives of the optimal control law for online implementations.

Key-words: Optimal Control Theory, Discrete Linear Quadratic Regulator, Equation Hamilton-Jacobi-Bellman, Adaptive Dynamic Programming, Multivariable Models.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos e Notação

ADP	Programação Dinâmica Aproximada
ARE	Equação Algébrica de Riccati
CLQR	Regulador Linear Quadrático Contínuo
EDO's	Equações Diferencias Lineares Ordinárias
DARE	Equação Algébrica Discreta de Riccati
DLQR	Regulador Linear Quadrático Discreto
DP	Programação Dinâmica
HDP	Programação Dinâmica Heurística
HJB	Hamilton-Jacob-Bellman
GA	Algoritmo Genético
LS	Mínimos Quadrados
LPV	Sistemas Lineares a Parâmetros Variáveis
LQR	Regulador Linear Quadrático
LTI	Sistemas Lineares Invariantes no Tempo
LTV	Sistemas Lineares Variantes no Tempo
MIMO	Várias Entradas e Várias Saídas
MLP	Perceptron de Multicamadas
NN	Rede Neural
PDM	Processo de Decisão Markoviano
PI	Política de Iteração
RADP	Programação Dinâmica Aproximada Robusta
RBF	Fução de Base Radial
RL	Aprendizagem por Reforço
RLS	Mínimos Quadrados Recursivos
RND	Rede Neuronal Direta
TD	Diferenças Temporais

A	Matriz de estado contínuo
Ad	Matriz de estado discreto
B	Matriz de entrada contínua
Bd	Matriz de entrada discreta
C	Matriz de saída contínua
Cd	Matriz de saída discreta
D	Matriz de transmissão direta contínua
Dd	Matriz de transmissão direta discreta
y	Variável observada
$\hat{y}$	Variável estimada
y_i	i-ésima variável observada
φ	Variáveis independentes ou regressores
θ	Parâmetros a serem determinados
$\hat{\theta}$	Parâmetro teta estimado
θ^T	Vetor de parâmetros transposto
$V(\theta, t)$	Função de custo
$V_{(xk)}^{kj}$	Função valor do DLQR
$\sum$	Somatório
K	Matriz de ganho
P_{rls}	Matriz de covariância do processo RLS
$P(t)$	Matriz de correlação
Q	Matriz de ponderação de estado
R	Matriz de ponderação de entrada
$\Re$	Conjunto dos números reais
χ_ε	Matriz de erro de Projeção
K^*	Matriz de controle ótimo de malha fechada
K_{rlsP}^{i+1}	Ganho do estimador RLS-Projeção
K_{rlsK}^{i+1}	Ganho do estimador RLS-Kaczmarz
α	Constante positiva
γ	Fator de ajuste do comprimento do passo do parâmetro
λ	Fator de esquecimento
ε	Erro
X	Espaço de estado
U	Espaço de ação
V_k	Função objetivo
V_{kL}	Função objetivo Lagrangeana
f	Função de transição de estado

r	Função de utilidade
u_j^*	Controle ótimo
V_{kL}	Função objetivo Lagrangeana
f	Função de transição de estado
r	Função de utilidade
u_j^*	Controle ótimo
K_{rlsK}	Ganho do RLS-Kaczmarz
K_{rls}	Ganho do RLS-Canônico
Λ	autovalor da matriz $\chi(t)$

Sumário

Capítulo 1

Introdução

A teoria de controle ótimo tem obtido grande sucesso em aplicações realizadas em sistemas lineares, porém este sucesso, está restrito a algumas aplicações *on-line*. De um modo geral, os métodos são off-line porque é necessário realizar a modelagem do sistema dinâmico, o que às vezes, é muito difícil ser realizada, devido as não linearidades ou restrições de tempo do processo. Uma abordagem alternativa para o controle ideal, é a solução baseada em conceitos e algoritmos matemáticos de otimização dinâmica, tais como, o método de Programação Dinâmica de Bellman (DP) e a equação Hamilton-Jacobi-Bellman (HJB)(BELLMAN, 1958) (LJUNG, 1999) e (BERTSEKAS; TSITSIKLIS, 1996). Os recursos computacionais associadas com a realização das formulações são elevados para a aplicação *on-line* (BERTSEKAS, 1995).

Os métodos de Programação Dinâmica Aproximada (ADP) associam a aprendizagem por reforço (controle) e o ambiente (processo). A Aprendizagem por Reforço (RL) é baseada na minimização do erro de Bellman e da HDP, por meio de um esquema que combina a aproximação da função valor do RLS (que é o melhor valor possível do objetivo, escrito como uma função de estado) com as políticas de melhoria. Esta Técnica é aplicada na solução *on-line* de controle ótimo.

Dentre as diversas áreas de aplicação dos métodos *on-line* que necessitam de controladores de alto desempenho, podemos citar os sistemas de controle de aeronaves, navios, veículos automotores, sistemas robóticos, controle de processos industriais, sistemas de controle de temperatura de edifícios, (VINTER, 2010) e (LEWIS; VRABIE, 2009b) na área médica como pode ser verificado em (LIU et al., 2011), na estimação *on-line* na área de transporte (VAHIDI; STEFANOPOULOU; PENG, 2005) e em estimadores de sistemas multivariáveis (LIU; DING, 2013). As técnicas de controle ótimo tem sido usado para projetar e implemantar leis de controle de

maneira off-line. Já as aplicações de forma *on-line* somente passaram a ser exploradas após o desenvolvimento das técnicas neuro-dinâmicas (ANDERSON; MOORE, 1990).

No contexto de Aprendizagem por Reforço, a importância da solução dos Mínimos Quadrados está associada com a Iteração de Política (PI) podendo ser observada no trabalho (BRADTKE; YDSTIE; BARTO, 1994a). Neste contexto, os autores afirmam que a PI do método dos Mínimos Quadrados (LS) é uma classe de algoritmo para aproximação de aprendizagem por reforço.

Nesta dissertação apresenta-se a proposta de uma metodologia para o projeto *on-line* de sistemas de controle ótimo que é constituída pelas soluções da equação Hamilton-Jacobi-Bellman (HJB) do Regulador Linear Quadrático Discreto (DLQR), Aprendizagem por Reforço (RL) e métodos paramétrico do tipo Mínimos Quadrados Recursivos (RLS). As soluções propostas para equação de HJB-Riccati do DLQR são classificadas em dois tipos:

1. Soluções aproximadas pelo método RLS que baseiam-se na transformação de Kronecker e dos vetores de estado para formação da matriz regressores.
2. Soluções baseadas em modelo que tem seus fundamentos nas recorrências de *Riccati* e *Lyapunov* com política de iteração (IP).

Apresenta-se também um método de análise de convergência para a função valor de estado-ação de algoritmos de Programação Dinâmica Heurística (HDP), baseado nas métricas estatísticas do estimador dos mínimos quadrados recursivos (RLS).

As propriedades do RLS são investigadas por aproximações da função de valor na HDP. O desempenho do algoritmo proposto é avaliado em termos da singularidade da matriz de regressores, consistência e convergência do estimador RLS quando associado a métodos HDP.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver uma metodologia para investigar o desempenho e a convergência de algoritmos da família *RLS* para aplicação *on-line* de Controle Ótimo *DLQR* com aplicações em sistemas *MIMO*.

1.1.2 Objetivos Específicos

1. Desenvolver o algoritmo RLS-Projeção e RLS-Kaczmarz para solução HJB-Riccati. Estes desenvolvimentos, encontram-se no contexto de programação dinâmica heurística dependente de ação;

2. Comparar o desempenho de convergência dos RLS-Projeção e RLS-Kaczmarz com os algoritmos RLS-Padrão e método de Schur;

3. Comparar as soluções aproximadas Independentes e Dependentes de modelos matemáticos de sistemas MIMO

1.2 Contribuições

As contribuições estão inseridas no contexto de automação e controle, no sentido de aplicações no setor industrial. Especificamente, as contribuições podem ser vistas como investigações, no contexto de estimação paramétrica, da convergência do processo iterativo para determinação das soluções aproximadas da Equação Algébrica de Riccati para aplicações de on-line de controle ótimo. A seguir, enumera-se as principais contribuições:

1. Tendo em vista aplicações industriais de automação e controle, tem-se como principal contribuição: as diretrizes para o projeto e implementação de procedimentos e algoritmos para realização do projeto *on-line* de sistemas de controle ótimo baseados na família do Método dos Mínimos Quadrados Recursivos e Programação Dinâmica Aproximada.

2. Algoritmos e Procedimento para a solução aproximada da Equação HJB matricial e quadrática que é conhecida como Equação Algébrica de Riccati para o solução do problema DLQR.

3. Procedimento para análise da convergência dos métodos da família RLS no contexto de aprendizado por reforço. Este procedimento baseia-se no comportamento do traço da matrizes de ponderação e dos autovalores do sistema dinâmico que descrevem o comportamento dinâmico do sistema, associando custos, sintonia e dinâmica especifica.

1.3 Estado da Arte

Nesta seção são apresentadas as sínteses dos trabalhos mais recentes que realizam investigações sobre as áreas de Programação Dinâmica Adaptativa, Controle Ótimo, Aprendizagem por Reforço. Desta forma é demonstrada a relevância das pesquisas realizadas nos Laboratórios, de Sistemas Embarcados e Controle Inteligente (LABSECI) e Controle de Processos (LCP), da Universidade Federal do Maranhão, diante do contexto das pesquisas realizadas pela comunidade científica internacional nas áreas de Controle Ótimo e Controle de Sistemas Inteligentes.

A Programação Dinâmica (DP) é uma abordagem para o cálculo da política de controle ótimo ao longo do tempo sob não-linearidade e incerteza, empregando o princípio da otimalidade introduzida por Richard Bellman. Esta ao invés enumerar todas as sequências possíveis de controle, somente procura estado e/ou ação de valores admissíveis que satisfaçam o princípio da otimalidade. Portanto, a complexidade computacional pode ser muito melhorada em relação ao método de enumeração direta. No entanto, os esforços computacionais e a necessidade de armazenamento de dados aumentam exponencialmente com a dimensionalidade do sistema, que se refletem nas três maldições: o espaço de estado, o espaço de observação e o espaço de ação. Assim, a abordagem tradicional DP limitou-se a resolver problemas de pequeno porte (SI et al., 2009).

Para a solução de problemas mais complexos, que necessitam de um conjunto de estratégias mais bem elaboradas, surge como alternativa a Programação Dinâmica Adaptativa ou Aproximada (ADP). Mesmo pequenos problemas podem ser de difíceis resoluções, caso não se tenha um modelo formal do processo, ou se não for conhecida a sua função de transição. Por exemplo, pode-se observar um adversário a jogar poker, mas não será possível descrever a sua lógica para a tomada de decisões, neste caso, não é possível definir a próxima jogada (POWELL, 2007).

No trabalho (WANG et al., 2011), é apresentada uma abordagem da DP para a solução do problema de Controle Ótimo para sistemas não-lineares discretos. É proposto a aplicação de um algoritmo iterativo, afim de obter a lei de controle ótimo que minimize ou maximize uma função de custo, dentro de uma faixa específica de erro, utilizando uma rede neural como aproximador da função de custo.

A ADP e suas aplicações para controle, nos últimos anos, tem apresentado rápidos progressos (JIANG; JIANG, 2013). Um novo horizonte tem sido explorado, sendo chamado de Programação Dinâmica Adaptativa Robusta (RADP), tem sido desenvolvidos para o projeto de controladores ótimos robustos para sistemas lineares e não-lineares, sujeitos a incertezas dinâmicas e paramétricas.

Alguns métodos de inteligência computacional de aprendizagem por reforço, foram reconhecidos como uma aproximação dos métodos de programação de *Bellman*, reconhecidamente, é uma ferramenta para a formulação de políticas de controle ótimo para sistemas não-lineares.

Para a minimização de uma função de custo para qualquer sistema não-linear, são apresentadas duas alternativas, a primeira é denominada de método direto, onde a equação HJB deve ser resolvida através de técnicas matemáticas e a segunda é chamada de método inverso, deve ser considerada a função de controle de Lyapunov e uma lei de controle realimentado. Uma função de custo será projetada de modo que a lei de controle apresentada será ótima para calcular a função de custo (RASTEGARPOUR; SHAHMANSOORIAN; MAZINAN, 2013). Nesta dissertação é proposta uma nova metodologia, de maneira que não há a necessidade de resolver a equação HJB e o controlador subótimo, estes serão projetados sem o uso de métodos numéricos. A função de custo para o sistema, a função de Lyapunov e a lei de controle subótimo são projetados simultaneamente, sendo que a função de controle de Lyapunov será projetada separadamente, através de um algoritmo de Otimização Enxame de Partículas (PSO) ou um Algoritmo Genético(GA).

No trabalho desenvolvido por (SONG; XIAO; LUO, 2013), é proposto um algoritmo ADP iterativo com aproximação de erros finitos, para a solução de problemas de controle ótimo para sistemas não lineares com restrições de controlador. Levando-se em conta as limitações do controlador e um índice de desempenho não-quadrático é introduzido. O algoritmo é usado para obter uma lei de controle iterativo que faz com que as funções de índice de desempenho iterativos possam convergir.

Na Aprendizagem por Reforço baseada na realimentação das saídas e estados para o projeto de controladores crítico-adaptativos, é proposto o uso de aproximadores *on-line* para sistemas discretos MIMO com distúrbios limitados. Uma rede de ação é projetada para produzir o sinal ótimo e um crítico realizará a avaliação de desempenho. O crítico é responsável por calcular a função de *cost-to-go* que está sintonizada *on-line*, através de equações recursivas derivadas da Programação Dinâmica Heurística (HDP). Aqui as Redes Neurais(NN) são utilizadas tanto para o ator como para o crítico, considerando qualquer aproximador *on-line*, tais como: Funções de Base Radial(RBF), Lógica Fuzzy e Algoritmos Genéticos. Para a realimentação de saída, uma NN adicional é utilizada como observador de estados(LIU; WEI, 2014).

Um Regulador Recursivo Robusto para sistemas lineares no tempo discreto, sujeitos a incertezas paramétricas, é proposto em (TERRA; CERRI; ISHIHARA, 2014). A característica do regulador ideal desenvolvido é a ausência de parâmetros de ajuste em aplicações *on-line*. Para alcançar este propósito, uma função de custo quadrática com base na combinação de função de

penalidade e métodos de mínimos quadrados robustos ponderados é formulado.

Um projeto de controlador ótimo contínuo e discreto com base no Regulador Linear Quadrático no tempo contínuo e discreto (CLQR e DLQR). Os controladores foram implementadas em um sistema de acionamento pneumático do pêndulo invertido, que apresenta uma dinâmica não-linear. O objetivo principal é projetar os controladores ideais com a especificação dada para o sistema e para estudar o impacto das matrizes de ponderação sobre o comportamento do sistema, tanto para a discreta como para a contínua (HALDER et al., 2013).

Inspirado na ADP e na Equação algébrica de Riccati (ARE), o trabalho apresentado por (XIE; LUO; TAN, 2013), investiga uma nova estratégia de controle de rastreamento ótimo para uma classe de problemas do regulador linear quadrático (LQR) no tempo discreto com perturbação. O problema de rastreamento ótimo é convertido em um projeto do regulador horizonte infinito ótimo para a dinâmica de erro de rastreamento via transformação do sistema. Em seguida, é calculada a política de controle de rastreamento ótimo, que pode ser considerado como uma forma de resolver o problema de controle ótimo de tempo discreto para o tempo adiante. O algoritmo iterativo ADP via a técnica de Programação Dinâmica Heurística (HDP), é introduzido para resolver a função valor do sistema controlado. Para verificar sua robustez, uma perturbação é adicionado ao sistema controlado.

No trabalho desenvolvido por (PANNOCCHIA et al., 2013), é apresentado um método computacional eficiente para resolver o problema do LQR no tempo contínuo, com restrições nas entradas, horizonte finito e com uma tolerância especificada em projeto. A trajetória de entrada de dimensão infinita é aproximada por meio de uma função linear por partes, em uma discretização de tempo finito, para garantir e satisfazer a restrição de entrada. Este problema de aproximação, é um programa quadrático padrão de dimensão finita que será resolvido por meio de métodos convencionais, gerando um limite superior para função valor ótimo. A discretização tempo finito é refinada, subdividindo os intervalos estimados para causar o maior decréscimo na função de custo.

Na dissertação de (SILVA; LUIS, 2008), foram realizados estudos sobre a aplicação de dois métodos para a solução da Equação Algébrica de Riccati, o primeiro método proposto teve a aplicação de uma Rede Neuronal Direta (RND), que tem a função de erro associada a (ARE). O segundo método proposto, utilizou uma Rede Neuronal Recorrente (RNR), que converte um problema de otimização restrita ao modelo de espaço de estados, em outro de otimização convexa, em função da (ARE) e do fator de *Cholesky*, de maneira a usufruir das propriedades de convexidade e condições de otimalidade. Neste trabalho foi realizada uma proposta para definir os parâmetros para a Rede Neural Recorrente, que foram utilizadas para a solução da equação

algébrica de Riccati por meio da geração de superfícies com a variação paramétrica da RNR, desta forma, realizando uma melhor sintonia da Rede Neuronal e assim obtendo um melhor desempenho.

No trabalho apresentado por (NETO; ANDRADE, 2011), foi proposto a utilização da Função de Base Radial (RBF) para realizar a estimação de parâmetros de maneira *on-line* para encontrar a solução aproximada da equação algébrica de Riccati. A RBF diferencia-se do Perceptron de Multicamadas(MLP) por alguns aspectos, dentre eles destacamos:

- a presença de apenas uma camada intermediária, enquanto que na MLP apresentam-se várias;
- sua ativação é realizada por funções de base radial, no MLP é realizada por funções sigmóides, tangentes hiperbólicas ou outras;
- A ativação interna de cada neurônio é obtida a partir da norma euclidiana ponderada da diferença entre o vetor de entrada e o vetor de centro. Na RBF decrescente, quanto maior a distância, menor é a ativação. Na MLP a ativação se dá pelo produto escalar dos vetores;
- Os neurônios da saída da RBF, são sempre lineares. O uso da RBF para a aproximação segundo (NETO; ANDRADE, 2011), vem da possibilidade de utilizar uma determinada classe de funções por meio da combinação linear de um conjunto de funções não lineares.

1.4 Estrutura da Dissertação

Esta Disssertação está organizada em 6 (seis) capítulos e 2(dois) apêndices. O conteúdo contempla técnicas de aproximação da função valor usando o método dos mínimos quadrados, esquemas de melhorias de políticas de Programação Dinâmica Heurística, desenvolvimento de algoritmos de aproximação para a família dos mínimos quadrados recursivos e análise de convergência do sistema dinâmico.

No Capítulo 2 são tratados conceitos de Programação Dinâmica (PD), conhecida também como otimização recursiva, que baseada no princípio da otimalidade, busca a resposta de como um sistema pode aprender a melhorar o seu desempenho. As definições de Programação Dinâmica Aproximada (ADP), são de fundamental importância no processo de decisões *on-line* para o aproximador, e no processo de aprendizagem por reforço.

No Capítulo 3 são tratatados os conceitos referentes ao método dos mínimos quadrados, que consiste em uma técnica que realiza a minimização de uma função de custo, em seguida, o método dos mínimos quadrados recursivos é apresentado como ferramenta para que seja realizada uma aproximação *on-line* para a solução da Equação Algébrica de Riccati, e por fim, as estrutura dos algoritmos RLS-projeção e RLS-Kaczmarz, que constituem a família dos algoritmos *RLS*.

No Capítulo 4 apresenta-se um método para o projeto *on-line* de sistemas de controle ótimo que baseia-se na aproximação da Equação de Hamilton-Jacobi-Bellman do tipo DARE.

No Capítulo 5 são descritos os procedimentos aplicados na pesquisa. Neste capítulo é descrito o método de parametrização para o regulador linear quadrático através do esquema ator-crítico, aproximações RLS e convergência para algoritmos de programação dinâmica heurística, o projeto do Regulador Linear Quadrático Discreto (DLQR) e os experimentos computacionais, além disso, o conjunto de dados obtidos pelos experimentos computacionais que serão comparados com o conjunto de dados que foram gerados pelo sistema, quando submetido ao Método de Schur em seu regime permanente.

No capítulo 6 encontra-se a conclusão, neste estão as contribuições deste trabalho, no que tange aos resultados obtidos com a análise de convergência dos sistemas: a) terceira ordem, controle longitudinal de uma aeronave, b) quarta ordem, neste caso especificamente, um circuito elétrico RLC de quarta ordem, e c) oitava ordem, o sistema de controle de um helicóptero. Sobre a dinâmica destes casos são realizadas as avaliações de desempenho dos algoritmos de estimação aproximada, bem como, as perspectivas de trabalhos futuros e publicação com aceite obtido.

Nos apêndices estão inseridos tópicos que auxiliam o processo de compreensão das técnicas aplicadas no desenvolvimento deste trabalho. No apêndice A é apresentado o lema da matriz inversa e no apêndice B que trata da programação dinâmica são apresentados os conceitos de otimalidade e as formulações ineretes a DP.

Nas Referências Bibliográficas, lista-se o acervo utilizado na construção das bases sólidas necessárias ao desenvolvimento desta dissertação.

Capítulo 2

Programação Dinâmica Adaptativa e Controle Ótimo

Este capítulo trata da otimização recursiva, cujo o interesse é buscar a resposta de como um sistema pode aprender a melhorar o seu desempenho baseando-se no princípio da otimalidade de Bellman. Em seguida será apresentado um método computacional, escolhido dentre outras técnicas, para a solução Recursiva da Equação Algébrica de Riccati Discreta (DARE). Para a interação contínua com o ambiente, será utilizada a aprendizagem por reforço que minimizará um índice escalar de desempenho e um controle adaptativo será responsável pelos ajustes do modelo de maneira *on-line* durante a operação do sistema. Esse modelo adaptativo é baseado em observações do sistema até o instante atual. A função valor é constituída a partir do reforço atual e dos futuros, de maneira que seja escolhido pelo agente uma determinada política de controle ótimo. Esta política de controle ótimo será aplicada ao ambiente por meio do ator, que é uma estrutura da Aprendizagem por Reforço (RL). Por fim, tem-se a parametrização do Regulador Linear Quadrático Discreto (DLQR), que são mapeamentos lineares e são representados por meio de combinadores lineares dos estados e das saídas do sistema

A Programação Dinâmica (DP), conhecida também como otimização recursiva, é uma técnica que trata de situações nas quais as decisões são tomadas em etapas, com o resultado de cada decisão sendo previsível até certo ponto, antes que a próxima decisão seja tomada. Um aspecto chave destas situações, é que nenhuma decisão pode ser tomada isoladamente, em vez disso, deve-se ponderar o desejo de um baixo custo no presente em relação a altos custos indesejáveis no futuro (SIMON, 2001). Esta se baseia no princípio da otimalidade desenvolvido por *Richard Ernest Bellman* em 1953, que é descrito como:

> "Uma política ótima tem a propriedade que, quaisquer que sejam o estado inicial e a decisão inicial, as decisões restantes devem constituir uma política ótima em relação ao estado resultante da primeira decisão."

Isto portanto, significa dizer que seguir uma política ótima a partir de um estado inicial até um estado final, passando por um estado intermediário, é equivalente a seguir a melhor política do estado inicial até o intermediário, seguida da melhor política deste, até o estado final.

A consequência imediata deste princípio é que, para se definir uma política ótima ou de ações ótimas de um sistema que se encontra no estado x_i, basta encontrar a ação u_i que leva ao melhor estado x_{i+1} e a partir deste, seguir a política ótima até o estado final, o que pode ser observado na Figura 2.1. Esta definição nos permite a construção de diversos algoritmos recursivos para solucionar o problema de otimização multiestágiousando a Programação Dinâmica.

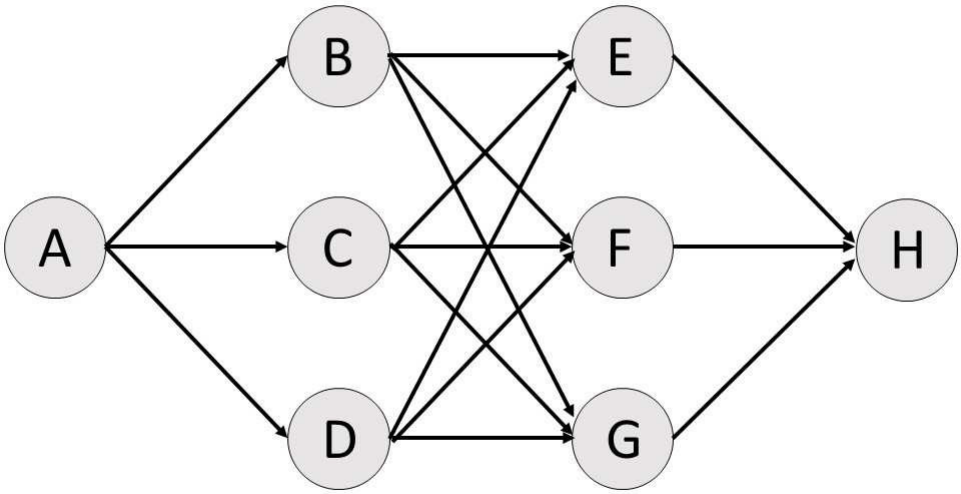

Figura 2.1: Programação Dinâmica

Frequentemente, as soluções são obtidas por um processo regressivo, trabalhando o problema do final para o início. Com isso a dificuldade será reduzida, ao se decompor o problema em uma sequência de problemas inter-relacionados mais simples (BERTSEKAS; TSITSIKLIS, 1996) e (SIMON, 2001). A programação dinâmica tem como questão fundamental, o interesse em buscar a resposta de como um sistema pode aprender a melhorar o seu desempenho a longo prazo, quando isto pode exigir o sacrifício do desempenho atual.

A Programação Dinâmica Heurística(HDP) se apresenta como sendo uma estrutura básica para o desenvolvimento de métodos de Programação Dinâmica Adaptativa(ADP). Tem-se que, independentemente de modelo, um *Crítico* realiza a estimação da *função valor* $V(x_k)$, baseando-se diretamente no estado x_k da planta e o treinamento do controlador necessita encontrar as derivadas a cada instante k. Portanto, o algoritmo HDP se utiliza do modelo do sistema, apenas para realizar a atualização do controlador.

2.1 Processo de Decisão Markoviano

Uma forma de se realizar a modelagem de sistemas ou processos, onde as transições entre estados são probabilísticas, é denominado de Processo de Decisão Markoviano(PDM). Neste é possível observar em que estado se encontra o processo, sendo possível interferir no processo de maneira periódica, executando ações. Cada ação tem uma recompensa (ou custo), que diretamente depende do estado em que se encontra o processo. De maneira alternativa, pode-se definir recompensas por estados apenas, sem que estas dependam da ação executada. Estes são conhecidos de Markov, porque os processos modelados obedecem a propriedade de Markov que afirma: o efeito de uma ação em um estado depende apenas da ação e do estado atual do sistema e não de como o processo chegou a tal estado, e são definidos de processo de decisão porque modelam a possibilidade de um agente interferir periodicamente no sistema executando ações.

O Processo de Decisão Markoviano pode ser representado pela seguinte tupla

$$\Upsilon \;=\; \{X, U, f, r\}\,, \tag{2.1}$$

sendo X, U, f e r são espaço de estado, espaço de ação, função de transição de estado e recompensa ou utilidade, respectivamente. Estes elementos estão associados com a função de custo (também chamada de função valor) que é dada por

$$V(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, u_i). \tag{2.2}$$

Expandindo o lado direito e realizando a manipulação da Equação(2.2), obtém-se uma equa-

ção à diferença, chamada de Equação de Bellman, que é dada por

$$V(x_k) = r(x_k, u_k) + \gamma V(x_{k+1}) \tag{2.3}$$

2.2 Controle Ótimo-Adaptativo e Aprendizagem por Reforço

Na Aprendizagem por Reforço (RL), o aprendizado de um mapeamento de entrada-saída é realizado através da interação contínua com o ambiente, e que tem como objetivo minimizar um índice escalar de desempenho (SIMON, 2001).

A estrutura típica de um sistema de aprendizagem por reforço (RL), apresentada na Figura 2.2, é constituída basicamente de um agente interagindo em um ambiente via sensores e atuadores, que representam a percepção e a ação, respectivamente. A ação tomada muda de alguma forma o ambiente, afetando o estado na tentativa de alcançar o seu objetivo, e as mudanças são comunicadas ao agente através de um sinal de reforço e o próximo estado.

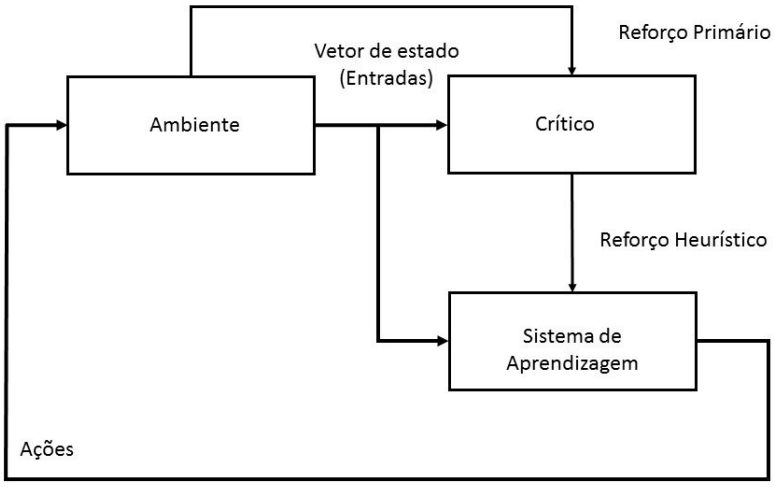

Figura 2.2: Aprendizagem por Reforço (Simon 2001).

Efetivamente a principal diferença entre a aprendizagem por reforço e a aprendizagem supervisionada está vinculada ao tipo de resposta recebida do meio ambiente. Na aprendizagem supervisionada, está disponível uma função que conhece a saída correta para cada uma das saídas do agente, e a aprendizagem é baseada nos dados de erro da saída. Na aprendizagem por reforço, não há o conhecimento da saída correta, o agente recebe apenas a informação do ambiente por um reforço e aplica uma ação de maneira a aumentar a quantidade de recompensa que ele recebe ao longo do tempo. Um fator interessante em relação ao aprendizado supervisionado, é que no desempenho *on-line* a avaliação do sistema ocorre paralelamente à aprendizagem. A aprendizagem por reforço é aplicada quando não é possível utilizar a aprendizagem supervisionada padrão, pois a RL requer o conhecimento das entradas e saídas, que são obtidas através de observações realizadas sobre o ambiente, enquanto que a aprendizagem supervisionada requer o conhecimento de toda a dinâmica do processo, ou seja , para cada entrada conhecida tem-se uma saída conhecida (BUSONIU et al., 2010) (BUSONIU et al., 2011) (SIMON, 2001)(LEWIS; VRABIE, 2009b).

O método de aprendizagem por reforço apresenta um vetor de ação e controle, que é montado a partir do número de ações em sua saída, este vetor é usado pelo agente para influenciar o ambiente. A *RL* tem como objetivo estabelecer uma política de controle ótimo de forma a minimizar o custo acumulado no tempo.

O agente apresenta variações à medida que vai acumulando experiência a partir das interações com o ambiente. O aprendizado pode ser expresso em termos da convergência até uma política ótima que conduza a solução do problema de forma ótima.

O reforço se apresenta na forma de um sinal escalar, este é devolvido pelo ambiente ao agente, sendo emitido, assim que uma ação tenha sido realizada e uma transição de estado tenha sido efetivada. As funções de reforço tem como meta, expressar o objetivo que o agente deve alcançar, desta forma o agente deve minimizar a quantidade de reforços acumulados.

2.2.1 Função Valor

A função valor ou mapeamento do estado, ou ainda par estado-ação, é contituído a partir do reforço atual e dos reforços futuros. Somente o estado x_k é considerado pela função valor, sendo portanto, chamado de $V(xk)$ e assim, denominado de função valor-estado. A função que considera o par estado-ação (x_k, u_k) é denotada por $Q(x_k, u_k)$ e denominada função valor-ação.

Desta maneira, para todo x_k, será escolhido pelo agente uma determinada política de controle ótima $h(x_k) = u_k$. A interação do valor de estado x_k sobre uma política ótima $h(x_k)$, representa

a expectativa de reforço quando se aplica a ação proposta pela política $h(x_i)$, pode ser definida como, segundo (MACIEL, 2012):

$$V_h(x_k) = E\left\{\langle\sum_{i=k}^{\infty}\gamma^{i-k}r(x_i,h(x_i))|x_0=x\rangle\right\} \quad \forall x \in X, \tag{2.4}$$

assumindo que $E\{r(x_k,h(x_k))\} = R(x_k,u_k)$, a Equação(2.4) pode ser reescrita da seguinte forma:

$$V_h(x_k) = R(x_k,u_k) + \gamma\sum_{y\in X}P_{xy}(h(x_k))V_h(y), \tag{2.5}$$

sabe-se que existe uma política ótima $h^*(x_k)$, que define:

$$V^*(x_k) \geqslant V_h(x_k) \quad \forall x \in X, \forall h, \tag{2.6}$$

sendo o valor ótimo:

$$V^*(x_k) = \max_h E\{\langle\sum_{i=k}^{\infty}\gamma^{i-k}r_i\rangle\}. \tag{2.7}$$

A função $V^*(x_k)$ satisfaz a Equação(2.8) , que no caso do horizonte infinito, é conhecida como a equação de *Otimalidade de Bellman*:

$$V^*(x_k) = \max_{u\in U_x}\{R(x_k,u_k) + \gamma\sum_y P_{xy}(u_k)V_h^*(y)\}, \qquad \forall x \in X. \tag{2.8}$$

Considerando a Equação(2.7), obtem-se a política de controle $h^*(x_k)$ dada por:

$$h^*(x_k) = arg\max_{u\in U_x}\{R(x_k,u_k) + \gamma\sum_y P_{xy}(u_k)V_h^*(y)\}. \tag{2.9}$$

Um conjunto de métodos são disponibilizados pela programação dinâmica, que constituem uma metodologia para a solução de problemas ótimos, estes métodos levam em consideração as propriedades estocásticas no qual o estado do processo no futuro depende apenas do estado do

processo e da decisão escolhida no presente, ou seja, este é conhecido como processo de decisão markoviano. Os principais métodos considerados na resolução são:

- Iteração de Política: opera considerando de maneira alternada a iteração de política e a avaliação da política;
- Iteração de Valor: baseia-se em aproximações sucessivas para resolver a equação de otimização de Bellman.

2.2.2 Aprendizagem por Ator-Crítico

A Aprendizagem por Reforço e o Controle Adaptativo estão relacionados fortemente. Dentro da Aprendizagem por Reforço há uma estrutura Ator-Crítico Figura 2.3, sendo o *ator* responsável em aplicar uma política de controle para o ambiente, e o *crítico* é responsável em efetuar a avaliação do novo valor.

A seguir, apresenta-se a estrutura da aprendizagem por ator-crítico, esta é dada em duas etapas, sendo:

- Avaliação de política;
- Melhoria da política.

A avaliação é realizada pelo crítico, sendo concretizada por meio de observações sobre o ambiente, dos resultados e das ações em curso. Para essa avaliação, existe o maior interesse que os sistemas de *RL* que utilizam o esquema Ator-Crítico, sejam baseados no critério da otimalidade de Bellman.

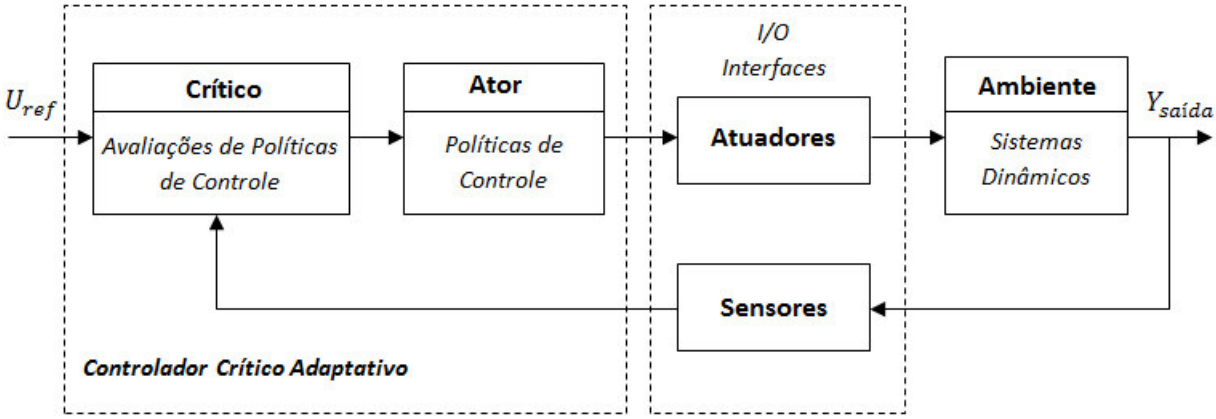

Figura 2.3: Aprendizagem por Ator-Crítico (RÊGO, 2014)

O método de aprendizagem ator-crítico, é uma técnica de aprendizagem que usa as diferenças temporais (TD) que utiliza estruturas separadas de memória para demonstrar a política e a função valor distintamente. O conhecimento prévio do agente é usado para definir qual a melhor decisão a ser tomada. A atualização da função valor irá ocorrer a cada instante de tempo, não necessitando de uma estimativa confiável da função de reforço. O algoritmo utilizado para atualizar $V(K_x)$ em um determinado instante k a uma ação $h(x_k) = u_k$ que é responsável pela transição de estado x_k para x_{k+1} para um reforço imediato $r_k = (x_k, u_k)$, é dado por:

$$V_h(x_k) = V_h(x_k) + \epsilon[r_k + \gamma V_h(x_{k+1}) - V_h(x_k)], \tag{2.10}$$

como ϵ é a taxa de aprendizagem, e expandindo a Equação (3.34), tem-se:

$$V_h(x_k) = E\{r(x_k, h(x_k)) + \gamma\langle \sum_{i=k+1}^{\infty} \gamma^{i-(k+1)} r(x_i, h(x_i)) \ \ |x_0 = x\rangle\}, \tag{2.11}$$

portanto,

$$V_h(x_k) = E\{r(x_k, h(x_k)) + \gamma V_h(x_{k+1}) - V_h(x_k)\}. \tag{2.12}$$

A função valor quando atualizada, ocorre uma aproximação da Equação(2.12) através da utilização de V_k ao invés de V_h. Isto pode ser utilizado como alvo na aprendizagem por diferença temporal, desta forma, o V_k é atualizado a partir da sua diferença com a aproximação

de V_h. Neste caso a forma do erro TD é assumido pelo crítico que é computado através da Equação(2.10). Então, o erro TD será dado:

$$e_k = r(x_k, h(x_k)) + \gamma V_h(x_{k+1}) - V_h(x_k). \tag{2.13}$$

Sendo assim, a regra de aprendizagem é dada pela seguinte relação:

$$V_h(x_k) = V_h(x_{k+1}) + e_k. \tag{2.14}$$

Tem-se, portanto, a regra de aprendizagem via TD. O que se pode considerar, ser um erro de previsão entre as estimativas $V_h(x_k)$ e $V_h(x_{k+1})$, a primeira representa a função valor atual considerada pelo crítico. Após cada decisão tomada pelo ator, o crítico avalia o novo estado do ambiente, de maneira a avaliar se a nova decisão é ou não adequada. Em cada política, o crítico mantém funções de valor diferentes, de forma a manter coerência em seus resultados e permitindo que diferentes políticas possam ser aplicadas para diferentes estados do ambiente. Após cada iteração, o crítico atualizará a função valor conforme a Equação(2.10).

2.3 Parametrização do Problema LQR Discreto

A parametrização do problema de controle ótimo do tipo Regulador Linear Quadrático Discreto (DLQR) está caracterizada como sendo um processo de decisão de Markov (PDM) que é um tipo especial de processo estocástico que possui propriedades de que as probabilidades associadas com o processo num dado instante do futuro dependem somente do estado presente, sendo portanto, independentes dos eventos no passado. No PDM é observado que o modelo do sistema e a política de controle são mapeamentos lineares e são representados por meio de combinadores lineares dos estados e das saídas do sistema (POWELL; RYZHOV, 2012) e (MACIEL, 2012). Tem-se que as paramentrizações dos estados e da política de decisão, respectivamente, $f(x_k, u_k)$ e $h(x_k)$, são dadas por:

$$f(x_k, u_k) = A_d x_k + B_d u_k \tag{2.15}$$

e

$$h(x_k) = -Kx_k, \tag{2.16}$$

sendo $A \in \mathbb{R}^{n\times n}$, tem-se que n é a ordem do sistema, $B \in \mathbb{R}^{n\times n_e}$, sendo que n_e representa o número de entradas da planta e $K \in \mathbb{R}^{n_e\times n}$ é a matriz de ganho de realimentação. Considerando-se que (A, B) é estabilizável, portanto, há uma matriz K que garante que o sistema de malha fechada seja assintóticamente estável. O sistema de malha fechada é dada pela seguinte relação:

$$x_{k+1} = (A - BK)x_k. \tag{2.17}$$

A função de utilidade r associada a este sistema se apresenta sob uma forma quadrática, sendo, portanto, representada da seguinte maneira

$$r(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k, \tag{2.18}$$

tem-se as matrizes simétricas de ponderação Q e R , onde $Q \in \mathbb{R}^{n\times n}$ e $R \in \mathbb{R}^{n_e\times n_e}$.

O DLQR tem como objetivo, definir uma política de controle K que minimize a função de custo a seguir:

$$V(x_k) = \gamma^{i-k} \sum_{i=k}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \tag{2.19}$$

$$V(x_k) = \gamma^{i-k} \sum_{i=k}^{\infty} x_i^T (Q + K^T R K) x_i, \qquad \forall x_k \in X \tag{2.20}$$

Neste caso, o valor ótimo da função de custo assume a seguinte forma quadrática (LEWIS FRANK L E VRABIE, 2012):

$$V(x_k) = x_k^T P x_k, \tag{2.21}$$

para alguma matriz $P \in \mathbb{R}^{n\times n}$ simétrica. Sendo assim, a equação de Bellman para o regulador linear quadrático discreto é dada por:

$$x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + \gamma(x_i^T P x_{k+1}). \tag{2.22}$$

Em função do ganho de realimentação, será dada por:

$$x_k^T P x_k = x_k^T [Q + K^T R K + \gamma (A - BK)^T P (A - BK)] x_k. \tag{2.23}$$

A Equação(2.23) é satisfeita para todos os estados de x_k se:

$$\gamma (A - BK)^T P (A - BK) - P + Q + K^T R K = 0. \tag{2.24}$$

Quando o ganho K é fixado, tem-se que a equação é conhecida como *Lyapunov*. Dado um ganho k estabilizante, a solução desta equação fornece $P = P^T > 0$, tal que $V(x_k) = x_k^T P x_k$ é o custo da política k, portanto:

$$\begin{aligned} V(x_k) &= \sum_{i=k}^{\infty} x_i^T (Q + K^T R K) x_i \\ &= x_k^T P x_k. \end{aligned} \tag{2.25}$$

Desta forma, é definida a função valor em função da política de controle K obtendo a matriz de P de *Riccati*.

2.4 Formulação e Solução do Problema DLQR

2.4.1 Formulação do Problema DLQR

O regulador discreto ótimo de tempo finito é dado por:

$$minJ = G(x_n) + \sum_{k=0}^{N-1} F(x_k, u_k), \tag{2.26}$$

sujeito a seguinte restrição

$$x_{k+1} = A_d x_d + B_d u_k, \tag{2.27}$$

sendo

$$G(x_n) = \frac{1}{2} x_n^T S x_n \tag{2.28}$$

e

$$\sum_{k=0}^{N-1} F(x_k, u_k) = \frac{1}{2} x_N^T \hat{Q} x_N + x_k^T M u_k + \frac{1}{2} u_k^T \hat{R} u_k, \tag{2.29}$$

com o estado inicial x_0, conhecido.

2.4.2 Soluções Exatas e Aproximadas do Problema do DLQR

Desta forma, é expressa a equação de *Ricatti*, segundo o princípio da otimalidade de Bellman. Esta técnica é definida como Programação Dinâmica.

Por conveniência, será considerado $\hat{Q} = Q, \hat{R} = R$, e $M = 0$. Desta forma, simplificaremos as Eqs.(B.47) e (B.48). Portanto, as seguintes formas são obtidas

$$P_j = Q + A_d^T P_{j+1} A_d - (B_d^T P_{j+1} A_d)^T (R + B_d^T P_{j+1} B_d)^{-1} (B_d^T P_{j+1} A_d) \tag{2.30}$$

e

$$u_j^* = (R + B_d^T P_{j+1} B_d)^{-1} (B_d^T P_{j+1} A_d) x_j^*. \tag{2.31}$$

Capítulo 3

Soluções Aproximadas e Dependentes de Modelo da Equação HJB-Riccati

O principal objetivo deste Capítulo, é apresentar os resultados de uma investigação em métodos para solução da Equação HJB-Riccati, no intuito de obter-se as melhores alternativas para resolver o problema de otimalidade que está orientado para o projeto *on-line* dos controladores ótimos do tipo DLQR. Para o desenvolvimento de projeto *on-line* de controladores ótimos, métodos aproximados e dependentes de modelo são investigados para solução recursiva da Equação HJB-Riccati, conforme Figura 3.1.

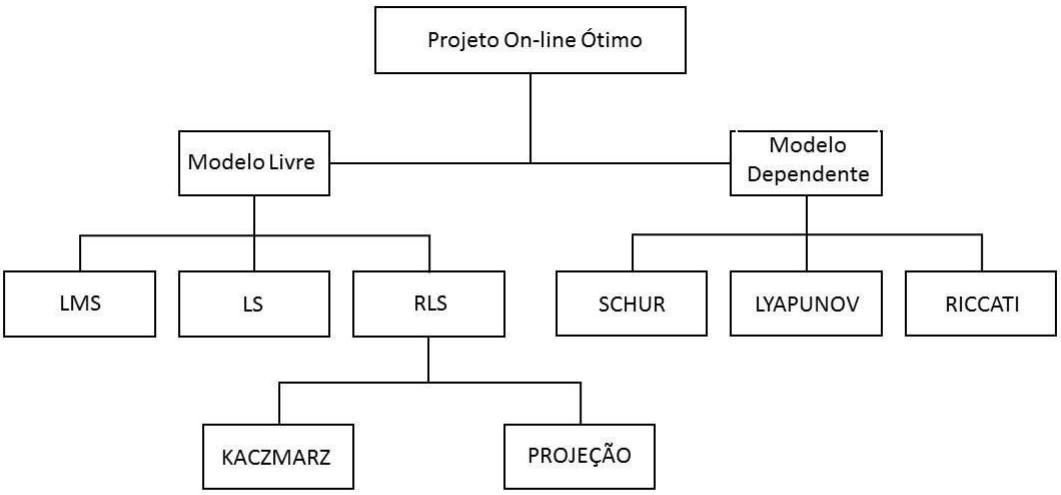

Figura 3.1: Paradigmas para o Projeto de Controladores Ótimos do Tipo DLQR.

Estas investigações são conduzidas no sentido de fomentar o desenvolvimento de formulações que constituem o núcleo dos algoritmos de sintonia ou adaptação dos ganhos de controladores de sistemas dinâmicos. A escolha do método para solução da Equação HJB-DARE está relacionado com o comportamento do sistema dinâmico que é controlado, i.e., leva-se em consideração o tempo disponível para realizar *on-line* a sintonia ou adaptação de ganhos do sistema de controle que é relacionada com os pólos dominantes do sistema dinâmico.

A identificação de sistemas realizado por meio de algoritmos de estimação recursiva, constitui umas das áreas mais importantes em sistemas e processamentos de sinais (SAYED, 2008)(LI; WEN, 2011) e (LI; ZHENG; LIN, 2014). Dentre todos os algoritmos aplicados no processo de identificação recursiva, o algoritmo dos Mínimos Quadrados Recursivos (RLS) é um dos mais populares, o que o faz ser encontrado em aplicações das mais diversas áreas, a qual podemos verificar seu emprego em áreas de controle adaptativo (ASTROM; WITTENMARK, 1994), processamento de sinais e comunicação de redes sem fio (*Wi-fi*)(LI; ZHENG; LIN, 2014).

Neste capítulo é apresentada a fundamentação teórica do Método dos Mínimos Quadrados (LS), que consiste de uma técnica de busca local para realizar minimização da função de custo. Em seguida, apresenta-se a forma recursiva dos estimadores LS, conhecido como estimadores RLS, devido ao algoritmo proposto realizar uma aproximação *on-line* para as soluções da Equação de Riccati Discreta dada por

$$\gamma(A_d^T P A_d) - P + Q - \gamma[A_d^T P B_d (R/\gamma + B_d^T P B_d)^{-1} B_d^T P A_d] = 0. \tag{3.1}$$

A Equação(3.1) que é um caso particularizado da Equação de *Hamilton-Jacobi-Bellman* (HJB) no tempo discreto.

Finalmente, os algoritmos simplificados de *Kaczmarz* e *Projeção*, estes algoritmos fazem parte da família dos algoritmos RLS que visam a minimização do custo computacional em sistemas de controle *on-line*.

3.1 Soluções Aproximadas da Equação HJB-DARE

A solução aproximada da HJB-DARE tem suas origens na vetorização da funcional do problema de controle ótimo discreto apresentada na Seção 2.3 do Capítulo 2.

3.1.1 Formulação do Método dos Mínimos Quadrados

O Método dos Mínimos Quadrados(LS) tem sua origem no século $XVIII$, mais precisamente em 1795, por Carl Friedrich Gauss ao realizar seus estudos astronômicos sobre as órbitas dos planetas e asteróides (ÅSTRÖM; WITTENMARK, 2008), (LJUNG, 1999), (AGUIRRE, 2004) et al.

Trata-se de uma abordagem padrão para a solução aproximada de sistemas sobredeterminados, ou seja, o conjunto de equações apresentam mais variáveis que incógnitas. Quando aplicado a sistemas não lineares, o LS é utilizado para ajustar um conjunto de observações m com um modelo de n parâmetros desconhecidos (ÅSTRÖM; WITTENMARK, 2008).

Carl Friedrich Gauss afirmou que, de acordo com o método LS, todos os parâmetros desconhecidos de um modelo matemático que descreve a dinâmica de um sistema, deve ser escolhido de tal forma que

> "A soma dos quadrados das diferenças entre os valores observados e os valores calculados, multiplicado por um número que mede o grau de precisão, é o mínimo."

Para um modelo matemático, a formulação dos Mínimos Quadrados faz uso da equação de regressão representada da seguinte forma:

$$y(i) = \varphi_1(i)\theta_1 + \varphi_2(i)\theta_2 + \cdots + \varphi_n(i)\theta_n, \tag{3.2}$$

podendo ser dada pela seguinte forma simplificada

$$y(i) = \varphi_{(i)}^T\theta, \tag{3.3}$$

sendo y a variável observada, θ é o vetor de parâmetros, $\theta = [\theta_1, \ldots, \theta_n]^T$, do modelo a serem determinados e φ são as variáveis independentes ou regressores. Caso a matriz de regressores seja não singular, é perfeitamente possível determinar o vetor de parâmetros invertendo-a, ou seja,

$$\theta = \varphi^{-1}(i)y(i). \tag{3.4}$$

3.1.2 Algoritmo RLS Canônico

Para estimação do *RLS*, é assumido que a matriz $\varphi(t)$ tem rank completo, e que, $\varphi^T(t)\varphi(t)$ é não singular, para todo $t \geq t_0$. Dados $\widehat{\theta}(t_0)$ e $P(t_0) = (\varphi(t_0)\varphi(t_0))^{-1}$, o estimador $\widehat{\phi}(t)$ satisfaz as equações recursivas (ÅSTRÖM; WITTENMARK, 2008):

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + K(t)(y(t) - \varphi^T\widehat{\theta}(t-1), \tag{3.5}$$

$$\begin{aligned} K_{rls}(t) &= P(t)\varphi(t) \\ &= P(t-1)\varphi(t)\left(I + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1}, \end{aligned} \tag{3.6}$$

e

$$\begin{aligned} P(t) &= P(t-1) - P(t-1)\varphi(t)\left(I + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1}\varphi(t)P(t-1) \\ &= \left(I - K(t)\varphi^T(t)\right)P(t-1). \end{aligned} \tag{3.7}$$

3.1.3 Algoritmos Simplificados

Os algoritmos simplificados constituem uma boa alternativa na busca da solução de problemas *on-line*, apesar destes apresentarem uma velocidade um pouco menor no processo de convergência, requerem um esforço computacional muito menor, se comparados ao método dos Mínimos Quadrados Recursivos (ÅSTRÖM; WITTENMARK, 2008) e (ISERMANN; MUNCHHOF, 2011).

No algoritmo visto na Subseção 3.3.2, dois conjuntos de elementos de variáveis de estado, θ e P, recebem uma atenção especial, pois devem ser atualizados a cada passo. Para um n muito grande a matriz P é responsável pela maior carga computacional exigida pelo processamento. Desta forma, faz-se necessário a utilização de outros métodos ou algoritmos, que visem evitar a atualização da matriz P, o que significa de maneira imediata um custo mais elevado, levando o algoritmo a uma convergência mais lenta. Como alternativa, são apresentados os métodos de *KACZMARZ* e de *PROJEÇÃO*, como soluções tecnicamente viáveis.

Algoritmo Simplificado de KACZMARZ

Para que seja descrito este algoritmo, é necessário realizar a consideração de que o parâmetro deconhecido seja um elemento do $\mathbb{R}^n$. Uma observação estimada é dada por

$$\hat{y}(t) = \varphi^T(i)\hat{\theta}, \tag{3.8}$$

determina a projeção do vetor de parâmetros $\hat{\theta}$ sobre o vetor $\varphi(i)$. Portanto, para isso, é estritamente perceptível que as n medições, onde $\varphi(1) \dots \varphi(n)$ é espaço no $\mathbb{R}^n$, são unicamente necessárias para realizar a determinação do vetor de parâmetros θ. Admitindo que o vetor de estimação $\widehat{\theta}(t-1)$ é obtido, uma vez que a medida $y(t)$ somente contenha a direção φ, é natural que seja escolhido um novo valor estimado que minimize a função objetivo dada

$$V_k||\hat{\theta}(t) - \hat{\theta}(t-1)||, \tag{3.9}$$

sujeito à seguinte restrição:

$$y(t) = \varphi^T(i)\hat{\theta}(t). \tag{3.10}$$

Introduzindo o multiplicador *Lagrangeano* $\bar{\alpha}$ para trabalhar com a restrição, que irá minimizar a função:

$$V_{kL} = \frac{1}{2}\left(\hat{\theta}(t) - \hat{\theta}(t-1)\right)^T\left(\hat{\theta}(t) - \hat{\theta}(t-1)\right) + \bar{\alpha}\left(y(t) - \varphi^T(t)\hat{\theta}(t)\right), \tag{3.11}$$

calculando as derivadas em relação a $\theta(t)$ e $\overline{\alpha}$, obtém-se o seguinte conjunto de equações

$$\hat{\theta}(t) - \hat{\theta}(t-1) - \bar{\alpha}\varphi(t) = 0 \tag{3.12}$$

e

$$y(t) - \varphi^T(t)\hat{\theta}(t) = 0. \tag{3.13}$$

Resolvendo estas equações (ÅSTRÖM; WITTENMARK, 2008), tem-se que o parâmetro

estimado ótimo é dado por

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\varphi(t)}{\varphi^T(t)\varphi(t)} \left(y(t) - \varphi^T(t)\hat{\theta}(t-1) \right). \tag{3.14}$$

Para que se possa realizar a aceleração no ajuste do parâmetro, é introduzido um fator γ. O que resulta em

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\gamma\varphi(t)}{\varphi^T(t)\varphi(t)} \left(y(t) - \varphi^T(t)\hat{\theta}(t-1) \right), \tag{3.15}$$

tem-se que o ganho, é dado por

$$K_{rlsK} = \frac{\gamma\varphi(t)}{\varphi^T(t)\varphi(t)}. \tag{3.16}$$

Algoritmo Simplificado de PROJEÇÃO

Em alguns sistemas de controle, é passível a ocorrência de uma situação onde $\varphi(t)$ assume o valor de zero, fazendo que o K da Equação(3.16) tenda para o infinito. Para garantir que isto não ocorra, é acrescido ao denominador da Equação(3.16) uma constante α para garantir que o denominador não seja nulo. Desta forma, modifica-se o algoritmo de *Kaczmarz*, dando origem ao algoritmo de *Projeção* (ÅSTRÖM; WITTENMARK, 2008), sendo assim, tem-se

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\gamma\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)} \left(y(t) - \varphi^T(t)\hat{\theta}(t-1) \right). \tag{3.17}$$

Portanto, os fatores α e γ devem assumir valores dentro das seguinte faixas:

$$\alpha \geq 0 \tag{3.18}$$

e

$$0 < \gamma < 2.$$

O parâmetro γ tem o seu valor definido a partir da seguinte análise. Considere que os dados são gerados pela Equação(3.8) com o parâmetro $\theta = \theta^0$. E então, a partir da Equação(3.17) que o parâmetro de erro é dado por

$$\tilde{\theta}(t) = \theta^0 - \hat{\theta}(t), \tag{3.19}$$

que deve satisfazer a equação

$$\tilde{\theta} = \chi(t)\tilde{\theta}(t-1), \tag{3.20}$$

sendo

$$\chi(t) = I - \frac{\gamma\varphi(t)\varphi^T(t)}{\alpha + \varphi^T(t)\varphi(t)}. \tag{3.21}$$

O autovalor da matriz $\chi(t)$ é dado por

$$\Lambda = \frac{\alpha + (1-\gamma)\varphi^T\varphi}{\alpha + \varphi^T\varphi(t)}, \tag{3.22}$$

o valor encontrado, em módulo, é inferior a 1 se $0 < \gamma < 2$. Os demais autovalores assumidos por $\chi(t)$ tem como resultado 1.

O algoritmo simplificado assume que os dados gerados não apresentam erro. Caso seja adicionada uma variável de erro aleatório, o algoritmo assume a forma dada por

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right), \tag{3.23}$$

sendo

$$P(t) = \left(\sum_{i=1}^{t} \varphi^T(i)\varphi(i)\right)^{-1}, \tag{3.24}$$

desta maneira, pode-se afirmar que este é um algoritmo com aproximação estocástica.

3.2 Convergência de Algoritmos RLS

Um dos pontos principais deste trabalho, é o estudo sobre as propriedades de convergência e estabilidade numérica da família de Algoritmos dos Mínimos Quadrados Recursivos, na estimativa dos parâmetros para sistemas multivariáveis (MIMO), que podem ser parametrizados em uma classe de modelos de regressão linear. A sua análise de desempenho indica o quanto que os erros de estimativa convergem para zero.

Nos algoritmos de Mínimos Quadrados Recursivos convencionais, é verificado um problema de precisão numérica, que está associado ao mau condicionamento da matriz de covariância, devido a perda da propriedade de positividade da matriz (POTTER, 1963) citado por (RÊGO, 2014). Diversos tipos de procedimentos foram propostos com a finalidade de melhorar a precisão ou exatidão numérica e preservar a positividade da matriz de covariância do RLS. Diversos autores, entre eles destacamos (LIAVAS; REGALIA, 1999), (LJUNG; LJUNG, 1985) e (SLOCK, 1992) citados por (RÊGO, 2014) desenvolveram técnicas de análise que apresentam explicações para a origem e a propagação do fenômeno de instabilidade numérica.

3.2.1 Convergência de Algoritmos RLS para Esquemas DLQR

Os resultados obtidos para a análise de convergência para esquemas de Programação Dinâmica Heurística (HDP), utilizados para a solução aproximada da Equação Algébrica de Riccati Discreta (DARE), baseados no Método dos Mínimos Quadrados, tem suas provas apresentadas em (ROBINETT et al., 2005) para os problemas que envolvam o Regulador Linear Quadrático (LQR). O teorema a seguir, que pode ser encontrado em (ROBINETT et al., 2005), mostra que a sequência $\{K_j\}_{j=1}^{\infty}$ de políticas geradas pela família de algoritmos HDP-RLS-DLQR convergem para uma política ótima(MACIEL, 2012).

Teorema: Partindo da suposição de que $\{A, B\}$ é um par controlável, K_0 é um controle estabilizante, e o vetor φ_k é excitado persistentemente segundo a equação dada por:

$$\varepsilon_0 I \leq \frac{1}{N} \sum_{i=1}^{N} \varphi_{k-i} \varphi_{k-i}^T \leq \bar{\varepsilon}_0 I, \tag{3.25}$$

para todo $k \geq N_0$, $N \geq N_0$, $\varepsilon_0 \leq \bar{\varepsilon}_0$, com ε_0 e $\bar{\varepsilon}_0$ inteiros positivos e N_0 uma constante positiva. Então existe um intervalo de estimação $N < \infty$ de modo que o esquema de iteração de política aproximada descrita anteriormente, gera uma sequência $\{K_j\}_{j=1}^{\infty}$ de controles estabilizantes, de forma que

$$\lim_{j\to\infty} \|K_j - K^*\| = 0, \tag{3.26}$$

sendo K^* a matriz de controle de realimentação ótima.

A falta de excitação persistente pode ser entendida, no domínio do tempo quando a matriz de autocorrelação é singular. No projeto de HDP utilizando o RLS, uma maneira de se obter uma solução para o problema de falta de excitação persistente consiste em aplicar o processo de revitalização de estados, sendo assim, os estados são constantementes reinicializados a cada intervalo de recorrência. A realização deste procedimento garante que os autovalores da matriz de correlação sejam positivos, ϕ seja não singular.

Para que se possa realizar a mensuração que define a qualidade do condicionamento da matriz de correlação, é o fator de positividade p, que é definido por

$$p = 1 - K_{rls}^T(N)\phi_k \tag{3.27}$$

Sabendo que a definição do vetor de ganho e conhecendo que a matriz de correlação $P(t)$ é simétrica, o que significa dizer que $P(t) = P^T(t)$, desta forma a a Equação(3.27) é reescrita da seguinte forma:

$$p = \frac{\mu}{\mu + \phi_k^T P(t-1)\phi_k} \tag{3.28}$$

3.3 Soluções Dependentes de Modelo da Equação HJB-DARE

Nesta seção serão explorados dois métodos para a solução dependente de modelo da equação HJB-DARE. Os métodos propostos são o de Schur e a recorrência de Riccati. O método de Schur apresenta-se como uma variação dos métodos clássicos, enquanto que a segunda proposta, utiliza dos métodos computacionais para calcular de maneira recursiva a equação de Riccati.

3.3.1 Método de Schur

O método de Schur é uma das técnicas clássicas de autovetores que fornece a solução da equação algébrica de Riccati. Schur fornece uma técnica confiável para a solução numérica da Equação Algébrica de Riccati Discreta (DARE).

Seja U uma matriz ortogonal que transforma a matriz *Hamiltoniana*, na forma real de Schur,

$$T = U^T H U = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \tag{3.29}$$

sendo T_{11} e T_{22} matrizes superiores quase-triangular. Considerando que os blocos da diagonal principal são de ordem $n \times n$, a Equação(3.29) não apresenta solução única, portanto, é possível escolher um valor de U para que os autovalores de T_{11} tenham parte real negativa, enquanto que os autovalores de T_{22}, apresentam parte real positiva. Portanto, a matriz particionada U, é dada por

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \tag{3.30}$$

Considerando que a matriz é não singular e a solução da DARE é semidefinida positiva, tem-se que

$$P(t+1)_{Schur} = U_{21} U_{11}^{-1}. \tag{3.31}$$

3.3.2 Método da Recorrência de Ricatti

Para realizar a solução da equação algébrica de Riccati (DARE) é necessário o uso de métodos computacionais, métodos recursivos ou a utilização de métodos de autovalores e autovetores.

Está no escopo desta seção, mostrar uma técnica capaz de calcular de maneira recursiva, utilizando método computacional, a equação não-linear de Riccati. A Programação Dinâmica (DP) é um método que proporciona a solução recursiva do problema de controle ótimo.

A matriz de ganho de realimentação do DLQR é apresentada a seguir

$$K_j = (R + B_d^T P_{j+1} B_d)^{-1} (B_d^T P_{j+1} A_d). \tag{3.32}$$

O controle ótimo é dado por

$$u_j^* = -K_j x_j^*. \tag{3.33}$$

A solução recorrente da equação de Riccati é obtida a partir da simplificação da Equação(B.49), apresentada no Apêndice B, desta forma, tem-se

$$P_j = Q + A_d^T P_{j+1} A_d - (B_d^T P_{j+1} A_d)^T K_j. \tag{3.34}$$

A partir do pressuposto de que a condição limite $P_N = S$ conforme exposto nas Eqs.(B.24) e (B.34), o método da recursividade resolve facilmente a Equação(3.32) e a Equação(3.34) (LEWIS FRANK L E VRABIE, 2012).

3.3.3 Método da Recorrência de Lyapunov

As principais ferramentas para o desenvolvimento do algoritmo de iteração política (PI) aproximada, para o projeto de controle DLQR, é o método de solução aproximada da Equação de *Lyapunov*, Equação (3.36), e o processo de melhoria da política de controle que é estabelecida por

$$K_{j+1} = \gamma (R + \gamma B^T \widehat{P}^{K_j} B)^{-1} B^T \widehat{P}^{K_j} A. \tag{3.35}$$

O método de aproximação de *Lyapunov* $\widehat{P}^{K_j}$ associado à política atual K_J é o de minimizar o erro na equação

$$\gamma (A - BK)^T P (A - BK) - P + Q + K^T R K = 0, \tag{3.36}$$

no sentido dos mínimos quadrados (LU; FISHER, 1989).

Capítulo 4

Projeto *on-line* de Sistemas de Controle Ótimo

Neste capítulo são apresentados os algoritmos da família dos mínimos quadrados recursivos que serão aplicados na solução aproximada da equação Hamilton-Jacob-Bellman-Riccati(HJB-Riccati) através dos métodos de Schur, RLS-Projeção e RLS-Kaczmarz. Algoritmos de políticas de iteração são utilizados para selecionar uma política de controle ótimo que minimize a função de custo. Em seguida serão apresentados métodos como solução do mínimos quadrados recursivos(RLS) através da aproximação da equação Hamilton-Jacob-Bellman(HJB).

4.1 Descrição do Problema

Os algoritmos HDP são desenvolvidos com base na abordagem dos Mínimos Quadrados Recursivos (RLS) (ASTROM; WITTENMARK, 1994) para aproximar a função de valor do estado. Nesta seção, a estimação RLS e suas variantes são formuladas para aproximar a política de controle ótimo do Regulador Linear Quadrático Discreto(DLQR). O projeto do DLQR é usado como uma parametrização de mapeamentos do Processo de Decisão de Markov(MDP). Nesta dissertação um estudo das equações e as mudanças no método RLS é realizada para resolver de maneira *on-line* a equação HJB. Detalhes sobre a dedução das equações podem ser obtidas a partir das referências (ASTROM; WITTENMARK, 1994).

$$V^{K_j}(x_k) = \varphi^T(x_k)\theta_j, \tag{4.1}$$

e deve concordar com a condição de que a consistência é dada por

$$V^{K_j}(x_k) = r(x_k, h_j(x_k)) + \gamma V^{K_j}(f(x_k, h_j(x_k))). \tag{4.2}$$

4.1.1 O Problema do Tempo para Tomada de Decisão

O problema do tempo para tomada de decisão está associado com o tempo necessário para tomar uma decisão de controle u_k, de forma a adaptar o comportamento do sistema a uma nova situação ou compensar problemas de oscilação. O tempo de decisão tem com parâmetro a frequência de Nyquist do sistema.

4.1.2 O Problema da Seleção do Algoritmo para Solução HJB-DARE

O problema da seleção do algoritmo para solução da equação HJB depende da especificidade do comportamento do sistema dinâmico. Este comportamento, no nosso trabalho é classificado dentro da taxonomia de modelos lineares que são conhecidos como:

- Sistemas lineares a parâmetros variáveis (LPV) - quando os parâmetros (estados, influências externas ou características físicas) podem ser variantes no tempo, ou dependentes de outros fenômenos, alterando a função de transferência ou a representação em espaço de estados do modelo;
- Sistemas lineares invariantes no tempo (LTI) - quando o modelo obedece as propriedades da linearidade (satisfaz o princípio da superposição) e a invariância dos seus parâmetros no tempo;
- Sistemas lineares variantes no tempo (LTV) - quando a variação do vetor de parâmetros pode ser representada como uma função do tempo.

Estes modelos tem como objetivo, realizar a representação do comportamento do sistema nos parâmetros de uma mesma estrutura de equações diferencias lineares ordinárias(EDO's).

4.1.3 O Problema da Complexidade na Escolha do Harwdare

O problema da complexidade na escolha do hardware aborda integração a escolha do melhor hardware associado com os algoritmos HDP-DLQR e o esquema ator-critico da Figura 2.3.

Observa-se que o sistema de aquisição de dados desempenha uma papel relevante na sintonia *on-line* dos ganhos dos controladores ótimos, devido ao nível de interação com o sistema dinâmico necessitar de forma explicita os estados do sistema dinâmico. Enquanto que nos procedimentos dependente de modelos, a iteração do sistema dinâmico é intensa com o hardware do sistema de controle.

4.2 Procedimentos para Projeto Livre de Modelo

O método para o projeto livre de modelo, consiste de uma abordagem para a sintonia de controladores ótimos, em que a solução da equação HJB é uma aproximação, que é obtida por meio de algoritmos de estimação paramétrica, tal como o Método dos Mínimos Quadrados ou o Filtro de Kalman.

4.3 Formulação do Problema

A formulação do modelo de regressores para o problema de aproximação da equação HJB usando a forma de equação de regressão é dada por

$$\theta_{i+1} = \theta_i + K^i_{rls}\left(d_i - \varphi_i^T\theta_i\right) \tag{4.3}$$

$$K^{i+1}_{rls} = P^{i+1}_{rls}\varphi_i\left(\lambda + \varphi_i^T P^{i+1}_{rls}\varphi_i\right)^T \tag{4.4}$$

$$P^{i+1}_{rls} = \frac{1}{\lambda}\left(I - K^i_{rls}\varphi_i^T\right)P^i_{rls}, \tag{4.5}$$

sendo θ_i a variável que representa os parâmetros desconhecidos, ou seja, a solução para a equação HJB. A variável P_{rls} representa a matriz de covariância de busca dos parâmetros θ_i, os valores de P^{i+1}_{rls} no lado direito da Equação(4.4) significa que eles foram atualizados pelo lado direito da Equação(4.5), a atualização de K^i_{rls} ocorre de forma similar. O valor de observação representa os componentes do vetor gradiente da função de custo $V(x)$ em relação ao Estado, e $d_i \in \mathbb{R}$.

A forma quadrática da função de custo é dada por

$$V^K(x_k) = x_k^T P x_k, \tag{4.6}$$

sendo representada em termos do produto de *Kronecker* de estados e em termos da matriz de vetorização P, que é a solução para a equação *Hamilton-Jacob-Bellman*, demosntrada na Equação(3.36), sendo mantida para um melhor entendimento:

$$\gamma(A^T PA) - P + Q - \gamma[A^T PB(R/\gamma + B^T PB)^{-1} B^T PA] = 0.$$

A solução ótima, na sua forma vetorizada é dada por

$$V^{K_j}(x_k) = x_k^T P_j x_k = \overline{x}_k^T vec(P_j), \tag{4.7}$$

sendo $\overline{x}_k \in \Re^{n(n+1)/2}$ é um vetor que de acordo com a definição do produto de *Kronecker* é dado por

$$\overline{x}_k^T = x_k^T \otimes x_k^T = [x_{1k} x_k^T \ \ \ldots \ \ x_{nk} x_k^T]. \tag{4.8}$$

4.3.1 Solução Proposta

Os métodos RLS-Projeção e RLS-Kaczmarz são investigados como uma alternativa que compõem o elemento central, que tem como função atualizar o ganho do estimador RLS que é dado, respectivamente, por

$$K_{rls_P}^{i+1} \ = \ \mu_{rls} \frac{\varphi_i}{\alpha_{rls} + \varphi_i^T \varphi_i} \tag{4.9}$$

e

$$K_{rls_K}^{i+1} \ = \ \mu_{rls} \frac{\varphi_i}{\varphi_i^T \varphi_i}, \tag{4.10}$$

sendo $\alpha_{rls} \geq 0$ o parâmetro usado para resolver o problema de sobrecarga no algoritmo RLS, que são derivados da ocorrência de singularidades do denominador $\varphi_i^T \varphi_i$ e do fator de ajuste de passo μ_{rls} que assume os valores de $0 < \mu_{rls} < 2$.

Assim, a Equação(4.3) da variável desconhecida e a Equação(4.4) de atualização de ganho, são as equações utilizadas para montar o núcleo do algoritmo RLS, a equação de atualização P_{rls} não é necessária para estes dois algoritmos. Estes dois algoritmos têm por função minimizar o esforço computacional, durante a tarefa de estimação dos parâmetros com relação ao método RLS com fator de esquecimento λ, sendo que este assumirá um valor fixo, conforme a subseção (5.1.1). Pode-se observar que α_{rls} realiza a mesma função desempenhada pelo parâmetro λ na equação do método RLS padrão, mas o seu efeito não apresenta o mesmo impacto provocado pelo parâmetro λ, que é o mais abrangente, produzindo desta forma impactos na matriz de atualização P_{rls}.

Os resultados da convergência em esquemas HDP via aproximação RLS, tem sido investigados por (BRADTKE; YDSTIE; BARTO, 1994b) para o regulador linear quadrático (LQR). De acordo com o teorema apresentado em (BRADTKE; YDSTIE; BARTO, 1994b) e (BRADTKE, 1994), a sequência $\{K_j\}_{j=1}^{\infty}$ de políticas geradas pelo algoritmo RLS-HDP-DLQR, convergem para uma política ótima K^*, conforme foi apresentado na Equação(B.27).

4.4 Solução Dependente de Modelo

Nesta seção será apresentado um método para determinar uma política de iteração(PI) aproximada para realizar o projeto de controle do DLQR, que será associado a equação de *Bellman*. Para a concepção do controlador será utilizada como ferramenta, a solução aproximada da equação de *Lyapunov*. A política de iteração pode ser definida como sendo uma função que determina uma decisão, mediante a informação do estado.

4.4.1 Algoritmo de Iteração de Política com DLQR

O Regulador Linear Quadrático Discreto está inserido no contexto do Processo de Decisão Markoviano (MDP) para o projeto HDP proposto. A parametrização do MDP para o problema DLQR e sua associação com a equação de *Bellman* para desenvolver algoritmos de iteração de política (PI) aproximada. As principais etapas dos algoritmos aproximados HDP-PI e ADHDP-PI para DLQR são comentadas em termos de esquemas de ator-crítico(SANTOS et al., 2014), (RÊGO; FONSECA; FERREIRA, 2013) e (ALI; ALI, 2011).

4.4.2 Algoritmo HDP-PI para o DLQR

As principais ferramentas para o desenvolvimento do algoritmo de iteração política (PI) aproximada, para o projeto de controle DLQR, é o método de solução aproximada da Equação de *Lyapunov*(2.24) e o processo de melhoria da política de controle que é estabelecida por

$$K_{j+1} = \gamma(R + \gamma B^T \widehat{P}^{K_j} B)^{-1} B^T \widehat{P}^{K_j} A. \tag{4.11}$$

O método de aproximação de*Lyapunov* $\widehat{P}^{K_j}$ associado à política atual K_J é o de minimizar o erro na Equação(2.24) no sentido dos mínimos quadrados (LU; FISHER, 1989).

Os primeiros blocos do PI para o DLQR são os parâmetros do sistema e configurações de simulação. O segundo bloco implementa as regras para avaliação do PI aproximado. O terceiro bloco implementa as melhorias da política. Estes passos são apresentados no algoritmo 1 (SANTOS et al., 2014) que implementa a aproximação PI para o DLQR.

Algoritmo 1

1 ▷ - **Setup - Condições Iniciais**
2 ▷ Ponderação e Matrizes do Sistema Dinâmico
3 $[Q, R, A, B] \leftarrow [\quad\quad]$
4 ▷ Valores Iniciais de P e K
5 $[P_{j0}, K_{j0}] \leftarrow [\quad\quad]$
6 ▷ Processo Iterativo
7 Para $j \rightarrow N : -1 : 1$
8 Faça
9 ▷ Recorrência de Lyapunov
10 $P_{j+1}(x_k) \rightarrow (A_d - B_d K_j)^T P_j((A_d - B_d K_j) + Q +$
11 $K_j^T R K_j$
12 $P_{j0+1} \leftarrow [\quad\quad]$
13 ▷ Feedback - Ganho Ótimo
14 $K_{j+1} \leftarrow (R + B_d^T P_{j+1} B_d)^{-1} B_d^T P_{j+1} A_d$
15 fim do para
16 ▷ - **Fim do processo Iterativo**

A seguir é exposto um algoritmo que apresenta uma metodologia para resolver o Regulador Linear Quadrático Discreto utilizando a Programação Dinâmica:

ALGORITMO 2 - DLRQ UTILIZANDO PD

1 ▷ - **Inicialização**
2 Sistema Dinâmico Discreto
3 A_d, B_d, C_d e D_d
4 Condição Limite
5 Matrizes de Ponderação
6 Q e R
7 Estados Iniciais
8 ▷ -**Processo Iterativo**
9 Para j de N até passo - 1
10 ▷ -**Ganho Ótimo de Realimentação**
11 $k_j = (R + B_d^T P_{j+1} B_d^{-1} (B_d^T P_{j+1} A_d)$
12 ▷ -**Recorrência de Riccati**
13 $P_{j+1} = (A_d - B_d K_j)^T P_j (A_d - B_d k_j) + k_j^T R k_j + Q$
14 ▷ -**Controle Ótimo**
15 $x_{j+1} = (A_d - B_d k_j) x_j$
16 $u_j^* = -k_j x_j$
17 fim do para
18 ▷ - **Fim do processo Iterativo**
19 ▷ -**Fim do Algoritmo 2**

Deve-se observar a inicialização do algoritmo que leva em consideração a condição limite, que nada mais é, o valor no instante final N assumido pela matriz P da solução da AREu, os estados iniciais, e ainda as matrizes de ponderação Q e R.

Capítulo 5

Testes de Validação e Análise de Convergência dos Métodos Propostos

Os testes de validação e a análise de convergência dos métodos propostos estão organizados em estudos de casos, sendo que as plantas, em cada estudo de caso é representado por modelos matemáticos que descrevem os sistemas dinâmicos multivariáveis (MIMO). A seguir, é apresentada uma breve descrição de cada planta:

1. Modelo matemático de ordem três que representa uma aeronave;
2. Modelo matemático de ordem quatro, que representa um circuito elétrico;
3. Modelo matemático de ordem oito, que representa um helicóptero.

5.1 Estudo de Caso I

A terceira parte do procedimento, é de interesse pelos testes e análises para avaliar a convergência e a precisão dos algoritmos *on-line* RLS-Canônico, RLS-Projeção e o RLS-Kaczmarz, para um modelo de sistema de referência de terceira ordem referenciado em (STEVENS; LEWIS, 2003), o modelo teve as suas equações da dinâmica modificadas em (SILVA; NETO; SOUZA, 2014), para incluir uma segunda ação de controle. Assim, a dinâmica longitudinal da aeronave é representada por um modelo linear MIMO com duas entradas e três saídas. As condições inicias, bem como os experimentos são discutidos nesta seção.

5.1.1 Condições Iniciais

As condições iniciais são organizadas de acordo com sua funcionalidade no processo de busca da solução da Equação Algébrica de Riccati Discreta(DARE). Estas condições que são apresentadas, estão associadas com o projeto do sistema dinâmico, método iterativo de aproximação da DARE, bem como as condições iniciais, e contituem uma interface entre o sistema dinâmico e o método de aproximação da DARE: o ganho K_{HDP} gera o novo vetor de estado, que por sua vez foi estimado pelo RLS. A configuração dos parâmetros é dada por

1. As condições iniciais do vector de estado tem os seguintes componentes $x_1 = 0,22;\ x_2 = -0,25\ e\ x_3 = -0,005$.
2. Solução Inicial da DARE é $P_{HDP} = param_{HDP} I_{n\times n}$ que gera as condições iniciais da matriz de ganho, dada por $K_{HDP} = (R + B_d^T P_{HDP} B_d)^{-1} Bd^T P_{HDP} A_d$.
3. As matrizes de ponderação para o projeto do DLQR, são dadas por $R = I_{2\times 2}$ e $Q = I_{3\times 3}$.

Condições iniciais para o RLS:

1. As condições iniciais para os parâmetros desconhecidos θ são $\theta_i = 0$, com $i = 1, \ldots, 6$.
2. O fator de esquecimento $\lambda = 0,96$.
3. A matriz de covariância do processo RLS é dado por $P_{rls} = param_{rls} I_{6\times 6}$, e o processo RLS é dado por $K_{rls} = param_{rls} I_{6\times 6}$ são associados durante o processo de atualização.

5.1.2 Configuração do Processo Iterativo

A configuração do processo iterativo, consiste em estabelecer os melhores parâmetros para a solução de uma determinada aplicação, tendo como referência o empirismo do projetista e/ou as correlações entre as variáveis relacionadas do sistema com os métodos. Os parâmetros que desempenham um papel relevante no método convergência são: ordem do sistema n ($n = 3$), intervalo de amostragem T_{amost} ($T_{amost} = 0,1s$), fator de esquecimento (RLS) λ ($\lambda = 0,982$) e os parâmetros de projeção α_{rls} e μ_{rls}.

5.1.3 Análise de Desempenho dos Algoritmos RLS-HDP

Nesta subseção será utilizado um modelo de terceira ordem, aeronave F-16, para avaliar o desempenho da família de algoritmos RLS, para o projeto de controle ótimo.

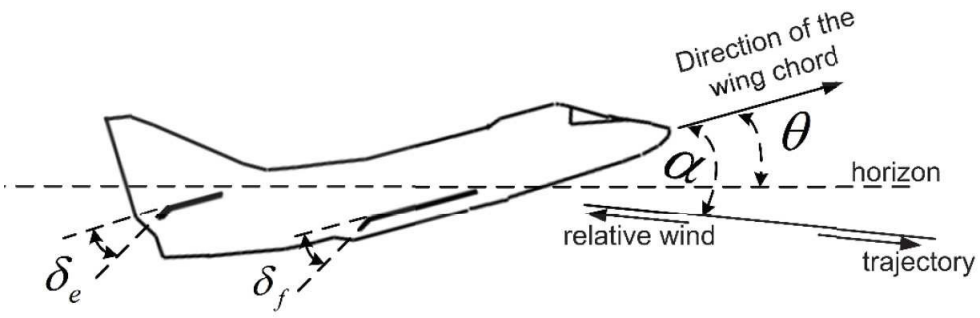

Figura 5.1: Movimento Longitudinal da Aeronave F-16, para Levantar e Baixar o Nariz (SILVA; NETO; SOUZA, 2014)

Para o modelo proposto, apresenta-se as seguintes variáveis de entrada:

- Taxa de elevação q, que representa a variação do ângulo de elevação θ em relação a linha do horizonte;
- Ângulo de ataque α, corresponde a diferença entre a trajetória e o curso real da aeronave;
- Ângulo de defecção do elevador δ_e.

E as seguintes variáveis de saída:

- μ_1 atuação em relação ao sistema de elevação definido por δ_e, age sobre a cauda da aeronave;
- μ_2 executa ação de controle sobre os flaps das asas.

Os parâmetros do sistema dinâmico foram discretizados, para isso foi aplicada uma taxa de amostragem de ($T_{amost} = 0,1s$), sendo portanto as matrizes, de estado (A), de entrada (B), de saída (C) e de transmissão direta (D), dadas por:

$$A = \begin{bmatrix} 0,9124 & 0,0829 & 0,0006 \\ 0,3724 & 0,9428 & -0,0103 \\ 0 & 0 & 0,3679 \end{bmatrix}, \tag{5.1}$$

$$B = \begin{bmatrix} -0,0066 & 0,8295 \\ -0,1034 & 9,4277 \\ 3,6788 & 0 \end{bmatrix}, \tag{5.2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.3}$$

e

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.4}$$

A solução de estado estacionário do RLS-HDP de *Riccati/Lyapunov*, baseada nas matrizes A, B, C e D, é dada por

$$P_{HDP}^{final} = \begin{bmatrix} 4,0154 & 0,0070 & 0,0001 \\ 0,0070 & 1,0097 & -0,0006 \\ 0,0001 & -0,0006 & 1,0098 \end{bmatrix}. \tag{5.5}$$

A matriz de ganho RLS-HDP é dada por

$$K_{HDP}^{final} = \begin{bmatrix} 0,0006 & -0,0001 & 0,0932 \\ 0,0710 & 0,0989 & -0,0001 \end{bmatrix}. \tag{5.6}$$

A solução da equação HJB discreta pelo método de ***Schur*** e os ganhos ótimos associados, são utilizados como referência para comparação de precisão, com os valores aproximados de P e K, respectivamente, supondo que os valores de *Schur* são verdadeiros. A solução discreta através do método de *Schur*, é dada por

$$P_{HDP_S}^{final} = \begin{bmatrix} 4,0154 & 0,0070 & 0,0001 \\ 0,0070 & 1,0097 & -0,0006 \\ 0,0001 & -0,0006 & 1,0098 \end{bmatrix}. \tag{5.7}$$

A matriz de ganho discreta é dada por

$$K_{HDP_S}^{final} = \begin{bmatrix} 0,0006 & -0,0001 & 0,0932 \\ 0,0710 & 0,0989 & -0,0001 \end{bmatrix}. \tag{5.8}$$

Para os métodos de *Projeção* e *Kaczmarz*, tem-se os seguintes valores para a solução discreta e as matrizes de ganho, respectivamente de cada método:

Solução discreta obtida a partir do método de *Projeção*

$$P_{HDP_P}^{final} = \begin{bmatrix} 4,0154 & 0,0070 & 0,0001 \\ 0,0070 & 1,0097 & -0,0006 \\ 0,0001 & -0,0006 & 1,0098 \end{bmatrix}. \tag{5.9}$$

A matriz de ganho discreta definida pelo método de *Projeção*, é dada por

$$K_{HDP_P}^{final} = \begin{bmatrix} 0,0006 & -0,0001 & 0,0932 \\ 0,0710 & 0,0989 & -0,0001 \end{bmatrix}. \tag{5.10}$$

A solução discreta para o método *Kaczmarz*

$$P_{HDP_S}^{final} = \begin{bmatrix} 4,0154 & 0,0070 & 0,0001 \\ 0,0070 & 1,0097 & -0,0006 \\ 0,0001 & -0,0006 & 1,0098 \end{bmatrix}. \tag{5.11}$$

A matriz de ganho discreta do método de *Kaczmarz*, é dada por

$$K_{HDP_S}^{final} = \begin{bmatrix} 0,0006 & -0,0001 & 0,0932 \\ 0,0710 & 0,0989 & -0,0001 \end{bmatrix}. \tag{5.12}$$

Os valores de estado estáveis da matriz P^{θ} dada por (5.7),(5.9) e (5.11) são comparados com a solução dada pela HJB (5.5) que é obtida através do método de *Schur*. O erro calculado entre cada elemento da matriz é inferior a 10^{-4}. A avaliação da precisão numérica mostrou que o vetor de parâmetro estimado $\widehat{\theta}$ converge para o valor verdadeiro do vetor θ^0. De acordo com as propriedades do RLS, esta ocorrência é uma indicação de que o estimador não polarizado.

A avaliação do processo de iteração para a solução da equação HJB-Riccati através do algoritmo RLS-HDP é apresentado nas Figuras 5.2 e 5.5, para um valor de 500 ciclos de iteração. As curvas (a)-(c) da Figura 5.2 representam o comportamento de convergência dos elementos p_{11}, p_{22} e p_{33} da matriz P, que correspondem respectivamente aos elementos θ_1, θ_4 e θ_6 do vetor de parâmetros θ, ou seja, a diagonal principal da matriz. As curvas (a)-(c) da Figura 5.5, apresentam a evolução do comportamento dos elementos p_{12}, p_{13}, e p_{23} da matriz P que estão associados aos elementos θ_2, θ_3 e θ_5 do vetor θ, de forma recíproca.

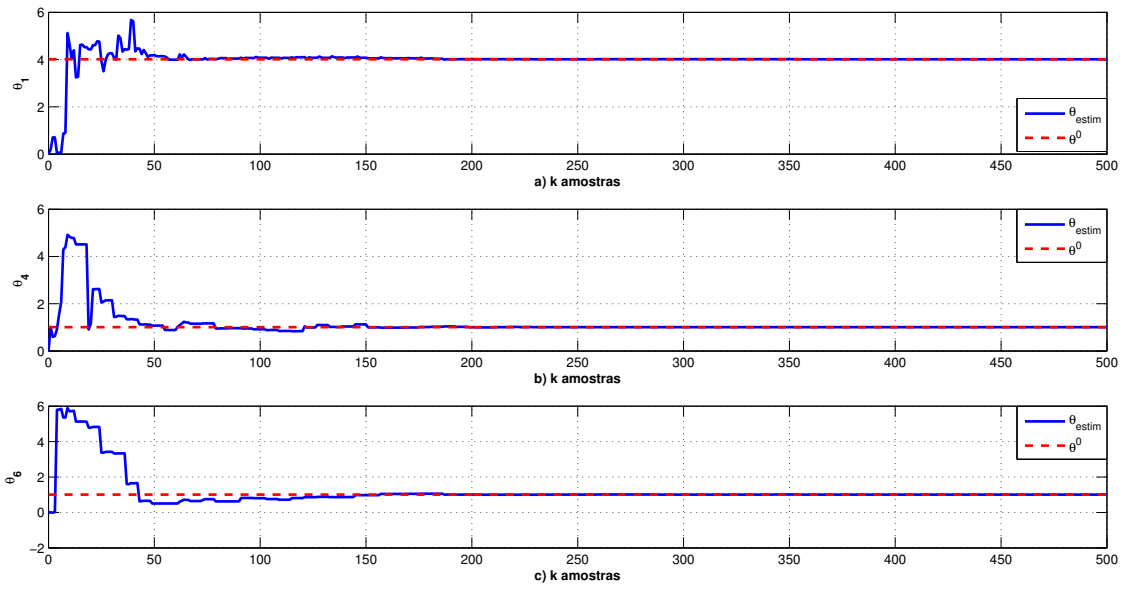

Figura 5.2: Convergência dos Parâmetros da Diagonal Principal pelo Método RLS-Canônico do Modelo de 3^a ordem

Na Figura 5.2 que apresenta a evolução do comportamento do método RLS-Canônico (gráfico contínuo) em relação ao método de Schur (gráfico tracejado), tem-se a evolução dos parâmetros da diagonal principal e a sua convergência. O sistema, em sua dinâmica, responde de maneira satisfatória e apresenta convergência por volta de 200 amostras.

Na Figura 5.3, tem-se a evolução dos parâmetros da diagonal principal pelo método de *RLS-Projeção* (gráfico contínuo) em relação ao método de Schur (gráfico tracejado), e a sua convergência é atingida por volta de 170 amostras.

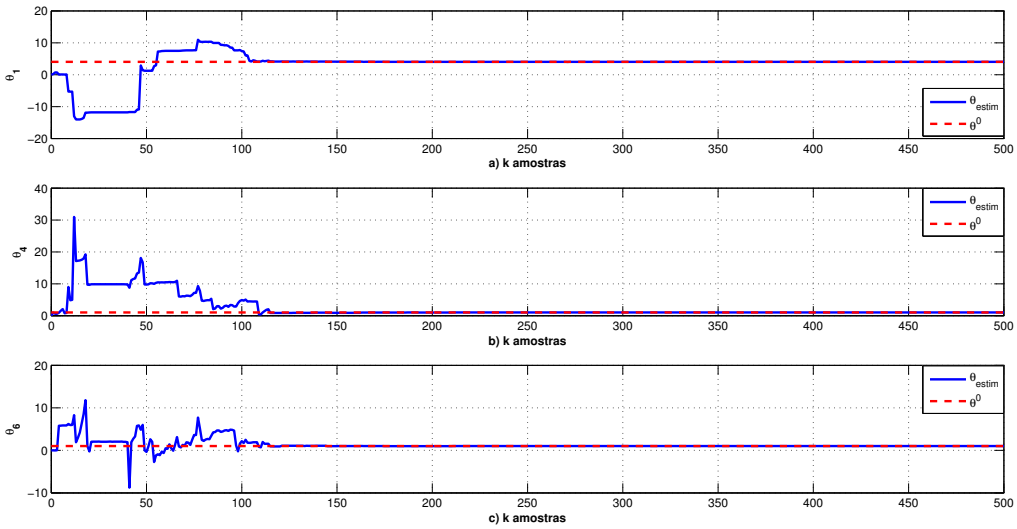

Figura 5.3: Convergência dos Parâmetros da Diagonal Principal pelo Método RLS-Projeção do Modelo de 3^a ordem

Na Figura 5.4, tem-se a evolução do comportamento do método *RLS-Kaczmarz* (gráfico contínuo), de sua diagonal principal, bem como a sua convergência que foi atingida por volta de 110 amostras. Portanto, o método RLS-kaczmarz apresentou um melhor desempenho de convergência para os elementos da diagonal principal.

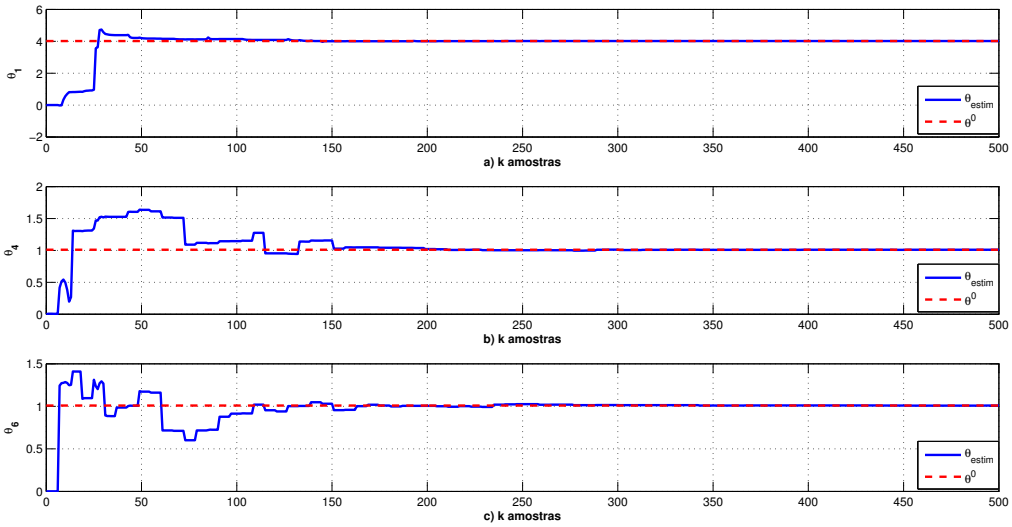

Figura 5.4: Convergência dos Parâmetros da Diagonal Principal pelo Método RLS-Kaczmarz do Modelo de 3^a ordem

Nas Figuras 5.5, 5.6 e 5.7, tem-se respectivamente, os resultados pelo método RLS-Canônico, Projeção e Kaczmarz referentes aos θ_2, θ_3 e θ_5 que são comparados com os resultados obtidos pelo método de Schur. Na devida ordem, tem-se que as convergências ocorreram por volta das 180, 110 e 250 amostras, respectivamente.

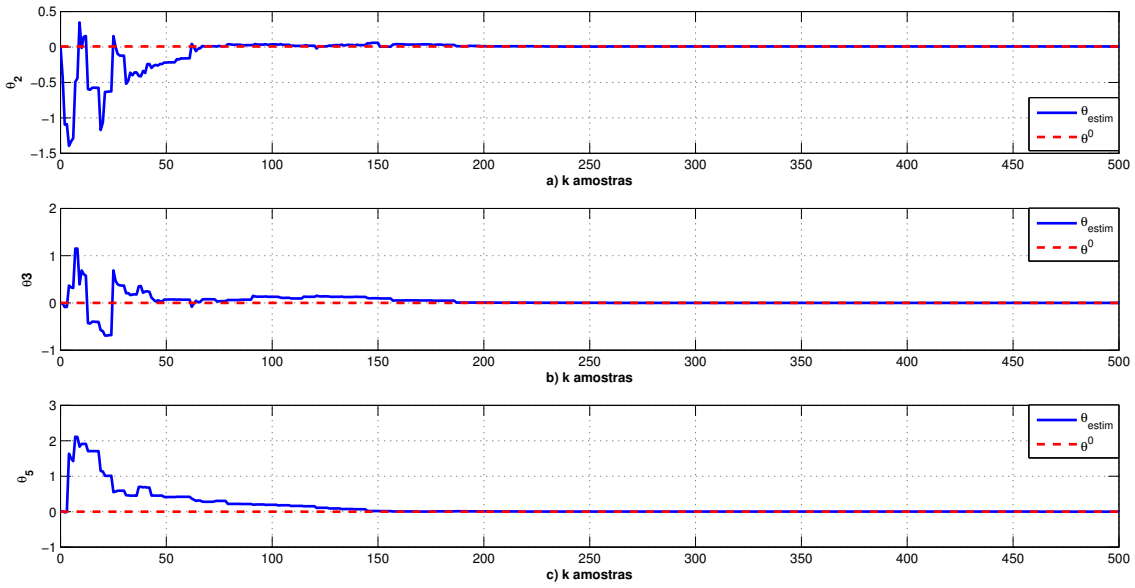

Figura 5.5: Convergência dos Parâmetros θ_2, θ_3 e θ_5 pelo Método RLS-Canônico do Modelo de 3^a ordem

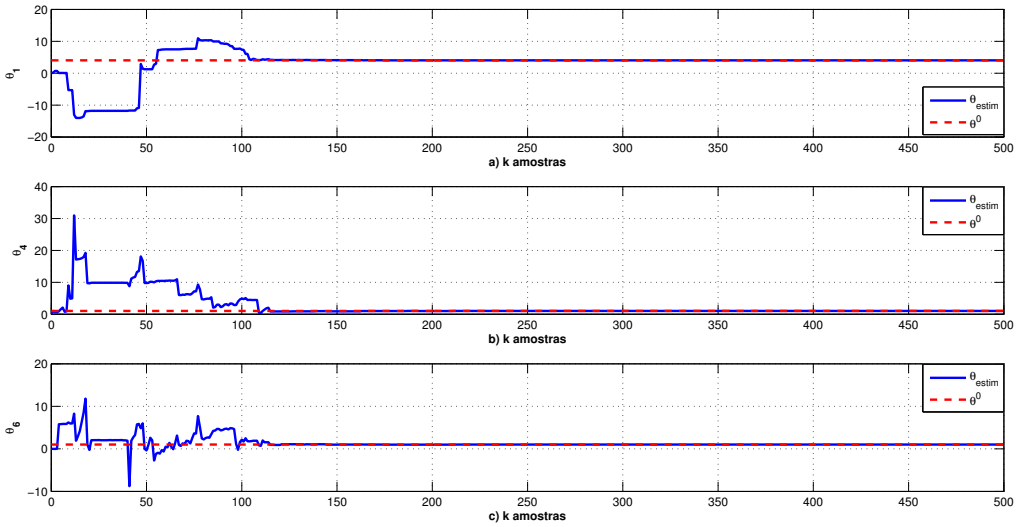

Figura 5.6: Convergência dos Parâmetros θ_2, θ_3 e θ_5 pelo Método RLS-Projeção do Modelo de 3^a ordem

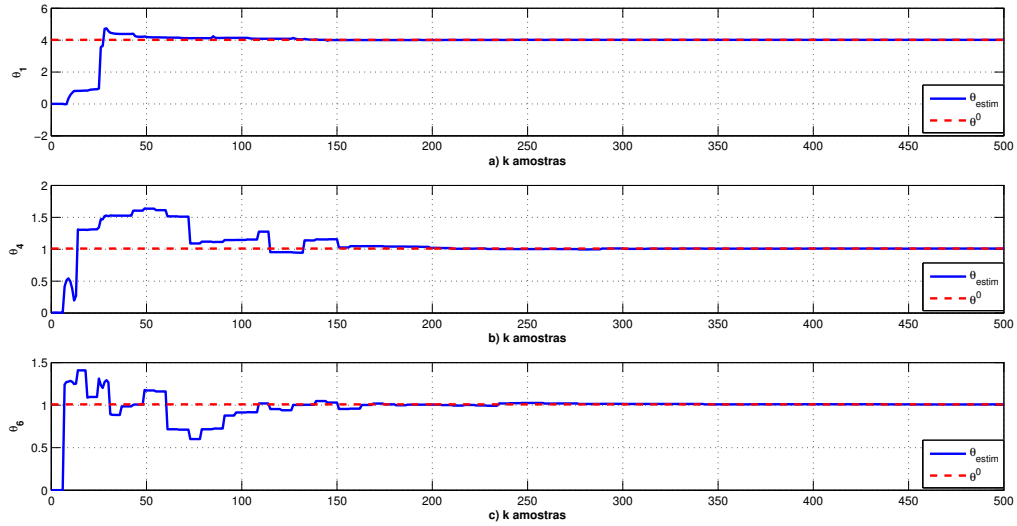

Figura 5.7: Convergência dos Parâmetros θ_2, θ_3 e θ_5 pelo Método RLS-Kaczmarz do Modelo de 3^a ordem

As figuras representam o comportamento dos estados e as de controle do sistema dinâmico, à medida que o tempo decorre é verificado que as iterações do algoritmo ocorrem, os estados apresentam valores oscilatórios, o que é característico do processo de revitalização e da condição de excitação persistente.

As Figuras 5.8, 5.9 e 5.10 apresentam a evolução das ações de controle para os métodos RLS-Canônico, RLS-Projeção e RLS-Kaczmarz, respectivamente, para o modelo da Seção 5.1 apresentado.

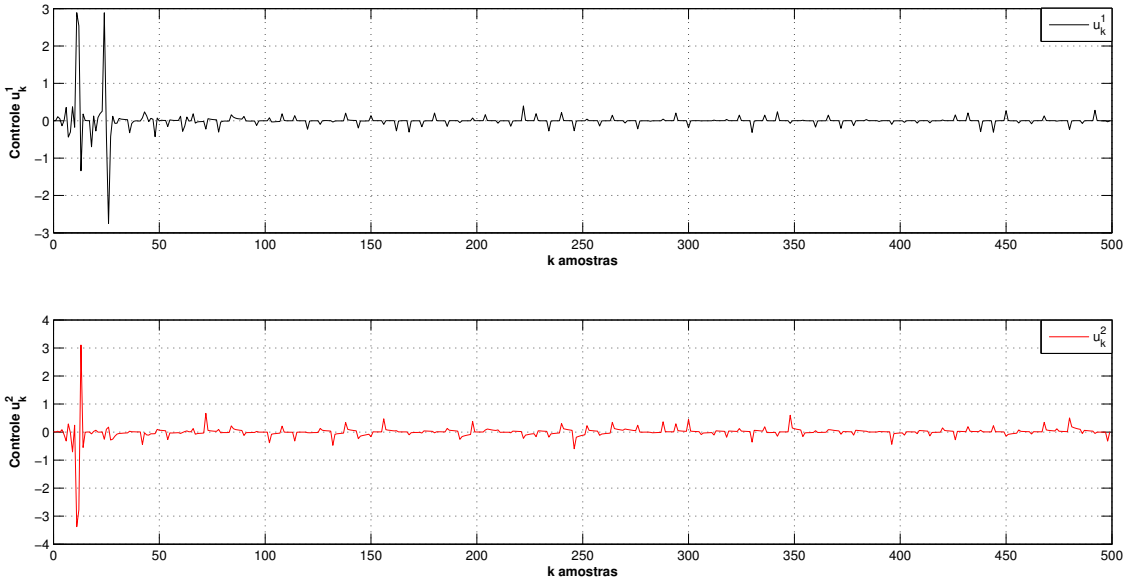

Figura 5.8: Gráfico da Ação de Controle u_k pelo Método RLS-Canônico do Modelo de 3^a ordem

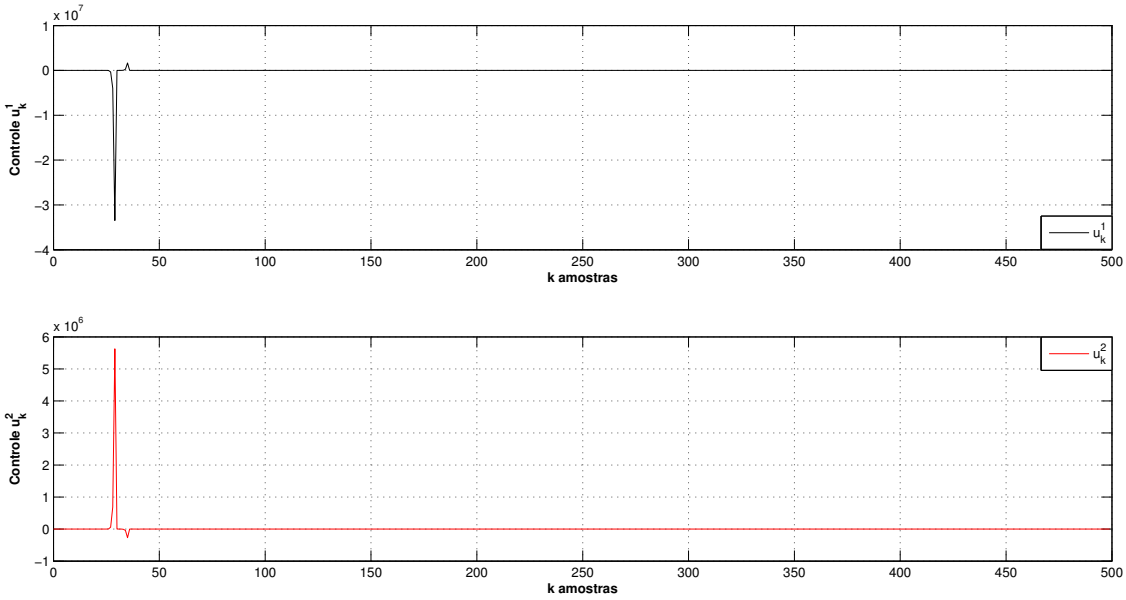

Figura 5.9: Gráfico da Ação de Controle pelo Método RLS-Projeção do Modelo de 3^a ordem

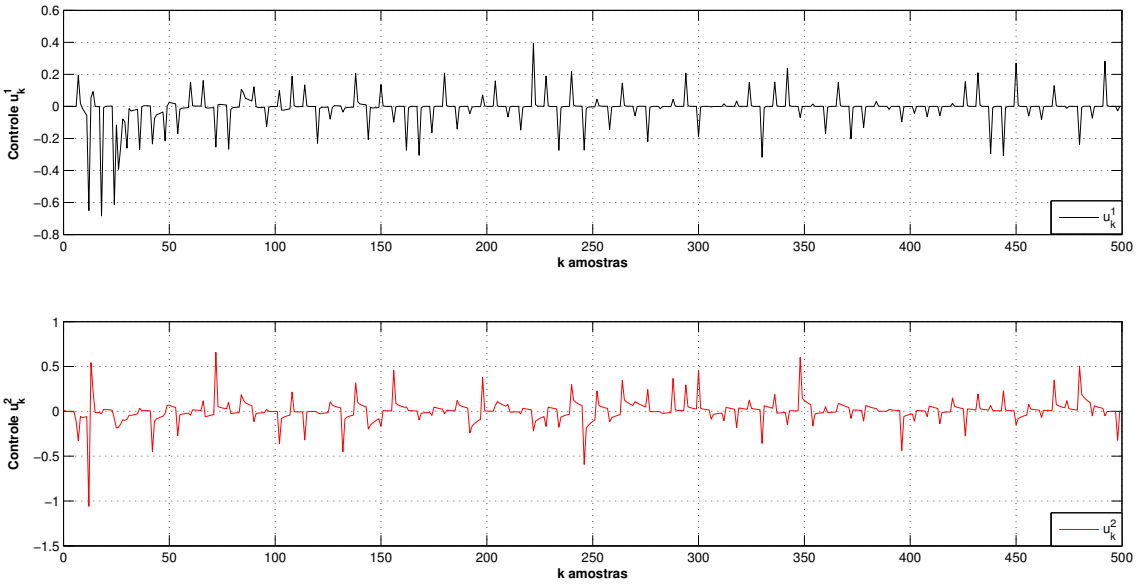

Figura 5.10: Gráfico da Ação de Controle pelo Método RLS-Kaczmarz do Modelo de 3^a ordem

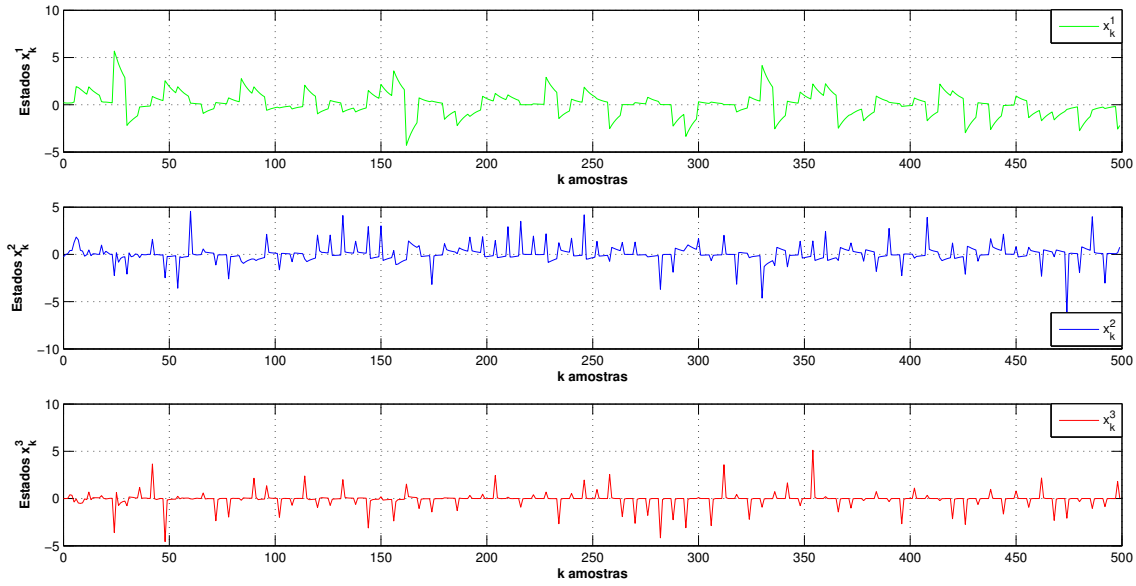

Figura 5.11: Gráfico dos Estados RLS-Canônico do Modelo de 3^a ordem

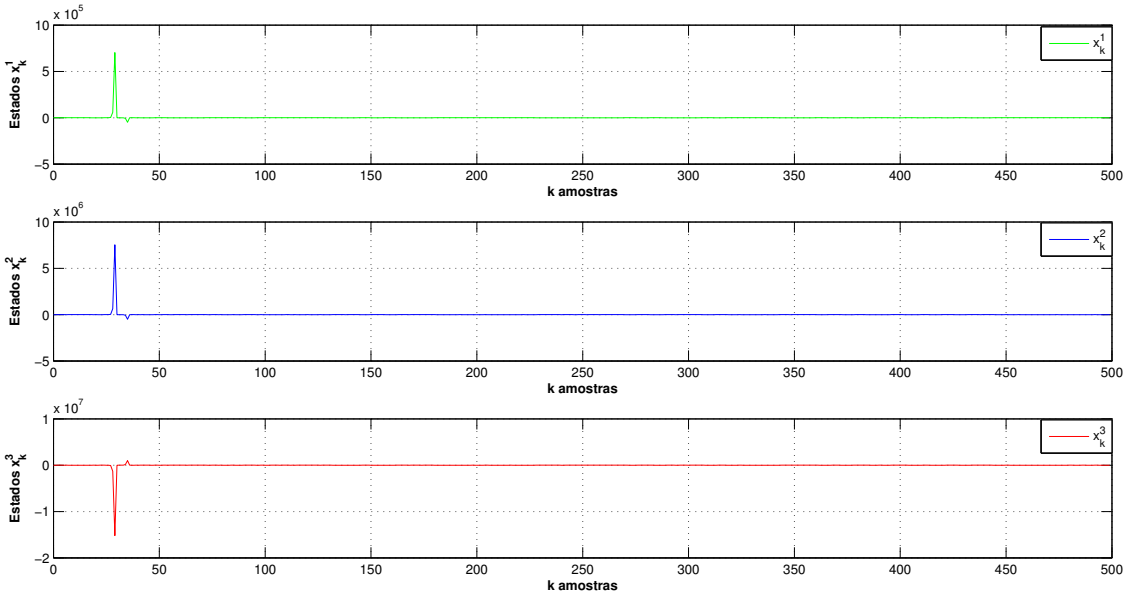

Figura 5.12: Gráfico dos Estados RLS-Projeção do Modelo de 3^a ordem

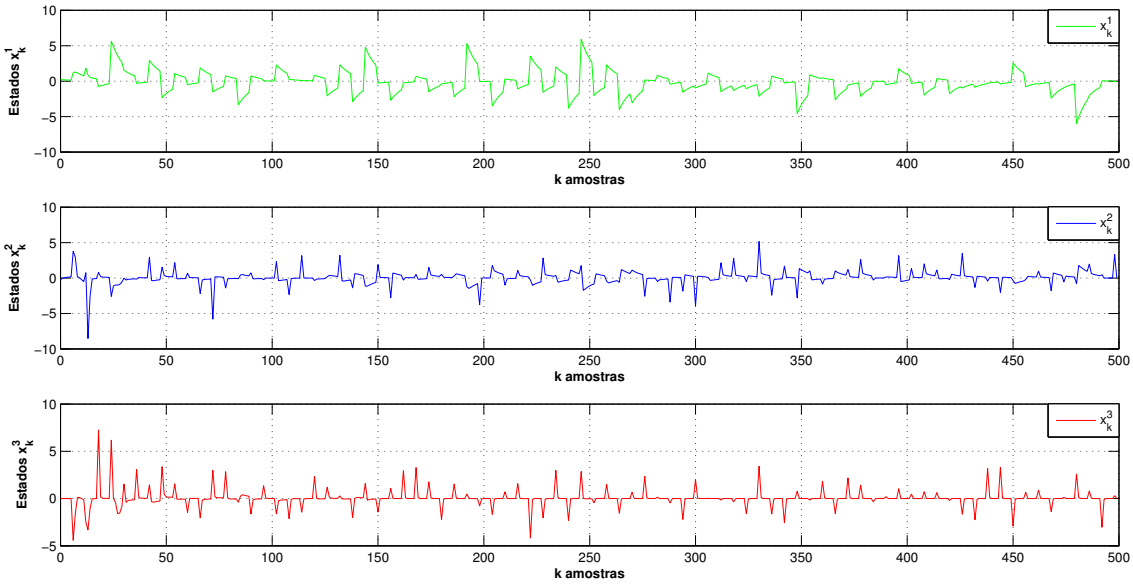

Figura 5.13: Gráfico dos Estados RLS-Kaczmarz do Modelo de 3^a ordem

O traço da matriz e os autovalores do sistema em malha fechada podem ser verificados a cada iteração de acordo com as Figuras 5.14, 5.15 e 5.16, respectivamente dos métodos RLS-Canônico, RLS-Projeção e RLS-Kaczmarz.

A medida que o processo de iteração ocorre, verifica-se que os autovalores apresentam menores variações de seus valores, isto se dá, à medida que o computo, pelos algoritmos, durante o processo iterativo aproxima-se da solução ótima da matriz P da equação Riccati.

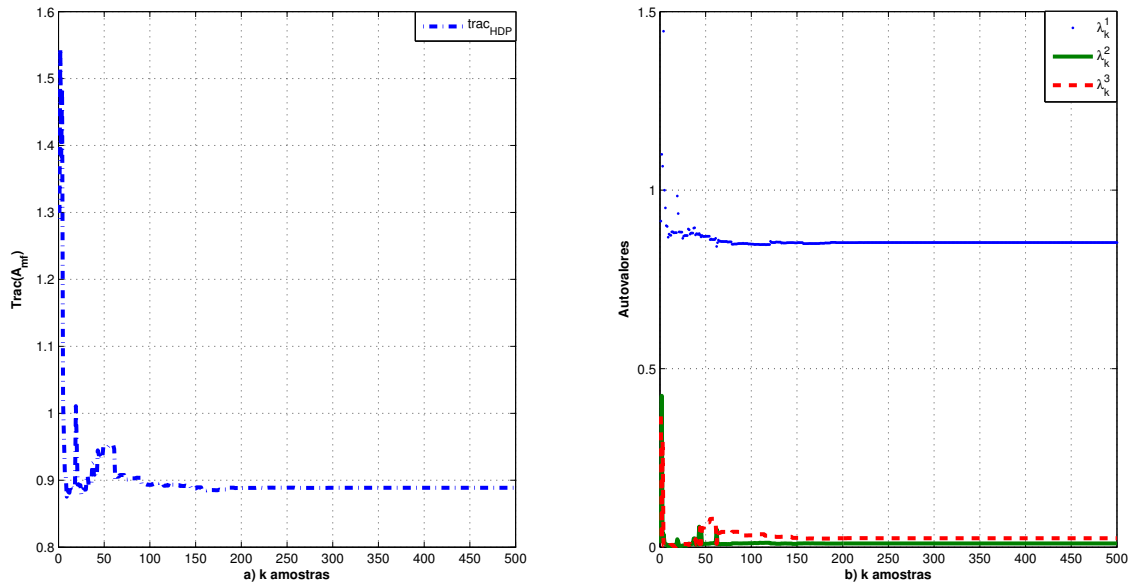

Figura 5.14: Gráfico do Traço e dos Autovalores RLS-Canônico do Modelo de 3^a ordem

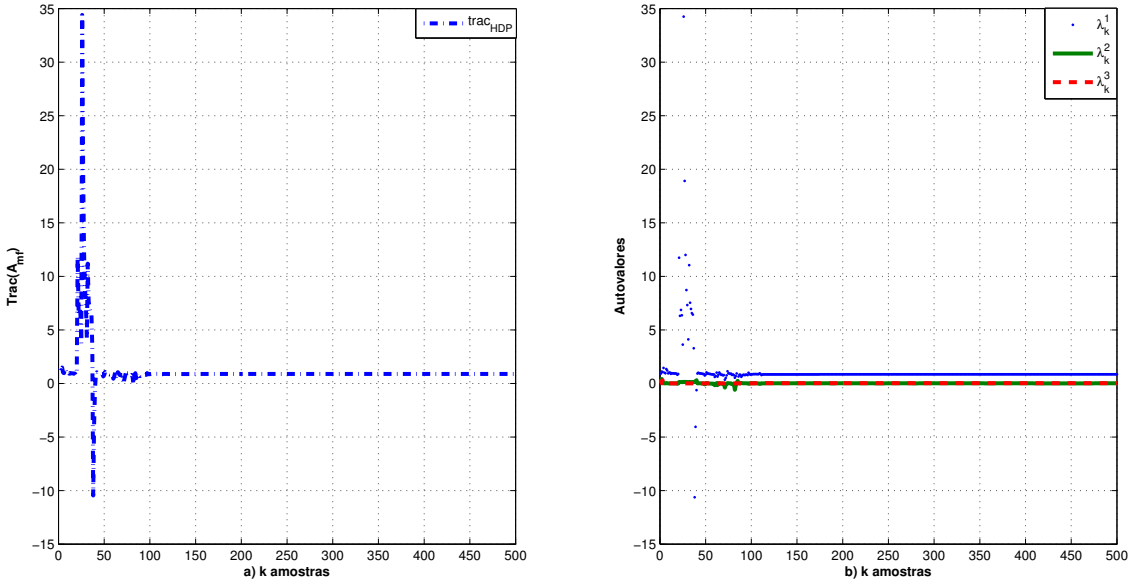

Figura 5.15: Gráfico do Traço e dos Autovalores RLS-Projeção do Modelo de 3^a ordem

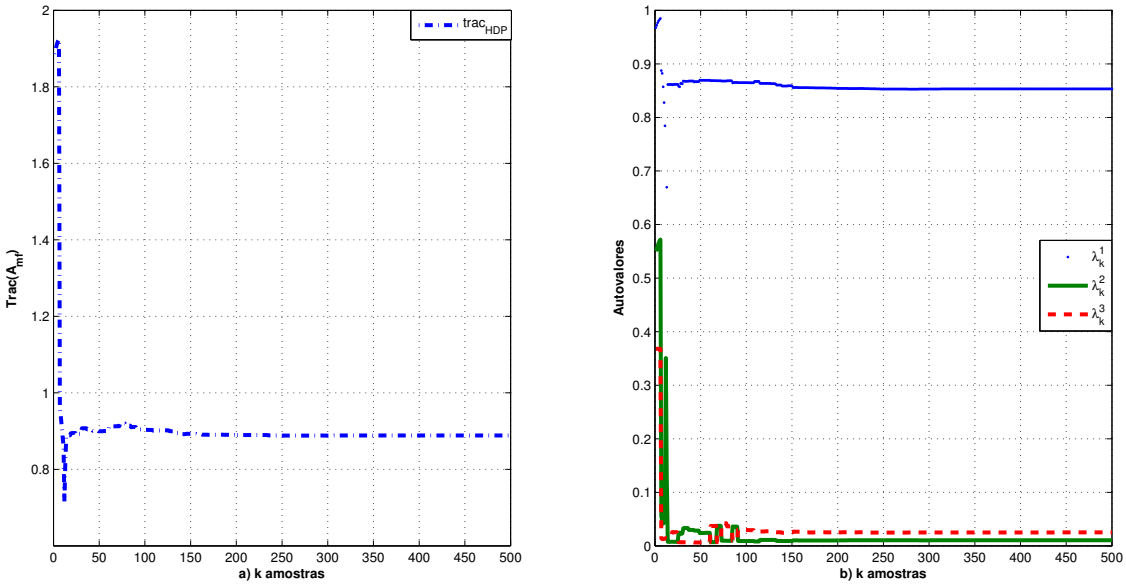

Figura 5.16: Gráfico do Traço e dos Autovalores RLS-Kaczmarz do Modelo de 3^a ordem

A partir do traço da matriz e do gráfico dos autovalores gerados pelo modelo de terceira ordem, da aeronave F-16, tem-se a Tabela 5.4 com os seguinte parâmetros estatísticos.

Tabela 5.1: Parâmetros - Estatísticos

Parâmetros	θ_1	θ_4	θ_6
Associação	p_{11}	p_{22}	p_{33}
θ^0	4,0154	1,0097	1,0098
$\widehat{\theta}$	4,0154	1,0097	1,0098
Mediana de θ	4,0158	1,0105	1,0094
Mínimo	-0,3696	-0,2830	-3,2151
Máximo	8,1293	6,6994	7,1145

5.2 Estudo de Caso II

Nesta seção são realizados procedimentos de testes e análises para avaliar a convergência e a precisão dos algoritmos *on-line*, como os realizados na Seção 5.1, neste estudo é utilizado um modelo de sistema de referência de quarta ordem investigado em (LEWIS; VRABIE, 2009a). O modelo proposto é apresentado na Figura 5.17, este apresenta quatro elementos armazenadores

de energia (dois indutores e dois capacitores), duas entradas e duas saídas conforme a Tabela 5.2.

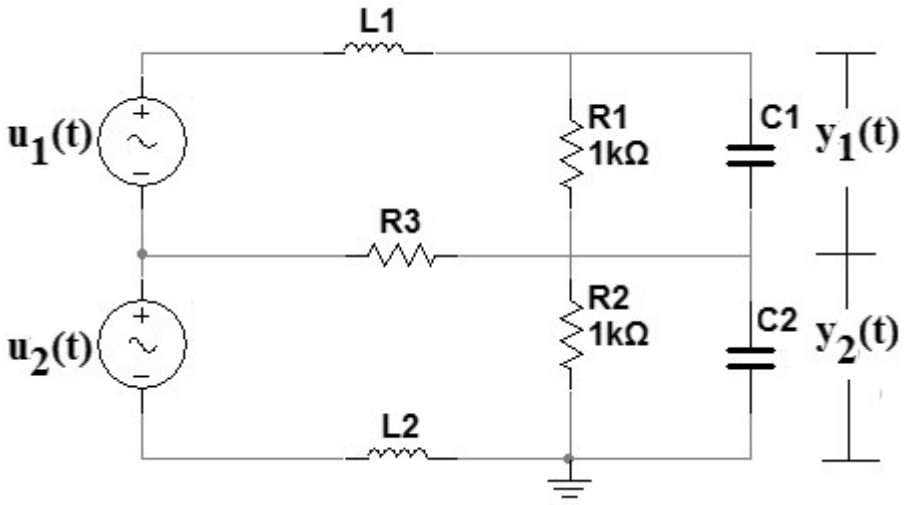

Figura 5.17: Circuito Elétrico de 4^a Ordem

Tabela 5.2: Entradas e Saídas do Modelo de 4^a Ordem

Entradas	$u_1(t)$	$u_2(t)$
Saídas	$y_1(t)$	$y_2(t)$

Para este estudo de caso, são admitidos os seguintes valores, conforme pode ser verificado na Tabela 5.3, para os componentes do circuito apresentado na Figura 5.17:

Tabela 5.3: Valores dos Componentes do Circuito de 4^a ordem

Componente	**Valor**	**Unidade**
$R1 - R2 - R3$	1	Ohm (Ω)
$C1 - C2$	1	Farad (F)
$L1 - L2$	1	Henry (H)

Para a modelagem matemática, é necessário que sejam aplicadas as Leis de Kirchhoff das correntes e das tensões, o teorema da superposição e por fim a transformada de Laplace para que se possa determinar as equação característica do modelo para que se possa extrair na estrutura do espaço de estado as matrizes A, B, C e D, respectivamente, matriz de estado, matriz de

entrada, matriz de saída e matriz de transmissão direta, que são utilizadas em nosso método de aproximação. Para maiores detalhes da modelagem, verificar (MACIEL, 2012).

As matrizes A, B, C e D para o modelo contínuo são dadas por

$$A = \begin{bmatrix} -1/(Cap*R1) & 0 & 1/Cap & 0 \\ 0 & -1/(Cap*R1) & 0 & -1/Cap \\ -1/Lind & 0 & -R2/Lind & -R2/Lind \\ 0 & 1/Lind & -R2/Lind & -R2/Lind \end{bmatrix}, \tag{5.13}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/Lind & 0 \\ 0 & 1/Lind \end{bmatrix}, \tag{5.14}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{5.15}$$

e

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.16}$$

5.2.1 Condições Iniciais

As condições iniciais são organizadas de acordo com sua funcionalidade no processo de busca da solução da equação algébrica de Riccati (DARE). Estas condições apresentadas, estão associadas com o projeto do sistema dinâmico (MIMO) e com o método iterativo de aproximação da solução da DARE. Constituem uma interface entre o sistema dinâmico e o método de aproximação da DARE: o ganho K_{HDP} que gera o novo vetor de estado, que por sua vez é estimado pelo RLS. A configuração dos parâmetros, é portanto, dada por:

1. As condições iniciais do vetor de estado, foram estimadas através de simulações do circuito elétrico da Figura 5.17 com os seguintes valores utilizados, $x_1 = 0,22; x_2 = -0,25; x_3 = -0,005$ *e* $x_4 = 0,005$.

2. Solução Inicial da DARE é $P_{HDP} = param_{HDP} I_{n\times n}$ que gera as condições iniciais da matriz de ganho, dada por $K_{HDP} = (R + B_d^T P_{HDP} B_d)^{-1} Bd^T P_{HDP} A_d$.

3. As matrizes de ponderação para o projeto do DLQR são definidas através de heurísticas, porém, seguem uma metodologia para a sua variação, estas são dadas pelas matrizes identidades com suas respectivas ordens $R = I_{2\times2}$ e $Q = I_{4\times4}$.

Condições iniciais para o RLS:

1. As condições iniciais para os parâmetros desconhecidos θ são $\theta_i = 0$, com i assumindo valores inteiros inteiros de 1 até 10.

2. O fator de esquecimento $\lambda = 0,96$, foi obtido através de experimentos, ou seja, tentativa e erro, até que fosse encontrado um valor que melhor contribuísse para o desempenho do algoritmo.

3. A matriz de covariância do processo RLS é dado por $P_{rls} = param_{rls} I_{6\times6}$, e o processo RLS é dado por $K_{rls} = param_{rls} I_{6\times6}$ são associados durante o processo de atualização.

5.2.2 Configuração do Processo Iterativo

A configuração do processo iterativo, consiste em estabelecer os melhores parâmetros para a solução de uma determinada aplicação, tendo como referência o empirismo do projetista ou as relações entre as variáveis relacionadas do sistema com os métodos. Os parâmetros que desempenham um papel relevante no método de convergência são: ordem do sistema n ($n = 4$), intervalo de amostragem T_{amost} ($T_{amost} = 0,1s$), fator de esquecimento (RLS) λ ($\lambda = 0,96$) e os parâmetros de projeção α_{rls} e μ_{rls}.

5.2.3 Análise de Desempenho dos Algoritmos RLS-HDP

Um modelo de circuito elétrico de quarta ordem dado pela Figura 5.17, é utilizado como sistema para avaliar o desempenho dos algoritmos RLS pelos métodos RLS-Canônico, RLS-Projeção e RLS-Kaczmarz, para o projeto de controle ótimo.

A solução de estado estacionário do RLS-HDP de *Riccati/Lyapunov*, baseada nas matrizes A, B, C e D, apresentadas na Seção 5.2, é dada por

$$P_{HDP}^{final} = \begin{bmatrix} 11,6401 & -7,1142 & 8,0334 & 0,7888 \\ -7,1142 & 11,6401 & -0,7888 & -8,0334 \\ 8,0334 & -0,7888 & 20,9536 & -15,5464 \\ 0,7888 & -8,0334 & 15,5464 & 20,9535 \end{bmatrix}. \tag{5.17}$$

A matriz de ganho RLS-HDP é dada por

$$K_{HDP}^{final} = \begin{bmatrix} 0,8033 & -0,0789 & 1,9954 & -1,5546 \\ 0,0789 & -0,8033 & -1,5546 & 1,9954 \end{bmatrix}. \tag{5.18}$$

A solução da equação HJB discreta pelo método de ***Schur*** e os ganhos ótimos associados, são utilizados como referência para comparação de precisão com os valores aproximados de P e K, respectivamente, supondo que os valores de *Schur* são verdadeiros. A solução discreta através do método de *Schur*, é dado por

$$P_{HDP_S}^{final} = \begin{bmatrix} 11,6401 & -7,1142 & 8,0334 & 0,7888 \\ -7,1142 & 11,6401 & -0,7888 & -8,0334 \\ 8,0334 & -0,7888 & 20,9536 & -15,5464 \\ 0,7888 & -8,0334 & 15,5464 & 20,9535 \end{bmatrix}. \tag{5.19}$$

A matriz de ganho discreta é dada por

$$K_{HDP_S}^{final} = \begin{bmatrix} 0,8033 & -0,0789 & 1,9954 & -1,5546 \\ 0,0789 & -0,8033 & -1,5546 & 1,9954 \end{bmatrix}. \tag{5.20}$$

Para os métodos de *Projeção* e *Kaczmarz*, tem-se os seguintes valores para a solução discreta e as matrizes de ganho, respectivamente de cada método:

Solução discreta obtida a partir do método de *Projeção*

$$P_{HDP_S}^{final} = \begin{bmatrix} 11,6401 & -7,1142 & 8,0334 & 0,7888 \\ -7,1142 & 11,6401 & -0,7888 & -8,0334 \\ 8,0334 & -0,7888 & 20,9536 & -15,5464 \\ 0,7888 & -8,0334 & 15,5464 & 20,9535 \end{bmatrix}. \tag{5.21}$$

A matriz de ganho discreta definida pelo método de *Projeção* é dada por

$$K_{HDP_S}^{final} = \begin{bmatrix} 0,8033 & -0,0789 & 1,9954 & -1,5546 \\ 0,0789 & -0,8033 & -1,5546 & 1,9954 \end{bmatrix}. \tag{5.22}$$

A solução discreta para o método *Kaczmarz*

$$P_{HDP_S}^{final} = \begin{bmatrix} 11,6401 & -7,1142 & 8,0334 & 0,7888 \\ -7,1142 & 11,6401 & -0,7888 & -8,0334 \\ 8,0334 & -0,7888 & 20,9536 & -15,5464 \\ 0,7888 & -8,0334 & 15,5464 & 20,9535 \end{bmatrix}. \tag{5.23}$$

A matriz de ganho discreta do método de *Kaczmarz*, é dada por

$$K_{HDP_S}^{final} = \begin{bmatrix} 0,8033 & -0,0789 & 1,9954 & -1,5546 \\ 0,0789 & -0,8033 & -1,5546 & 1,9954 \end{bmatrix} \tag{5.24}$$

Os valores de estado estáveis da matriz P^{θ} dada por (5.23) são comparados com a solução dada pela HJB (5.17) que é obtida através do método de *Schur*. O erro calculado entre cada elemento da matriz é inferior a 10^{-4}. A avaliação da precisão numérica mostrou que o vetor de parâmetro estimado $\widehat{\theta}$ converge para o valor verdadeiro do vetor θ^0. De acordo com as propriedades do RLS, esta ocorrência é uma indicação que o estimador é não polarizado.

A avaliação do processo iterativo para a solução da equação HJB-Riccati através do algoritmo RLS-HDP é apresentado nas Figuras 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.25 e 5.26 para um valor de 3000 ciclos de iteração. As curvas (a)-(d) da Figuras 5.18, 5.19, 5.20, representa o comportamento de convergência dos elementos p_{11}, p_{22}, p_{33} e p_{44} da matriz P, para os métodos RLS-Canônico, RLS-Projeção e RLS-Kaczmarz que são comparados com os valores obtidos para o método de Schur, correspondem respectivamente aos elementos θ_1, θ_5, θ_8 e θ_{10} do vetor de

parâmetros θ, ou seja a diagonal principal da matriz. As curvas (a)-(c) das Figuras 5.21, 5.22 e 5.23 apresentam a evolução do comportamento dos elementos p_{12}, p_{13}, e p_{14} da matriz P que estão associados aos elementos θ_2, θ_3 e θ_4 do vetor θ, de forma recíproca. Por fim as curvas (a)-(c) da Figuras 5.25, 5.25 e 5.26 descrevem a evolução do comportamento dos elementos p_{23}, p_{24}, e p_{34} da matriz P que estão associados aos elementos θ_6, θ_7 e θ_9 do vetor θ.

Na Figura 5.18 que apresenta o comportamento do método RLS-Canônico, tem-se a evolução dos parâmetros da diagonal principal e a sua convergência. O sistema, em sua dinâmica, responde de forma satisfatória e apresenta sua convergência por volta de 2100 amostras.

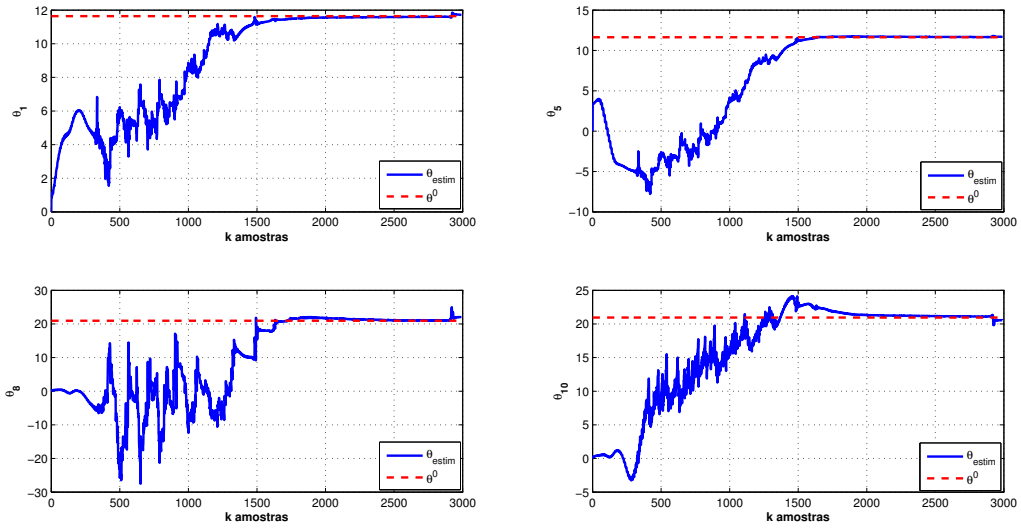

Figura 5.18: Convergência dos Parâmetros da Diagonal Principal pelo Método RLS-Canônico do Modelo de 4^a Ordem

Na Figura 5.19, tem-se a evolução dos parâmetros da diagonal principal pelo método de *RLS-Projeção* e a sua convergência, que foi atingida por volta de 2200 amostras.

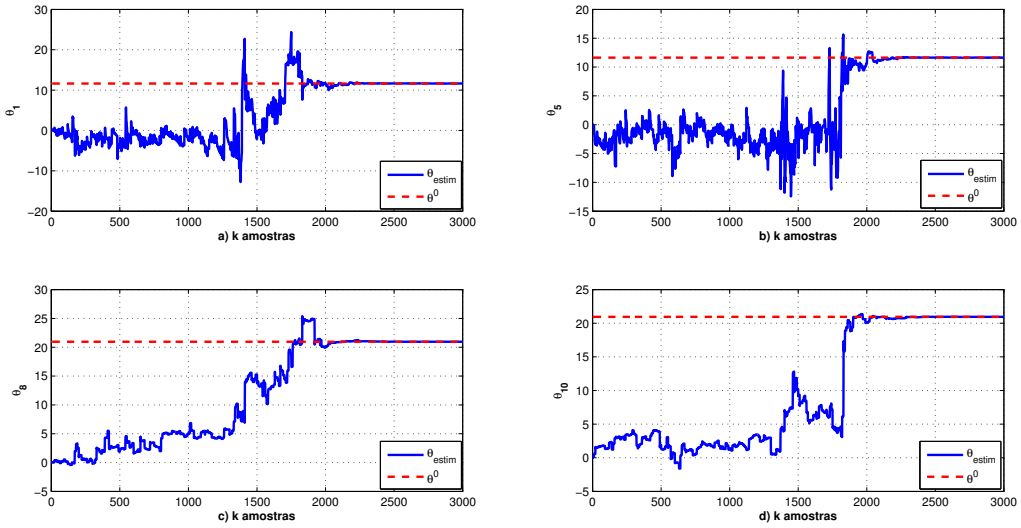

Figura 5.19: Convergência dos Parâmetros da Diagonal Principal pelo Método RLS-Projeção do Modelo de 4^a Ordem

Na Figura 5.20, tem-se a evolução do comportamento do método *RLS-Kaczmarz*, de sua diagonal principal, bem como a sua convergência que foi atingida por volta de 850 amostras, portanto, o método RLS-kaczmarz apresentou um melhor desempenho de convergência para os elementos da diagonal principal.

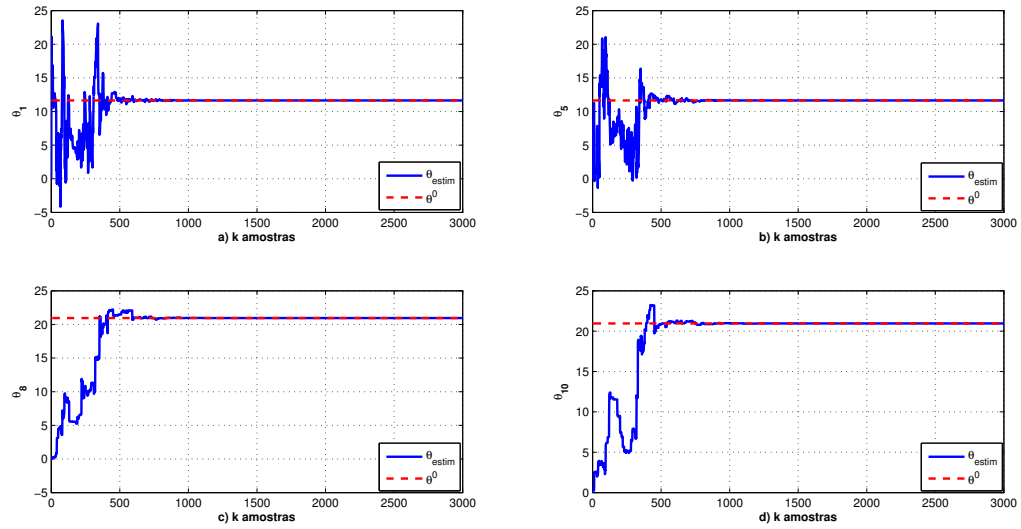

Figura 5.20: Convergência dos Parâmetros da Diagonal Principal pelo Método RLS-Kaczmarz do Modelo de 4^a Ordem

Nas Figura 5.21, 5.22 e 5.23, tem-se os resultados pelo método RLS-Canônico, RLS-Projeção e RLS-Kaczmarz referentes aos θ_2, θ_3 e θ_4, respectivamente. Na devida ordem, as convergências ocorreram por volta das 1700, 2250 e 750, para cada um dos métodos.

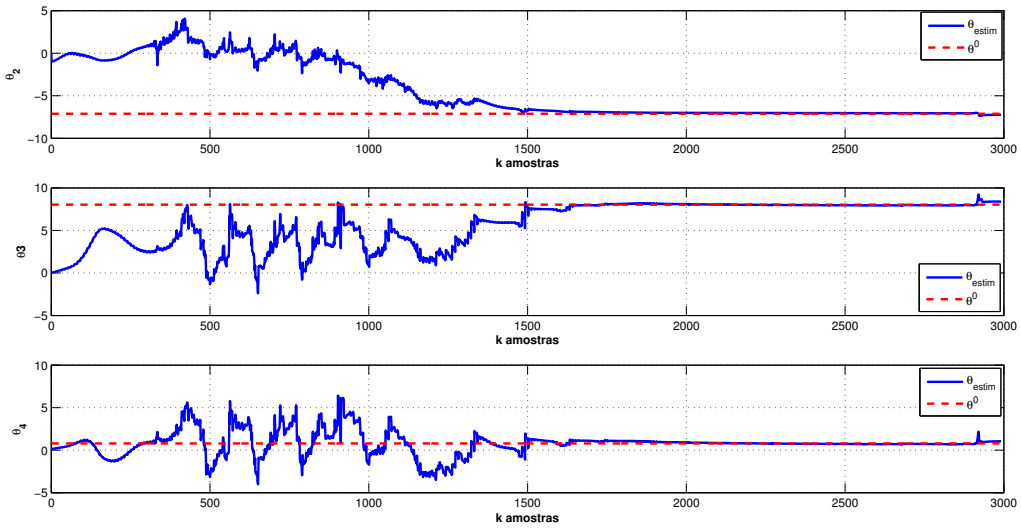

Figura 5.21: Convergência dos Parâmetros θ_2, θ_3 e θ_4 pelo Método RLS-Canônico do Modelo de 4^a Ordem

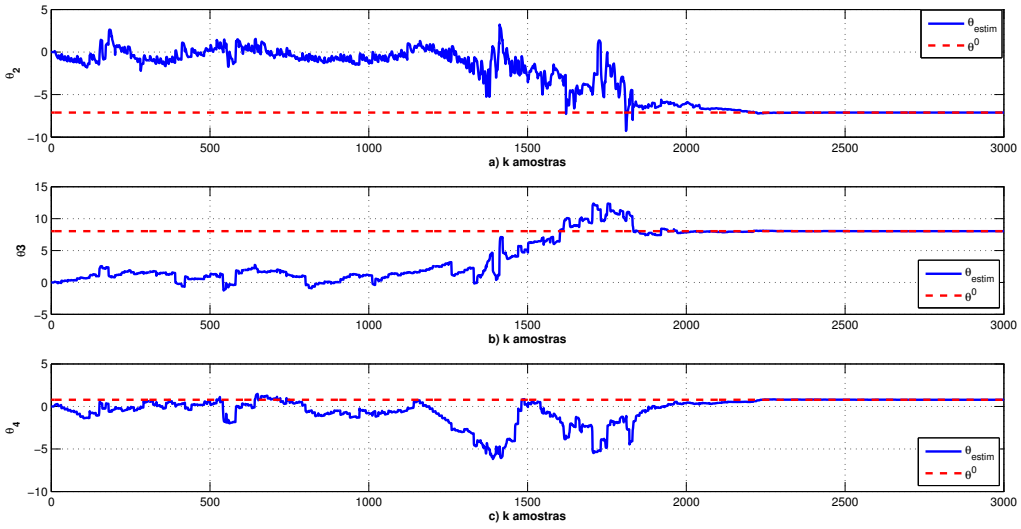

Figura 5.22: Convergência dos Parâmetros θ_2, θ_3 e θ_4 pelo Método RLS-Projeção do Modelo de 4^a Ordem

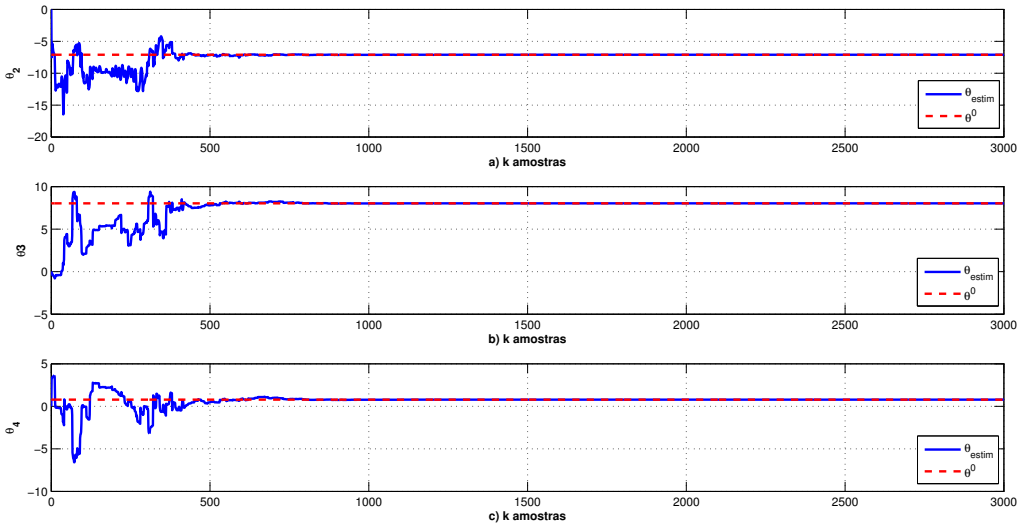

Figura 5.23: Convergência dos Parâmetros θ_2, θ_3 e θ_4 pelo Método RLS-Kaczmarz do Modelo de 4^a Ordem

Nas Figura 5.24, 5.25 e 5.26, tem-se respectivamente, os resultados pelo método RLS-Canônico, RLS-Projeção e RLS-Kaczmarz referentes aos θ_6, θ_7 e θ_9 que são comparados aos valores dados pela solução de Schur. Na devida ordem, tem-se que as convergências ocorreram por volta das 1600, 2250 e 750 amostras.

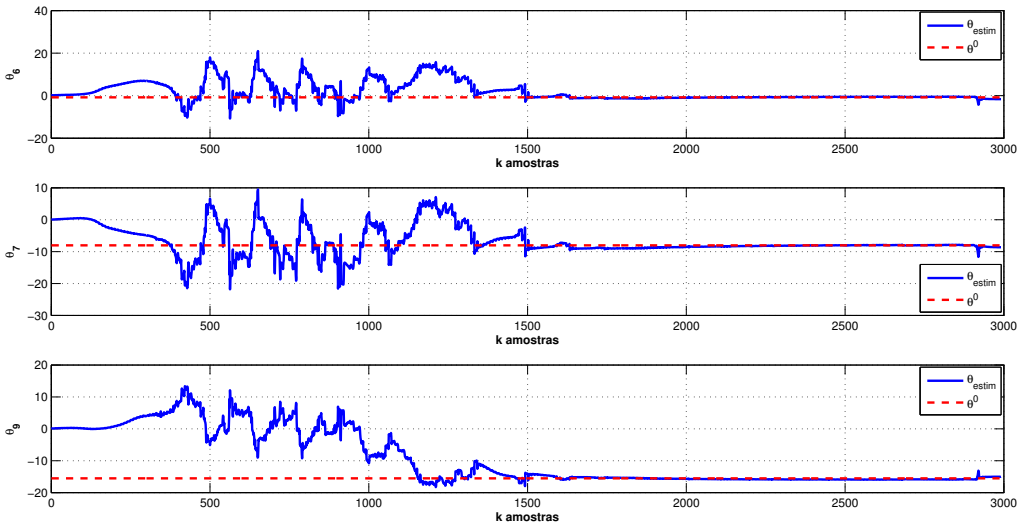

Figura 5.24: Convergência dos Parâmetros θ_6, θ_7 e θ_9 pelo Método RLS-Canônico do Modelo de 4^a Ordem

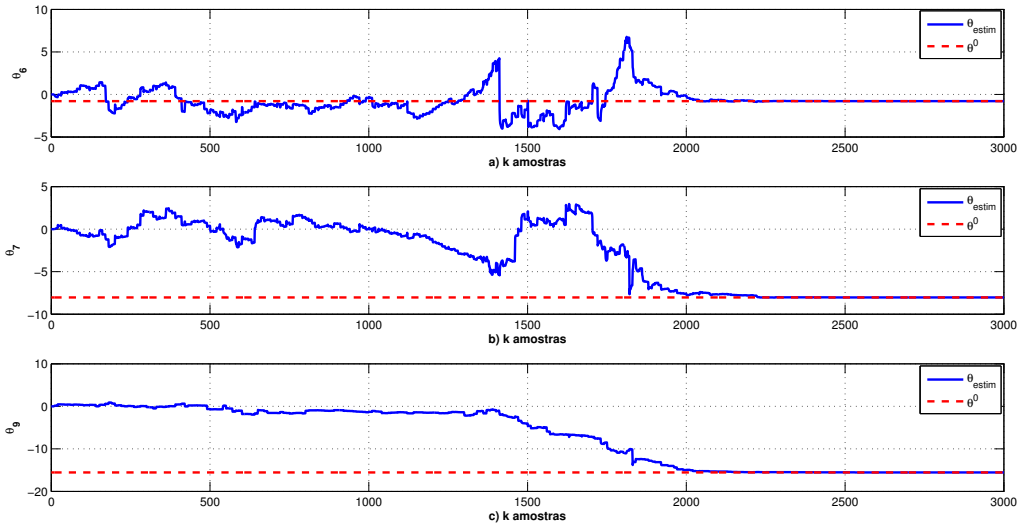

Figura 5.25: Convergência dos Parâmetros θ_6, θ_7 e θ_9 da Matriz P pelo Método RLS-Projeção do Modelo de 4^a Ordem

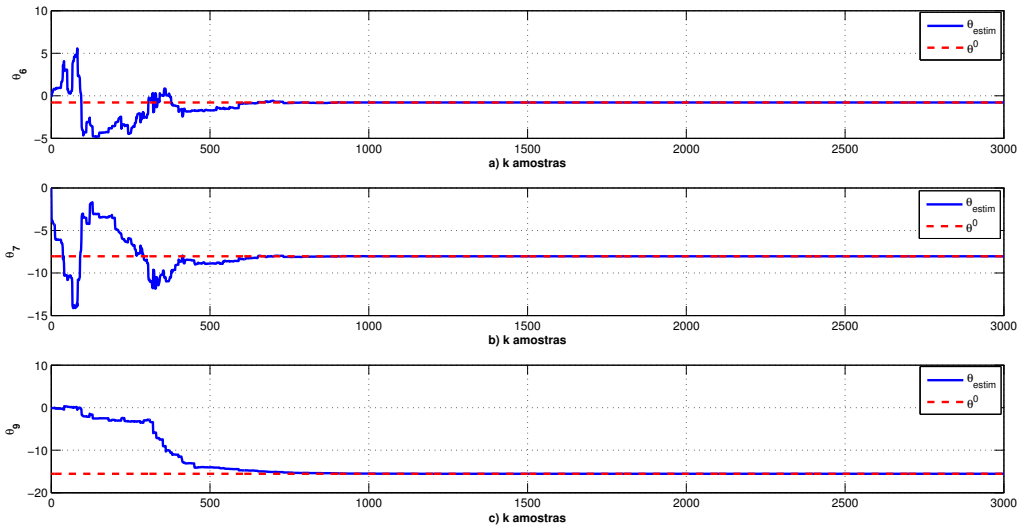

Figura 5.26: Convergência dos Parâmetros θ_6, θ_7 e θ_9 da Matriz P pelo Método RLS-Kaczmarz do Modelo de 4^a Ordem

As figuras de estados e as de controle do sistema dinâmico, à medida que o tempo decorre é verificado que as iterações do algoritmo ocorrem, os estados apresentam valores oscilatórios, o que é característico do processo de revitalização e da condição de excitação persistente, foi atribuído ao algoritmo uma variável radômica na revitalização x_{revit} de estados, esta variável assume valores $0 \leq x_{revit} \leq 1$.

As Figuras 5.27, 5.28 e 5.29 apresentam a evolução das ações de controle, respectivamente para o método de RLS-Canônico, RLS-Projeção e RLS-Kaczmarz, respectivamente, para o circuito da Figura 5.17 apresentado na Subseção 4.4.2.

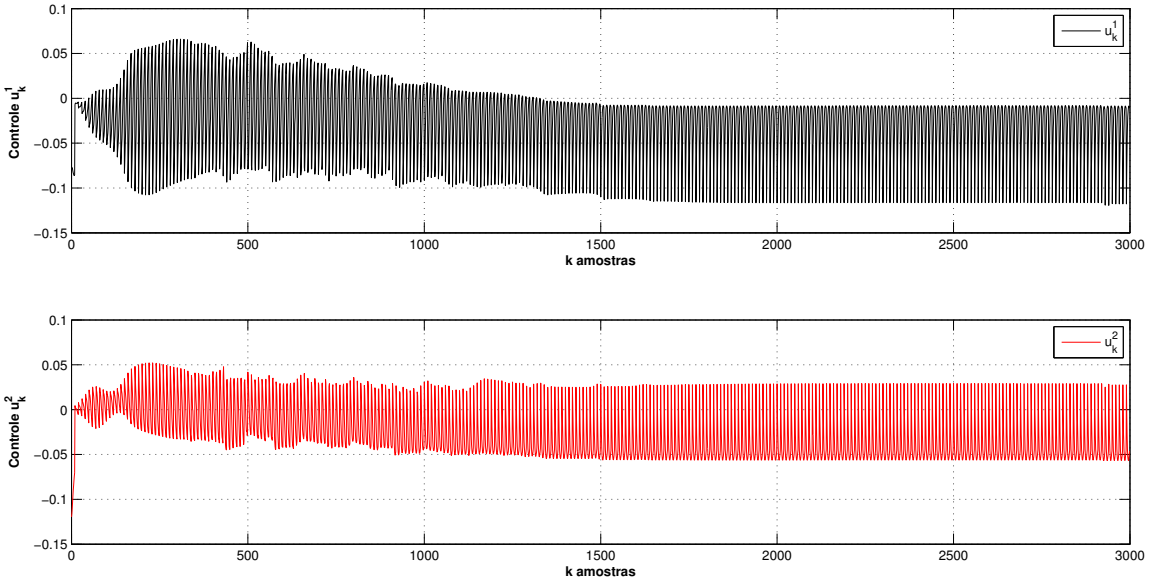

Figura 5.27: Gráfico da Ação de Controle u_k pelo Método RLS-Canônico do Modelo de 4^a Ordem

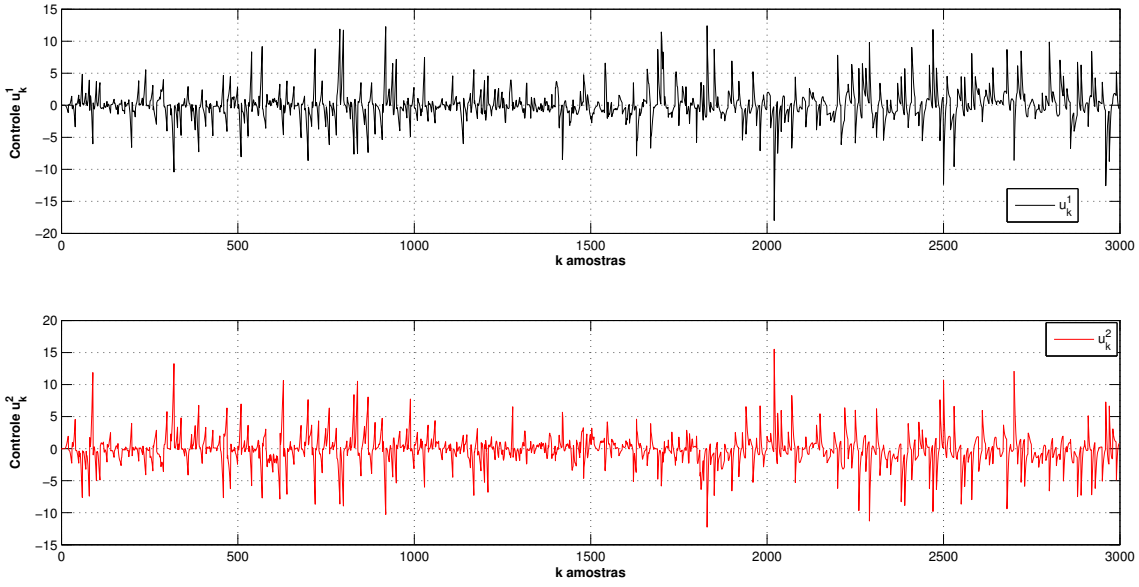

Figura 5.28: Gráfico da Ação de Controle pelo Método RLS-Projeção do Modelo de 4^a Ordem

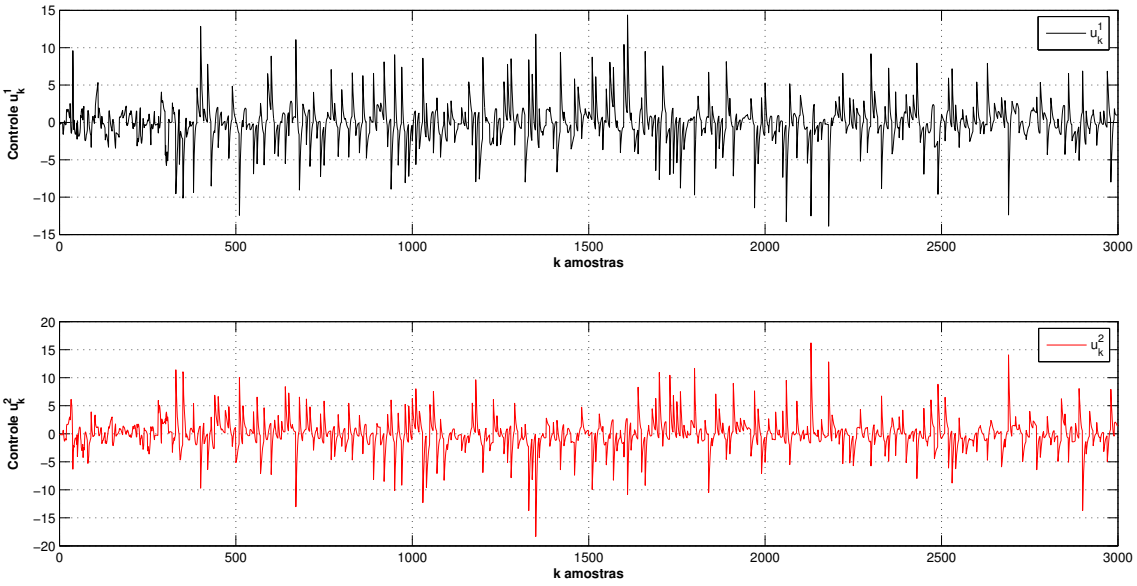

Figura 5.29: Gráfico da Ação de Controle pelo Método RLS-Kaczmarz do Modelo de 4^a Ordem

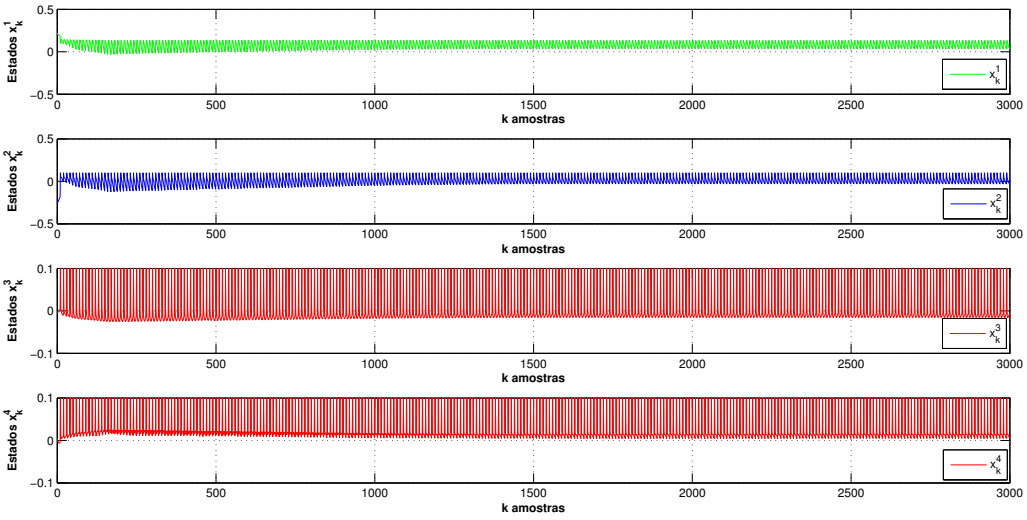

Figura 5.30: Gráfico dos Estados RLS-Canônico do Modelo de 4^a Ordem

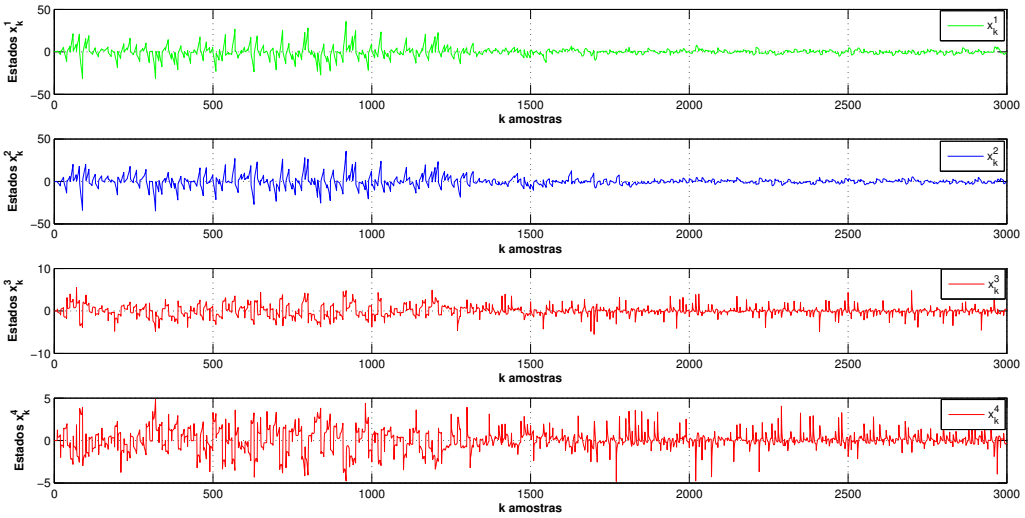

Figura 5.31: Gráfico dos Estados RLS-Projeção do Modelo de 4^a Ordem

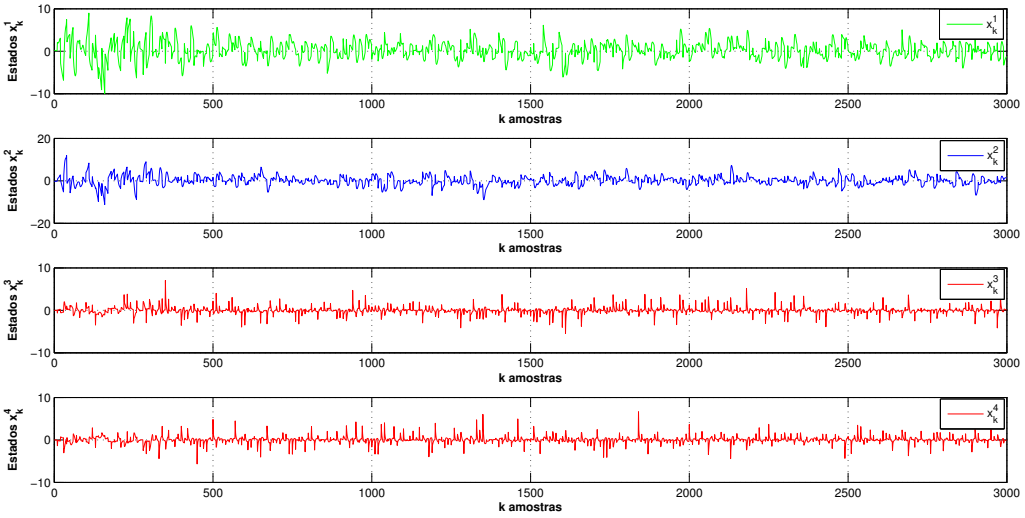

Figura 5.32: Gráfico dos Estados RLS-Kaczmarz do Modelo de 4^a Ordem

O traço da matriz e os autovalores do sistema em malha fechada podem ser verificados a cada iteração de acordo com as Figuras 5.33, 5.34 e 5.35, respectivamente dos métodos RLS-Canônico, RLS-Projeção e RLS-Kaczmarz.

A medida que o processo iterativo ocorre, é verificado que os autovalores apresentam menores variações de seus valores, isto se dá à medida em que os algoritmos se aproximam da solução ótima da matriz P da equação de Riccati.

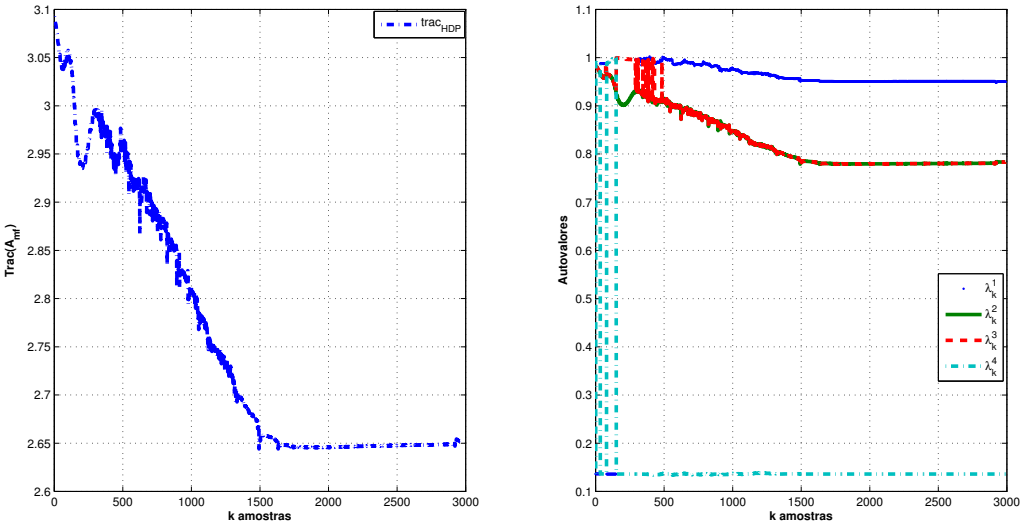

Figura 5.33: Gráfico do Traço e dos Autovalores RLS-Canônico do Modelo de 4^a Ordem

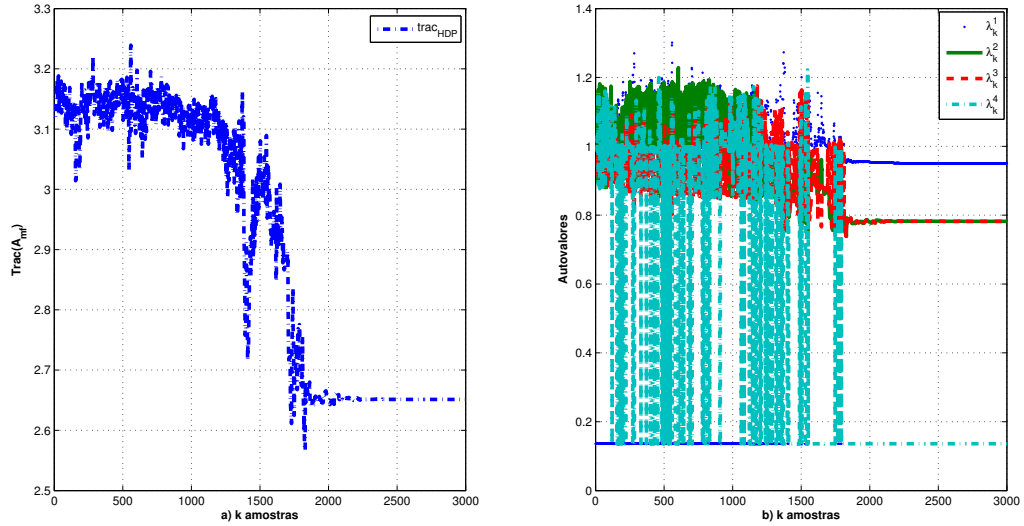

Figura 5.34: Gráfico do Traço e dos Autovalores RLS-Projeção do Modelo de 4^a Ordem

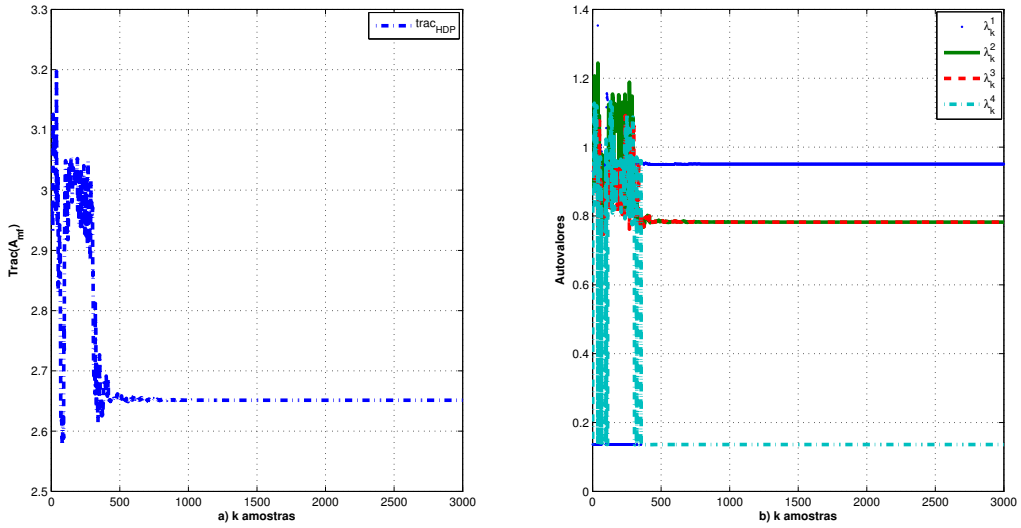

Figura 5.35: Gráfico do Traço e dos Autovalores RLS-Kaczmarz do Modelo de 4^a Ordem

A partir do traço da matriz e do gráfico dos autovalores gerados pelo circuito elétrico de quarta ordem, tem-se a Tabela 5.4 com os seguinte parâmetros estatísticos.

Tabela 5.4: Parâmetros - Estatísticos do Modelo de 4^a Ordem

Parâmetros	θ_1	θ_5	θ_8	θ_{10}
Associação	p_{11}	p_{22}	p_{33}	p_{44}
θ^0	11,64	11,64	20,95	20,95
$\widehat{\theta}$	11,64	11,64	20,95	20,95
Mediana de θ	11,23	11,24	17,66	21,11
Mínimo	-8,08	-4,91	-0,44	-0,37
Máximo	20,94	19,41	23,37	21,67

5.3 Estudo de Caso III

Nesta seção, procedimentos de testes e análises para avaliar a convergência e a precisão dos algoritmos *on-line*, são realizados. Um modelo de sistema de referência de oitava ordem investigado em (FIRMINO, 2008) será aqui utilizado.

Para um melhor entendimento deste estudo de caso, são apresentados os três tipos de movimentos realizados pelo helicóptero:

- ARFAGEM: ângulo que a aeronave faz com o eixo longitudinal x. É produzido no deslocamento do helicóptero para a frente, que faz a aeronave inclinar a frente para baixo. Sua orientação pode ser dada pela regra da mão direita: com os dedos esticados e o polegar aberto na direção do eixo lateral, o sentido positivo é o sentido que a mão fecha.

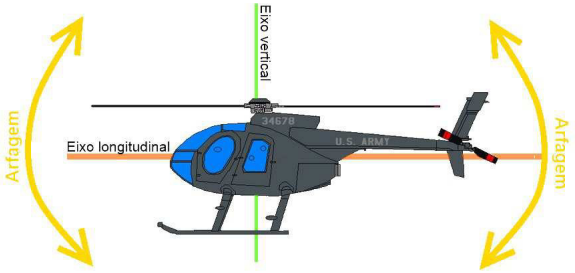

Figura 5.36: Movimento de Arfagem.

- ROLAGEM: ângulo que a aeronave faz com o eixo lateral y. É produzido no deslocamento do helicóptero para as laterais, que faz a aeronave inclinar-se para o lado. Sua orientação também pode ser dada pela mão direita: com o polegar na direção do eixo longitudinal, o sentido positivo é quando a mão fecha.

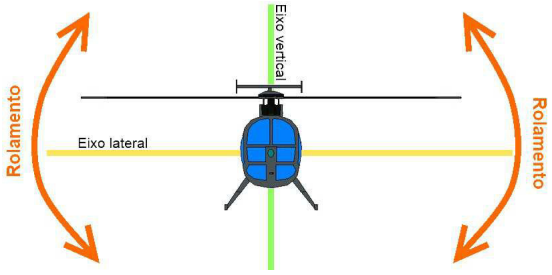

Figura 5.37: Movimento de Rolagem.

- GUINADA: movimento de rotação em torno do eixo z. O sentido positivo é dado quando o dedo polegar da mão direita está na direção do eixo vertical, pontando para baixo.

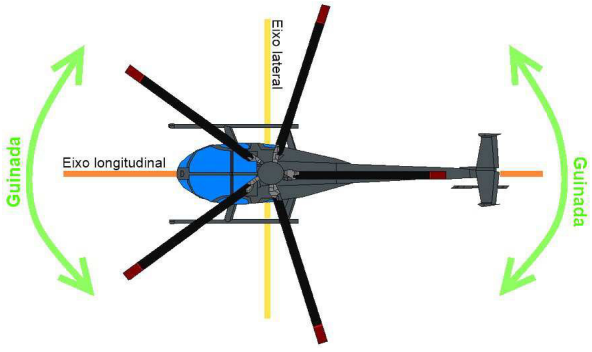

Figura 5.38: Movimento de Guinada.

A seguir é apresentada uma tabela que exibe relação entre os movimentos de *arfagem, rolagem e guinada*, com os seus respectivos eixos de rotação, bem como a sua simbologia.

Tabela 5.5: Simbologia dos Movimentos do Helicóptero

Ângulo/Variáveis	**Eixo**	**Símbolo**
Arfagem	x	ψ
Rolagem	y	ϕ
Guinada	z	γ

O modelo de helicóptero de (SAMBLANCAT, 1991) encontra-se representado sob a forma de espaço de estado, composto por sete modelos lineares e invariantes no tempo, correspondentes a sete pontos de operação diferentes. Em termos físicos, estes modelos correspondem ao helicóptero se deslocando em vôo longitudinal com altura constante nas velocidades de 0, 50, 100, 150, 200, 250 e 300 km/h.

Como os modelos são do tipo multivariável e possuem 8 estados, 3 entradas e 5 saídas. As saídas correspondem aos estados x_1, x_2, x_4, x_5 e x_8. A descrição dos estados $x(t) = [x_1 \quad x_2 \quad \cdots \quad x_8]$ estão dispostas na Tabela 5.6. As velocidades são dadas em metros por segundo e os ângulos em grados.

Tabela 5.6: Estados do Modelo de 8^a Ordem

Estado	**Símbolo**	**Descrição**	**Unidade**
x_1	v_x	velocidade longitudinal	m/s
x_2	v_z	velocidade vertical	m/s
x_3	q	velocidade angular de arfagem	$grad/s$
x_4	ψ	ângulo de arfagem	$grad$
x_5	v_y	velocidade lateral	m/s
x_6	P	velocidade angular de rolagem	$grad/s$
x_7	r	velocidade angular de guinada	$grad/s$
x_8	ϕ	ângulo de rolagem	$grad$

5.3.1 Condições Iniciais e Parâmetros HJB

As condições iniciais estão relacionadas com as variáveis de controle (o par ator-crítico) e os parâmetros da Equação HJB com os coeficientes do modelo matemático e os coeficientes da função de utilidade do processo de decisão markoviano. A seguir é apresentada uma descrição

destes parâmetros no contexto da solução da equação HJB para o modelo do sistema dinâmico (MIMO). No caso, as condições iniciais são dada por

a) **Par ator-crítico:**

1. Solução P_0 da Equação HJB, tal que $P > 0$ e do valor de ganho K_0.
2. Solução Inicial da DARE é $P_{HDP} = param_{HDP} I_{n\times n}$ que gera as condições iniciais da matriz de ganho, dada por $K_{HDP} = (R + B_d^T P_{HDP} B_d)^{-1} Bd^T P_{HDP} A_d$.

b) **Parâmetros:**

1. Coeficientes do modelo matemático no espaço de estados ($x_{k+1} = Ax_x + Bu_k$).
2. Coeficientes da função de utilidade do processo.
   (a) As matrizes de ponderação para o projeto do DLQR, são dadas por $R = I_{3\times 3}$ e $Q = I_{8\times 8}$.

Os parâmetros para a avaliação do método proposto são: ordem do sistema n ($n = 8$) e intervalo de amostragem T_{amost} ($T_{amost} = 0,1s$).

5.3.2 Solução HJB via Método de Schur

Dentre os pontos de operação, que se referem a cada uma das velocidades da aeronave investigados por (FIRMINO, 2008), o modelo três, que foi idealizado para a velocidade de $100km$, apresentou um melhor desempenho. Devido a este melhor comportamento, foram utilizadas as matrizes A, B e C, observadas neste ponto de operação. As matrizes do modelo matemático são dados por

$$A = \begin{bmatrix} -0,0218 & -0,01470 & 0,61800 & -9,81000 & 0,000566 & 0,23400 & 0,01300 & 0 \\ -0,12700 & -0,73500 & 27,9000 & 0,04100 & 0,01100 & 0,33900 & -0,44400 & -0,131 \\ 0,00523 & -0,05030 & -1,30000 & 0 & -0,02330 & -0,265 & 0,07480 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -0,01270 & 0 \\ -0,00413 & -0,00362 & 0,23100 & 0,00052 & -0,13100 & -0,700 & -27,4000 & 9,810 \\ 0,02890 & -0,17400 & 1,07000 & 0 & -0,16500 & -5,590 & 0,1440 & 0 \\ 0,01530 & -0,02300 & 0,07570 & 0 & 0,02980 & -0,586 & -0,7070 & 0 \\ 0 & 0 & -0,00006 & 0 & 0 & 1 & -0,00471 & 0 \end{bmatrix}, \quad (5.25)$$

$$B = \begin{bmatrix} 10,200 & -1,350 & 0,241 \\ 18,500 & -0,198 & -0,544 \\ -16,400 & 2,570 & -0,914 \\ 0 & 0 & 0 \\ -1,140 & -10,100 & -4,460 \\ -10,700 & -83,400 & -3,480 \\ -1,130 & -9,890 & 7,040 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.26}$$

e

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.27}$$

A solução HJB-Riccati via método de Schur é utilizada como referência para avaliar o desempenho dos algoritmos que são baseados nas recorrência de *Riccati* e *Lyapunov*. A solução da Equação HJB para o sistema de oitava ordem obtida pelo método de *Schur*, baseado nas matrizes (5.25), (5.26) e (5.27), é dado por

$$P_{Schur} = \begin{bmatrix} 0,0153 & 0,0036 & 0,0129 & -0,1009 & -0,0001 & -0,0000 & 0,0014 & -0,0009 \\ 0,0036 & 0,0053 & 0,0070 & -0,1143 & -0,0000 & 0,0000 & 0,0011 & -0,0004 \\ 0,0129 & 0,0070 & 0,0167 & -0,1894 & -0,0001 & 0,0000 & 0,0021 & -0,0010 \\ -0,1009 & -0,1143 & -0,1894 & 3,2773 & 0,0007 & -0,0005 & -0,0298 & 0,0100 \\ -0,0001 & -0,0000 & -0,0001 & 0,0007 & 0,0011 & -0,0000 & 0,0000 & 0,0001 \\ -0,0000 & 0,0000 & 0,0000 & -0,0005 & -0,0000 & 0,0010 & -0,0000 & -0,0000 \\ 0,0014 & 0,0011 & 0,0021 & -0,0298 & 0,0000 & -0,0000 & 0,0013 & -0,0001 \\ -0,0009 & -0,0004 & -0,0010 & 0,0100 & 0,0001 & -0,0000 & -0,0001 & 0,0085 \end{bmatrix} \times 10^3. \tag{5.28}$$

O ator pelo método de *Schur* é dada por

$$K_{Schur} = \begin{bmatrix} 0,0238 & -0,0107 & -0,0569 & -0,3738 & 0,0036 & -0,0017 & -0,0076 & -0,0010 \\ -0,0027 & 0,0034 & 0,0097 & 0,0124 & 0,0010 & -0,0114 & -0,0034 & -0,0184 \\ 0,0015 & 0,0009 & 0,0024 & -0,0366 & -0,0389 & -0,0087 & 0,1132 & -0,0525 \end{bmatrix}. \tag{5.29}$$

5.3.3 Solução HJB via Recorrência de Riccati com PI

As Figuras 5.39, 5.40 e 5.41, apresentam a evolução do comportamento dos coeficientes da diagonal principal da matriz P, para um número de 100 amostras de iteração, da solução HJB via Recorrência de Riccati com iteração de política (PI).

Na Figura 5.39 tem-se os elementos P_{11}, P_{22}, P_{33} e P_{88} que estão associados aos elementos θ_1, θ_9, θ_{16} e θ_{36} do vetor de parâmetros θ^0. Verificamos que o processo iterativo atinge convergência com aproximadamente 90 amostras de iterações e que a dinâmica do transitório é bastante suave.

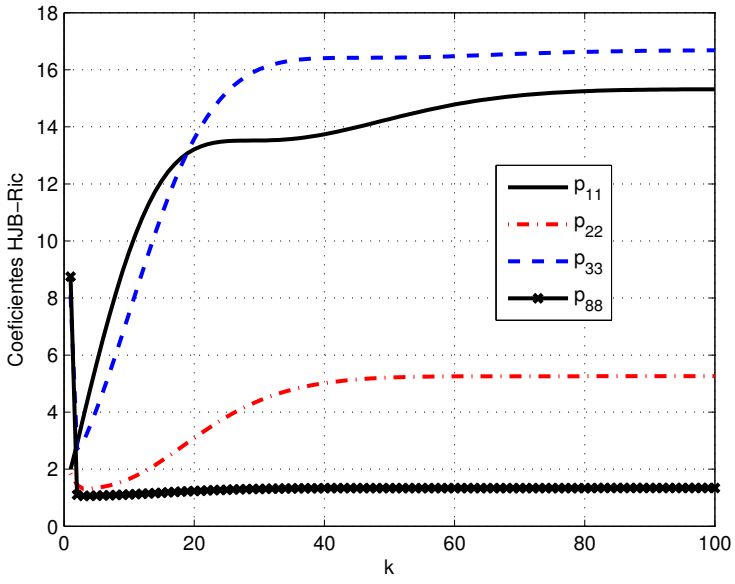

Figura 5.39: Convergência dos Parâmetros θ_1, θ_9, θ_{16} e θ_{36} da Solução Dependente de Modelo HJB-Riccati via PI e Recorrência de Riccati.

Na Figura 5.40 apresenta-se o comportamento dos coeficientes P_{44}, P_{55} e P_{77}, que correspondem a θ_{22}, θ_{27} e θ_{34} da solução da equação matricial de Riccati, observamos que a convergência ocorre com 50 amostras de iteração.

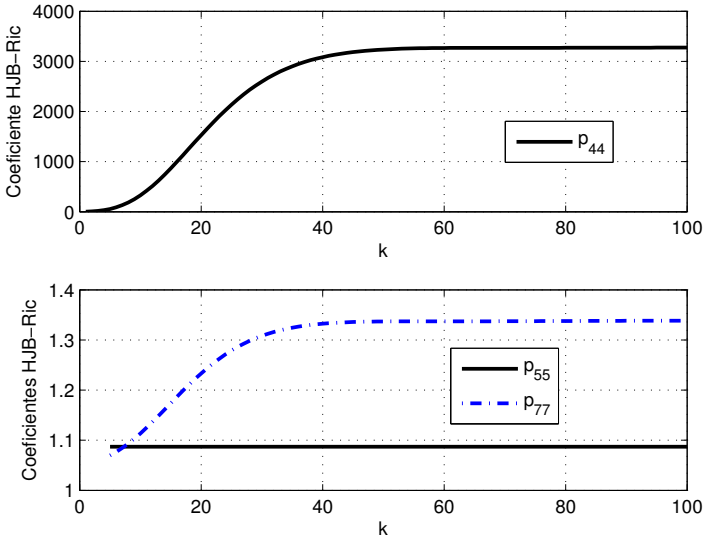

Figura 5.40: Convergência dos Parâmentros θ_{22}, θ_{27} e θ_{34} da Solução Dependente de Modelo HJB-Riccati via PI e Recorrência de Riccati.

Na Figura 5.41 apresenta-se o comportamento dos coeficientes P_{31} que está associado ao elemento θ_{31} do vetor de parâmetros θ^0 da solução da equação matricial de Riccati, apresentando convergência por volta de 70 amostras de iteração.

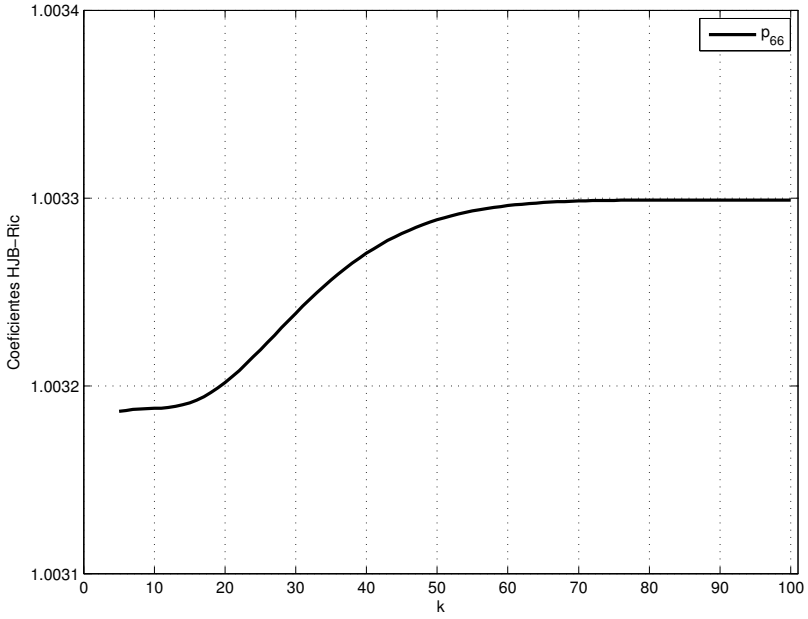

Figura 5.41: Convergência do Parâmetros θ_{31} da Solução Dependente de Modelo HJB-Riccati via PI e Recorrência de Riccati.

5.3.4 Solução HJB via Recorrência de Lypunov com GPI

Na Figura 5.42 apresenta-se o comportamento dos coeficientes da solução da equação matricial de Riccati.As Figuras 5.42, 5.43 e 5.44, apresentam a evolução do comportamento dos coeficientes da diagonal principal da matriz P, para um número de 100 amostras de iteração, da solução HJB via Recorrência de Riccati com iteração de política (GPI).

Na Figura 5.42 tem-se os elementos P_{11}, P_{22}, P_{33} e P_{88} que estão associados aos elementos θ_1, θ_9, θ_{16} e θ_{36} do vetor de parâmetros θ^0. Verificamos que o processo iterativo atinge convergência com 30 amostras de iterações e que a dinâmica do transitório é bastante suave

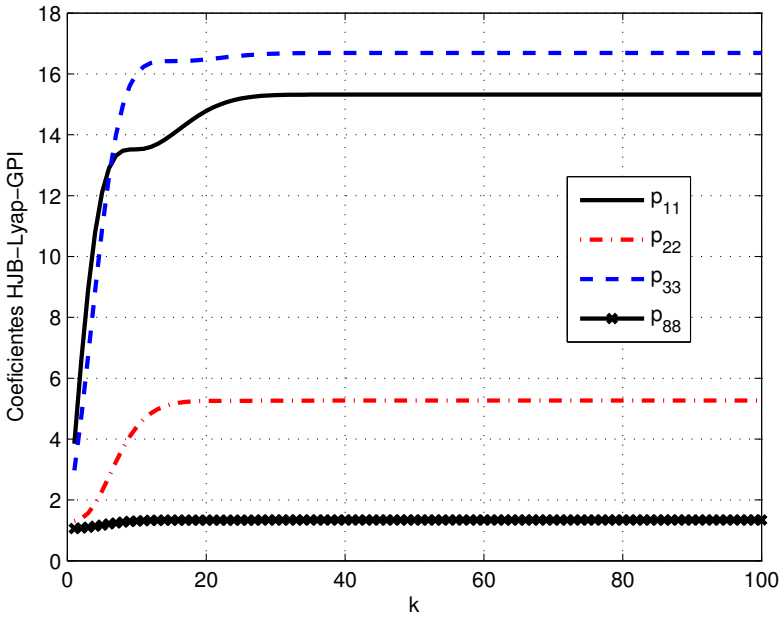

Figura 5.42: Convergência dos Parâmetros θ_1, θ_9, θ_{16} e θ_{36} da Solução Dependente de Modelo HJB-Riccati via GPI e Recorrência de Lyapunov.

Na Figura 5.43 apresenta-se o comportamento dos coeficientes P_{44}, P_{55} e P_{77}, que correspondem a θ_{22}, θ_{27} e θ_{34} da solução da equação matricial de Riccati via GPI, onde é observada que a convergência ocorre em torno de 20 amostras de iteração.

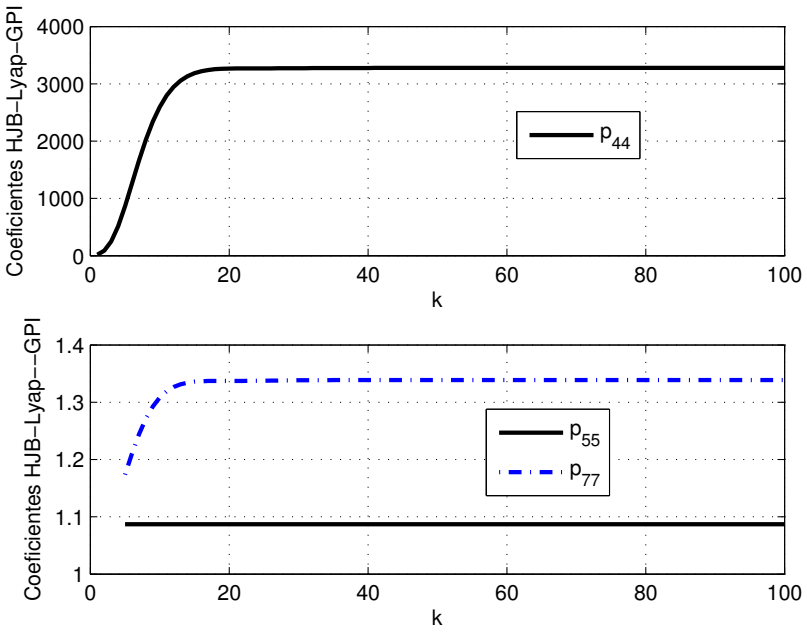

Figura 5.43: Convergência dos Parâmetros θ_{22}, θ_{27} e θ_{34} da Solução Dependente de Modelo HJB-Riccati via GPI e Recorrência de Lyapunov.

Na Figura 5.44 apresenta-se o comportamento do coeficiente P_{31} que está associado ao elemento θ_{31} do vetor de parâmetros θ^0 da solução da equação matricial de Riccati, apresentando convergência por volta de 25 amostras de iteração.

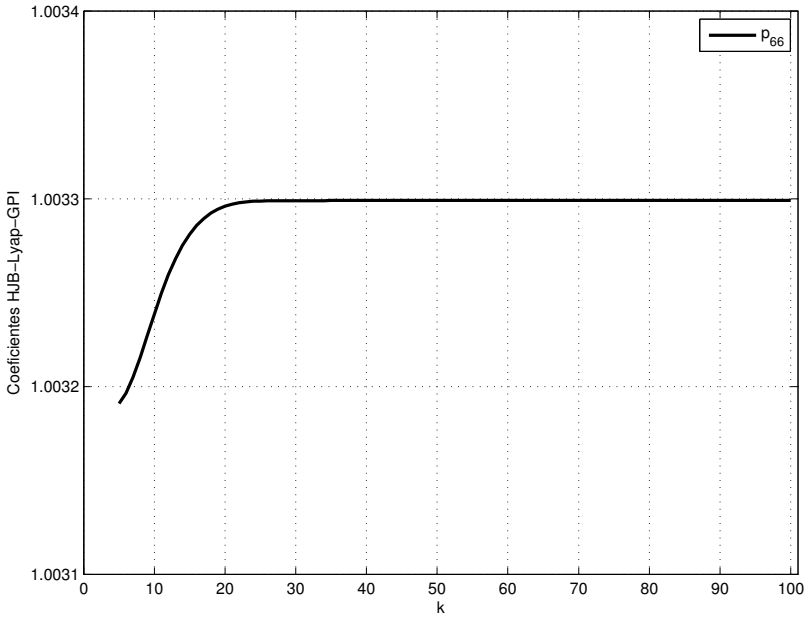

Figura 5.44: Convergência do Parâmetro θ_{31} da Solução Dependente de Modelo HJB-Riccati via GPI e Recorrência de Lyapunov.

Mediante as observações realizadas sobre o comportamento da convergência dos parâmetros θ^0 da matriz P, concluímos que a Solução HJB via Recorrência de Lyapunov com Iteração de Política Gulosa(GPI), apresenta uma convergência mais rápida que a obtida pela Solução HJB via Recorrência de Riccati com Iteração de Política(PI).

5.3.5 Análise dos Algoritmos Dependentes de Modelo

A análise dos algoritmos dependentes de Modelos está associada aos modos de operação dos algoritmos e com os resultados apresentados nas Figuras 5.39 - 5.41 e nas Figuras 5.42 - 5.44. Estes gráficos estão relacionados com as recorrências de Riccati e Lyapunov via iteração de politica generalizada como 2 passos. Conclui-se que a velocidade para que se obtenha a solução por meio dos referidos algoritmos está relacionada com os pólos dominantes do sistema dinâmico ou a quanto de tempo tem-se disponível, antes de tomar uma dada decisão, para realizar o ajuste em função das variações paramétricas no modelo matemático.

5.4 Comparações e Comentários da Análise dos Algoritmos Dependentes e Livres de Modelo

As comparações, comentários e trade-offs dos métodos para solução da Equação HJB-Riccati, por meio de algoritmos dependentes de modelos e aproximações, são direcionadas para o projeto *on-line* de sistemas de controle ótimo. Associados com as almejadas implementações destas abordagens, os seguintes itens são observados durante os experimentos computacionais:

1. Quantidade operações para o cálculo do par ator-crítico;
2. Associar a constante de tempo do sistema com o tempo de cálculo do par ator-crítico junto com o tempo de comunicação entre as partes do sistema de controle e a planta de malha fechada (sensor, controle, atuador e planta).
3. A solução aproximada HJB é sensível a adaptação no sentido de regulação. Enquanto que a solução dependente de modelo garante uma boa aproximação para o problema de controle ótimo, mesmo que o modelo matemático apresente alguma perturbação do tipo ruído branco. Desta forma o modelo matemático é dado por

$$x_{k+1} = Ax_k + Bu_k + \Gamma w_k,$$

sendo w_k e Γ sinal ruído branco e a ponderação, respectivamente.

Capítulo 6

Conclusões e Perspectivas

Nesta dissertação foi apresentado o desenvolvimento de algoritmos de programação dinâmica aproximada (ADP), independentes de modelo, baseados na família de estimadores de Mínimos Quadrados Recursivos (Canônico - Projeção - Kaczmarz), algoritmos dependentes de modelo baseados nas recorrências de *Riccati* e *Lyapunov*, e um método de análise de convergência montado para avaliar o desempenho dos algoritmos de estimação RLS para aproximar as políticas ótimas com base nas seguintes etapas:

- Ajuste dos ganhos do controle *on-line*;
- Análise de convergência do estimador;
- Os impactos sobre o desempenho do controle do sistema dinâmico.

A solução da equação HJB foi avaliada para o RLS via política de iteração (PI) dependente de estado e abordagens de programação dinâmica heurística aproximada.

O desenvolvimento de algoritmos HDP de estado é apresentado em uma formulação que combina a aproximação da função valor por meio de estrutura RLS (mínimos quadrados recursivos) e política de iteração (PI) para aplicações de projeto ótimo *on-line*. A Programação Dinâmica Adaptativa demonstra ser uma alternativa viável para realização de projeto de sistema de controle sem parametrizações do meio ambiente e com a política de decisão. Os testes de avaliação mostraram que a metodologia RLS apresenta um desempenho satisfatório para estimar a política ótima no esquema Ator-Crítico de aprendizado por reforço.

Os algoritmos dependentes de modelo, aplicados no controle ótimo sobre o estudo de *caso III*, apresentaram um comportamento muito suave em sua dinâmica, tanto para a recorrência de

Riccati, quanto para a recorrência de *Lyapunov*. No entanto, é observado que o método baseado em *Lyapunov* atingiu a convergência com aproximadamente 25 amostras de iteração, portanto o seu desempenho, quanto a velocidade, é superior aos resultados obtidos pela recorrência de *Lyapunov*, que é atingida por volta de 80 amostras de iteração.

Perspectivas

Apesar desta dissertação apresentar contribuições e estar bem avaliada, conforme pode ser verificado nas Subseções (5.1.3), (5.2.3) e (5.3.5), abre-se espaço para a realização de trabalhos futuros, conforme segue:

- Realizar melhorias e ou adaptações nos algoritmos apresentados, de maneira que possam ser aplicados a sistemas de ordens superiores aos aqui apresentados;
- Desenvolver e implementar em hardware o sistema proposto.

Publicações

Santos, Watson R. M., Queiroz, Jonathan A.,Neto, João V. da Fonseca, Rêgo, Patrícia H. M., Santana, Ewaldo e Andrade, Gustavo. *RLS Algorithms and Convergence Analysis Method for on-line DLQR Control Design via Heuristic Dynamic Programming*, UKSim-AMSS 16th International Conferece on Moddeling and Simulation,2014.

Apêndice A

Álgebra Linear

A.1 Lema de Inversão de Matrizes

Lema 1 (Lema de Inversão de Matrizes): Sejam A, B, C e D matrizes quadradas não singulares, de tal forma que A, C e $(A + BCD)$ sejam inversíveis, então

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \tag{A.1}$$

Prova Multiplicando os dois lados da Equação(A.1) por $(A + BCD)$, temos

$$\begin{aligned}(A + BCD)(A + BCD)^{-1} &= (A + BCD)A^{-1} - \\ &A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ I = I + BCDA^{-1} &- (I + BCDA^{-1})B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ I = I + BCDA^{-1} &- (B + BCDA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}DA^{-1}.\end{aligned}$$

Colocando-se BC em evidência, temos

$$\begin{aligned}I = I + BCDA^{-1} - BC(C^{-1} + DA^{-1}B)&(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ I = I + BCDA^{-1} - +BCDA^{-1},&\end{aligned}$$

portanto,

$$I = I.$$

Apêndice B

Programação Dinâmica

B.1 Princípio da Otimalidade e Programação Dinâmica

A Programação Dinâmica (DP), conhecida também como otimização recursiva, é uma técnica que trata de situações nas quais as decisões são tomadas em etapas, com o resultado de cada decisão sendo previsível até certo ponto, antes que a próxima decisão seja tomada. Um aspecto chave destas situações, é que nenhuma decisão pode ser tomada isoladamente, em vez disso, deve-se ponderar o desejo de um baixo custo no presente em relação a altos custos indesejáveis no futuro (SIMON, 2001). Esta se baseia no princípio da otimalidade desenvolvido por *Richard Ernest Bellman* em 1953, que é descrito como:

Frequentemente as soluções são obtidas por um processo regressivo, trabalhando o problema do final para o início. Com isso a dificuldade será reduzida, ao se decompor o problema em uma sequência de problemas inter-relacionados mais simples (BERTSEKAS; TSITSIKLIS, 1996) e (SIMON, 2001). A programação dinâmica tem como questão fundamental, o interesse em buscar a resposta de como um sistema pode aprender a melhorar o seu desempenho a longo prazo, quando isto pode exigir o sacrifício do desempenho atual.

Um sistema dinâmico é representado em sua forma contínua por

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{B.1}$$

sendo o vetor de estado $x(t)_{n\times 1}$ e o vetor de controle $u(t)_{p\times 1}$. Sendo o vetor de controle definido por

$$u(t) = u(kT), \tag{B.2}$$

sendo que t, deve assumir valores de maneira que $kT \leq t \leq (k+1)T$.

O objetivo é definir o valor ótimo para $u^*(KT)$, para um dado $k = 0, 1, 2, 3, \ldots, N-1$ que minimize o índice de desempenho dado por

$$J = \frac{1}{2}(tF)^T S x(tF) + \frac{1}{2}\int_0^{tF} [x(t)^T Q x(t) + u(t)^T R u(t)], \tag{B.3}$$

sendo N é o número de amostras, T corresponde ao período de amostragem e as matrizes Q e R do índice de desempenho são definidas, respectivamente, positiva e semi-positiva. Portanto, a equação de estado discretizada é apresentada da seguinte forma

$$x[(k+1)T] = A_d x(kT) + B_u(kT), \tag{B.4}$$

sendo que A_d e B_d são definidas por

$$A_d = e^{AT} \tag{B.5}$$

e

$$B_d = \int_0^T A_d(T-\tau) B d\tau. \tag{B.6}$$

O índice de desempenho discretizado, é dado em termos dos operadores

$$\begin{aligned} J_{DLQR} &= \frac{1}{2}x(NT)^T S x(NT) + \frac{1}{2}\sum_{k=0}^{N-1}[x(kT)^T \hat{Q} x(kT) + \\ &\quad + 2x(kT)^T M(T) u(kT) + u(kT)^T \hat{R} u(kT)] \end{aligned} \tag{B.7}$$

sendo que

$$\hat{Q}(T) = \int_{kT}^{(k+1)T} A_d^T(t-kT) Q A_d(t-kT) dt, \tag{B.8}$$

$$M(T) = \int_{kT}^{(k+1)T} A_d^T(t - kT)QB_d(t - kT)dt \tag{B.9}$$

e

$$\hat{R}(T) = \int_{kT}^{(k+1)T} [B_d^T(t - kT)QB_d(t - kT) + R]dt \tag{B.10}$$

A matriz $M(T)$ é a matriz de ponderação do acoplamento entre o estado e a entrada. Como de maneira geral, o índice de desempenho quadrático é descrito sem a matriz M, temos o índice descrito da seguinte forma

$$J_{DLQR} = \frac{1}{2}x_k^T S x_n + \frac{1}{2}\sum_{k=0}^{n-1}(x_k^T Q x_k + u_k^T R u_k). \tag{B.11}$$

Considerando o tempo infinito, ou seja, $N = \infty$, temos o índice de desempenho igual a

$$J_{DLQR} = \frac{1}{2}\sum_{k=0}^{\infty}(x_k^T \hat{Q} x_k + x_k^T 2M u_k + u_k^T R u_k). \tag{B.12}$$

Observe que como $N \to \infty$, o custo final pode ser desprezado, visto que, o estado final x_n tende ao estado de equilíbrio. Para o projeto do DLQR de tempo infinito, é necessário que o sistema seja assintoticamente estável, quando em malha fechada.

O sistema dado pela equação a seguir, precisa ser controlável e observável.

$$x_{k+1} = f(x_k) + g(x_k)u_k. \tag{B.13}$$

A solução para o DLQR infinito é obtido, de maneira que $k \to \infty$. Assumindo que $N \to \infty$, a matriz P_k dda solução da DARE é o ganho e assume um valor constante, que é dada por

$$\lim_{k\to\infty} P_k = P. \tag{B.14}$$

O controle ótimo é dado por

$$u_k^* = -(\hat{R} + B_d^T P B_d)^{-1}(B_d^T P A_d + M^T)x_k^*. \tag{B.15}$$

Portanto, a matriz de realimentação é constante, e é dada por

$$K = (\hat{R} + B_d^T P B_d)^{-1}(B_d^T P A_d + M^T) \tag{B.16}$$

O índice de desempenho ótimo para $N = \infty$ através da minimização da Equação(B.13), é dada por:

$$J_\infty^* = \frac{1}{2} x_0^T P x_0 \tag{B.17}$$

B.2 Solução do DLQR via Programação Dinâmica

Para o projeto do DLQR de tempo infinito, faz-se necessário que o sistema seja controlável ou estável na realimentação de estados. É exigido que a controlabilidade seja superior a estabilidade, visto que um sistema não controlável pode ser estabilizado se os estados não controláveis sejam estáveis.

O regulador discreto ótimo de tempo finito é dado por:

$$minJ = G(x_n) + \sum_{k=0}^{N-1} F(x_k, u_k) \tag{B.18}$$

sujeito a seguinte restrição

$$x_{k+1} = A_d x_d + B_d u_k \tag{B.19}$$

sendo

$$G(x_n) = \frac{1}{2} x_n^T S x_n \tag{B.20}$$

$$\sum_{k=0}^{N-1} F(x_k, u_k) = \frac{1}{2} x_N^T \hat{Q} x_N + x_k^T M u_k + \frac{1}{2} u_k^T \hat{R} u_k \tag{B.21}$$

com o estado inicial x_0, conhecido.

Assumindo que $J_{N-j}(x_j)$ seja o índice de desempenho no intervalo de $[j, N]$, tem-se que

$$J_{N-j}(x_j) = G(x_N) + \sum_{k=0}^{N-1} F_k(x_k, u_k) \tag{B.22}$$

com $j = 0, 1, 2, \ldots, N$.

O mínimo valor de $J_{N-j}(x_j)$ é descrito por

$$f_{N-j}(x_j) = \min_{u_{N-1}} J_{N-j}(x_j). \tag{B.23}$$

Admitindo que $j = N$, a Equação (B.23) representa o índice de desempenho no instante 0, então:

$$f_0(x_N) = G(x_N) = \frac{1}{2} x_N^T S x_N. \tag{B.24}$$

Para os demais valores de j, o índice é dado por

$$\begin{aligned} f_1(x_{N-1}) &= \min_{u_{N-1}} J_{N-j} \\ &= \min_{u_{N-1}} G(x_N) + F_{N-1}(x_{N-1}, u_{N-1}). \end{aligned} \tag{B.25}$$

Sabendo-se que

$$G(x_N) = \frac{1}{2}(A_d x_{N-1} + B_d u_{N-1})^T S (A_d x_{N-1} + B_d u_{N-1}). \tag{B.26}$$

Considerando a Equação(B.25), tem-se

$$\begin{aligned} f_1(x_{N-1}) &= \min_{u_{N-1}} \left(\frac{1}{2} x_{N-1}^T \left(\hat{Q} + A_d^T S A_d \right) x_{N-1} \right) \\ &\quad + \min_{u_{N-1}} \left(x_{N-1}^T \left(M + \frac{1}{2} A_d^T S B_d \right) u_{N-1} \right) \\ &\quad + \min_{u_{N-1}} \left(\frac{1}{2} u_{N-1}^T B_d^T S A_d x_{N-1} \right) \\ &\quad + \min_{u_{N-1}} \left(\frac{1}{2} u_{N-1}^T \left(\hat{R} + B_d^T S B_d \right) u_{N-1} \right). \end{aligned} \tag{B.27}$$

Então,

$$f_1(x_{N-1}) = \min_{u_{N-1}} J_1(x_{N-1}). \tag{B.28}$$

Para calcular o mínimo, tem-se

$$\frac{\partial J_1(x_{N-1})}{\partial u_{N-1}} = 0, \tag{B.29}$$

o resultado será dado por

$$\left[\left(M + \frac{1}{2} A_d^T S B_d \right) + \frac{1}{2} B_d^T S A_d \right] x_{N-1}^* + (\hat{R} + B_d^T S B_d) u_{N-1}^*, \tag{B.30}$$

desta forma o controle ótimo será definido por

$$u_{N-1}^* = -(\hat{R} + B_d^T S B_d)^{-1} \left(M + \frac{1}{2} A_d^T S B_d \right) x_{N-1}^*. \tag{B.31}$$

Aplicando a Equação(B.31) na Equação(B.27), e realizando ass devidas simplificações, temos

$$\begin{aligned} f_1(x_{N-1}) &= \frac{1}{2} x_{N-1}^T [\hat{Q} + A_d^T S A_d - \\ &\quad \left(M^T + B_d^T S A_d \right)^T \left(\hat{R} + B_d^T S B_d \right)^{-1} \left(M^T + B_d^T S A_d \right)] x_{N-1}. \end{aligned} \tag{B.32}$$

Como a matriz P_R de Riccati, corresponde a matriz S, tem-se que

$$P_{N-1} = \hat{Q} + A_d^T S A_d - (M^T + B_d^T S A_d)^T (\hat{R} + B_d^T S B_d)^{-1} (M^T + B_d^T S A_d), \tag{B.33}$$

desta forma, tem-se respectivamente:

$$f_0(x_N) = \frac{1}{2} x_N^T P_N x_N \tag{B.34}$$

e

$$f_1(x_{N-1}) = \frac{1}{2} x_{N-1}^T P_{N-1} x_{N-1}. \tag{B.35}$$

Para caracterizar a otimização para os dois últimos estágios, segue que

$$f_2(x_{N-2} = \min_{u_{N-2,N-1}} J_2(x_{N-2}), \tag{B.36}$$

então,

$$f_2(x_{N-2}) = \min_{u_{N-2},u_{N-1}} F_{N-2}(x_{N-2}, u_{N-2}) + F_{N-1}(x_{N-1}, u_{N-1}) + G(x_N). \tag{B.37}$$

Sabe-se que:

$$f_2(x_{N-2}) = \min_{u_{N-2}} F_{N-2}(x_{N-2}) + f_1(x_{N-1}) \tag{B.38}$$

$$F_{N-2}(x_{N-2}, u_{N-2}) = \frac{1}{2} x_{N-2}^T \hat{Q} x_{N-2} + x_{N-2}^T M u_{N-2} + \frac{1}{2} u_{N-2}^T \hat{R} u_{N-2} \tag{B.39}$$

$$f_1(x_{N-1}) = \frac{1}{2} (A_d x_{N-2} + B_d u_{N-2})^T P (A_d x_{N-2} + B_d u_{N-2}). \tag{B.40}$$

Para calcular o mínimo, tem-se que

$$\frac{\partial J_2(x_{N-2})}{\partial u_{N-2}} = 0, \tag{B.41}$$

portanto, o controle é dado por:

$$u^*_{N-2} = -(\hat{R} + B_d^T P_{N-1} B_d)^{-1}(M^T + B_d^T P_{N-1} A_d) x^*_{N-2} \tag{B.42}$$

e

$$\begin{aligned} f_2(x_{N-2}) &= \frac{1}{2} x_{N-2}^T [\hat{Q} + A_d^T P_{N-1} A_d - (M^T + B_d^T P_{N-2} A_d)^T \\ &\quad (\hat{R} + B_d^T P_{N-2} B_d)^{-1}(M^T + B_d^T P_{N-1} A_d)] x_{N-2}. \end{aligned} \tag{B.43}$$

Considerando,

$$\begin{aligned} P_{N-2} &= \hat{Q} + A_d^T P_{N-1} A_d - (M^T + B_d^T P_{N-1} A_d)^T \\ &\quad (\hat{R} + B_d^T P_{N-1} B_d)^{-1}(M^T + B_d^T P_{N-1} A_d), \end{aligned} \tag{B.44}$$

a Equação(B.43), após ser simplificada, é dada por

$$f_2(x_{N-2}) = \frac{1}{2} x_{N-2}^T P_{N-2} x_{N-2}. \tag{B.45}$$

De uma forma geral, tem-se que

$$f_{N-j}(x_j) = \frac{1}{2} x_j^T P_j x_j, \tag{B.46}$$

sendo

$$\begin{aligned} P_j &= \hat{Q} + A_d^T P_{j+1} A_d - (M^T + B_d^T P_{j+1} A_d)^T \\ &\quad (\hat{R} + B_d^T P_{j+1} B_d)^{-1}(M^T + B_d^T P_{j+1} A_d), \end{aligned} \tag{B.47}$$

o controle ótimo é dado pela seguinte equação:

$$u_j^* = -(\hat{R} + B_d^T P_{j+1} B_d)^{-1}(M^T + B_d^T P_{j+1} A_d) x_j^* \tag{B.48}$$

Desta forma, tem-se a equação de *Ricatti*, segundo o princípio da otimalidade de Bellman. Esta técnica é definida como Programação Dinâmica.

Por conveniência, será considerado $\hat{Q} = Q, \hat{R} = R$, e $M = 0$. Desta forma, simplificaremos as Equações(B.47) e (B.48). Portanto, as seguintes formas são obtidas

$$P_j = Q + A_d^T P_{j+1} A_d - (B_d^T P_{j+1} A_d)^T (R + B_d^T P_{j+1} B_d)^{-1}(B_d^T P_{j+1} A_d) \tag{B.49}$$

e

$$u_j^* = (R + B_d^T P_{j+1} B_d)^{-1}(B_d^T P_{j+1} A_d) x_j^*. \tag{B.50}$$

Bibliografia

AGUIRRE, L. A. *Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais*. [S.l.]: editora UFMG, 2004.

ALI, A.; ALI, R. L. An improved gain vector to enhance convergence characteristics of recursive least squares algorithm. *International Journal of Hybrid Information Technology*, v. 4, n. 2, 2011.

ANDERSON, B. D.; MOORE, J. B. *Optimal control: linear quadratic methods*. [S.l.]: Prentice Hall Englewood Cliffs, NJ, 1990.

ASTROM, K. J.; WITTENMARK, B. *Adaptive control*. 2nd. ed. Boston - USA: Addison-Wesley Longman Publishing Co., Inc, 1994.

ÅSTRÖM, K. J.; WITTENMARK, B. *Adaptive control*. [S.l.]: Courier Dover Publications, 2008.

BELLMAN, R. Dynamic programming and stochastic control processes. *Information and control*, Elsevier, v. 1, n. 3, p. 228–239, 1958.

BERTSEKAS, D. P. *Dynamic programming and optimal control*. [S.l.]: Athena Scientific, 1995.

BERTSEKAS, D. P.; TSITSIKLIS, J. N. Neuro-dynamic programming (optimization and neural computation series, 3). *Athena Scientific*, v. 7, p. 15–23, 1996.

BRADTKE, S. J. Incremental dynamic programming for on-line adaptive optimal control. In: *CMPSCI Technical Report 94-62*. [S.l.: s.n.], 1994.

BRADTKE, S. J.; YDSTIE, B. E.; BARTO, A. G. Adaptive linear quadratic control using policy iteration. In: IEEE. *American Control Conference, 1994*. [S.l.], 1994. v. 3, p. 3475–3479.

BRADTKE, S. J.; YDSTIE, B. E.; BARTO, A. G. Adaptive linear quadratic control using policy iteration. In: *Computer Science Departmant University of Massachusetts*. [S.l.: s.n.], 1994.

BUSONIU, L. et al. *Reinforcement learning and dynamic programming using function approximators*. [S.l.]: CRC press, 2010.

BUSONIU, L. et al. Approximate reinforcement learning: An overview. In: IEEE. *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*. [S.l.], 2011. p. 1–8.

FIRMINO, F. L. *Simulação e controle de um helicóptero a partir de modelos linearizados em sete pontos de operação*. Tese (Doutorado) — Instituto Militar de Engenharia, 2008.

HALDER, K. et al. Impact of weighting matrices in the design of discrete optimal controller based on lqr technique for non-linear system. In: *Computer Communication and Informatics (ICCCI), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 1–6.

ISERMANN, R.; MUNCHHOF, M. *Identification of dynamic systems, an introduction with applications*. [S.l.]: Springer Verlag, Berlin, Heidelberg, 2011.

JIANG, Z.-P.; JIANG, Y. Robust adaptive dynamic programming: Recent results and applications. In: *Control Conference (CCC), 2013 32nd Chinese*. [S.l.: s.n.], 2013. p. 968–973.

LEWIS, F. L.; VRABIE, D. Reinforcement learning and adaptive dynamic control. *Circuits and Systems Magazine, IEEE*, n. 3, p. 32 – 50, 2009.

LEWIS, F. L.; VRABIE, D. Reinforcement learning and adaptive dynamic programming for feedback control. *Circuits and Systems Magazine, IEEE*, IEEE, v. 9, n. 3, p. 32–50, 2009.

LEWIS FRANK L E VRABIE, D. e. S. V. L. *Optimal control*. [S.l.]: Wiley.com, 2012.

LI, G.; WEN, C. Convergence of normalized iterative identification of hammerstein systems. *Systems & Control Letters*, Elsevier, v. 60, n. 11, p. 929–935, 2011.

LI, J.; ZHENG, Y.; LIN, Z. Recursive identification of time-varying systems: Self-tuning and matrix rls algorithms. *Systems & Control Letters*, Elsevier, v. 66, p. 104–110, 2014.

LIAVAS, A. P.; REGALIA, P. A. On the numerical stability and accuracy of the conventional recursive least squares algorithm. *Signal Processing, IEEE Transactions on*, IEEE, v. 47, n. 1, p. 88–96, 1999.

LIU, D.; WEI, Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE, 2014.

LIU, S.-j. et al. Extraction of fetal electrocardiogram using recursive least squares and normalized least mean squares algorithms. In: IEEE. *Advanced Computer Control (ICACC), 2011 3rd International Conference on*. [S.l.], 2011. p. 333–336.

LIU, Y.; DING, F. Convergence properties of the least squares estimation algorithm for multivariable systems. *Applied Mathematical Modelling*, Elsevier, v. 37, n. 1, p. 476–483, 2013.

LJUNG, L. *System identification*. [S.l.]: Wiley Online Library, 1999.

LJUNG, S.; LJUNG, L. Error propagation properties of recursive least-squares adaptation algorithms. *Automatica*, Elsevier, v. 21, n. 2, p. 157–167, 1985.

LU, W.; FISHER, D. G. Rls parameter convergence with overparameterized models. *Systems and Control*, p. 133–138, 1989.

MACIEL, A. J. F. *Convergência de estimador RLS para algoritmos de programação dinâmica heurística*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Maranhão, 2012.

NETO, J. Viana da F.; ANDRADE, G. de. Design of decision making unit for neuro-fuzzy control of dynamic systems. In: *Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on*. [S.l.: s.n.], 2011. p. 116–121.

PANNOCCHIA, G. et al. A parsimonious algorithm for the solution of continuous-time constrained lqr problems with guaranteed convergence. In: IEEE. *Control Conference (ECC), 2013 European*. [S.l.], 2013. p. 1553–1558.

POTTER, J. New statistical formulas,". *Space Guidance Analysis Memo*, v. 40, p. 1309–1314, 1963.

POWELL, W. B. *Approximate dynamic programming: Solving the curses of dimensionality*. [S.l.]: John Wiley & Sons, 2007.

POWELL, W. B.; RYZHOV, I. O. *Optimal learnig*. [S.l.]: John Wiley & Sons, 2012.

RASTEGARPOUR, S.; SHAHMANSOORIAN, A.; MAZINAN, A. Designing an optimal control law for nonlinear system by using intelligent algorithm. In: *Measurement, Information and Control (ICMIC), 2013 International Conference on*. [S.l.: s.n.], 2013. v. 02, p. 830–836.

RÊGO, P. H. M. *Aprendizagem por reforço e programação dinâmica aproximada para controle ótimo: Uma abordagem para o Projeto Online do regulador linear quadrático discreto com programação dinâmica heuística*. Tese (Doutorado) — UFMA, 2014.

RÊGO, P. H. M.; FONSECA, J. a. V.; FERREIRA, E. M. Convergence of the standard rls method and udut factorisation of covariance matrix for solving the algebraic riccati equation of the dlqr via heuristic approximate dynamic programming. *International Journal of Systems Science*, p. 1–19, October 2013.

ROBINETT, R. D. et al. *Applied dynamic programming for optimization of dynamical systems*. [S.l.]: SIAM, 2005.

SAMBLANCAT, C. *Commande robuste multivariable. Applications à l'hélicoptère*. Tese (Doutorado), 1991.

SANTOS, W. R. M. et al. Rls algorithms and convergence analysis method for online dlqr control design via heuristic dynamic programming. In: *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. Washington, DC, USA: IEEE Computer Society, 2014. (UKSIM '14), p. 77–83. ISBN 978-1-4799-4922-9. Disponível em: <http://dx.doi.org/10.1109/UKSim.2014.109>.

SAYED, A. H. *Adaptive filters*. [S.l.]: John Wiley & Sons, 2008.

SI, J. et al. Approximate dynamic programming for continuous state and control problems. In: *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*. [S.l.: s.n.], 2009. p. 1415–1420.

SILVA, F. N. da; LUIS, S. Métodos neuronais para a soluçao da equaçao algébrica de riccati e o lqr. 2008.

SILVA, M. E. G.; NETO, J. V. d. F.; SOUZA, F. de. Pnlms-based algorithm for online approximated solution of hjb equation in the context of discrete mimo optimal control and reinforcement learning. *UKSim-AMSS 16th International Conferece on Moddeling and Simulation - IEEE*, 2014.

SIMON, H. *Redes neurais: princípios e prática*. [S.l.]: Porto Alegre: Bookman, 2001.

SLOCK, D. T. Backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms. *Optical Engineering*, International Society for Optics and Photonics, v. 31, n. 6, p. 1153–1169, 1992.

SONG, R.; XIAO, W.; LUO, Y. Optimal control for a class of nonlinear system with controller constraints based on finite-approximation-errors adp algorithm. In: IEEE. *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2013 IEEE Symposium on*. [S.l.], 2013. p. 19–23.

STEVENS, B. L.; LEWIS, F. L. *Aircraft control and simulation*. [S.l.]: John Wiley & Sons, 2003.

TERRA, M.; CERRI, J.; ISHIHARA, J. *Optimal robust linear quadratic regulator for systems subject to uncertainties*. 2014. 1-1 p.

VAHIDI, A.; STEFANOPOULOU, A.; PENG, H. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, Taylor & Francis, v. 43, n. 1, p. 31–55, 2005.

VINTER, R. B. *Optimal control*. [S.l.]: Springer Science+ Business Media, 2010.

WANG, F.-Y. et al. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with-error bound. *Neural Networks, IEEE Transactions on*, IEEE, v. 22, n. 1, p. 24–36, 2011.

XIE, Q.; LUO, B.; TAN, F. Discrete-time lqr optimal tracking control problems using approximate dynamic programming algorithm with disturbance. In: IEEE. *Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on*. [S.l.], 2013. p. 716–721.