UNIVERSIDADE FEDERAL DO MARANHÃO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
MESTRADO EM ENGENHARIA DE ELETRICIDADE
ÁREA: AUTOMAÇÃO E CONTROLE

Gustavo Araújo de Andrade

Programação Dinâmica Heurística Dual e Redes de Funções de Base Radial para Solução da Equação de Hamilton-Jacobi-Bellman em Problemas de Controle Ótimo

São Luís
2014

Gustavo Araújo de Andrade

Programação Dinâmica Heurística Dual e Redes de Funções de Base Radial para Solução da Equação de Hamilton-Jacobi-Bellman em Problemas de Controle Ótimo

Dissertação apresentada ao de Programa de Pós Graduação em Engenharia de Eletricidade da UFMA, como requisito parcial para a obtenção do grau de MESTRE em Engenharia de Eletricidade.

Orientador: João Viana da Fonseca Neto

Doutor em Engenharia Elétrica - UFMA

São Luís

2014

Andrade, Gustavo Araújo de

Programação Dinâmica Heurística Dual e Redes de Funções de Base Radial para Solução da Equação de Hamilton-Jacobi-Bellman em Problemas de Controle Ótimo / Gustavo Araújo de Andrade - São Luís

110f.

Impresso por computador (fotocópia)

Orientador: João Viana da Fonseca Neto

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

1. Controle ótimo 2. Aprendizagem por reforço 3. Programação dinâmica aproximada 4. Programação heurística dual. I.Título.

CDU 681.5.015.24

PROGRAMAÇÃO DINÂMICA HEURÍSTICA DUAL E REDES DE FUNÇÕES DE BASE RADIAL PARA SOLUÇÃO DA EQUAÇÃO DE HAMILTON-JACOBI-BELLMAN EM PROBLEMAS DE CONTROLE ÓTIMO

Gustavo Araújo de Andrade

Dissertação aprovada em 28 de abril de 2014.

Prof. João Viana da Fonseca Neto, Dr.
(Orientador)

Prof. Roberto Célio Limão de Oliveira, Dr.
(Membro da Banca Examinadora)

Prof. Ginalber Luiz de Oliveira Serra, Dr.
(Membro da Banca Examinadora)

Dedicado integralmente à
memória de Raimunda Araújo Barros.

Resumo

Neste trabalho o principal objetivo é apresentar o desenvolvimento de algoritmos de aprendizagem para execução *online* para a solução da equação algébrica de Hamilton-Jacobi-Bellman. O conceitos abordados se concentram no desenvolvimento da metodologia para sistemas de controle, por meio de técnicas que tem como objetivo o projeto *online* de controladores adaptativos são projetados para rejeitar ruídos de sensores, variações paramétricas e erros de modelagem. Conceitos de programação neurodinâmica e aprendizagem por reforço são abordados para desenvolver algoritmos onde a contextualização de determinado ponto de operação faz com que o sistema de controle se adapte e, dessa forma, apresente o desempenho de acordo com as especificações de projeto. Desenvolve-se métodos para a estimação *online* do crítico adaptativo concentrando os esforços em técnicas de estimação do gradiente da função valor do ambiente.

Palavras Chaves: Controle Ótimo, Aprendizagem por Reforço, Programação Dinâmica Aproximada, Programação Heurística Dual, Redes de Função de Base Radial.

Abstract

In this work the main objective is to present the development of learning algorithms for online application for the solution of algebraic Hamilton-Jacobi-Bellman equation. The concepts covered are focused on developing the methodology for control systems, through techniques that aims to design online adaptive controllers to reject noise sensors, parametric variations and modeling errors. Concepts of neurodynamic programming and reinforcement learning are are discussed to design algorithms where the context of a given operating point causes the control system to adapt and thus present the performance according to specifications design. Are designed methods for online estimation of adaptive critic focusing efforts on techniques for gradient estimating of the environment value function.

Keywords: Optimal Control, Reinforcement Learning, Approximate Dynamic Programming,Dual Heuristic Programming, Radial Basis Function Neural Networks.

Agradecimentos

Ao meu orientador Professor João Viana pelo apoio e pela experiência compartilhada através dos anos de convivência e para com quem tenho muita gratidão e amizade.

Aos meus colegas do LabSECI pelas discussões, pelo apoio e amizade nesta árdua porém prazerosa caminhada.

A todos de que contribuíram de forma direta ou indiretamente para minha formação ao longo de todo este tempo de vida acadêmica.

A Coordenação de Aperfeiçoamento de Pessoal (CAPES) pelo apoio financeiro.

A minha namorada Hellen Rose pelo companheirismo e por ficar ao meu lado nos piores e nos melhores momentos.

A minha família, meus irmãos Adva Barros e Antonio Andrade, razão para continuar sempre em frente sem deixar que os obstáculos que a vida impõe me façam desistir.

A minha mãe Raimunda Barros, aonde meus pensamentos estão sempre concentrados nos bons momentos.

Onde quer que estejas, cave profundamente!

Em baixo esta a fonte!

Deixe os homens sombrios a gritar:

“Em baixo é sempre o inferno!”.

Friedrich Nietzsche - A Gaia Ciência.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1

INTRODUÇÃO

Conteúdo

1.1 Introdução

A capacidade do ser humano de tomar decisões com base na sua interação com o ambiente e fazer com que decisões futuras sejam tomadas levando em conta a experiência de interação homem-ambiente anterior é uma das características que faz com que o ser humano se diferencie de outros animais com o mesmo ou maior tempo de evolução. O aprendizado, principal ferramenta evolutiva de muitos sistemas biológicos, se tornou um campo de pesquisa das mais diversas áreas tais como psicologia, filosofia, e neurociência em geral e é explorado neste trabalho como uma ferramenta importante para desenvolvimento de sistemas inteligentes.

O aprendizado é determinado por experiências de inserção no ambiente em que o indivíduo pretende se adaptar como bem explorado em (HAYKIN, 1999, 2012). Este indivíduo sofre com as características intrínsecas do ambiente, e "aprende" a responder as tais características e por sua vez, em uma próxima oportunidade, o individuo responderá àquela situação usando o aprendizado adquirido e minimizando os eventuais erros de interação proveniente do indivíduo com o ambiente.

Para os sistemas de controle o aprendizado esta relacionado com os critérios que o controlador usa para se adaptar as condições atuais do sistema. Para tal situação existem ligeiras diferenças em relação ao aprendizado para outras finalidades. Neste caso o tempo de processamento da informação mediada tem restrições em relação à decisão que influenciou esta consequência.

O processamento de dados em um tempo razoável e levando em conta uma alta gama de variáveis envolvidas, requer de de sistemas microprocessados uma grande capacidade de cálculos por ciclo de amostragem , o que pode tornar o processamento de tal sistema demasiado custoso e até inviável. Por processamento de dados temos como objetivo o controle de sistemas dinâmicos ou a filtragem de dados de um determinado comportamento. Desta forma é necessário encarar não somente os problemas de "computabilidade" envolvido neste processo, como também o comportamento estável do controlador ou filtro a ser projetado (CARNIELLI e EPSTEIN, 2005).

Neste trabalho são abordados problemas envolvendo a especificação, projeto *online* de controle ótimo e conceitos gerais de controladores para sistemas não lineares, de alta ordem, ou com dinâmica de ruídos não modeláveis. A partir dos conceitos de programação neuro-

dinâmica tem se a utilização da ferramenta de inteligência computacional inserida no contexto de controle de sistemas dinâmicos para resolver os problemas citados anteriormente (SUTTON e BARTO, 1998).

A inteligência computacional é uma metodologia na busca por soluções de problemas em que a solução de maneiras clássicas ou até mesmo com ferramentas matemáticas mais arrojadas pode ser custosa. Para a concepção de sistemas de tais tipos é necessário o conhecimento de algoritmos bio inspirados, ou seja, sistemas que tem como principal inspiração o comportamento e a arquitetura da natureza, seja ela comportamental ou biológica. Dessa forma os sistemas baseados em inteligência computacional tem como principal aspecto a busca pela "semelhança" com características da natureza, da fisiologia humana e ate mesmo da forma de pensar do ser humano assim como o seu comportamento em uma determinada sociedade.

O trabalho apresentado a seguir tem como fonte de inspiração, como citado inicialmente, a capacidade de aprendizado do ser humano, afim de se usar os conceitos de aprendizagem por reforço para se obter sistemas de controle ótimo com capacidade de rejeição a pertubações na planta e distúrbios do ambiente em geral. A ferramenta usada para desenvolver o processo de aprendizado, de forma que o mesmo aconteça através das próprias ocorrências, ou seja, *online*. Dessa forma teremos um sistema capaz de aproximar uma solução de controle ótimo e aumentar, em um determinado instante, as margens de estabilidade deste sistema.

Nas seções a seguir são mostrados de maneira objetiva os principais tópicos que motivaram o desenvolvimento de uma metodologia para o projeto e avaliação de controladores.

1.2 Motivação

Este trabalho é motivado por problemas de controle ótimo, aprendizagem por reforço e as soluções apresentadas em (BERTSEKAS, 1995a), (BERTSEKAS, 1995b), (BUSONIU et al., 2010), e ainda em relação a estabilidade robusta em (IOANNOU e SUN, 1996). Nestes trabalhos temos a concepção de algoritmos de programação dinâmica para a solução de problemas de controle ótimo, o regulador linear quadrático no tempo discreto (DLQR). Algumas questões levantadas em (MITCHELL, 1997) e (OMIDVAR e ELLIOTT, 1997) tem como principal aspecto os problemas causados por tais algoritmos enquanto se eleva a escala do processamento de dados a ser realizados. Aplicado a este trabalho utiliza-se uma breve análise da

características computacionais dos algoritmos. Logo, neste caso, serão usados algoritmos desenvolvidos para que o sistema possa ter uma viabilidade mínima de se operar embarcado, seja ele de alta ordem ou não linear, isto é fatores que influenciam na complexidade computacional considerados no desenvolvimento do projeto.

Para os sistemas não lineares é importante uma estratégia de controle que tenha a capacidade de operar em todas as regiões desse sistema, contudo, se a lei de controle for baseada em um mapeamento não linear, esse requisito pode ser facilmente estabelecido. Tais critérios, podem ser aplicados para sistemas de ordem alta, pois não havendo a possibilidade de redução ou aproximação do modelo do mesmo é necessário trabalhar um um grande número de variáveis acopladas o que torna, voltando a este mesmo ponto, o sistema de difícil controle pois o sistema em malha fechada terá obviamente ordem superior ao de malha aberta.

Desta forma o problema de controle a solucionar para a proposta neste trabalho é melhor detalhada nos capítulos seguintes. Estes são tópicos incentivadores para obtenção de resultados no sentido de apresentar uma metodologia de projeto e implementação eficientes para sistemas de controle complexos e no tempo discreto.

1.3 Objetivos

A partir de critérios gerais e específicos estabelecidos no decorrer da pesquisa foram determinados um conjunto de objetivos que traduzem o foco principal do trabalho e são organizados desta maneira para que, desde já, os critérios a serem atingidos fiquem claros para uma boa compreensão de quais são as propostas, a metodologia e os resultados para fins de se esclarecer a contribuição da pesquisa.

1.3.1 Objetivo Geral

O principal objetivo deste trabalho é desenvolver uma metodologia de controle para projeto e análise de sistemas que tenham sobre o ambiente a característica de se adaptar e aprender para operar em regiões transitórias e com margem de estabilidade com algum grau de robustez.

Ainda como objetivo geral é necessário destacar que os sistemas aqui citados e testa-

dos são, da maneira mais realista possível, frutos de observações experimentais em plataformas computacionais, e que serão investigados parâmetros de projeto e implementação para que o sistema de controle tenha aplicabilidade do ponto de vista de engenharia.

1.3.2 Objetivos Específicos

Os objetivos específicos destacados, baseando-se nas premissas gerais, são:

- Projetar uma arquitetura de controle utilizando-se programação dinâmica para controle ótimo;
- Projeto e implementação computacional do sistema de aprendizado online para controle ótimo;
- Desenvolver e implementar algoritmos *online* de controle ótimo via Programação Heurística Dual baseados em Redes de Base Radial;
- Investigar a estabilidade do sistema na região transitória do ciclo de aprendizagem;
- Desenvolver método para avaliação de sistemas de controle ótimo online via aproximação da equação de HJB.

1.4 Estado da Arte

Um dos grande entraves nas questões que abordam aprendizado de máquina e sistemas de inteligência computacional em geral é a adoção de políticas que habilitam uma máquina a capacidade de poder responder a estímulos que não estão previstos, pelo menos de forma direta, em sua programação. O aprendizado de máquina tem como paradigma a busca por estratégias que possam combinar o enorme poder de processamento dos dispositivos atuais e a ideia "humana" de quantificar, qualificar, e tomar decisões não somente baseada em dados pontuais como também em características que tornam tais decisões acertadas e baseadas em uma experiência usualmente atribuído a seres humanos. Como notamos em (MITCHELL, 1997) e (OMIDVAR e ELLIOTT, 1997) podemos estimular sistemas computacionais com uma larga escala de dados obtidos a partir de interfaces com o ambiente. A composição primária do problema é justamente a decisão que deve se incorrer a partir de tais dados, ou seja, o processo

que fará com que a máquina interfira no ambiente e possa, de alguma forma, adequado ao seu próprio funcionamento.

A partir da década de 1950 surgiram novos formas de tratar problemas de otimização de sistemas dinâmicos. A partir de tais mudanças aplicadas ao desenvolvimento de processadores mais poderosos a fim de aplicar os algorítimos obtidos. Desta maneira a pesquisa se utiliza da teoria de aprendizagem por reforço moderna ou a programação dinâmica aproximada,(do inglês, *Approximate Dynamic Programming* - ADP) é aplicada a problemas de controle ótimo multivariável afim de se obter, de forma conjunta, características de sistemas de controle ótimo e de controle adaptativo com resposta robusta a situações adversas e erros de modelagem e de medição. Pode se observar nos trabalhos de (FAIRBANK et al., 2013),(ARSLAN, 2004) que as correntes de pesquisa em programação neurodinâmica se aplicam aos modos de como o critico adaptativo é desenvolvido para obter este mesmo nível de robustez.

Um grande esforço para desenvolver tais técnicas vem sendo explorado de (SUTTON e BARTO, 1998),(BARTO, 2004) (WIERING e VAN OTTERLO, 2011) através de técnica que se utilizam da programação dinâmica para resolver problemas relacionados com o aprendizado de máquina, especificado em (HAYKIN, 1999, 2012) como programação neurodinâmica que tem como fundamento o aprendizado de máquina pela aprendizagem por reforço do inglês (Reinforcement Learning, RL). As técnicas específicas de envolvem o aprendizado não supervisionado, bem explorado em (BARTO, 2004) (SUTTON e BARTO, 1998) e ainda as arquiteturas neurais aplicadas a programação dinâmica observadas em (BERTSEKAS, 1995b).

Desta maneira iniciou-se a aplicação de poderosos algorítimos de otimização para a maior variedade de sistemas dinâmicos, obviamente, neste contexto histórico, tais sistemas estavam aliados a ferramentas que envolviam equipamentos bélicos e aeroespaciais (BOUDAREL et al., 1971, BRYSON e HO, 1975).

Logo tem-se como uma máxima que rege a programação dinâmica, desenvolvida por Ernest Richard Bellman em seus trabalhos (BELLMAN, 1957), que resulta, explicitado de forma bem coerente em (HAYKIN, 1999) como princípio da otimalidade de *Bellman*. As equações de otimalidade são base fundamental para a programação dinâmica e ainda podendo o sistema linearizado em pontos de operações aonde é possível se determinar funções quadráticas para o estados e ação de controle, amplamente discutidas em(BERTSEKAS, 1995a) e (BERTSEKAS, 1995b) com o estabelecimento de critérios práticos para a determinação de críticos

adaptativos para sistemas de aprendizagem e ainda o estabelecimento de critérios quadráticos apresentados em (BRADTKE, 1993) e (LEWIS e SYRMOS, 1995).

No contexto de controle ótimo é importante observar que dentre as características que são observadas como relevantes para a pesquisa esta a contribuição em relação ao teste e implementação de arquiteturas de críticos adaptativos ja desenvolvidos, vide (LENDARIS et al., 2001),(LENDARIS et al., 2002),(AL-TAMIMI et al., 2007a) e na maioria dos capítulos de (LEWIS e LIU, 2013). Trabalhos como (AL-TAMIMI et al., 2007b),(DING et al., 2011),(LIN et al., 2006) e (LIN e YANG, 2007) exploram tais esquemas de crítico adaptativo e mostram as vantagens e desvantagens que cada esquema tem, seja em relação a ser livre de modelo ou seja em relação a complexidade computacional para um número maior de parâmetros a se estimar. Ate mesmo os estudo que necessitam de investigação apurada em relação a estabilidade numérica dos sistemas, tema investigado recentemente em (REGO et al., 2013) e (da FONSECA NETO and REGO, 2014). Em relação a estabilidade e convergência do algoritmos apresentados trabalhos como (KHAN et al., 2012), (LIU et al., 2011) e (LIU et al., 2012) consideram a possibilidade de aproximar a função de custo de maneira a se obter desempenho robusto e considerar características de ruídos ainda visualizando a ótica do projeto de uma aproximador para tal custo, seja ele da família LS ou neuro adaptativo. Neste aspecto (JIANG e JIANG, 2012, 2013, 2014) trabalha com o caso de aproximação robusta do crítico adaptativo considerando um aproximador de função neural.

As características de RNA's de funções de base radial (do inglês,*Radial Basis Function* - RBF), arquitetura explorada neste trabalho auxiliam na solução onde problemas em que o contexto é aproximação de funções através de um espaço não linear de alta dimensão, desta forma são exploradas estas características em (LENDARIS et al., 2002),(YU et al., 2011) e (KARAYIANNIS, 1998).

Por fim o trabalho se baseia na literatura clássica de controle digital discreto e de controle adaptativo (ASTROM e WITTEMMAK, 1989) ,(ASTROM e WITTEMMAK, 1997) para estudos de implementação e concepção de sistemas robóticos industriais com a principal finalidade da obtenção de estabilidade numérica e de aplicabilidade a sistemas computacionais dedicados, usando assim conceitos de projetos de sistemas embarcados seja em projeto de microcontroladores e/ou linguagens de descrição de hardware, do inglês (Hardware Description Language, HDL) (DANIEL D. GAJSKI, 2009),(BERGER, 2001), observando ainda os avanços

e as vantagens relacionadas em (HU et al., 2008),(WAN et al., 2007),(WANG et al., 2009),(KIS et al., 2008) e (WU et al., 2009).

1.5 Organização da Dissertação

A dissertação foi organizada de forma a manter, ao mesmo tempo, a coerência da obtenção de resultados, e a boa prática da apresentação dos conceitos para que o leitor possa compreender como a teoria atual esta engajada nesta pesquisa e qual é a contribuição da mesma.

No Capítulo 1 tem-se os conceitos gerais envolvidos, a motivação do trabalho, assim como a especificação dos objetivos da pesquisa e a apresentação dos mesmos. Já na Seção 1.4 tem-se o estado da arte do tema proposto na dissertação, organizados em tópicos pertinentes e esta pesquisa relacionando com as publicações e o contexto delas dentro da contribuição aqui proposta.

No Capítulo 2 tem-se a apresentação da teoria de programação neurodinâmica para que possa ser apresentada posteriormente, a teoria em discussão na pesquisa de programação dinâmica aproximada. No Capítulo 3 tem-se a apresentação da teoria de redes neurais, ferramenta de aprendizado para o nosso sistema de controle, necessariamente as RNA's serão introduzidas como o sub bloco *crítico*, o sistema responsável pela lei de adaptação do sistema de controle. No mesmo Capítulo 3 tem-se a apresentação da arquitetura *online* desenvolvida para este sistema de aprendizado. Por fim, no Capítulo 4, apresenta-se desenvolvimento dos algoritmos desenvolvidos, os testes para projetos de sistemas embarcados assim como os resultados computacionais obtidos. No Capítulo 5 tem-se as considerações finais, tratados sobre a contribuição do trabalho e uma perspectiva de trabalhos futuros para a pesquisa.

2

APRENDIZAGEM POR REFORÇO PARA CONTROLE

Conteúdo

2.1 Introdução

A programação neurodinâmica estabelece princípios para aplicação e solução de problemas para aprendizagem de sistemas, seja no que se refere a tomada de decisão ou seja em relação a problemas de controle e rastreamento.

Aprendizagem por reforço remete a questões de como um agente autônomo tem a capacidade de sentir e atuar de maneira ótima em seu ambiente de interação. Considerando o sistema de controle como um agente, ou ator, tem-se a ótica da necessidade da tomada de decisão para determinadas situações. A necessidade de aprender esta no contexto de que é necessário ao controlador operar em regiões de otimalidade global, desta forma o sistema pode se adaptar de maneira ótima as condições de operação que são apresentadas ao mesmo.

Sistemas de controle ótimo são aplicados aos problemas de controle que necessitam de especificações de desempenho bem propostas e de uma solução de política de controle que minimiza os custos que o sistema de controle tem para operar a planta em regiões lineares e estáveis. As vezes essas garantias de desempenho e estabilidade levam determinados controladores projetados por meios clássicos a ter que operar em regiões aonde o sistema não pode se estabilizar ou ate mesmo não pode garantir que o mesmo rejeite pertubações e falhas no modelo do processo.

O processo de projeto de sistemas multivariáveis envolve uma série de análises sobre as condições que este sistema apresenta quando operando em malha fechada. Por tal motivo é interessante, a partir destas análises, propor uma metodologia de projeto de controladores discretos que realizam sua política de controle considerando as incertezas e pertubações presentes em qualquer sistema sujeito a intempéries da natureza. Um dos focos que se pretende é garantir, em um processo de aprendizagem online que este controlador se adapta, em um intervalo de tempo desejado, as condições que o ambiente apresenta neste instante.

2.2 Aprendizagem por Reforço

A aprendizagem por reforço é um processo de aprendizagem baseado no comportamento do sistema, ou seja, a programação neurodinâmica aplica os conceitos de tomada de decisão baseada exclusivamente na interação do sistema de aprendizagem e o ambiente no qual

o mesmo esta inserido, levando em conta ainda a presença de incertezas na operação do sistema de aprendizagem em relação ao ambiente. Neste processo de aprendizagem temos dois importantes blocos que integram o sistema, como mostrado na Figura 2.1.

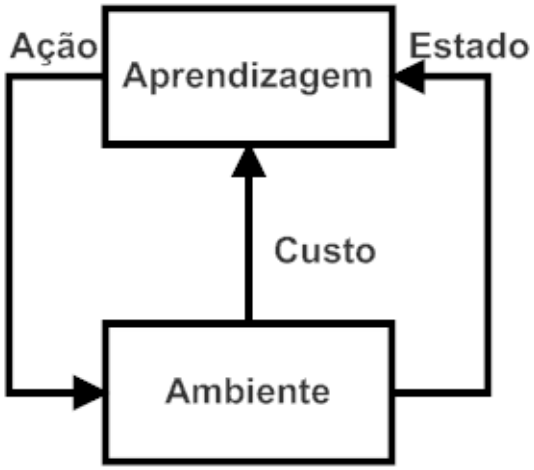


Figura 2.1: Sistema de Aprendizagem e Interação com o Ambiente

Como observado na Figura 2.1 um sistema de controle baseado em aprendizagem por reforço necessita basicamente da parte sensora do sistema para a tomada de decisões. Há um contexto controverso no que se diz respeito de que AR pode está intrinsecamente ligada a programação dinâmica. Em (BERTSEKAS, 1995a) AR é uma classe de algoritmos que necessariamente aplica-se a tomada de decisão em virtude de medições a aproximações de custo. Já em (POWELL, 2008, 2011) o contexto de AR é excluído totalmente da programação dinâmica, observando que esta última necessita inevitavelmente de conhecimento prévio sobre o comportamento do ambiente.

2.3 Aprendizagem por Reforço para Controle Ótimo

A programação dinâmica, diferente de técnicas clássicas de controle ótimo, deforma que a estimativa dos custos do sistema venha a se tornar o crítico adaptativo do sistema. Um dos pontos fortes do projeto do crítico adaptativo em programação dinâmica é que tais formulações podem se aplicar a problemas envolvendo sistemas estocásticos, pois a parametrização da função de aprendizagem pode ser utilizada na forma de uma função de distribuição de probabilidade. Tem-se o ambiente do sistema, ou seja, a dinâmica representada por f e as políticas de controle representadas por g e, mesmo assumindo que este é um Processo

de Decisão Markoviano a dinâmica do sistema será então

$$\begin{aligned} x_{k+1} &= f(x_k, y_k) \\ y_k &= g(x_k). \end{aligned} \tag{2.1}$$

A principal função da programação dinâmica é explorar a Função Valor do sistema, denotada por $V(x)$ na qual os modelos do sistema de recompensas para os estados x usando-se a política de controle g. Considerando f e g funções lineares de parâmetros **A**,**B** e **E** tem-se

$$\begin{aligned} x_{k+1} &= \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{E}w_k \\ y_k &= x_k, \end{aligned} \tag{2.2}$$

sendo $x \in \Re^n$ os estados do sistema de ordem n no tempo discreto, $u \in \Re^{nxm}$ o vetor de m entradas, $w \in \Re^{nxm}$ o vetor de pertubações de entrada e $y \in \Re^p$ o vetor das p saídas do ambiente.

Portanto o trabalho a se desenvolver é aproximar a função de custo $V(x_k)$ ótima de maneira a se obter a tomada de decisão otimizada. Desta forma a Programação Dinâmica Aproximada surge como uma maneira de restringir o esforço computacional para a tomada de decisão do PDM de forma Desta maneira tem-se as equações do comportamento linear do sistema. Observa-se portanto, que o sistema é modelado de maneira a se obter a resposta a entradas ruidosas de forma que o mesmo apresente comportamento com margem de estabilidade robusta. Tem -se portanto a função valor

$$V^*(x_k) = \min_u \max_w \sum_{i=k}^{\infty} \left[x_i^T \mathbf{Q} x_i + u_i^T \mathbf{R} u_i - \gamma^2 w_i^T w_i \right], \tag{2.3}$$

sendo $\mathbf{Q} > 0$ a matriz de ponderação de estados, $\mathbf{R} > 0$ a matriz de ponderação de controle e γ um valor de projeto relacionado a atenuação de pertubações. Agora as equação da função valor é tratada de forma a se observar o princípio da otimalidade de Bellman, logo

$$V^*(x_k) = \min_u \max_w \{ r(x_k, u_k, w_k) + V^*(x_{k+1}) \} = \max_w \min_u \{ r(x_k, u_k, w_k) + V^*(x_{k+1}) \} \tag{2.4}$$

sendo $r(x_k, u_k, w_k) = x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k - \gamma^2 w_k^T w_k$ a função de recompensa quadrática do ambiente, dado por

$$V^*(x_k) = x_k^T \mathbf{P_k} x_k, \tag{2.5}$$

sendo $\mathbf{P} > 0$ a solução da Equação Algébrica de Riccati Generalizada,(EARG) expandida da seguinte forma para o jogo de soma zero em (AL-TAMIMI et al., 2007a) e dada por:

$$\mathbf{P}_{k+1} = \mathbf{A^T P}_k\mathbf{B} + \mathbf{Q} - \left[\begin{array}{cc} \mathbf{A^T P}_k\mathbf{B} & \mathbf{A^T P}_k\mathbf{E} \end{array}\right] \times \tag{2.6}$$

$$\left[\begin{array}{cc} \mathbf{I} + \mathbf{B^T P}_k\mathbf{B}\mathbf{R}^{-1} & \mathbf{B^T P}_k\mathbf{E} \\ \mathbf{E^T P}_k\mathbf{A} & \mathbf{E^T P}_k\mathbf{E} - \gamma^2\mathbf{I} \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{B^T P}_k\mathbf{A} \\ \mathbf{E^T P}_k\mathbf{A} \end{array}\right], \tag{2.7}$$

e para o ponto de sela único obtém-se as seguintes restrições

$$\begin{array}{c} \mathbf{I} - \gamma^2\mathbf{E^T P E} > \mathbf{0} \\ \mathbf{I} + \mathbf{B^T P B} > \mathbf{0}. \end{array} \tag{2.8}$$

Aplicando o princípio da otimalidade de Bellman tem-se

$$\begin{array}{c} V^*(x_k) = \min\limits_u \max\limits_w \left\{ r(x_k, u_k, w_k) + V^*(x_{k+1}) \right\} \\ = \min\limits_u \max\limits_w \left\{ x_k^T\mathbf{Q}x_k + u_k^T\mathbf{R}u_k - \gamma^2 w_k^T w_k + x_{k+1}^T\mathbf{P}x_{k+1} \right\}. \end{array} \tag{2.9}$$

Substituindo na dinâmica f e g, na Eq.(2.2) e aplicando a um sistema linear dado na Eq. (2.2) tem-se

$$\begin{array}{c} x_k^T\mathbf{P}_{k+1}x_k = \min\limits_u \max\limits_w \left\{ r(x_k, u_k, w_k) + V^*(x_{k+1}) \right\} \\ = \min\limits_u \max\limits_w \left\{ x_k^T\mathbf{Q}x_k + u_k^T\mathbf{R}u_k - \gamma^2 w_k^T w_k + (\mathbf{A}x_k + \mathbf{B}u_k + \mathbf{E}w_k)^{\mathbf{T}}\mathbf{P}_k(\mathbf{A}x_k + \mathbf{B}u_k + \mathbf{E}w_k) \right\}. \end{array} \tag{2.10}$$

Com o objetivo de maximizar a função de custo apresentada, em relação a w_k tem-se a seguinte condição de primeira ordem

$$\frac{\partial V_k}{\partial w_k} = 0 \tag{2.11}$$

tem-se como resultado a Eq.(2.11),

$$\frac{\partial V_k}{\partial w_k} = -2\gamma^2 w_k + 2\mathbf{E^T P}_k(\mathbf{A}x_k + \mathbf{B}u_k + \mathbf{E}w_k). \tag{2.12}$$

Desta forma a pertubação pode ser escrita da seguinte forma

$$w_k = \left(\gamma^2\mathbf{I} - \mathbf{E^T P}_k\mathbf{E}\right)^{-1}\left(\mathbf{E^T P}_k\mathbf{A}x_k + \mathbf{E^T P}_k\mathbf{B}u_k\right). \tag{2.13}$$

Agora, da mesma maneira determina-se a política de controle

$$\frac{\partial V_k}{\partial u_k} = 0, \tag{2.14}$$

e da mesma maneira escreve-se

$$u_k = -\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}\right)^{-1}\left(\mathbf{B^T P}_k\mathbf{A}x_k+\mathbf{B^T P}_k\mathbf{E}w_k\right), \tag{2.15}$$

logo substituindo a Eq. (2.13) na Eq. (2.15) tem-se a política ótima de controle

$$\begin{aligned} u_k^* = &\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}-\mathbf{B^T P}_k\mathbf{E}\left(\mathbf{E^T P}_k\mathbf{E}-\gamma^2\mathbf{I}\right)^{-1}\right)^{-1}\times \\ &\left(\mathbf{B^T P}_k\mathbf{E}\left(\mathbf{E^T P}_k\mathbf{E}-\gamma^2\mathbf{I}\right)^{-1}\mathbf{E^T P}_k\mathbf{A}+\mathbf{B^T P}_k\mathbf{A}\right)x_k, \end{aligned} \tag{2.16}$$

portanto a política ótima para a realimentação de estados será

$$\begin{aligned} \mathbf{K}_u = &\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}-\mathbf{B^T P}_k\mathbf{E}\left(\mathbf{E^T P}_k\mathbf{E}-\gamma^2\mathbf{I}\right)^{-1}\mathbf{E^T P}_k\mathbf{B}\right)^{-1}\times \\ &\left(\mathbf{B^T P}_k\mathbf{E}\left(\mathbf{E^T P}_k\mathbf{E}-\gamma^2\mathbf{I}\right)^{-1}\mathbf{E^T P}_k\mathbf{A}-\mathbf{B^T P}_k\mathbf{A}\right). \end{aligned} \tag{2.17}$$

De forma similar a realimentação da pertubação de entrada ótima tem-se

$$\begin{aligned} w_k^* = &\left(\mathbf{E^T P}_k\mathbf{E}-\gamma^2\mathbf{I}-\mathbf{E^T P}_k\mathbf{B}-\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}\right)^{-1}\mathbf{B^T P}_k\mathbf{E}\right)^{-1}\times \\ &\left(\mathbf{B^T P}_k\mathbf{E}\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}\right)^{-1}\mathbf{B^T P}_k\mathbf{A}-\mathbf{E^T P}_k\mathbf{A}\right)x_k, \end{aligned} \tag{2.18}$$

dessa forma o ganho ótimo sobre a pertubação w será

$$\begin{aligned} \mathbf{K}_w = &\left(\mathbf{E^T P}_k\mathbf{E}-\gamma^2\mathbf{I}-\mathbf{E^T P}_k\mathbf{B}-\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}\right)^{-1}\mathbf{B^T P}_k\mathbf{E}\right)^{-1}\times \\ &\left(\mathbf{B^T P}_k\mathbf{E}\left(\mathbf{I}+\mathbf{B^T P}_k\mathbf{B}\right)^{-1}\mathbf{B^T P}_k\mathbf{A}-\mathbf{E^T P}_k\mathbf{A}\right). \end{aligned} \tag{2.19}$$

Observa-se que os ganhos encontrados são a solução ótima para a realimentação de estados em relação a política de controle e a realimentação da pertubação na planta. Tais ganhos são obtidos de maneira a relaciona o ganho ótimo com as medições realizadas nos sensores. A seguir é mostrada a relação entre a custo instantâneo e o custo de relaciono na função valor, ou seja, se a função valor $V^*(x_k) = x_k^T\mathbf{P}x_k$ satisfaz a equação de HJB. Portanto, escrevendo a Eq. (2.10) da seguinte maneira

$$\begin{aligned} &x_k^T\mathbf{P}_{k+1}x_k = \\ &x_k^T\mathbf{Q}x_k+u_k^{*T}\mathbf{R}u_k^*-\gamma^2w_k^{*T}w_k^*+\left(\mathbf{A}x_k+\mathbf{B}u_k^*+\mathbf{E}w_k^*\right)^{\mathbf{T}}\mathbf{P}_k\left(\mathbf{A}x_k+\mathbf{B}u_k^*+\mathbf{E}w_k^*\right). \end{aligned} \tag{2.20}$$

Desta maneira a pode se isolar a solução de **P**, obtendo-se portanto

$$\begin{aligned}\mathbf{P}_{k+1} &= \mathbf{Q} + \mathbf{K}_u^{\mathbf{T}}\mathbf{K}_u - \gamma^2\mathbf{K}_w^{\mathbf{T}}\mathbf{K}_w + (\mathbf{A}+\mathbf{B}\mathbf{K}_u+\mathbf{E}\mathbf{K}_w)^{\mathbf{T}}\mathbf{P}_k(\mathbf{A}+\mathbf{B}\mathbf{K}_u+\mathbf{E}\mathbf{K}_w) \\ &= \mathbf{Q} + \mathbf{K}_u{}^{\mathbf{T}}\mathbf{K}_u - \gamma^2\mathbf{K}_w^{\mathbf{T}}\mathbf{K}_w + \mathbf{A}_{mf}^{\mathbf{T}}\mathbf{P}_k\mathbf{A}_{mf},\end{aligned} \tag{2.21}$$

sendo $\mathbf{A}_{mf} = (\mathbf{A}+\mathbf{B}\mathbf{K}+\mathbf{E}\mathbf{K}_w)$ a matriz de malha fechada com a politica de controle ótima e realimentação de estados. A equação (2.21) representa a equação de Riccati em malha fechada.

Os problemas tratados ate são uma abordagem inicial para o desenvolvimento de sistemas de controle baseados em programação neurodinâmica. Desta maneira o projeto do sistema de controle consiste em determinar $V^*(x_k)$ afim de se obter a solução da equação de algébrica de Riccati, ou seja, "aprender" no decorrer do funcionamento, as politicas de controle e realimentação de pertubações ótimas para tais regiões de operação.

2.4 O Problema da Programação Dinâmica

A programação Dinâmica (PD) consiste na solução do problema, basicamente de otimização, sendo a solução o resultado da busca no espaço de soluções possível para cada estágio da decisão posterior. É interessante notar que tais estágios são desenvolvidos por meio da aplicação de um Processo de Decisão Markoviano (PDM), modelado pelas decisões e os custos que acarretas as mesmas para a próxima transição de estados.

Assim uma das grandes vantagens do uso de programação dinâmica é que esta pode ser aplicada a problemas envolvendo formulações estocásticas, pois a função de aprendizagem $Q(x,u,w,\omega)$ é usada como uma função de probabilidade $p(u,w|x,\omega)$.

Aplicando as formulações mostradas nas Seção anterior tem-se a proposta de um sistema de controle adaptativo, baseado em programação dinâmica, como exemplificado na Figura 2.2.

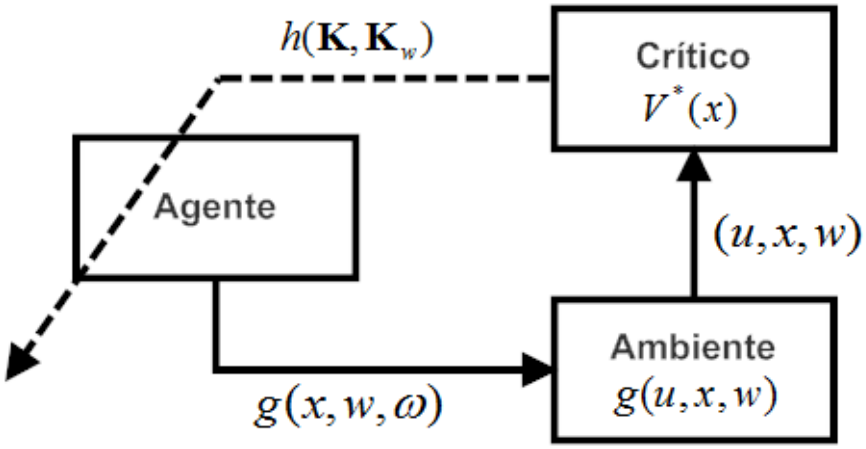


Figura 2.2: Diagrama de um Sistema de Controle, utilizando PD

Como bem observado, o sistema necessita de grande poder computacional para que os custos das transições estejam disponíveis para o agente, que é o subsistema responsável por implementar as políticas de controle ótimo. Para a solução de problema, que encontra uma política ótima de controle, há basicamente duas maneiras, por iteração de política e por iteração gulosa, detalhadas a seguir.

2.4.1 Iteração de Política

Neste método é providenciado um laço que alternada entre a avaliação e a melhoria de política. Primeiramente, o algoritmo avalia a política de controle g e obtém a função de valor correspondente que venha a satisfazer:

$$V_g(x) = r(x, g(x, w)) + V_g(f(x, g(x, w))). \tag{2.22}$$

A função mostrada na Eq.(2.22) é estimada de maneira a atualizar a política, através da melhoria da politica:

$$g(x) = \arg \min_u \max_w \left\{ r(x, g(x, w)) + V_g(f(x, g(x, w))) \right\}. \tag{2.23}$$

Estas duas medidas são aplicadas ao sistema ate haver a convergência. Em (BRADTKE, 1993) é mostrado que a convergência é verdadeira se as Eqs. (2.21) e (2.22) forem satisfeitas.

2.4.2 Iteração de Política Gulosa

Para o caso da Iteração de Política Gulosa, tanto a avaliação da política quanto a melhoria são executadas ao mesmo tempo. Desta maneira é necessário que exista uma avaliação de política inicial e um melhoria inicial, pois há interdependência nas Equações (2.22) e (2.23). Desta forma a política atual é considerada, para cada ciclo, a ser sempre a política ótima. A busca é feita utilizando-se técnicas iterativas e em cada iteração a parametrização da função valor é atualizada afim de se satisfazer a equação da otimalidade de Bellman o que resultara na atualização de $\hat{V}^*$ tendendo ao valor ótimo de V^*, logo

$$\begin{aligned} \hat{V}^*(x) &= r(x, \hat{g}^*(x,w)) + \hat{V}^*(f(x, \hat{g}^*(x,w))) \\ \hat{g}^*(x) &= \arg\min_u \max_w \left\{ r(x, g(x,w)) + \hat{V}^*(f(x, g(x,w))) \right\}. \end{aligned} \tag{2.24}$$

O método de iteração gulosa de política, a função valor estimada é tratada como ótima em relação a nova política estimada de de controle. Desta maneiras as funções da Eq. (2.24) são avaliadas na ordem inversa pois o foco do algoritmo é a política parametrizada $\hat{g}^*(x,w,\omega)$ em relação a parametrização ω e seus resultados induzirão a versão atualizada de $\hat{V}^*$.

2.5 Esquemas de Crítico Adaptativo

O problema de controle ótimo é resolvido de maneira a garantir a estabilidade do sistema em malha fechada, e as técnicas de soluciona-lo envolve a solução da realimentação ótima em relação ao parâmetros de uma dado modelo.

Os esquemas de crítico adaptativo, diferem desta solução usual pois tais técnicas utilizam da modelagem do função valor do sistemas e a recompensa posterior do sistema. Das muitas variações de esquemas de algoritmos de adaptativo critico presentes na literatura podemos destacar os métodos mais presentes neste campo, que são baseados em fatoração-TD e métodos baseados em em aprendizagem-Q (WATKINS, 1989). Tais métodos são usados em grande parte nas pesquisas que envolvem Aprendizagem por Reforço em sistemas de controle de processo e sistemas de robótica autônoma. Este métodos foram classificados por (WERBOS, 1992). A classificação se baseia principalmente na escolha da função de custo e na forma de parametriza-la. A Tabela 2.1 mostra os esquemas classificados e os respectivos modelos de críticos estimado para se obter a solução de Equação Algébrica de Riccati Discreta no tempo (DARE).

Tabela 2.1: Esquemas de Critico Adaptativo e os Modelos

Esquemas de Crítico Adaptativo		**Modelo**
HDP	Programação Dinâmica Heurística	$V(x)$
DHP	Programação Heuristica Dual	$\partial V/\partial x$
ADHDP	Programação Heuristica - Dependente de Ação	$Q(x,u)$
ADDHP	Programação Heuristica Dual - Dependente de Ação	$\partial Q/\partial x$ e $\partial Q/\partial u$

Os esquemas apresentados na Tabela 2.1 mostram as variáveis necessárias para a parametrização do alvo móvel do sistema para se obter a estimativa de $V(x)$. O Alvo móvel é necessariamente o custo instantâneo calculado para k, e desta forma, tem-se a solução **P**. A forma como é apresentada esse alvo, assim como a base no espaço de estados para obte-lo difere os esquemas e na capacidade e velocidade de convergência de cada. A seguir serão introduzidos os dois esquemas que foram utilizados nesse trabalho, um envolvendo a aproximação de V no tempo e o outro a aproximação da derivada de V em relação aos estados dos sistema.

Neste trabalho o esquema de projeto de critico adaptativo utilizado é a programação heurística dual. (LIN e YANG, 2007)(LENDARIS et al., 2001)(LEWIS e LIU, 2013)(POWELL, 2011)

2.5.1 Programação Dinâmica Heurística - Dual (DHP)

A programação dinâmica dual (DHP) é um método que tem como objetivo a estimação do gradiente do custo no estágio k para avaliar o desempenho das ações de controle em termos de satisfação das especificações de projeto, isto é, estimar o gradiente de função de valor para o crítico que é dada por

$$V_x = \frac{\partial V(x)}{\partial x}. \tag{2.25}$$

sendo V_x a função valor apresentada

$$V(x) = r(x,g(x)) + V(f(x),g(x)). \tag{2.26}$$

Derivando a Eq.(2.26) em relação a x, obtém-se a equação que viabiliza a determinação da aproximação da solução equação HJB. Esta equação é dada por

$$\frac{\partial V}{\partial x} = \frac{\partial r}{\partial x} + \frac{\partial r}{\partial g}\frac{\partial g}{\partial x} + \frac{\partial V}{\partial f}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}\right). \tag{2.27}$$

Para fins de realização da avaliação das ações, o problema de estimação da solução HJB utiliza as derivadas parciais da função de valor como amostras que de acordo como o lado direito da Eq.(2.27) são dadas por

$$r_x = \frac{\partial r}{\partial x}, \tag{2.28}$$
$$r_g = \frac{\partial r}{\partial g}, \tag{2.29}$$
$$g_x = \frac{\partial g}{\partial x}, \tag{2.30}$$
$$V_f = \frac{\partial V}{\partial f}, \tag{2.31}$$
$$f_x = \frac{\partial f}{\partial x}, \tag{2.32}$$
$$f_g = \frac{\partial f}{\partial g}. \tag{2.33}$$

Na formulação do problema é utilizada a notação representada pelas Eqs.(2.28-2.33). Consequentemente, o lado direito da Eq.(2.27) representa o alvo do algoritmo do gradiente que é aproximado por uma modelo parametrizado pela seguinte equação

$$d(r_x, r_g, f_x, f_g, f, w) = \frac{\partial r}{\partial x} + \frac{\partial r}{\partial g}\frac{\partial g}{\partial x} + \frac{\partial V}{\partial f}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}\right), \tag{2.34}$$

sendo w o vetor de parâmetros que minimiza o erro esperado no problema de alvo móvel dado por

$$d(r_x, r_g, f_x, f_g, f, w) = \arg\min_u E\left\{\left|V_x(x, w') - d(r_x, r_g, f_x, f_g, f, w)\right|^2\right\}. \tag{2.35}$$

O vetor da estimação $\frac{\partial V}{\partial x}$ é determinado por iteração de política ou iteração gulosa.

As parametrizações do vetor gradiente da função de custo em relação ao ator e ao critico são formulados para determinação on-line dos ganhos do sistemas de controle DLQR.

A formulação segue o padrão das definições do processo markoviano de decisão (MDP) que são estados, politica de controle, ambiente ou sistema dinâmico e função de utilidade. As parametrizações DLQR são formadas por modelos lineares e invariantes no tempo do sistema dinâmico, uma lei de controle que é uma combinação linear dos ganhos e dos estados, a função de utilidade que é definida pelo projetista e função de desempenho representa o custo processo para um dado horizonte de tempo. As parametrizações DLQR são dadas pelas Eq.(2.2) e (aplicando a um sistema linear) na Eq. (2.3) com os detalhes mostrados no Apêndice A.

$$r(x_k, u_k) = x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k \tag{2.36}$$

$$u_k(x_k) = \mathbf{K} x_k, \tag{2.37}$$

sendo $r(x,u)$ne $u(x)$ o custo instantâneo e a política de controle, respectivamente.

2.5.2 Política de Controle

O problema consiste na determinação dos parâmetros **P** da função de valor que traduz-se no rastreamento do alvo móvel. A lei de controle deve minimizar a função de custo. Deste forma, a lei de controle é obtida pela determinação do ponto extremo função de custo que é dada por

$$\frac{\partial V(x)}{\partial u} = 0. \tag{2.38}$$

A lei de controle ótimo u^* é obtida por meio da determinação do ponto extremo da função de valor em relação a u. Logo, a equação que permite a determinação do ponto extremo é dada por

$$\frac{\partial r}{\partial g} + \frac{\partial V}{\partial f}\frac{\partial f}{\partial g} = 0. \tag{2.39}$$

As derivadas parciais da Eq.(2.39) foram mostradas para a solução geral do problema DHP, nas Eqs. (2.17) e (2.19), aplicando-se desta forma a lei de controle para o regulador linear quadrático $u_k = \mathbf{K_k} x$.

2.5.3 O Arranjo do Problema de Estimação Paramétrica

As parametrizações da aproximação para o DLQR pelo método dos mínimos quadrados é dada por

$$2x^T\mathbf{P} \quad = \quad d_{alvo}, \tag{2.40}$$

sendo d_p as amostras dos estimador em função de P, conforme apresentado na Eq.(2.34). Consequentemente, a solução da Equação HJB é dada por

$$\mathbf{P} \quad = \quad (xx^T)^{-1}x \;\; d_{alvo}(r_x, r_g, f_x, f_g, f, P), \tag{2.41}$$

sendo as observações $d_p(r_x, r_g, f_x, f_g, f, P)$ dadas por

$$d_{alvo}(r_x, r_g, f_x, f_g, f, \mathbf{P}) = 2x^T\mathbf{P} + u^T\mathbf{RK} + x_{k+1}^T\mathbf{P}(\mathbf{A}+\mathbf{BK}). \tag{2.42}$$

Desta forma a partir da Eq. (2.42) é obtida a melhoria de política e a etapa do aprendizado aonde o ambiente sofre as consequências da tomada de decisão. Os atributos destas deduções serão melhores explorados no próximo capitulo a partir da obtenção de um núcleo de estimação online de parâmetros do problema nas equações (2.25) e (2.40).

2.5.4 Formulação para Estimação Paramétrica

O problema DHP é formulado como estimação paramétrica *online* do gradiente da função de custo em k. A estimação da solução da Equação HJB por meio do método dos mínimo quadrados é apresentada em duas etapas. Na primeira etapa, substitui-se a parametrização gradiente $\frac{\partial V(x)}{\partial x}$ no lado direito da Eq.(2.25) e lado esquerdo da Eq.(2.25), a seguir multiplica-se os lados esquerdo e direito pelo vetor de estado. Consequentemente, tem-se n problemas de mínimos de quadrados que são representados por

$$x^T\mathbf{P} = \begin{bmatrix} V_{x_1} & V_{x_2} & \dots & V_{x_n} \end{bmatrix}. \tag{2.43}$$

sendo **P** o vetor de parâmetros desconhecidos que representam a solução da equação HJB. No segundo passo, a Eq.(2.43) é multiplicada pelo vetor de estados que é dado por

$$xx^T\mathbf{P} \;=\; \frac{1}{2}x\begin{bmatrix} V_{x_1} & V_{x_2} & \dots & V_{x_n} \end{bmatrix}. \tag{2.44}$$

Substituindo os vetores de estado em termos de n componentes, obtém-se n sistemas de ordem n com $n(n+1)/2$ incógnitas que é dado por

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\begin{bmatrix} V_{x_1} & V_{x_2} & \dots & V_{x_n} \end{bmatrix}. \tag{2.45}$$

A forma do novo sistema é dada por

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \dots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} = \begin{bmatrix} x_1V_{x1} & x_1V_{x2} & \dots & x_1V_{xn} \\ x_2V_{x1} & x_2V_{x2} & \dots & x_2V_{xn} \\ \vdots & \vdots & \ddots & \vdots \\ x_nV_{x1} & x_nV_{x2} & \cdots & x_nV_{xn} \end{bmatrix}. \tag{2.46}$$

Desta maneira, tem-se n sistemas lineares, dados por

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}\begin{bmatrix} p_{1j} \\ p_{2j} \\ \vdots \\ p_{nj} \end{bmatrix} = \begin{bmatrix} x_1V_{xj} \\ x_2V_{xj} \\ \vdots \\ x_nV_{xj} \end{bmatrix}, \tag{2.47}$$

e os n-sistemas lineares são representados por

$$\begin{bmatrix} \mathbf{x}_{1n}^T p_1 & \mathbf{x}_{1n}^T p_2 & \dots & \mathbf{x}_{1n}^T p_n \\ \mathbf{x}_{2n}^T p_1 & \mathbf{x}_{2n}^T p_2 & \dots & \mathbf{x}_{2n}^T p_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{nn}^T p_1 & \mathbf{x}_{nn}^T p_2 & \cdots & \mathbf{x}_{nn}^T p_n \end{bmatrix} = \begin{bmatrix} x_1V_{x1} & x_1V_{x2} & \dots & x_1V_{xn} \\ x_2V_{x1} & x_2V_{x2} & \dots & x_2V_{xn} \\ \vdots & \vdots & \ddots & \vdots \\ x_nV_{x1} & x_nV_{x2} & \cdots & x_nV_{xn} \end{bmatrix}, \tag{2.48}$$

sendo $\mathbf{x}_{in} = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{in} \end{bmatrix}^T$, com $i = 1,\dots n$ and p_j com $j = 1,\dots n$ são as respectivas colunas de $\mathbf{P}$.

Os n-sistemas possuem n^2 variáveis desconhecidas e estas não estão presentes simultaneamente em todos os n sistemas lineares, mas estão distribuídas nos $n-1$ sistemas subsequentes. Desta a forma, a ordem do sistema de equações lineares pode ser reduzida. Os vetores desconhecidos e de observação dos $n-1$ sistemas são dados por

$$\begin{aligned}
p_2 &= \begin{bmatrix} p_{12} & p_{22} & \cdots & p_{n2} \end{bmatrix}^T \\
&\vdots \\
p_n &= \begin{bmatrix} p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix}^T \\
d_{x_2} &= \begin{bmatrix} x_1 V_{x_2} & x_2 V_{x_2} & \cdots & x_n V_{x_2} \end{bmatrix}^T \\
&\vdots \\
d_{x_n} &= \begin{bmatrix} x_1 V_{x_n} & x_2 V_{x_n} & \cdots & x_n V_{x_n} \end{bmatrix}^T.
\end{aligned} \tag{2.49}$$

2.5.5 Formulação via Gradiente de V_x

Das propriedades do produto de kronecker:

$$vec(XAY) = (Y^T \otimes X)vec(A) \tag{2.50}$$

é importante ressaltar que maiores informações sobre o produto de kronecker estão disponíveis no Apêndice B. Portanto tem-se o gradiente da função de custo

$$V(x) = x^T \mathbf{P} x = (x^T \otimes x^T)vec(\mathbf{P}), \tag{2.51}$$

contudo a Eq. (2.51) pode ser escrita da seguinte forma

$$\left(\frac{\partial V(x)}{\partial x}\right)^T = vec(2x^T \mathbf{P}\mathbf{P}) = 2(I \otimes x^T)vec(\mathbf{P}), \tag{2.52}$$

sendo ainda

$$x^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \tag{2.53}$$

e a matriz

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}, \tag{2.54}$$

portanto, tem-se a seguinte estrutura

$$\bar{x} \cdot vec(\mathbf{P}) = d_{\text{alvo}}. \tag{2.55}$$

Expandindo a Eq.(2.55) para o exemplo ilustrado de um sistema de terceira ordem tem-se

$$vec(2x^T\mathbf{PI}) = 2 \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \right) \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{12} \\ p_{22} \\ p_{32} \\ p_{13} \\ p_{23} \\ p_{33} \end{bmatrix}, \tag{2.56}$$

e aplicando desta forma o tensor de kronecker

$$vec(2x^T PI) = \begin{bmatrix} 2x_1 & 2x_2 & 2x_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2x_1 & 2x_2 & 2x_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2x_1 & 2x_2 & 2x_3 \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{12} \\ p_{22} \\ p_{32} \\ p_{13} \\ p_{23} \\ p_{33} \end{bmatrix}, \tag{2.57}$$

logo, desenvolvendo-se é possível obter

$$vec(2x^T PI) = \begin{bmatrix} 2x_1p_{11} + 2x_2p_{21} + 2x_3p_{31} \\ 2x_1p_{12} + 2x_2p_{22} + 2x_3p_{32} \\ 2x_1p_{13} + 2x_2p_{23} + 2x_3p_{33} \end{bmatrix}. \tag{2.58}$$

Agora o sistema de equações é reduzido, satisfazendo-se a condição de que $\mathbf{P} = \mathbf{P^T}$, logo

$$vec(2x^T PI) = \begin{bmatrix} 2x_1 & x_2 & x_3 & 0 & 0 & 0 \\ 0 & x_1 & 0 & 2x_2 & x_3 & 0 \\ 0 & 0 & x_1 & 0 & x_2 & 2x_3 \end{bmatrix} \begin{bmatrix} p_{11} \\ 2p_{12} \\ 2p_{13} \\ p_{22} \\ 2p_{23} \\ p_{33} \end{bmatrix}. \tag{2.59}$$

2.6 Considerações Finais do Capítulo

Neste capítulo apresentou-se os conceitos a respeito da aplicação de controle ótimo e aprendizagem por reforço. A conceitualização dos paradigmas apresentados aqui é de importância vital para compreensão da contribuição do trabalho. Desta maneira foram observado conceitos gerais de aprendizagem por reforço, suas formulações e os esquemas utilizados para a solução do problema de programação dinâmica aproximada. Os esquemas apresentados aqui serão melhor explorados no próximos capítulo onde os métodos de solução da parametrização dos mesmo são apresentados por formulações utilizando redes neurais artificiais e mínimos quadrados recursivo.

3

ALGORITMOS PARA CONTROLE ONLINE

Conteúdo

3.1 Introdução

Neste capítulo serão abordados os conceitos desenvolvidos para que o esquema de crítico adaptativo apresentado no Capítulo 2 seja implementado em algorítimos de controle aonde o sistema opera em aprendizagem por reforço, ou seja o algorítimo é adaptativo com uma lei de adaptação baseada em medições online e tomada de decisões que levam em conta os estado futuros do sistema. O objetivo principal é apresentar o núcleo do sistema de controle baseado na solução aproximada para o crítico adaptativo. O núcleo consiste do estimador paramétrico que determina a melhoria da política e controle quando o sistema esta no estado transitório ou perturbado.

No capítulo anterior pode-se ver que o sistema foi estruturado para que se implementasse o núcleo de controle constituído pelo crítico adaptativo. Neste aspecto são explorados sistemas para aproximação da função de custo tais como Mínimos Quadrados Recursivos (RLS) e aproximadores de funções de dimensão não linear, neste caso as Redes Neurais de Função de Base Radial. Um algorítimo baseado nestes dois métodos foi desenvolvido e será mostrado a seguir assim como os resultados e seu desempenho é observado no capítulo posterior.

3.2 Adaptativo Crítico: Concepção do Núcleo

O conceito que se explora aqui é de que a arquitetura de controle já foi definida para um sistema linear ou linearizado em faixas estreitas de operação e desta forma é desenvolvida uma metodologia de controle baseado em Regulação Linear Quadrática no tempo Discreto (DLQR). Desta forma a metodologia de controle é definida como linear e deixa algumas particularidades a se explorar no projeto que tornam o sistema em malha fechada com boas respostas a erros de modelagem como mudanças de parâmetros e a ruídos intrínsecos dos sistemas de controle. De maneira geral o desenvolvimento deste núcleo consistem no projeto de um sistema de tomada de decisão que seja capaz de fornecer uma solução ótima para um determinado sistema em um intervalo de tempo suficiente para que o sistema de controle não fique sub amostrado e se torne instável.Todos os conceitos estão portanto ligados a forma que os esquemas serão aplicados. Neste caso para se implementar o DLQR foi feito o arranjo a partir do esquema de critico adaptativo baseado em Programação Heurística Dual (DHP) como já citado, um método que torna a convergência mais rápida sem necessariamente adicionar instabilidade ao algoritmo.

O diagrama mostrado na Figura 3.1 ilustra as variáveis que os algoritmos desenvolvidos manipulam afim de se obter uma política de controle ótima.

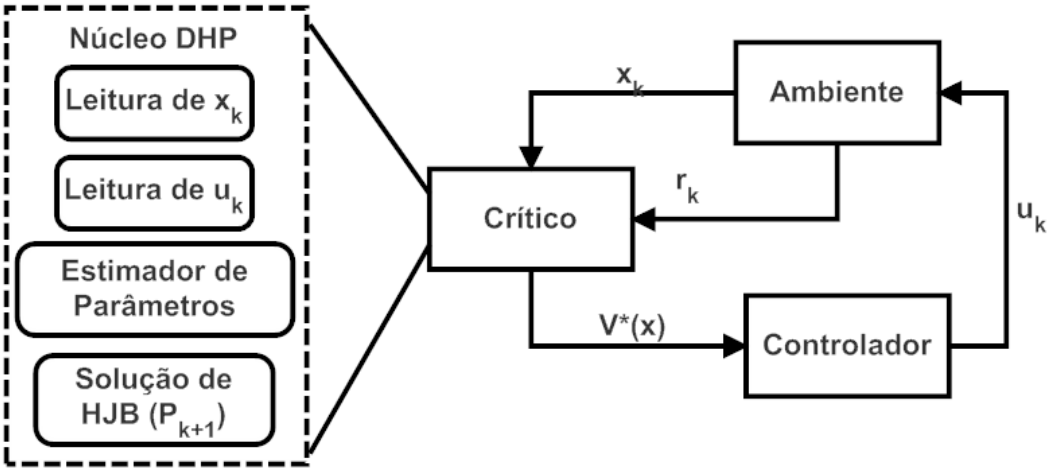


Figura 3.1: Diagrama do Núcleo do DHP, Variáveis Envolvidas na Tomada de Decisão

A Figura 3.1 mostra ainda nos detalhes do núcleo do crítico adaptativo as variáveis necessárias para a implementação do esquema DHP, observando, como mostrado no Capítulo 2 que o sistema interage com o ambiente e estabelece critérios para a tomada de decisão, desta forma assim operando como um algoritmo de aprendizagem por reforço.

3.3 Estimador de Mínimos Quadrados Recursivos (RLS)

O método dos mínimos quadrados tem sido utilizado desde sua formulação em 1692 por Gauss para a solução dos mais diversos problemas de engenharia e ciência. Apresenta-se aqui as principais equações do método RLS e suas variantes para resolver *on-line* para equação HJB. Detalhes a respeito da dedução da equações podem ser obtidos com mais detalhes em (ASTROM e WITTEMMAK, 1997).

Inicialmente, considera-se que o problema da aproximação da equação HJB para HDP dual pode ser formulado na forma de uma modulo Box-Jenkis (modelo de regressão) dada por

$$d^i_{\text{alvo}} \quad = \quad \varphi_i^T \theta_i, \tag{3.1}$$

sendo d^i_{alvo} , φ^T_i e θ_i o alvo móvel, transformada do vetor de estados (regressores) e os vetores desconhecidos que são dados por

$$
\begin{aligned}
d^i_{\text{alvo}} &= d_i, \\
\varphi^T_i &= \begin{bmatrix} \varphi_1 & \varphi_2 & \dots & \varphi_{n_\theta} \end{bmatrix}, \\
\theta_i &= \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_{n_\theta} \end{bmatrix}.
\end{aligned} \tag{3.2}
$$

A formulação da Eq.(2.40) é representada conforme a estrutura algébrica da Eq.(3.1). Desta forma, obtém-se a forma canônica da equação de regressão que viabiliza a determinação aproximada da solução HJB pelo conjunto de equações do método RLS que é dado por

$$\theta_{k+1} = \theta_k + \mathbf{K}^{RLS}_k \left(d_k - \varphi^T_k \theta_k \right), \tag{3.3}$$

que é a equação de atualização do vetor de parâmetros e

$$\mathbf{K}^{RLS}_{k+1} = \Gamma_k \varphi_k \left(\lambda + \varphi^T_k \Gamma_k \varphi_k \right)^{\mathbf{T}}, \tag{3.4}$$

é a equação de atualização do ganho do estimador e ainda a Eq.(3.5) para a atualização da matriz Γ de covariância do RLS.

$$\Gamma_{k+1} = \frac{1}{\lambda} \left(\mathbf{I} - \mathbf{K}^{RLS}_k \varphi^T_i \right) \Gamma_k, \tag{3.5}$$

sendo θ_i a variável que representa os parâmetros desconhecidos, isto é, as soluções da equação HJB. A variável Γ representa a matriz de covariância do processo de busca dos parâmetros θ_i, os valores de Γ_{k+1} no lado direito da Eq.(3.4) significa que foram atualizados pelo lado direito da Eq.(3.5), de maneira similar ocorre a atualização de $\mathbf{K}^{RLS}_k$. O valor da observação representa os componentes do vetor gradiente da função de custo $V(x,t)$ em relação ao estado, sendo d_i $\in R$.

Os métodos simplificados de *Kaczmarz* e da projeção são os métodos investigados como alternativa para formar o núcleo do elemento que atualiza o ganho do estimador RLS. A diferença entre estes métodos reside na forma da computação dos ganhos RLS da Eq.(3.4), a atualização do ganho no algoritmo de *Kaczmarz* é dada por

$$\mathbf{K}_k^{RLS} = \gamma_{RLS} \frac{\varphi_i}{\varphi_i^T \varphi_i}, \tag{3.6}$$

sendo $0 < \gamma_{RLS} < 2$ o parâmetro que regula o tamanho do passo de atualização das variáveis desconhecidas θ. O ganho algoritmo da projeção é dado por

$$\mathbf{K}_k^{RLS} = \gamma_{RLS} \frac{\varphi_i}{\alpha_{RLS} + \varphi_i^T \varphi_i}, \tag{3.7}$$

sendo $\alpha_{RLS} \geq 0$ o parâmetro utilizado para resolver o problema de *overflow* no algoritmo RLS que são oriundos de ocorrências de singularidades no produto $\varphi_i^T \varphi_i$ do denominador.

Desta forma, a Eq.(3.3) da variável desconhecida e a Eq.(3.4) da atualização do ganho são as equações utilizados para formar o núcleo do algoritmo RLS, a equação da atualização de Γ não é necessária para estes dois algoritmos. Estes dois algoritmos reduzem o esforço computacional na estimação dos parâmeros em relação ao método RLS com fator de esquecimento λ. Observa-se que α_{RLS} realiza o papel do parâmetro λ na equação do método RLS padrão, mas o seu efeito não é similar ao impacto provocado pelo parâmetro λ que é mais abrangente, produzindo impactos na atualização da matiz Γ.

A seguir será apresentado o algoritmo para implementação computacional do critico adaptativo **DHP-RLS-*online***.

3.4 Algoritmo RLS

Apresenta-se nesta seção a composição de uma algoritmo para implementação computacional de um sistema de controle com agente DLQR com realimentação de estados e pertubações e com um esquema de crítico adaptativo DHP, aonde as medições são realizadas a cada intervalo kT_s de tempo, sendo T_s a amostragem do ambiente. Observa-se no Algoritmo 1 o número de variáveis que o sistema baseado em aprendizagem por reforço necessita para para

a solução de HJB.

Algoritmo 1: Algoritmo para a solução RLS-DLQR-DHP-(*Online*)

Entrada: Medições (x,u,w), Inicializações $(P_0,Q,R,K_{u_0},K_{w_0})$

Saída: Solução de HJB: $(P \leftarrow \theta)$

1 **inicio**

    // Carrega Valores para Inicialização

2 (P_0,Q,R);

3 **repita**

        // Leitura das Variáveis de Entrada

4 Lê $\leftarrow (x,u,w)$;

5 Determina $(x_{k+1},w_{k+1}) \leftarrow (x,u,w,K_u,K_w)$;

6 Determina$(d_{alvo}) \leftarrow (P_k,Q,R,x,u)$;

        // Início do Estimador RLS

7 Determina $\varphi \leftarrow \frac{\partial x}{\partial \bar{x}}$

8 Determina $Y \leftarrow d_{alvo}$;

9 Determina $\mathbf{K}_k^{RLS}$ (Eq. (3.4));

10 Determina θ_{k+1} (Eq. (3.3));

11 Determina Γ_{k+1} (Eq. (3.5));

12 Determina $P_{k+1} \leftarrow matricializa(\theta_{k+1})$;

        // Melhoria de Política

13 Determina $K_u \leftarrow (P_{k+1}, f(x,u,w)$; (Eq. (2.17))

14 Determina $K_w \leftarrow P_{k+1}, f(x,u,w))$;(Eq. (2.19))

15 **até** *Fim do Aprendizado*;

Ainda no Algoritmo 1 podemos observar que as maiores demandas computacionais estão presentes no calculo de $\frac{\partial x}{\partial \bar{x}}$ (linha 7) e no alvo móvel d_{alvo} (linhas 6 e 8). Notas que o núcleo do RLS (Linhas 9 a 12) pode levar o sistema s instabilidade numérica, principalmente no que tange a matriz de covariância Γ, problema em discussão na comunidade e com avanço significantes em (REGO et al., 2013) e (da FONSECA NETO and REGO, 2014). Nestes termos notamos que as o sistema de aprendizagem por reforço lê as medidas do ambiente e precisa de dados do modelo somente na melhoria de política de controle (linhas 13 e 14).

3.5 Redes Função de Base Radial

Os métodos de regressão e aproximação de funções tem conquistado atenção de áreas como matemática, engenharia, geofísica, aprendizagem de máquina, mineração de dados entre outros. A teoria de aproximação de funções consiste em diminuirmos o erro entre uma função multivariada $f(.)$ e uma função aproximada $\hat{f}(x,\theta)$, dado um conjunto de parâmetros θ. Quando a função $\hat{f}(.)$ é escolhida o problema consiste então em se encontrar os parâmetros θ comparativo

As redes de função de base radial(RBF) são, de maneira geral, semelhantes as redes Perceptron Multicamadas (MLP) As redes RBF apresentam algumas diferenças em relação às redes MLP. A mais óbvia é que os neurônios da camada intermediária das redes RBF têm apenas funções de ativação de base radial, ao contrário dos MLP, que apresentam funções sigmoidais ou outras. As redes RBF sempre apresentam uma única camada intermediária, ao invés das múltiplas camadas do MLP. Os neurônios de saída das redes RBF são sempre lineares. Mas a principal diferença está na forma como as entradas são processadas pelos neurônios na camada intermediária. A ativação interna de cada neurônio é obtida a partir da norma euclidiana ponderada da diferença entre o vetor de entradas e o vetor de centros. No caso de uma RBF decrescente, por exemplo, quanto maior a distância entre a entrada e o centro, menor a ativação do neurônio. Nas redes MLPs, a ativação é dada pelo produto escalar entre o vetor de entradas e o vetor de pesos. Para o caso de aproximador de funções, especificamente um estimador paramétrico *online* a RBF se torna mais interessante para o problema de estimação paramétrica *online* pois a convergência é mais rápida apesar de uma número maior de parâmetros (se comparado um uma MLP de uma única camada escondida) e a capacidade de precisão da aproximação é maior pois as saídas da camada intermediaria são não lineares, tornado o ajuste de curva mair preciso para uma combinação linear dos neurônios da camada de saída. Trabalhos que exploram tais vantagens e desvantagens destas topologias podem ser analisados em (MASHOR, 2004), (DASH et al., 2010) e (PARK et al., 2002) e de forma mais concreta em (NISBET et al., 2009).

O uso das Redes de Funções de Base Radial para aproximação vem da possibilidade de utilizarmos uma determinada classe de funções por meio da combinação linear de de um conjunto de funções não lineares, as *funções de base* (de ALMEIDA REGO, 2010). O uso de RBF para aproximação de funções é denotado também pelo fato que a elevação da dimensão em relação as funções de base aumenta a possibilidade de classificarmos os pontos dentro de

uma determinada curva, como apresentado pelo teorema da separabilidade de Cover em 1965 e demonstrado por (HAYKIN, 1999) e (BISHOP, 1995). O teorema da separabilidade de Cover diz:

> *"Um problema complexo de classificação de padrões tratado não-linearmente em um espaço de dimensão alta é mais provável de ser linearmente separável que em um espaço de dimensão baixa."*

Para a concepção da rede as funções não lineares utilizadas aqui serão as funções de base radial definidas como um funcional $\Phi : \Re \rightarrow \Re^n$ é uma função radial se existir uma função univariada tal que

$$\Phi(x) = \varphi(r) = \|x - c\|, \tag{3.8}$$

sendo $\|.\|$ é uma norma euclidiana. Desta forma o valor de Φ é radialmente simétrico em relação a um centro c. Uma topologia geral para as redes de função de base radial é apresentada no diagrama da Figura 3.4:

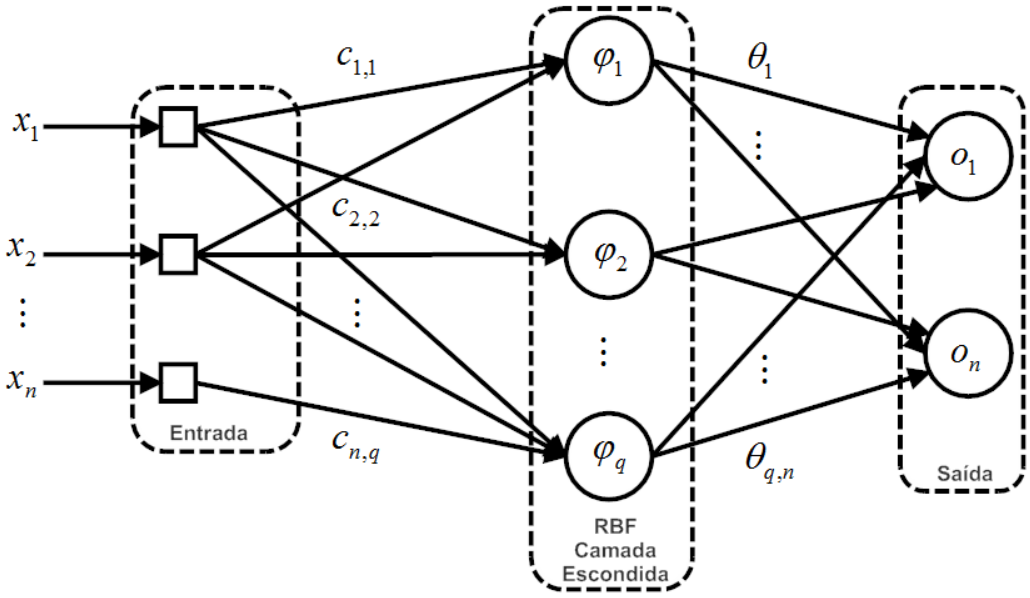


Figura 3.2: Topologia Geral de uma Rede RBF

É importante notar as variáveis envolvidas sendo os estados da entrada $x_1, x_2, \ldots x_n$, e $\varphi_1, \varphi_2, \ldots \varphi_q$ as funções de base e ainda $\theta_1, \theta_2, \ldots \theta_{n_\theta}$ os pesos da camada de saída, e para o

caso deste trabalho, os parâmetros da solução de HJB, sendo $o_1, o_2, \ldots o_n$ a saída multivariada estimada.Desta forma defini-se as funções de base gaussiana como

$$\varphi(r) = g(r) = e^{\left(-\frac{r^2}{2\sigma^2}\right)} = e^{\left(-\frac{\|x-c\|^2}{2\sigma^2}\right)}, \tag{3.9}$$

sendo r a distância euclidiana da entrada a um ponto centra c e σ a largura da função gaussiana como ilustrado na Figura 3.3.

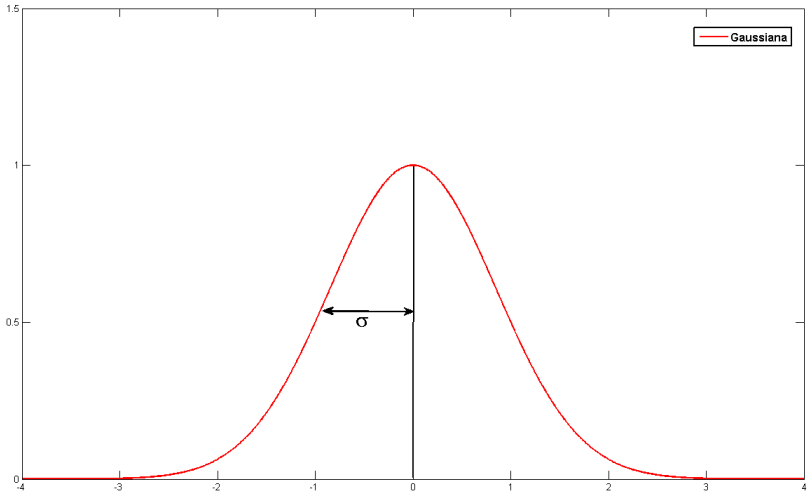


Figura 3.3: Exemplo de Função Gaussiana Unidimensional com um Centro c e espalhamento σ.

Considerando um modelo de regressão linear para se estimar uma função $\hat{f}$, é utilizado uma aproximação RBF da seguinte forma. O modelo de regressão:

$$f(.) \cong \hat{f}(x,\theta) = \theta_0 + \theta_1\varphi(x_1) + \theta_2\varphi(x_1) + \ldots + \theta_n\varphi(x_n), \tag{3.10}$$

sendo x e θ o vetor de entradas e o vetor de parâmetros da função a ser estimada. Agora o modelo é estendido para uma combinação linear de funções não lineares em relação as dados de entrada:

$$
\begin{aligned}
f_1 &= \theta_1 e^{\left(-\frac{\|x_1-c_1\|^2}{2\sigma^2}\right)} + \theta_2 e^{\left(-\frac{\|x_1-c_2\|^2}{2\sigma^2}\right)} + \cdots + \theta_n e^{\left(-\frac{\|x_1-c_q\|^2}{2\sigma^2}\right)} \\
f_2 &= \theta_1 e^{\left(-\frac{\|x_2-c_1\|^2}{2\sigma^2}\right)} + \theta_2 e^{\left(-\frac{\|x_2-c_2\|^2}{2\sigma^2}\right)} + \cdots + \theta_n e^{\left(-\frac{\|x_2-c_q\|^2}{2\sigma^2}\right)} \\
&\vdots \\
f_n &= \theta_1 e^{\left(-\frac{\|x_n-c_1\|^2}{2\sigma^2}\right)} + \theta_2 e^{\left(-\frac{\|x_n-c_2\|^2}{2\sigma^2}\right)} + \cdots + \theta_n e^{\left(-\frac{\|x_n-c_q\|^2}{2\sigma^2}\right)},
\end{aligned}
\tag{3.11}
$$

sendo $c = [c_1, c_2, \ldots c_n]$ o conjunto dos centro das funções de base, escolhidos de acordo com o comportamento dos estados de entrada, lembando sempre da necessidade de que c esteja em um ponto médio para englobar as variações de estados que ocorrem e formar matrizes de base não singulares.

Agora a Eq. (3.11) é rescrita como o seguinte conjunto de equações:

$$
\begin{cases}
g_{11}\theta_1 + g_{12}\theta_2 + \cdots + g_{1n}\theta_n = f_1 \\
g_{21}\theta_1 + g_{22}\theta_2 + \cdots + g_{2n}\theta_n = f_2 \\
\vdots \\
g_{n1}\theta_1 + g_{n2}\theta_2 + \cdots + g_{nn}\theta_n = f_n
\end{cases}
\tag{3.12}
$$

sendo g_{ij} as funções de base radial e w_n os parâmetro de ponderação de f. Agora a Eq.(3.12) é escrita na forma matricial:

$$
\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}
\begin{bmatrix}
g_{11} = e^{\left(-\frac{\|x_1-c_1\|^2}{2\sigma^2}\right)} & \cdots & g_{1n} = e^{\left(-\frac{\|x_1-c_n\|^2}{2\sigma^2}\right)} \\
\vdots & \vdots & \vdots \\
g_{n1} = e^{\left(-\frac{\|x_n-c_1\|^2}{2\sigma^2}\right)} & \cdots & g_{nn} = e^{\left(-\frac{\|x_n-c_n\|^2}{2\sigma^2}\right)}
\end{bmatrix}
\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix},
\tag{3.13}
$$

dessa forma, tem-se

$$
\begin{bmatrix}
g_{11} & g_{11} & \cdots & g_{1n} \\
g_{21} & g_{21} & \cdots & g_{2n} \\
\vdots & \vdots & \vdots & \vdots \\
g_{n1} & g_{n2} & \cdots & g_{nn}
\end{bmatrix}
\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}
=
\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix},
\tag{3.14}
$$

e de uma maneira compacta se obtém

$$\mathbf{G}\theta = f, \tag{3.15}$$

logo podemos obter os parâmetros estimados de seguinte forma

$$\mathbf{G}^{-1} f = \theta, \tag{3.16}$$

aplicando o teorema da pseudo inversa, para garantir que $\mathbf{G}$ seja inversível a seguinte relação é obtida

$$\theta_{k+1} = \left(\left(\mathbf{G_k}^{\mathbf{T}} \mathbf{G_k} + \lambda \mathbf{I} \right)^{-1} \mathbf{G_k}^{\mathbf{T}} \right) d_{alvo}, \tag{3.17}$$

considerando $0 < \lambda < 1$ que ao se aproximar de zero vira um problema de mínimos quadrados.Desta forma é formulado o estimador da solução de HJB utilizando-se o um aproximador baseado em redes de funções de base radial.

Desta maneira agora são determinados os parâmetros para um ciclo de aprendizado com um mapeamento baseado em funções de base radial. Agora as equações apresentadas anteriormente serão adaptadas para o algorítimo RBF-RLS, sendo d_{alvo} o algo móvel, ou seja o custo ótimo pra a transição de estados no momento k e θ o vetor de parâmetros da solução de HJB, e considerando a condição $\mathbf{P} = \mathbf{P}^{\mathbf{T}}$ o número de parâmetros é diminuído de $n \times n$ para $n \times (n+1)/2$. Para uma melhor visualização a rede RBF toma a topologia mostrada na Figura 3.4.

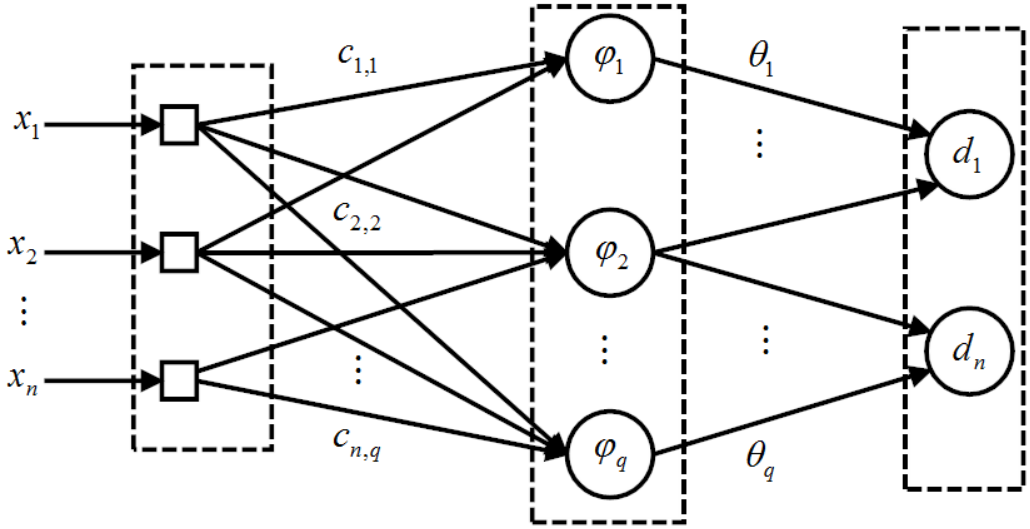


Figura 3.4: Arquitetura *online* da RBF

Ainda observando a nova topologia da rede RBF na Figura 3.4 nota-se que, para este caso, o número de neurônios da camada escondida, ou seja, a camada de funções de base radial é necessariamente igual ao quantidade de parâmetros estimados. Tal consideração acarreta a rede algumas consequências pois seu espaço não linear limitado pode reduzir o poder da aproximação da mesma ou mesmo dependendo do número de parâmetros esta camada pode esta super dimensionada. Um importante estágio do algoritmo é a determinação dos centros das funções de base, desta forma a determinação de g_{ij}, elementos da matriz da matriz $\mathbf{G}$ de funções de base. Desta forma esta etapa necessita de atualização de valores dos centros de forma *online* para que as funções gaussianas possam englobar no raio de atuação do mapeamento, as variações que ocorrem nos estados durante o aprendizado. Portanto assumi-se que os centros possuem valores médios para cada ciclo de aprendizado determinados por agrupamento, logo a regra de aprendizado para os centros sera

$$c_{k+1} = c_k + \eta \left(x_n - c_k \right), \tag{3.18}$$

sendo η limitada em $0 < \eta < 1$, uma constante de aprendizado para a atualização dos centros das funções de base. A seguir é apresentado o algoritmo RBF-DHP para a solução da equação de HJB, levando em conta os passos mostrados nesta seção e ainda observando que o sistema roda de forma *online*, ou seja, o algoritmo executa um ciclo de aprendizado, com excitação persistente para a convergência e o mesmo é aplicado ao sistema na forma de uma melhoria de política.

3.5.1 Algoritmo para Solução RBF-DHP

Apresenta-se nesta seção a composição de uma algoritmo para implementação computacional de um sistema de controle com agente DLQR com realimentação de estados e pertubações e com um esquema de crítico adaptativo DHP, aonde as medições são realizadas a cada intervalo kT_s de tempo, sendo T_s a amostragem do ambiente. Nesta variante o algoritmo de controle se difere do Algoritmo 2 em relação a base de x e a atualização θ_{k+1}. Observa-se ainda que no Algoritmo 2 o número de variáveis que o sistema baseado em aprendizagem por reforço necessita para para a solução de HJB em um determinado ciclo de aprendizagem. Nota-se que o sistema de aprendizagem é excitado por pertubações de estado e entrada para garantir a con-

vergência de **G** nas regiões de operações para o ciclo, produzindo desta forma uma solução sub ótima para o sistema de controle. .

Algoritmo 2: Algoritmo para a solução RBF-DLQR-DHP-(*Online*)

Entrada: Medições (x,u,w), Inicializações $(P_0,Q,R,K_{u_0},K_{w_0}),c_0,g_{ij_0}$

Saída: Solução de HJB: $(P \leftarrow \theta)$

1 **inicio**

`// Carrega Valores para Inicialização,` $q = n_\theta$

2 (P_0,Q,R,c_q,g_{nn});

3 **repita**

`// Leitura das Variáveis de Entrada`

4 Lê $\leftarrow (x,u,w)$;

5 Determina $(x_{k+1},w_{k+1}) \leftarrow (x,u,w,K_u,K_w)$;

6 Determina$(d_{alvo}) \leftarrow (P_k,Q,R,x,u)$;

`// Início do Estimador RBF`

7 Determina $Y \leftarrow d_{alvo}$; `// Determina as Centroides de` g_{ij}

8 **for** *i=1:q* **do**

9 Determina c_q (Eq. (3.18));

`// Determina os Elementos de` g_{ij}

10 **for** *i=1:q* **do**

11 **for** *j=1:n* **do**

12 Determina g_{ij} (Eq. (3.11));

13 Determina w_{k+1} (Eq. (3.17));

14 Determina $\theta_k \leftarrow w_{k+1}$;

15 Determina $P_{k+1} \leftarrow mattricializa(\theta_{k+1})$;

`// Melhoria de Política`

16 Determina $K_u \leftarrow (P_{k+1}, f(x,u,w)$ (Eq. (2.17));

17 Determina $K_w \leftarrow P_{k+1}, f(x,u,w)$(Eq. (2.19));

18 **até** *Fim do Aprendizado*;

O Algoritmo 2 demonstra a solução do problema HJB a partir de uma base do vetor de estados x em relação a funções gaussianas. Observa-se que o núcleo do estimador RBF atua das na determinação desta base de x (linhas 8 a 12). Nota-se ainda que existe um subnúcleo de

treinamento (linha 8) aonde a operação de atualização de c pela Eq. (3.18) ocorre na no mesmo ciclo em relação ao calculo da norma $||x-c||$ para se obter g_{ij} (linhas 10 a 12). Por fim as leis de melhoria de políticas são implementadas (linas 16 e 17) atravez das Eqs. (2.17) e (2.19).

3.6 Arquitetura Online RBF-RLS para DHP

Os algoritmos apresentados ate agora mostram a dedução para o esquema de crítico adaptativo baseado na derivada da função valor o DHP. Este esquema tem como principais vantagens maior velocidade de convergência e maior sensibilidade a mudanças na dinâmica da planta. Desvantagens são observadas no que se diz respeito a necessidade do algoritmo para a implementação da lei de controle, ou seja, neste estágio o sistema necessita de parâmetros da dinâmica da planta. Como será observado no Capítulo 5 tal entrave é vencido pela capacidade de adaptação e rápida convergência do algoritmo, pois na presença de incertezas estruturadas ou não estruturadas o sistema entra novamente no ciclo de aprendizado e para uma nova convergência para a solução da equação de Hamiton-Jacobi-Bellman.
O diagrama apresentado na Figura 3.5 mostra a topologia dos algoritmos apresentados levando-se em conta as variáveis dinâmicas e as constantes presentes na aprendizagem.

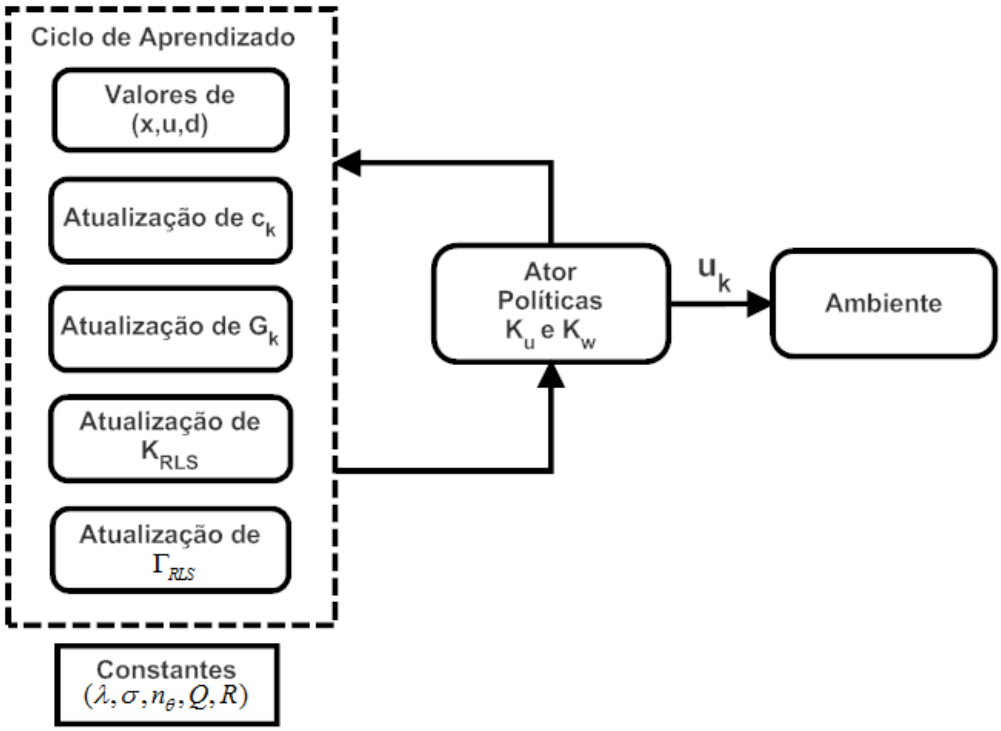


Figura 3.5: Arquitetura *online* do Sistema, Com enfase nas variáveis atualizadas. Constantes fazem parte do projeto, porém não são atualizadas no ciclos de aprendizagem

Ainda na análise do diagrama apresentado na Figura 3.5 observa-se que os parâmetros constantes do sistema são ou inerentes a planta, ou inerentes as características do controle via regulador linear quadrático no tempo discreto (DLQR). Os conceitos mostrado ate aqui exploram as características de convergência dos estimadores de mínimos quadrados recursivos (RLS) e a capacidade de robustez apresentada pelas rede neurais de funções de base radial.

No Algoritmo 3 é apresentado o sistema de aprendizado por reforço baseado em uma rede RBF introduzida no algoritmo RLS. A forma apresentada no Algoritmo 2 mostra a rede funcionando como um estimador de mínimos quadrados através da matriz **G** atuando como a regressora de funções de base. Portanto, ao se observar o Algoritmo 3 nota-se que a estrutura é basicamente a mesma do RLS, diferenciados-e necessariamente na formação da base do espaço de estados para a formação da matriz de regressores. Nota-se que o núcleo para a execução do aprendizado é implementado entre as linas (9 a 20), levando-se em conta a determinação da matriz de funções de base até a atualização da forma vetorizada da solução de HJB. O núcleo da RBF funciona na determinação da base g_{ij} (linhas 9 a 13).Para este núcleo são determinados os elementos das matriz **G** através da Eq. (3.11). Vale lembrar que para o caso do Algoritmo 3 este calculo determina a base φ para no instante *k* que assim como no caso anterior, o Algoritmo

2 é formada por funções de base centralizadas em valores médios c_q.

Portanto o algoritmo RLS permanece inalterado posteriormente (linhas 14 a 19) em relação ao Algoritmo 1. A atualização da solução de HBJ (linha 20) e as melhorias de política de realimentação de estados são mas mesmas definidas no Capítulo 3, lembrando que neste sub bloco (linhas 21 e 22) o algoritmo depende do modelo $f(x,u,w)$, para o caso linear **A**, **B** e **E**. Porém como citado anteriormente é parte intrínseca do esquema DHP a capacidade de obter uma nova solução para uma nova condição de operação. A convergência dos algoritmos se aplica aos Algoritmos 1 a 3 desde que a característica do esquema, ou seja a estimação através de $\partial V / \partial x$ e a solução seja dependente da otimização em relação a o estado atual, ainda levando em conta que $\gamma V_{k+1}(x,u,w)$ é determinado para o instantes k pelo cálculo de d_{target} (linha 6) e das melhorias de políticas, a convergência provada em (LANDELIUS, 1997), vale para tais resultados.

Algoritmo 3: Algoritmo para a solução RBF-RLS-DLQR-DHP-(*Online*)

Entrada: Medições (x,u,w), Inicializações $(P_0,Q,R,K_{u_0},K_{w_0}),c_0,g_{nn_0}$

Saída: Solução de HJB: $(P \leftarrow \theta)$

inicio

    // Carrega Valores para Inicialização, $q = n_\theta$

    (P_0,Q,R,c_q,g_{ij});

    repita

        // Leitura das Variáveis de Entrada

        Lê $\leftarrow (x,u,w)$;

        Determina $(x_{k+1},w_{k+1}) \leftarrow (x,u,w,K_u,K_w)$;

        Determina$(d_{alvo}) \leftarrow (P_k,Q,R,x,u,w)$;

        // Início do Estimador RBF-RLS

        Determina $Y \leftarrow d_{alvo}$;

        // Determina as Centroides de g_{ij}

        for *i=1:q* **do**

            Determina c_q (Eq. (3.18));

        // Determina os Elementos de g_{ij}

        for *i=1:q* **do**

            for *j=1:n* **do**

                Determina g_{ij} (Eq. (3.11));

        // Nucleo RLS

        Determina $\varphi \leftarrow g_{ij}$

        Determina $Y \leftarrow d_{target}$;

        Determina $\mathbf{K}_k^{RLS}$ (Eq. (3.4));

        Determina θ_{k+1} (Eq. (3.3));

        Determina Γ_{k+1} (Eq. (3.5)); Determina w_{k+1} (Eq. (3.17));

        Determina $\theta_k \leftarrow w_{k+1}$;

        Determina $P_{k+1} \leftarrow matricializa(\theta_{k+1})$;

        // Melhoria de Política

        Determina $K_u \leftarrow (P_{k+1}, f(x,u,w)$ (Eq. (2.17));

        Determina $K_w \leftarrow P_{k+1}, f(x,u,w)$(Eq. (2.19));

    até *Fim do Aprendizado*;

3.7 Considerações Finais do Capítulo

Neste Capítulo foram exploradas as concepções de algorítimos de aprendizagem para controle baseado em Regulador Linear Quadrático no tempo Discreto. É importante frisar que, como bem observado no início do Capítulo, o objeto de desenvolvimento, a contribuição, é justamente o núcleo inserido como estimadores paramétricos para controle adaptativo. Porém observa-se que a estratégia de controle utilizada busca sempre a solução ótima para a transição de estados em $k+1$. O objetivo preterido para a análise de performance dos algoritmos é justamente a estabilidade numérica, convergência paramétrica em tempo razoável e solução implementável, o que remete aos item citados anteriormente. Por fim é importante observar que foram apresentadas as equações para a solução dos esquemas DHP-RLS, DHP-RBF no tempo discreto. Desta forma no Capítulo 4 a seguir serão avaliados tais esquemas para que as conclusões apresentadas no Capítulo 5 sejam compatíveis com os objetivos primários apresentados no Capítulo 1. A convergência mostrada anteriormente é trabalhada no Apêndice C. Alguns aspectos sobre análise de complexidade computacional são apresentados no Apêndice D.

4

EXPERIMENTOS COMPUTACIONAIS

Conteúdo

4.1 Introdução

Neste capítulo serão discutidos os resultados dos algoritmos apresentados nos Capitulo 3 através de três estudos de caso. O sistemas foram escolhidos para possibilitar uma análise variada dos resultados assim como o comportamento dos algorítimos em diferentes ambientes numéricos. No primeiro caso serão explorados os resultados para um sistema de terceira ordem, o controle longitudinal pra para estabilidade lateral de um avião de caça do tipo F-16. Depois os resultados do controle de uma helicóptero de duas turbinas e três graus de liberdade é mostrado, levando em conta que a planta tem incertezas estruturadas e ainda é linearizada em um modelo de sexta ordem. Por fim é explorado o modelo de um processo industrial de décima ordem, um laminador a frio, modelo linear obtido no tempo discreto.

4.2 Avaliação dos Algoritmos

Os algoritmos desenvolvidos no Capítulo 3 foram portanto analisados e avaliados por meio de experimentos computacionais realizados de forma a se identificar três importantes parâmetros:

- Convergência paramétrica da solução de HJB;
- Estabilidade numérica e de malha fechada;
- Complexidade computacional e capacidade de implementação;

Portanto, é importante que seja estabelecida uma metodologia fixa de avaliação para o desenvolvimento dos algoritmos de controle, pois as características de simulação computacional e iteração com o ambiente "simulado" devem permanecer constantes para todos os casos testados para uma correta medição de performance dos algoritmos.

É apresentado na Figura 4.1 o esquema de implementação e avalização de algoritmos, elucidados em três diferentes níveis de abordagem.

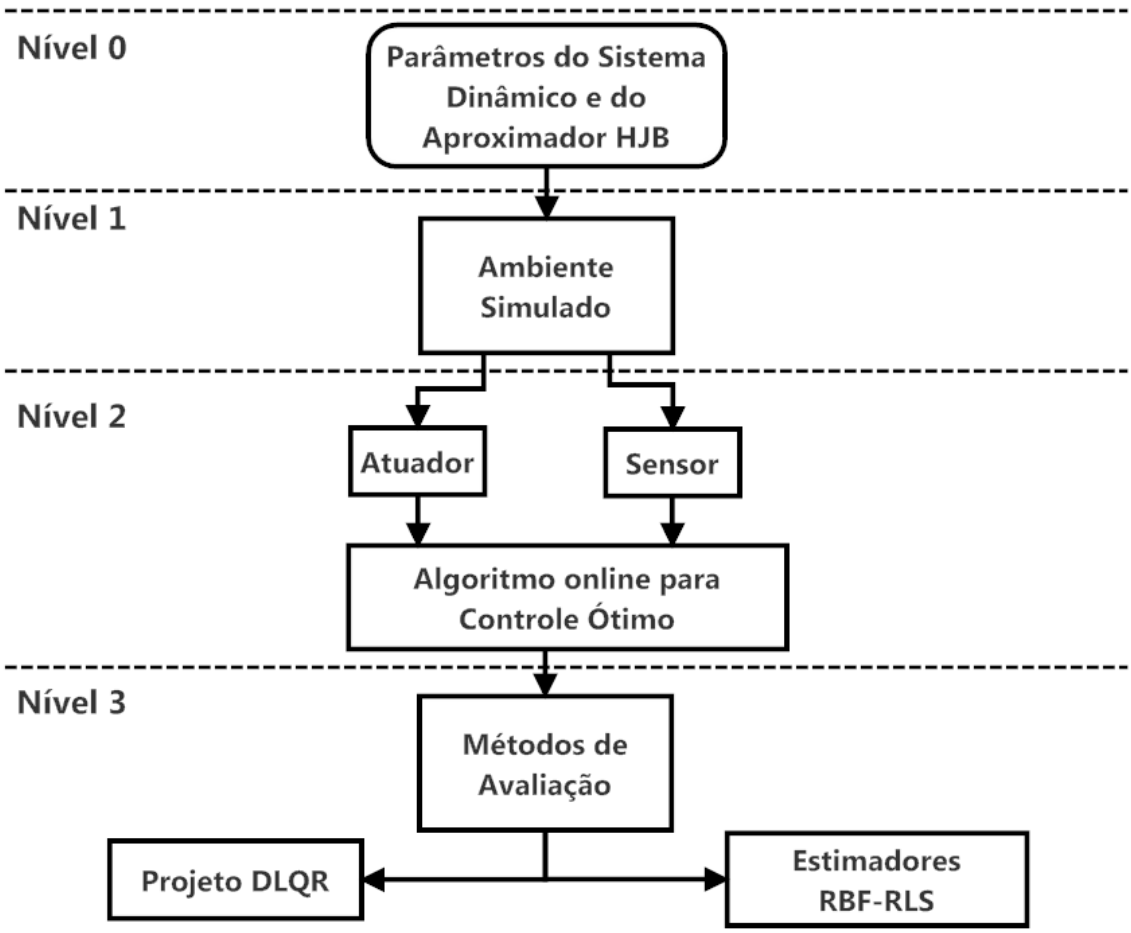


Figura 4.1: Diagrama para o Sistema de Avaliação de Altgoritmos.

Os níveis de abordagem observados na Figura 4.1 são necessários para que fique claro o contexto aonde esta inserida a estimação paramétrica online. Desta forma tem-se

Nivel 0 $\rightarrow$ Nível de inicialização dos parâmetros do estimador, parâmetros do modelo para emulação do ambiente;

Nivel 1 $\rightarrow$ Nível do ambiente simulado, implementação computacional dos modelos matemáticos e descrição da planta;

Nivel 2 $\rightarrow$ Nível do Sistema de controle online, abstração para dispositivos dedicados, e de tempo real;

Nivel 3 $\rightarrow$ Nível de avaliação de desempenho dos estimadores para o controle DLQR.

Os sistemas foram simulados levando-se em conta características de atuação no mundo real, tal como problemas de atuação , ou seja, restrições no sinal de controle, e ainda problemas se sensores tal como pertubações e sinais sub amostrados.

4.3 Estudo de Caso: Controle Longitudinal de um F-16

O modelo explorado neste experimento computacional é demonstrado com maiores detalhes em (STEVENS e LEWIS, 1992). O mesmo é um modelo linearizado da dinâmica longitudinal de um avião de caça do tipo F-10. As equações para o movimento longitudinal foram modificadas para que seja inclusa uma segunda entrada de controle tornando o modelo original em um sistema dinâmico MIMO de terceira ordem. Na Figura 4.2 são mostrados detalhes e abordados no trabalho de (MARCIO EDUARDO G. Silva, 2014)

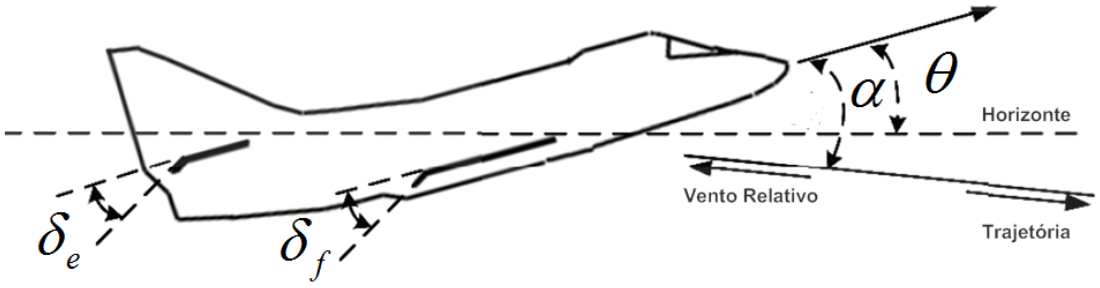


Figura 4.2: Diagrama de Corpo Livre do Avião de Caça F-16, Estabilidade Lateral (Provisório).

Portanto para o modelo tem-se as seguintes variáveis para o modelo longitudinal:

- A taxa de elevação q, ou seja a variação do angulo de elevação θ em relação a linha de horizonte, sendo $q = \dot{\theta}$;
- O angulo de ataque α, sendo a diferença entre a trajetória e o sentido atual do avião;
- O angulo de defecção do elevador δ_e.

As saídas do sistema são os próprios estados definidos por

$$x_k = \begin{bmatrix} \alpha & q & \delta_e \end{bmatrix}^{\mathbf{T}}, \tag{4.1}$$

e ainda os sinais de controle

$$u_k = \begin{bmatrix} \mu_1 & \mu_2 \end{bmatrix}, \tag{4.2}$$

sendo os mesmos a atuação em relação ao sistema de elevação determinado por δ_e e os *flaps* laterais das asas. Portanto os parâmetros do modelo dinâmico linearizado do sistema, discretizado para uma amostragem de cem milissegundos ($T_s = 100ms$), no espaço de estados serão

$$\mathbf{A} = \begin{bmatrix} 0.9124 & 0.0829 & -0.0007 \\ 0.3724 & 0.9428 & -0.0103 \\ 0.0000 & 0.0000 & 0.3679 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -0.0003 & 0.0427 \\ -0.0061 & 0.9682 \\ 0.6321 & 0.0000 \end{bmatrix}$$

e como os estados são as respectivas saídas tem-se

$$\mathbf{C} = \mathbf{I}$$

4.3.1 Inicialização da Simulação

A seguir é são mostrados os parâmetros de inicialização para os algoritmos do sistema. A Tabela 4.1 mostra os parâmetros necessários para a simulação do sistema assim como parâmetros para a inicialização do algorítimo de controle.

Tabela 4.1: Inicialização do Sistema de Controle e das Variáveis necessárias para a Simulação, para o Estudo de Caso I.

Parâmetros	Valores
N	500
λ	0.941
w_0	0.001
Q	$\begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$
R	$\begin{bmatrix} 0.1000 & 0.0000 \\ 0.0000 & 0.1000 \end{bmatrix}$
x_0	$\begin{bmatrix} 0.0300 & 1.2000 & 5.1000 \end{bmatrix}^T$

Sendo N o número de iterações, ou seja o ciclo de aprendizado mínimo para a convergência dos algoritmos testados, λ o fator de esquecimento para os algoritmos RBF e para o cálculo da pseudo inversa de **G**(foram utilizados os mesmos valores para que um algoritmo não tivesse seu aprendizado acelerado em relação ao outro), **Q** e **R** as matrizes de ponderação de estado e de controle, respectivamente, e ainda $\mathbf{P_0}$ a solução inicial de HJB (necessário para o calculo de u_1). A seguir os gráficos aonde são mostradas as evoluções de cada variável que interessa ao aprendizado do sistema.

4.3.2 Convergência do Parâmetros - Ciclo de Aprendizagem

Apresenta-se agora os gráficos para os experimentos computacionais realizados para os algoritmos desenvolvidos.

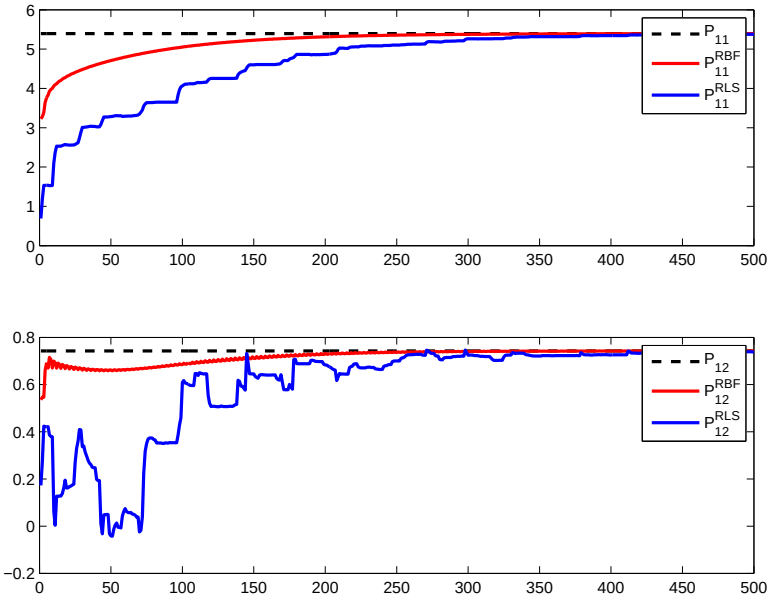


Figura 4.3: Evolução da solução de HJB para os Parâmetros P_{11} e P_{12}.

Na figura 4.3 é apresentada a evolução dos parâmetros P_{11} e P_{12} e sua convergência em relação ao tempo amostrado kT_s. Note que o sistema responde de forma satisfatória quando realizando a convergência numérica por volta de 450 iterações. A suavidade e a maior rapidez do algoritmo RBF-DHP pode ser notada. Lembando que a solução mostrada como RLS-DHP é equivalente para os Algoritmos 1 e 3.

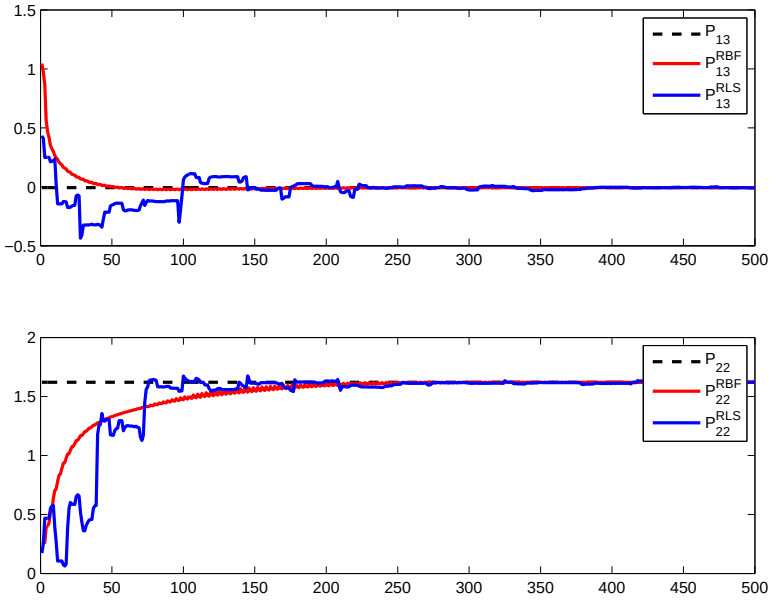


Figura 4.4: Evolução da solução de HJB para os Parâmetros P_{13} e P_{22}.

Nas Figuras 4.4 e 4.4 mostra-se os resultados para o sistema de terceira ordem em relação aos parâmetros de **P**, neste caso para P_{13} e P_{22} e ainda P_{23} e P_{33}, e ainda notando que a convergência de HJB por volta de 300 iterações.

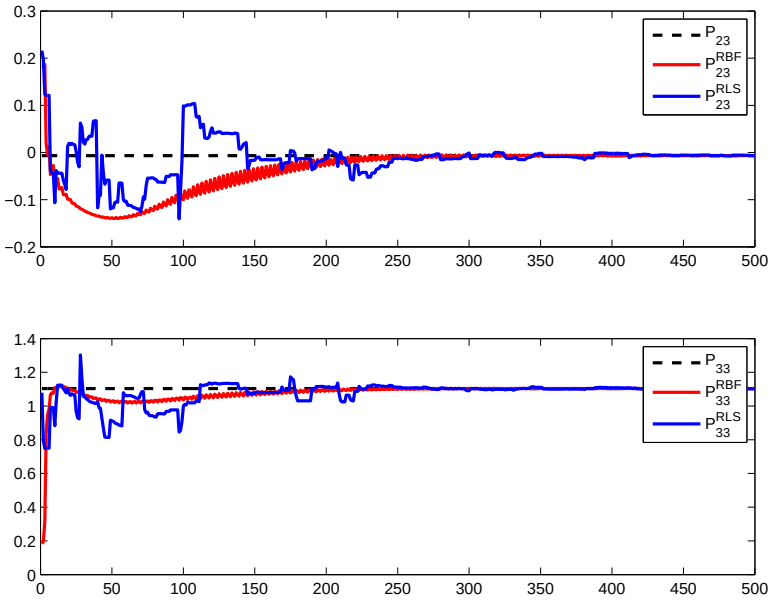


Figura 4.5: Evolução da solução de HJB para os Parâmetros P_{23} e P_{33}.

Na Figura 4.6 apresenta-se a observação e a estimativa do alvo móvel da solução HJB para RLS e RBF para o valor de λ dado como fator de esquecimento.

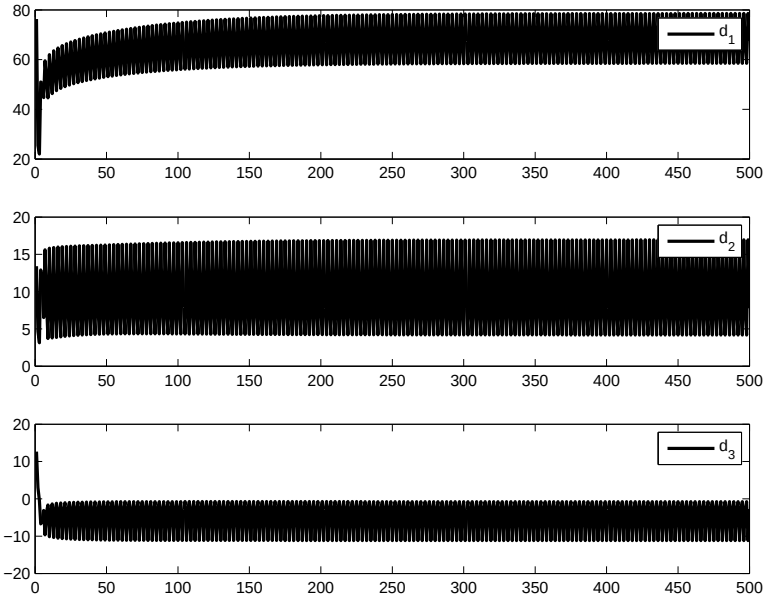


Figura 4.6: Estimação dos Alvos Móveis d_1, d_2 e d_3

Nota-se, ainda na Figura 4.6 que os e excitação com o ruído permanente no ciclo de aprendizagem produz o efeito no cálculo de d_{target}. Na Figura 4.7 podemos observar que o custo para o algoritmo RBF é maior para este sistema.

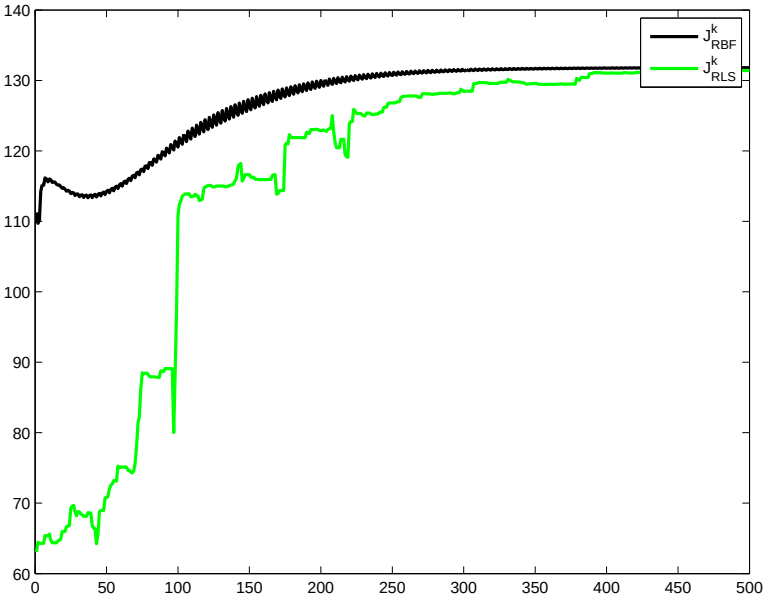
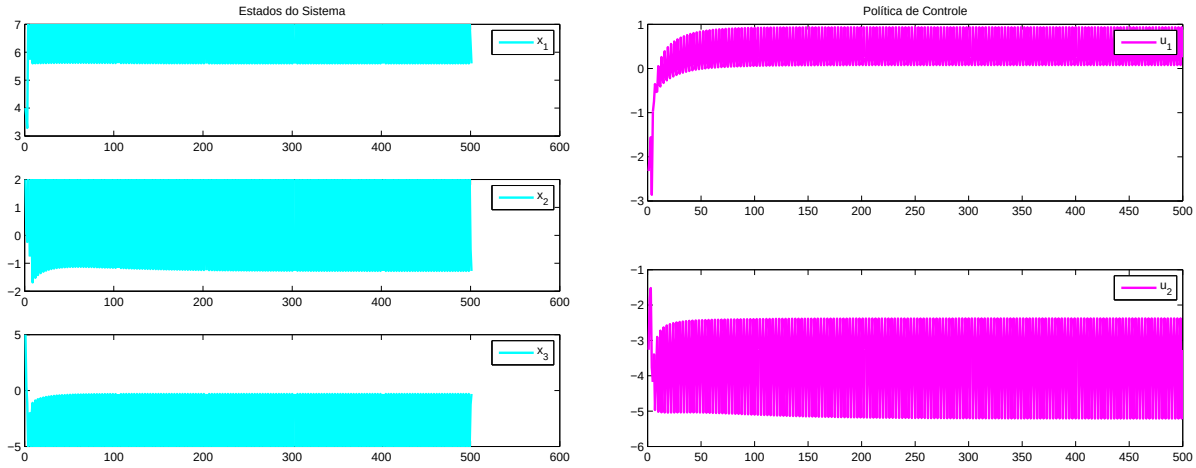


Figura 4.7: Custo para a iteração k da transição de estados

Na Figura 4.8(a) termos s evolução dos estados durante os ciclos de aprendizagem. É interessante observar que é aplicado as entradas do sistemas um ruído de média zero para que a excitação permanente do sistema torne o processo de aprendizagem mais robusto. Para os

estabelecimento dos mesmos critérios os estados são revitalizados a cada N_θ ciclos de amostragem, onde N_θ é o número de parâmetros a ser estimado.



(a) Estados (x_1, x_2, x_3) **(b)** Política de Controle $(u_1 e u_2)$

Figura 4.8: Comportamento dos Estados (x_1, x_2, x_3) e Política de Controle (u_1, u_2)

Já a Figura 4.8(b) trás os estados do sistema para o aprendizado e excitação de revitalização para uma dado conjunto de iterações.

Por fim a Figura 4.9 mostra a evolução dos autovalores de $(\mathbf{A} + \mathbf{BK_{DHP}})$ para o ciclo de aprendizado.

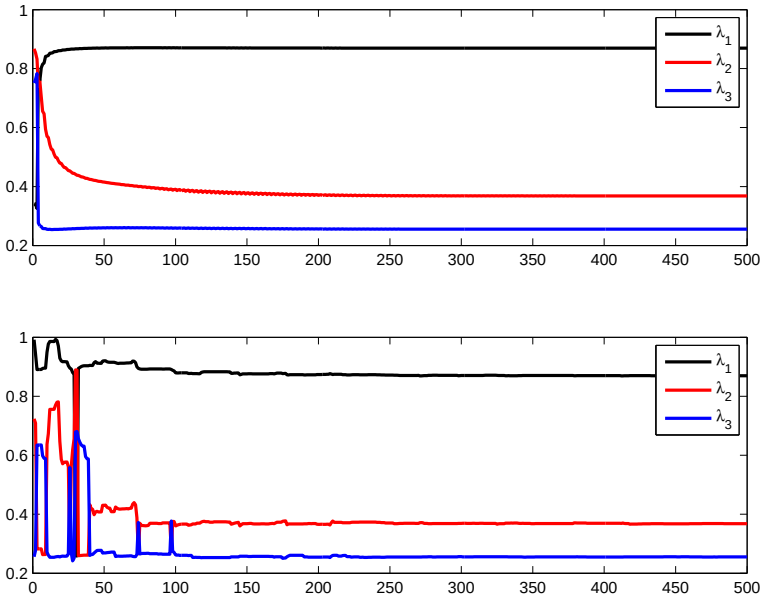


Figura 4.9: Evolução dos Autovalores de Malha fechada em $\mathbf{K_{DHP}}$ para o Algoritmo RBF e RLS, respectivamente.

Note ainda na Figura 4.9 que os autovalores de malha fechada são estáveis deste a iteração 50 do algoritmo, ou seja concluí-se que o sistema se estabiliza mais cedo para a solução ótima da HJB.Desta forma é possível observar que o sistema se mantem estável durante o ciclo de aprendizagem. É importante lembrar que

$$\mathbf{A}_{mf} = \mathbf{A} + \mathbf{B}\mathbf{K}_u + \mathbf{K}_w\mathbf{E}, \tag{4.3}$$

e

$$h\left(\mathbf{K}_u, \mathbf{K}_u\right) = f\left(\mathbf{G}_k^{RBF}\right), \tag{4.4}$$

ou ainda

$$h\left(\mathbf{K}_u, \mathbf{K}_u\right) = f\left(\mathbf{K}_k^{RLS}\right), \tag{4.5}$$

e desta forma a estabilidade de malha fechada do sistema de controle está intrinsecamente ligada a estabilidade numérica dos algorítimos de estimação paramétrica, tema abordado em (REGO et al., 2013).

4.4 Estudo de Caso: Controle de Helicóptero 3DOF

Nesta seção serão mostrados alguns aspectos relativos a planta, ou seja ao sistema de vôo baseado em um helicóptero com 3 graus de liberdade. Tal planta pertence a classe de sistemas não lineares sub-atuados pois possui três graus de liberdade e dois atuadores. O modelo da aeronave simula o comportamento típico de um helicóptero possibilitando assim a simulação do desempenho de sistemas de controle MIMO) viabilizando assim o teste do desempenho de controladores assim como a viabilidade de algumas técnicas.

O Helicóptero em questão é mostrado na Figura 4.10, sendo composto por quatro partes mecânicas triviais a sua dinâmicas que são: a base, o corpo, o braço principal e o braço secundário. O corpo do helicóptero é formado por uma pequena haste acoplado a uma dois motores de corrente contínua e hélices, sendo cada conjunto motor-hélice situado em uma extremidade da haste. A base fixa o equipamento a bancada. O braço principal conecta o helicóptero a base assim como possui um sistema de geração de ruído de massa, para a inserção de pertubações nos testes. Já este sistema de distúrbio consiste em uma massa presa em um parafuso, movimentado por um motor de corrente contínua, assim caracterizando uma pertubação de

ordem paramétrica. Já o braço secundário esta diretamente acoplado ao braço principal atuando como um contrapeso fazendo assim com que o esforço para se levantar vôo seja reduzido.

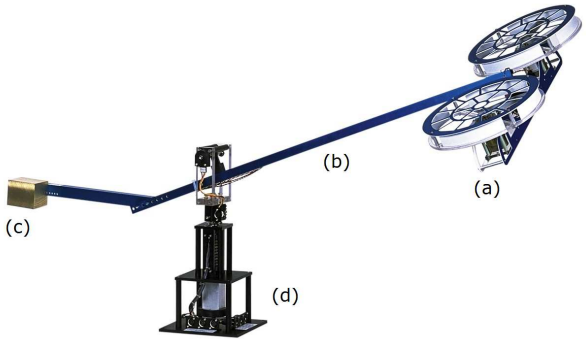


Figura 4.10: Partes Reais do Helicóptero com 3DOF (Quanser)

Pode-se identificar as partes citadas acima como: *a*- Corpo, *b* - Braço principal, *c* - Braço secundário, *d* - Base. O sistema do helicóptero é capaz de realizar três movimentos de rotação distintos através somente do controle dos dois atuadores, ou seja, através da tensão aplicada aos dois motores CC. Os três graus de liberdade mostrados permitem ao sistema a execução dos movimentos de elevação, arfagem e deslocamento.

O movimento de elevação consiste no movimento vertical do corpo do helicóptero, ou seja, corresponde a rotação do braço principal em torno do eixo horizontal. Para alterar a elevação do helicóptero, deve-se modificar as tensões fornecidas aos motores igualmente. O movimento de arfagem consiste no movimento de rotação do corpo do helicóptero em torno do eixo do braço principal. Mudanças no ângulo de arfagem são obtidas aplicando-se tensões diferentes a cada motor, proporcionando o desequilíbrio do sistema. O movimento de deslocamento consiste no movimento de rotação de todo o conjunto em torno do eixo vertical. Este movimento e influenciado pelo ângulo de arfagem, onde as forças geradas pelos motores se decompõem gerando uma força perpendicular ao braço secundário e proporcionando o movimento do helicóptero. Na Figura 4.11, mostra-se uma representação esquemática do movimentos do helicóptero com os seguintes ângulos de arfagem (P), elevação (E) e deslocamento (T).

As variáveis de controle são as tensões de entrada para os amplificadores de potência de cada um dos dois motores CC ligados nas hélices do helicóptero. A tensão máxima de entrada para os amplificadores e de $5V$. Três codificadores digitais permitem as medições dos ângulos de helicóptero. A resolução dos encoders e de aproximadamente de 0.44

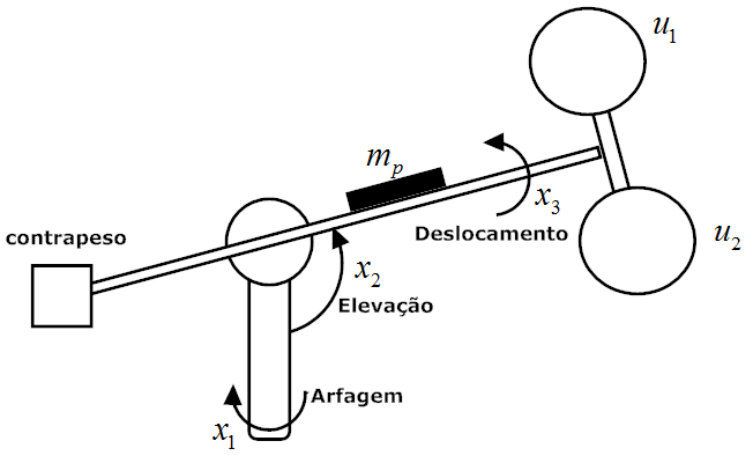


Figura 4.11: Diagrama de Corpo Livre do Sistema do Helicóptero 3DOF

4.4.1 O Modelo Dinâmico do Sistema

O modelo matemático do helicóptero 3DOF fabricado pela empresa Quanser Consulting foi obtido em vários trabalhos na literatura como em (APKARIAN e DAWES, 2000). Dentre os modelos propostos na representação da dinâmica do helicóptero escolheu-se o modelo apresentado na dissertação de (APKARIAN e DAWES, 2000) cujos resultados experimentais foram comparados com simulações no ambiente *MATLAB*. Este modelo possui características não lineares e de sexta ordem expressa pelo seguinte conjunto de equações diferenciais

$$
\begin{aligned}
&\dot{x}_1 = x_2 \\
&\dot{x}_2 = \xi_{16}\{\xi_1(u_1^2 - u_2^2) + \xi_2(u_1 - u_2) - 0.18x_2\} \\
&\dot{x}_3 = x_4 \\
&\dot{x}_4 = x_6^2\{\xi_3 \sin 2x_3 + \xi_4 \cos 2x_3\} + \xi_5 \sin x_3 + \\
&\xi_6 \cos x_3 + \cos x_1\{\xi_7(u_1^2 - u_2^2) + \xi_8(u_1 - u_2)\} \\
&\dot{x}_5 = x_6 \\
&\dot{x}_6 = \{\xi_{13} + \xi_{14} \sin 2x_3 + \xi_{15} \cos 2x_3\}^{-1}\{0.107 + 0.47x_6 \\
&+ [\xi_9(u_1^2 - u_2^2) + \xi_{10}(u_1 - u_2) \sin x_1] + x_4 x_6[\xi_{11} \sin 2x_3 + \xi_{12} \cos 2x_3]\}
\end{aligned}
\tag{4.6}
$$

sendo que x_1, x_3 e x_5 representam, respectivamente, os ângulos de arfagem, elevação e deslocamento em rad), x_2, x_4 e x_6 representam suas respectivas derivadas em rad/s), e u_1 e u_2 representam, respectivamente, as tensões de entrada dos amplificadores dos motores dianteiro e traseiro em Volts) do helicóptero.

Os demais parâmetros são constantes, relacionados com as dimensões físicas e mas-

sas dos diversos componentes do helicóptero, e os valores utilizados estão apresentados na Tabela 4.2 A convenção para a orientação do sentido positivo dos ângulos no modelo corresponde aquela indicada na Figura 4.11, que também assinala com linha pontilhada a posição em que os ângulos de arfagem e de elevação são nulos.

Tabela 4.2: Valores de Parâmetros

Parâmetro	***Valor***	***Unidade***
ξ_1	0.1117	N/V^2
ξ_2	0.044	N/V
ξ_3	-0.4843	rad
ξ_4	0.1153	rad
ξ_5	-1.038	$N/(m.kg)$
ξ_6	-1.3170	$N/(m.kg)$
ξ_7	0.0656	$N/(m.kg.V^2)$
ξ_8	0.0264	$N/(m.kg.V)$
ξ_9	-0.0718	$N.m/V^2$
ξ_10	-0.0289	$N.m/V$
ξ_11	1.0567	$kg\Delta m^2$
ξ_12	-0.2515	$kg\Delta m^2$
ξ_13	0.5454	$kg.m^2$
ξ_14	0.1258	$kg.m^2$
ξ_15	0.5283	$kg.m^2$
ξ_16	4.1832	$(m.kg)^{(-1)}$

4.4.1.A Linearização do Modelo

O modelo a ser utilizado no experimento computacional foi obtido mediante a linearização em torno de um ponto de equilíbrio. Tomando os parâmetros $\xi_1,\ldots,\xi_{16}$ e linearizando-se o modelo não-linear descrito pelas Equações diferenciais do modelos mostradas anteriormente torno dos seguintes pontos de equilíbrio ($x_{eq}=0$,$u_{1eq}=u_{2eq}=2.9735V$) o equilíbrio ocorre com o helicóptero na horizontal. Temos então as seguintes matrizes do modelo lineari-

zado

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.7521 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1.0389 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1.3428 & 0 & 0 & 0 & 0 & -0.4700 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 2.9633 & -2.9633 \\ 0 & 0 \\ 0.4168 & 0.4168 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Como explicado anteriormente, as variáveis de saídas são x_1, x_3 e x_5, que representam os ângulos de arfagem, elevação e deslocamento, respectivamente. Dessa forma, a matriz C é definida como:

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

Vale ressaltar ainda que, como não ha transmissão direta entre a entrada e a saída da planta,logo teremos a matriz $D = 0$ Além disso, tem se que os autovalores da matriz A são: $p_1 = 0 + 1.0193i$, $p_3 = p_4 = 0$,$p_5 = 0.4700$ e $p_6 = -0.7521$ que caracterizam a dinâmica descrita pelo modelo linearizado como marginalmente estável e de fase mínima.

4.4.2 Inicialização do Sistema

A seguir é são mostrados os parâmetros de inicialização para os algoritmos do sistema. A Tabela 4.3 mostra os parâmetros necessários para a simulação do sistema assim como parâmetros para a inicialização do algorítimo de controle.

Tabela 4.3: Inicialização do Sistema de Controle e das Variáveis necessárias para a Simulação, para o Estudo de Caso II.

Parâmetros	Valores
N	700
λ	0.972
w_0	0.001
Q	$\mathbf{I}_{6x6}$
R	$0.1x\mathbf{I}_{2x2}$
$\mathbf{P_0}$	$10x\mathbf{I}_{6x6}$

Sendo N o número de iterações, ou seja o ciclo de aprendizado mínimo para a convergência dos algoritmos testados, λ o fator de esquecimento para os algoritmos RBF e para o cálculo da pseudo inversa de **G**(foram utilizados os mesmos valores para que um algoritmo não tivesse seu aprendizado acelerado em relação ao outro), **Q** e **R** as matrizes de ponderação de estado e de controle, respectivamente, e ainda $\mathbf{P_0}$ a solução inicial de HJB (necessário para o calculo da política inicial u_1). A seguir os gráficos aonde são mostradas as evoluções de cada variável que interessa ao aprendizado do sistema.

4.4.3 Convergência do Parâmetros - Ciclo de Aprendizagem

Aplicando os resultados para os algorítimos apresentados no Capítulo 3 Segue na Figura com os parâmetros de inicialização apresentados na Tabela 4.3 tem-se portanto os resultados, para este caso, com um sistema tendo uma quantidade de 21 parâmetros somente na triangular superior, apresenta-se somente os elementos da diagonal.

Na figura 4.12 é apresentada a evolução dos parâmetros P_{11} e P_{22} e sua convergência em relação ao tempo amostrado kT_s. Note que o sistema responde de forma satisfatória quando realizando a convergência numérica por volta de 500 iterações. A suavidade e a maior rapidez do algoritmo RBF-DHP pode ser notada. Lembando que a solução mostrada como RLS-DHP é equivalente para os Algoritmos 1 e 3.

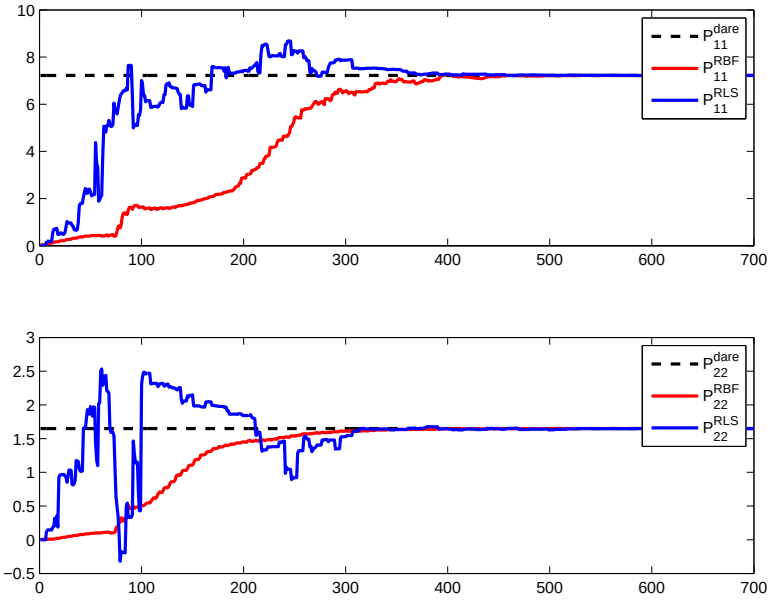


Figura 4.12: Evolução da solução de HJB para os Parâmetros P_{11} e P_{22}.

E Ainda o segundo lote de parâmetros da diagonal principal na Figura 4.13

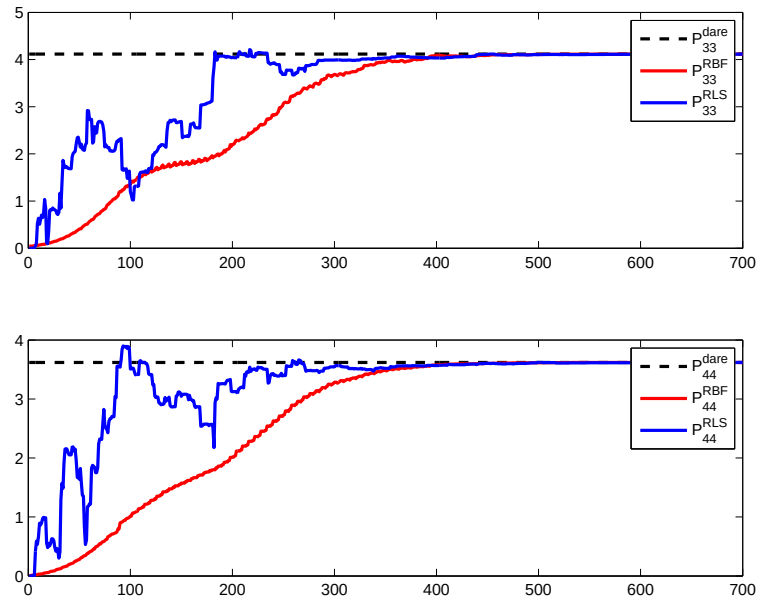


Figura 4.13: Evolução da solução de HJB para os Parâmetros P_{33} e P_{44}.

Por fim o terceiro lote de parâmetros da diagonal principal na Figura 4.14

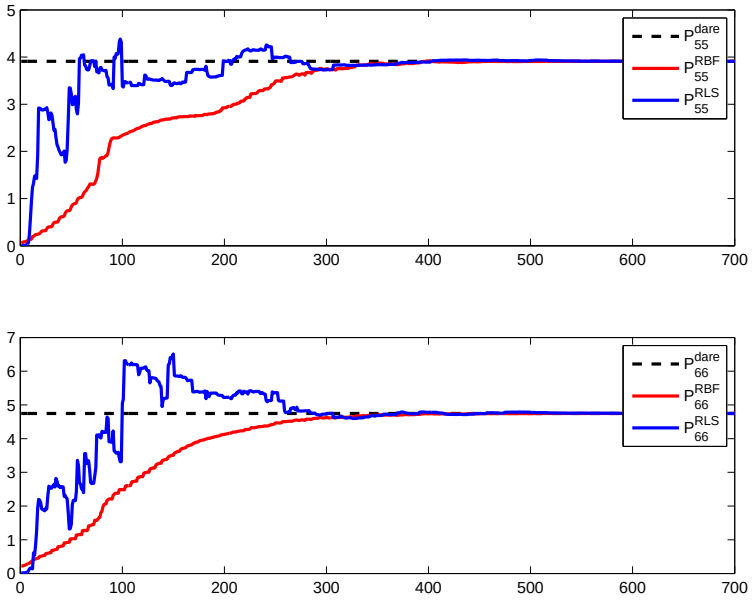


Figura 4.14: Evolução da solução de HJB para os Parâmetros P_{55} e P_{66}.

Na Figura 4.15 tem-se o custo instantâneo para o aprendizado. Observa-se que, para este caso, o sistema tem o mesmo custo instantâneo a partir da iteração 500, levando em conta a maior quantidades de estados pode-se concluir que a transição de estados para ambos algoritmos é menos custosa e mais estável para este sistema.

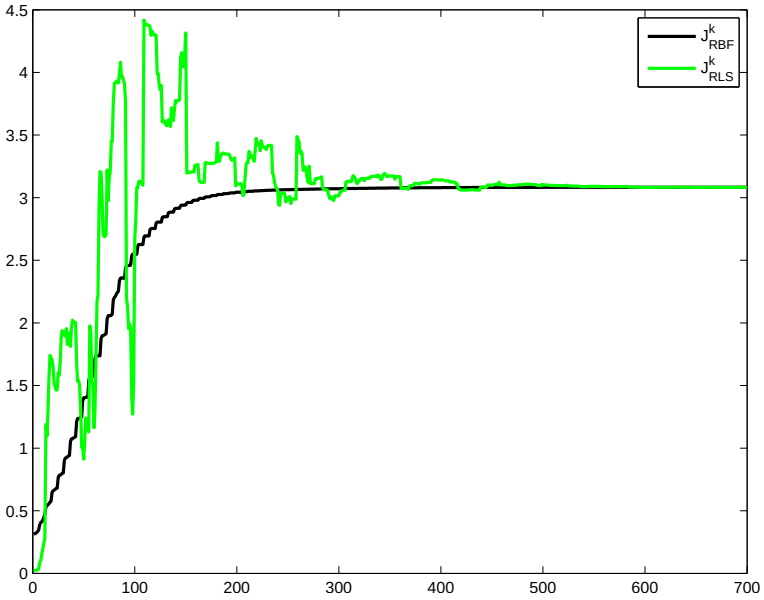
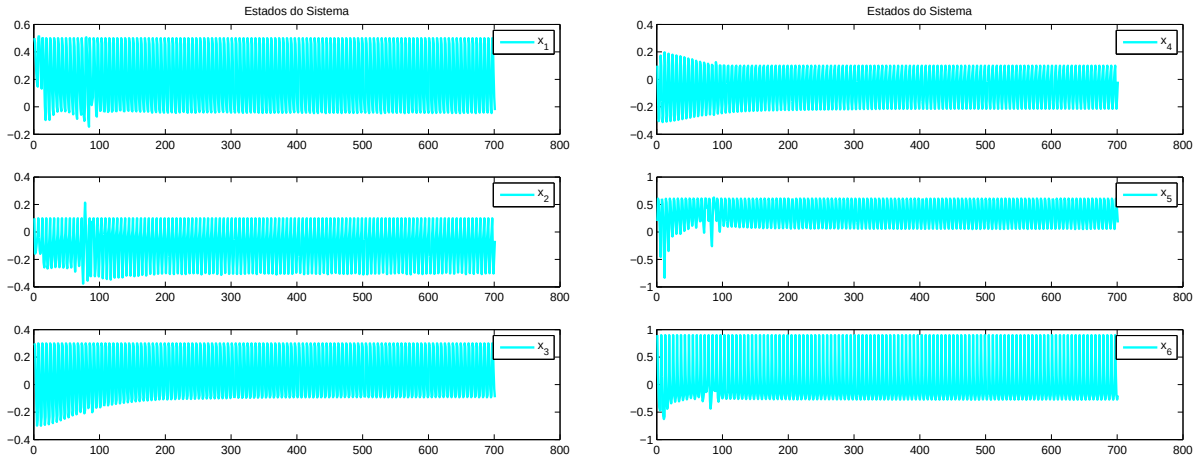


Figura 4.15: Custo Instantâneo para o aprendizado

Nas Figuras 4.16(a) e 4.16(b) tem-se o comportamento dos sinais dos estados durante os ciclos de aprendizado e levando em conta as revitalização com ruídos a cada conjunto

de iterações:



(a) Estados (x_1, x_2, x_3) **(b)** Estados (x_4, x_5, x_6)

Figura 4.16: Comportamento dos Estados $(x_1, x_2, x_3, x_4, x_5, x_6)$.

E por fim, nas Figuras 4.17 e 4.18 tem-se a evolução do alvo móvel em relação a cada estado do sistema:

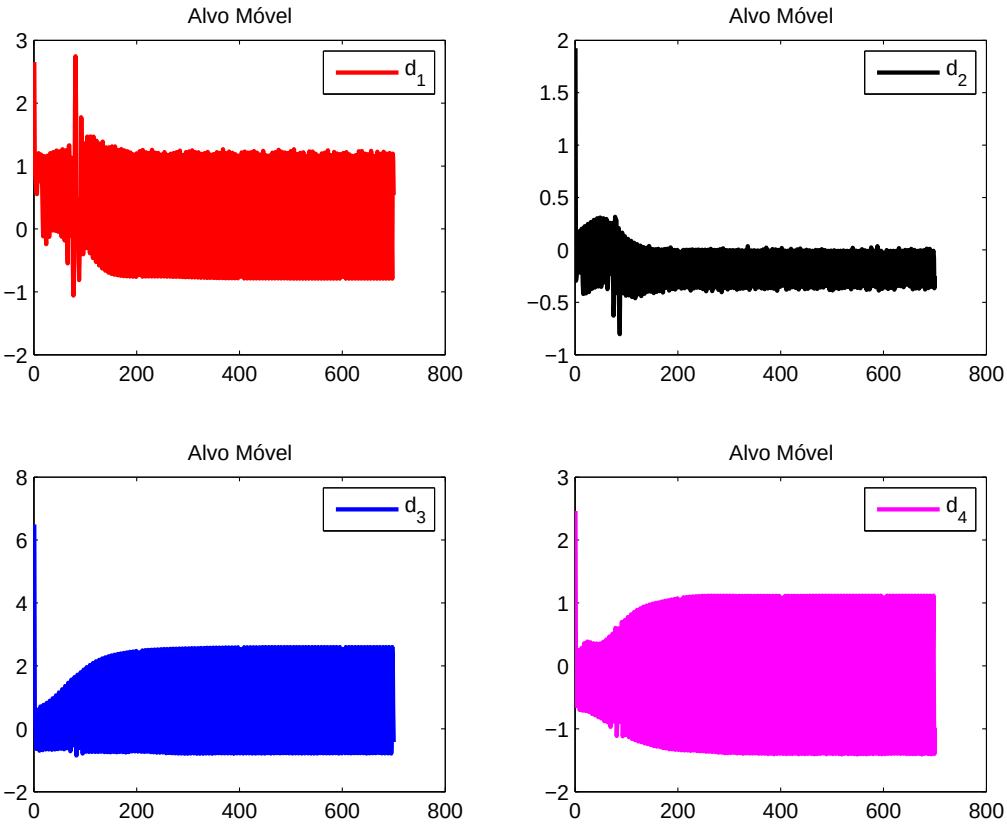


Figura 4.17: Evolução do Alvo Móvel (d_1, d_2, d_3, d_4) para Cada Estado.

e para o restante dos respectivos d_{target}

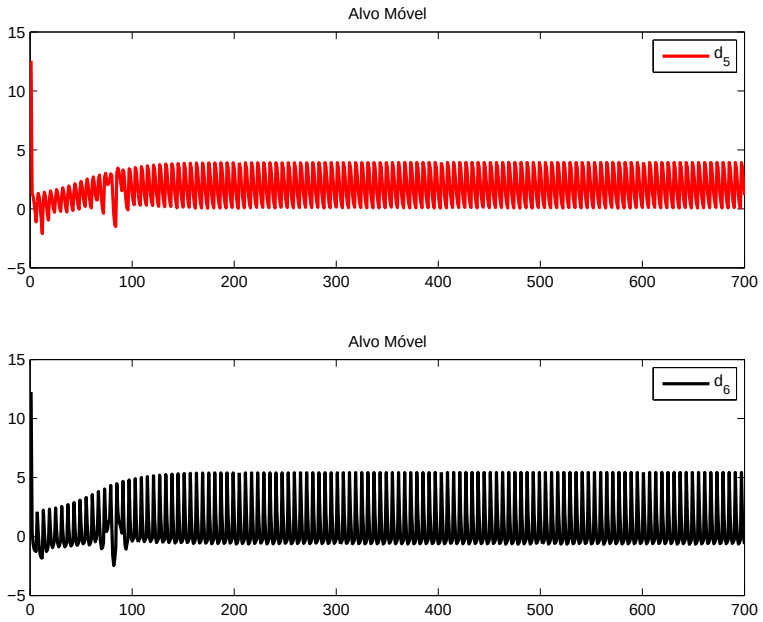


Figura 4.18: Evolução do Alvo Móvel para Cada Estado.

A Figura 4.19 mostra a evolução dos autovalores λ_n para o ciclo de convergência.

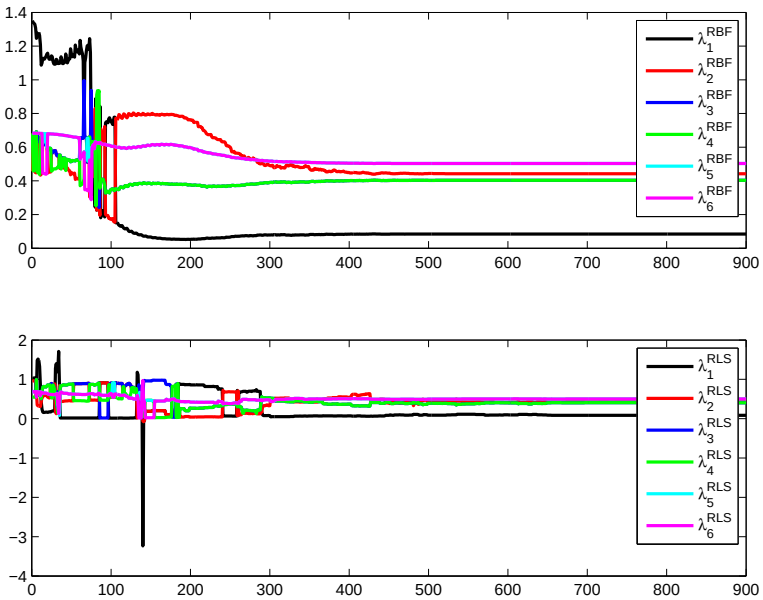


Figura 4.19: Evolução dos Autovalores de Malha fechada em $\mathbf{K_{DHP}}$ para o Algoritmo RBF e RLS, respectivamente.

Note ainda que na Figura 4.19 os autovalores se tornam menos oscilantes em relação ao período em que os algoritmos estão aproximando a solução ótima de **P**.

4.5 Estudo de Caso: Sistema de Controle para o Processo de Laminação a Frio

O processo de laminação a frio consiste de um sistema de duas etapas onde o produto do sistema é a altura fina da lâmina de saída. O resfriamento da lâmina ocorre ao final do segundo estágio para se obter um rolo de metal com a altura h_o desejada. O problema consistem em uma Prensa de Laminação a Frio (CRM) em dois estágios. Trata-se de um modelo dinâmico discreto de décima ordem, dominado pelo atraso de transporte com com uma alta interatividade das variáveis. O metal é aquecido anteriormente e no processo de laminação é resfriada para se obter a chapa de metal na espessura desejada, com maiores detalhes em (SMITH, 1969). Este sistema possui 3 entradas e 3 saídas e está sujeito a 3 entradas de perturbação (sendo 2 não mensuráveis). Existe, no processos restrições de erro da altura da lâmina assim como velocidades limites para a entrada da lâmina afim de se garantir a continuidade do rolo.

A Figura 4.20 mostra a descrição detalhada do processo para velocidades v_i e v_o de entrada e saída da lâmina respectivamente. O processo ocorre com a presença de uma força F nos estágios S_1 e S_2 do laminador.

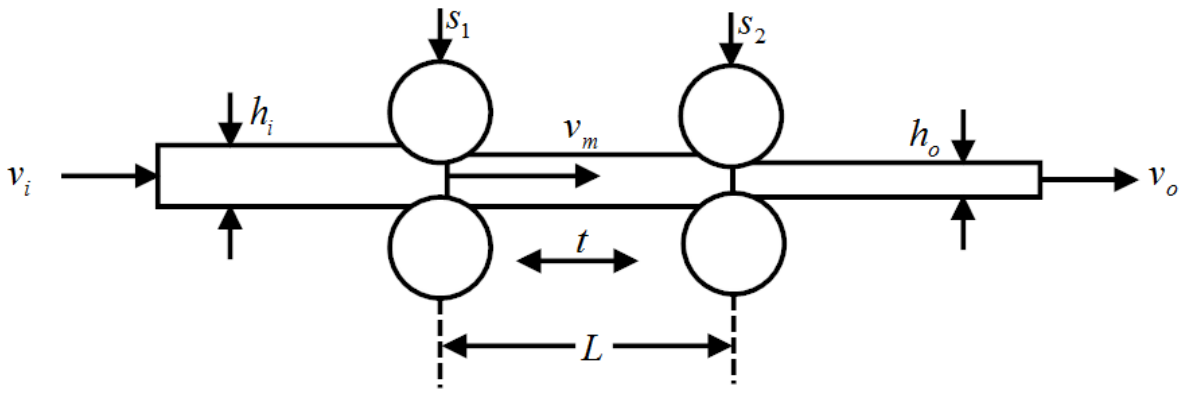


Figura 4.20: Diagrama do Processo de Laminação

Na Figura 4.20 temos os diagrama de corpo da planta. A chapa entra na compressão com uma altura h_i e velocidade v_i. Restrições estão também relacionadas com a tensão que a chapa pode sofrer no estágio intermediário. As Matrizes da dinâmica do sistema são:

$$\mathbf{A} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.1120 \\ 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 \end{bmatrix}$$

e ainda

$$\mathbf{B} = \begin{bmatrix} 2.7600 & 1.3500 & - 0.4600 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

4.5.1 Inicialização do Sistema

A seguir é são mostrados os parâmetros de inicialização para os algoritmos do sistema. A Tabela 4.4 mostra os parâmetros necessários para a simulação do sistema assim como parâmetros para a inicialização do algorítimo de controle.

Tabela 4.4: Inicialização do Sistema de Controle e das Variáveis necessárias para a Simulação, Estudo de Caso II.

Parâmetros	Valores
N	1200
λ	0.9403
w_0	0.1
$\mathbf{Q}$	$\mathbf{I}_{10x10}$
$\mathbf{R}$	$0.1x\mathbf{I}_{3x3}$
$\mathbf{P_0}$	$10x\mathbf{I}_{10x10}$

Sendo N o número de iterações, ou seja o ciclo de aprendizado mínimo para a convergência dos algoritmos testados, λ o fator de esquecimento para os algoritmos RBF e para o cálculo da pseudo inversa de $\mathbf{G}$(foram utilizados os mesmos valores para que um algoritmo não tivesse seu aprendizado acelerado em relação ao outro), $\mathbf{Q}$ e $\mathbf{R}$ as matrizes de ponderação de estado e de controle, respectivamente, e ainda $\mathbf{P_0}$ a solução inicial de HJB (necessário para o calculo da política inicial u_1). A seguir os gráficos aonde são mostradas as evoluções de cada variável que interessa ao aprendizado do sistema.

4.5.2 Convergência do Parâmetros - Ciclo de Aprendizagem

Aplicando os resultados para os algorítimos apresentados no Capítulo 3 Segue na Figura com os parâmetros de inicialização apresentados na Tabela 4.3 tem-se portanto os resultados, para este caso, com um sistema tendo uma quantidade de 55 parâmetros somente na triangular superior, apresenta-se somente os elementos da diagonal.

Na figura 4.21 é apresentada a evolução dos parâmetros P_{11} e P_{12} e sua convergência em relação ao tempo amostrado kT_s. Note que o sistema responde de forma satisfatória quando realizando a convergência numérica por volta de 150 iterações. A suavidade e a maior rapidez do algoritmo RBF-DHP pode ser notada. Lembando que a solução mostrada como RLS-DHP é equivalente para os Algoritmos 1 e 3.

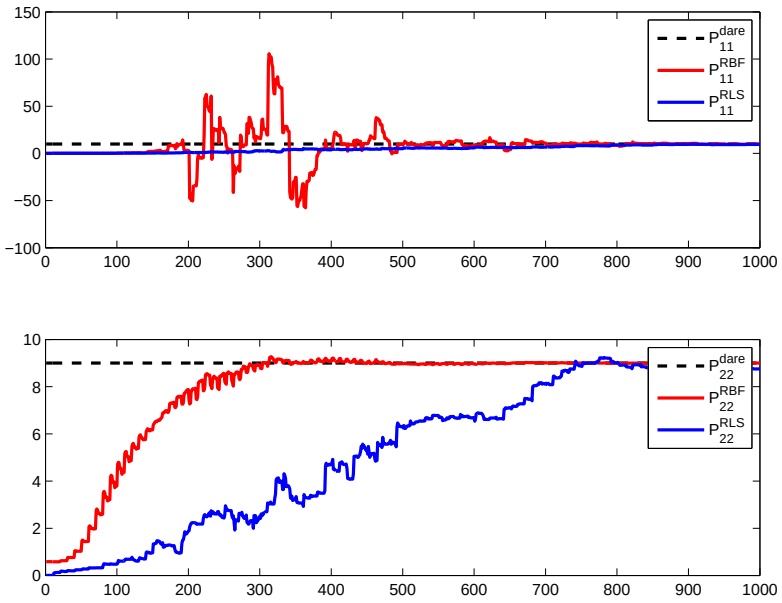


Figura 4.21: Evolução da solução de HJB para os Parâmetros P_{11} e P_{22}.

A Figura 4.22 apresenta s convergência dos parâmetros da diagonal P_{33} e P_{44}.

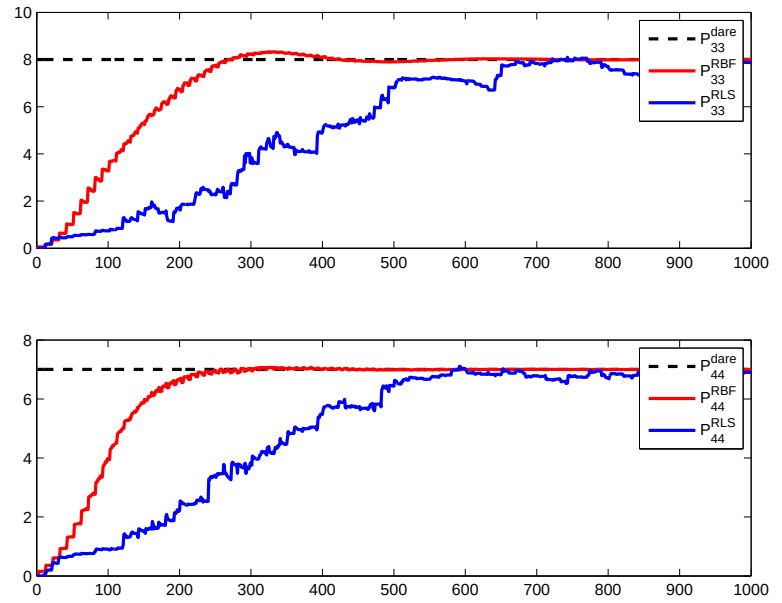


Figura 4.22: Evolução da solução de HJB para os Parâmetros P_{33} e P_{44}.

A Figura 4.23 apresenta s convergência dos parâmetros da diagonal P_{55} e P_{66}.

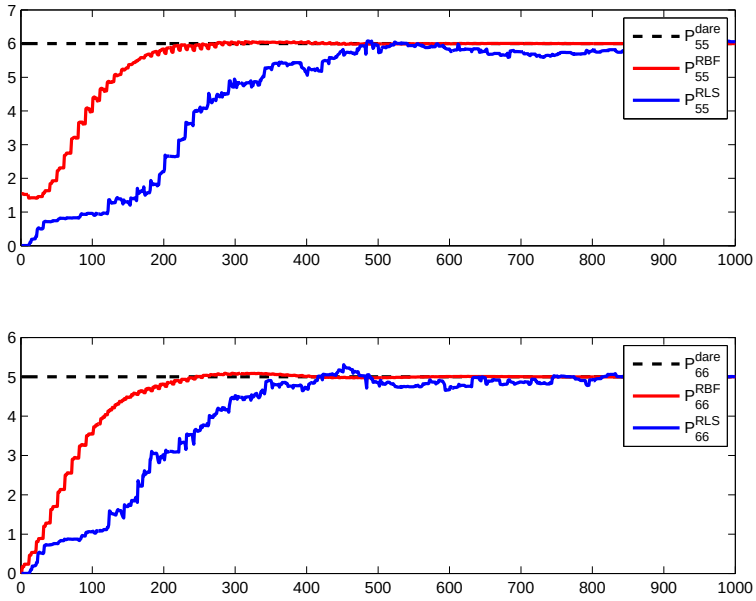


Figura 4.23: Evolução da solução de HJB para os Parâmetros P_{55} e P_{66}.

A Figura 4.24 apresenta s convergência dos parâmetros da diagonal P_{77} e P_{88}.

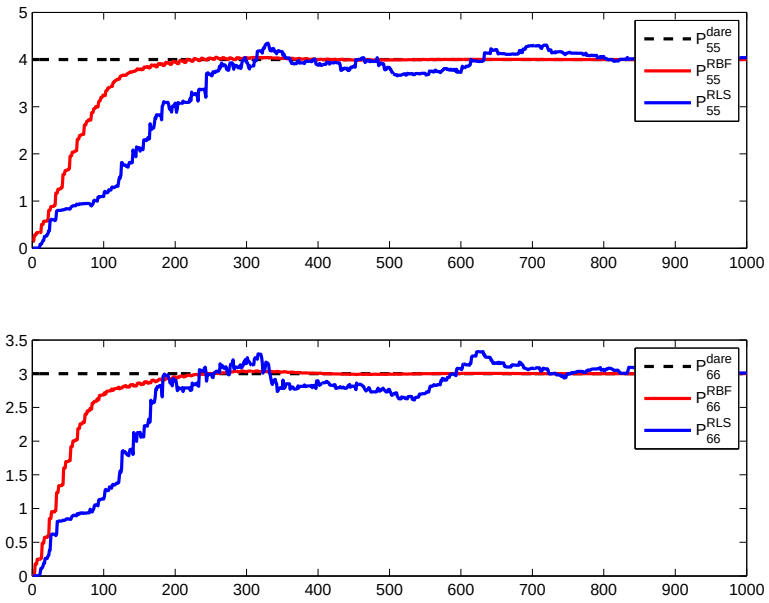


Figura 4.24: Evolução da solução de HJB para os Parâmetros P_{77} e P_{88}.

Por fim o ultimo conjunto de parâmetros apresentados na Figura 4.25.

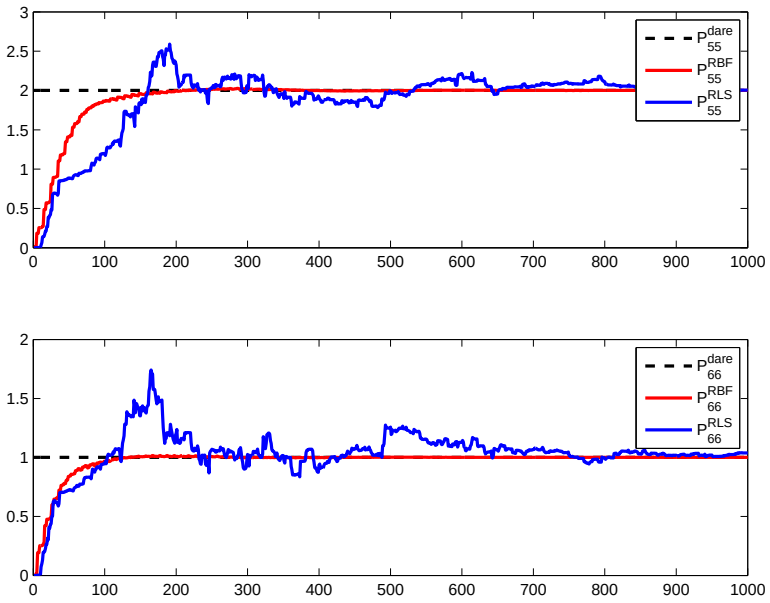


Figura 4.25: Evolução da solução de HJB para os Parâmetros P_{99} e P_{1010}.

Na Figura 4.26 tem-se o custo instantâneo para o aprendizado. Observa-se que, para este caso, o sistema tem o mesmo custo instantâneo a partir da iteração 650, levando em conta a maior quantidades de estados pode-se concluir que a transição de estados para ambos algoritmos é menos custosa e mais estável para este sistema.

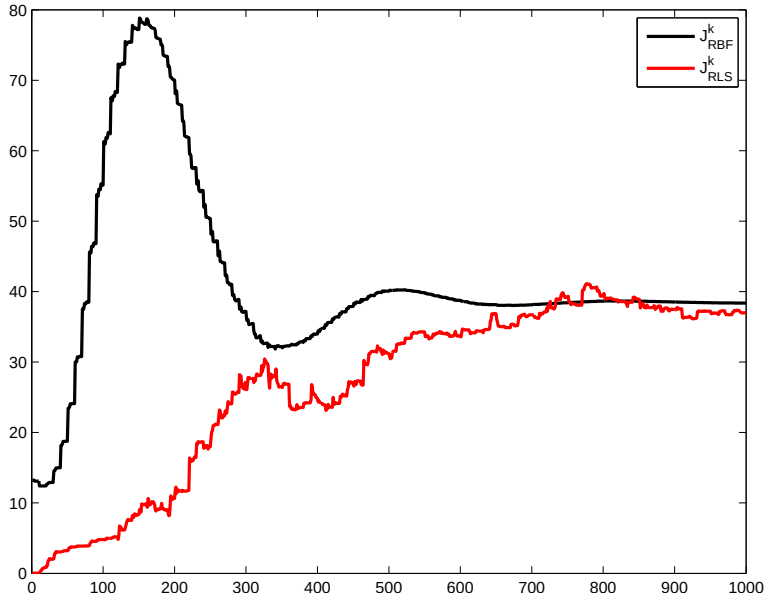


Figura 4.26: Custo Instantâneo para cada k no Ciclo de Aprendizado

E por fim, nas Figuras 4.27, 4.28 e 4.29 tem-se a evolução do alvo móvel em relação a cada estado do sistema:

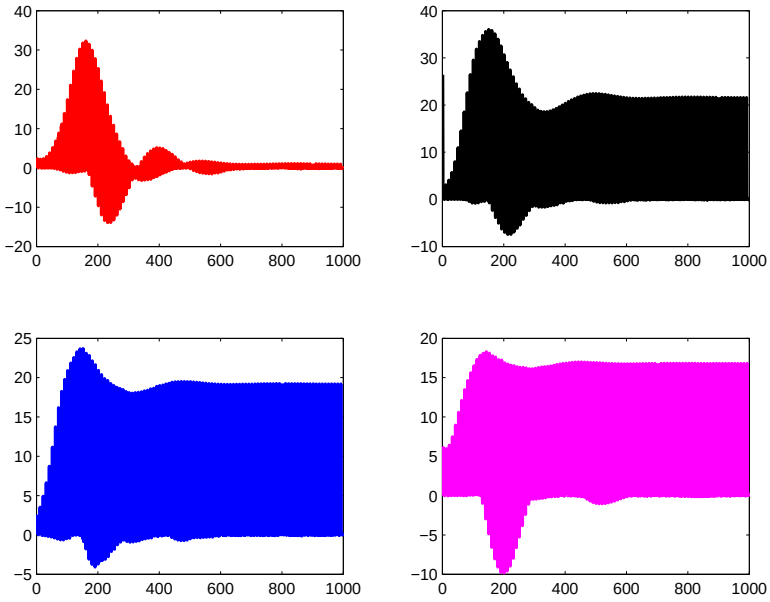


Figura 4.27: Evolução do Alvo Móvel para o Aprendizado online Relativo a (d_1, d_2, d_3, d_4)

e os d_{target} para os estados de x_5 a x_8

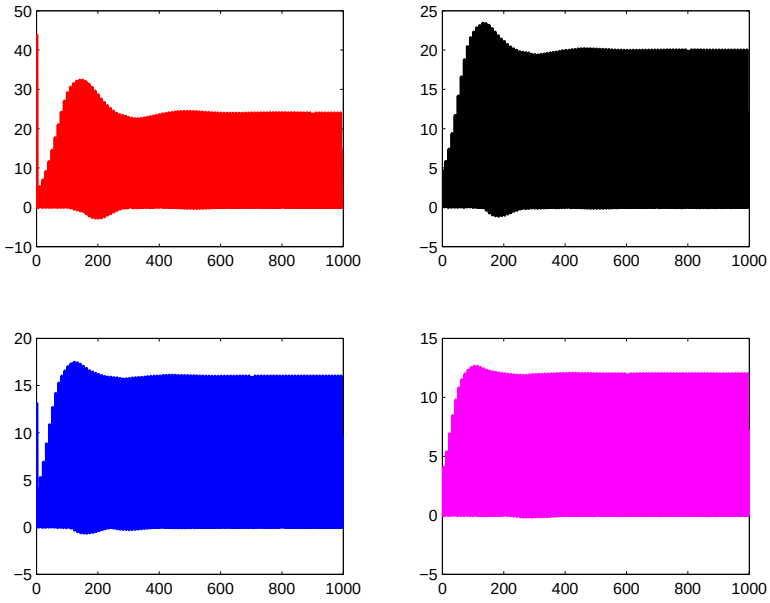


Figura 4.28: Evolução do Alvo Móvel para o Aprendizado online Relativo a (d_5, d_6, d_7, d_8)

Por fim d_{target} para os estados de x_9 e x_{10}.

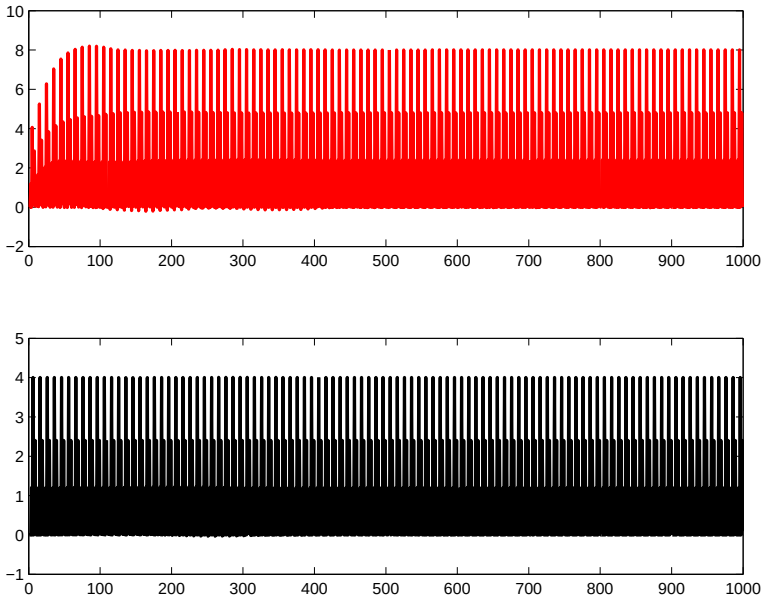


Figura 4.29: Evolução do Alvo Móvel para o Aprendizado online (d_9, d_{10})

Os valores de d_{alvo} estão relacionados ao custo de transição de estados e de controle na componente de cada estado, ou seja, tais valores são o custo final aproximado $V(x_{k+1})$ da contribuição de cada estado no instante k.

A Figura 4.30 mostra a evolução dos autovalores λ_n para o ciclo de convergência.

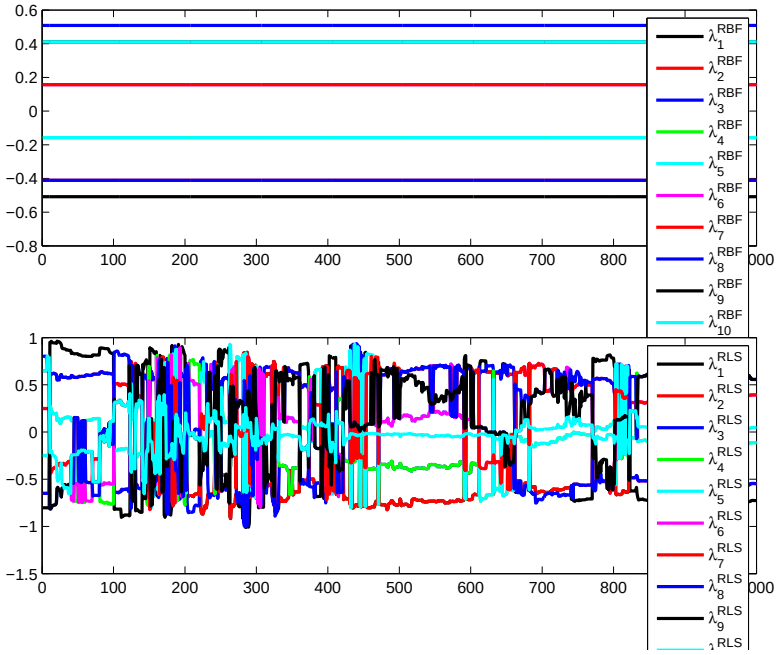


Figura 4.30: Evolução dos Autovalores de Malha fechada em $\mathbf{K_{DHP}}$ para o Algoritmo RBF e RLS, respectivamente.

Note ainda que na Figura 4.30 os autovalores se tornam menos oscilantes em relação ao período em que os algoritmos estão aproximando a solução ótima de **P**.

4.6 Considerações Finais do Capítulo

Neste capítulo foram apresentados o resultados dos sistemas propostos no capítulo anterior em relação a três diferentes sistemas, tais como plantas industriais e processos de estabilização da indústria aeroespacial. Os resultados mostraram que os algoritmos tem sua funcionalidade comprovada, para cada caso pode apresentar vantagens e desvantagens. As Figuras 4.9,4.19 e 4.30 mostram que o sistema é instável em malha fechada para os ganho estimado no tempo k.Os ciclos de aprendizagem mostrados correspondem a uma mudança na operação do sistema, seja esta mudança relativa ao ponto de operação em que o mesmo foi linearizado ou seja em relação a mudanças físicas nos parâmetros do modelo. Desta forma é possível dizer que esta dinâmica de convergência paramétrica vale para todas estas mudanças tornando o sistema de controle adaptativo. Como um sistema de controle inteiramente discreto é necessário ressaltar que para casos de ordem alta ,ou mesmo de melhoria de política não linear, o custo computacional pode se tornar elevado e faz-se então necessário o estudo e implementação da fatoração do algoritmos apresentados. As conclusões gerais a respeito do trabalho estão apresentadas no próximo capítulo.

5

CONCLUSÕES E CONTRIBUIÇÕES DO TRABALHO

Conteúdo

5.1 Conclusões Gerais

Neste trabalho apresentou a dedução de uma base no espaço de estados para a aplicação da estimação online dos parâmetros da solução da equação de Hamilton-Jacobbi-Belman-Riccati. A partir dessa dedução aplicamos a base na solução do algoritmo de projeto de critico adaptativo via programação heurística dual. Os resultados apresentaram uma boa taxa de convergência do sistema e um grande capacidade de adaptação a mudanças ou erros de modelagem a ainda um resposta suficiente a apresentação de ruídos na entrada. Os resultados da simulação podem ser conclusivos no que se diz respeito a solução imediata da aprendizagem do sistema o que pode se explorado em trabalhos futuros em forma de implementações em hardware e softwares de tempo real. Mostra-se uma aplicação expressiva para a aproximação da função de custo de um sistema dinâmico afim de se obter a solução da Equação de Hamilton-Jacobi-Bellman. Um importante resultado experimentado foi a robustez obtida com a inserção de uma rede de funções de base radial atuando como crítico adaptativo no sistema.

5.2 Principais Contribuições

Os resultados obtidos neste trabalho demonstram o desempenho de algoritmos de controle adaptativo utilizando heurísticas para a solução da equação ótima de HJB. O principal foco explorado aqui são as técnicas de estimação paramétrica de críticos adaptativos, assim como estabilização das soluções RLS-RBF para sistemas não lineares de alta ordem.

5.3 Trabalhos Futuros

Como trabalhos futuros pode-se enumerar o melhoramento do desempenho dos críticos adaptativos assim como a viabilização de implementação em hardware dos sistemas de controle aqui propostos.

Referências Bibliográficas

AL-TAMIMI, A., ABU-KHALAF, M., and LEWIS, F. (2007a). Adaptive critic designs for discrete-time zero-sum games with application to h_∞ control. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(1):240–247.

AL-TAMIMI, A., VRABIE, D., ABU-KHALAF, M., and LEWIS, F. (2007b). Model-free approximate dynamic programming schemes for linear systems. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 371–378.

APKARIAN, J. e DAWES, A. (2000). A interactive control education with virtual presence. In *American Control Conference*, volume 6, pages 3985 – 3990.

ARSLAN, A. N. (2004). Dynamic programming based approximation algorithms for sequence alignment with constraints. *INFORMS Journal on Computing*, 16(4):441–458.

ASTROM, K. J. and WITTEMMAK, B. (1989). *Adaptive Control*. Addison-Wesley.

ASTROM, K. J. and WITTEMMAK, B. (1997). *Computer-Controlled Systems: Theory and Design*. Information and System Sciences. Prentice Hall, New Jersey, 3 edition.

BARTO, A. G. (2004). *Handbook of Learning Dynamic Programming.* Wiley, Hoboken, New Jersey.

BELLMAN, R. E. (1957). *Dynamic Programming*. Princeton University Press.

BERGER, A. S. (2001). *Embedded Systems Design: An Introduction to Processes, Tools and Techniques*. CMP Books.

BERTSEKAS, D. P. (1995a). *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific.

BERTSEKAS, D. P. (1995b). *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific.

BISHOP, C. M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

BOUDAREL, R., DELMAS, J., and GUICHET, P. (1971). *Dynamic Programming and Its Application to Optimal Control*. Academic Press.

BRADTKE, S. J. (1993). Reinforcement learning applied to linear quadratic regulation. In Hanson, S. J., editor, *Advances in Neural Information Processing Systems 5*, pages 295–302. Morgan Kaufmann Publishers.

BREWER, J. W. (1978). Kronecker products and matrix calculus in system theory. *Circuits and Systems, IEEE Transactions on*, 25(9):772–781.

BRYSON, A. E., J. e HO, Y. C. (1975). *Applied Optimal Control*. Hemisphere, New York.

BUSONIU, L., BABUSKA, R., SCHUTTER, B. D., e ERNST, D. (2010). *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press.

CARNIELLI, W. e EPSTEIN, R. L. (2005). *Computabilidade, Funções Computáveis, Lógica e os Fundamentos da Matemática*. Editora UNESP, segunda edition.

da FONSECA NETO, J. V. e REGO, P. H. M. (2014). Qr-tunning and approximate-ls solutions of the hjb equation for online action-dependent heuristic dynamic programming. *International Journal of Innovative Computing, Information and Control*, 10(3):1071–1094.

Daniel D. GAJSKI, SAMAR Abdi, A. G. G. S. (2009). *Embedded System Design: Modeling, Synthesis and Verification*. Springer.

DASH, R., SUBUDHI, B., e DAS, S. (2010). A comparison between mlp nn and rbf nn techniques for the detection of stator inter-turn fault of an induction motor. In *Industrial Electronics, Control Robotics (IECR), 2010 International Conference on*, pages 251–256.

de ALMEIDA REGO, J. B. (2010). Identificação de uma planta de corrente de um motor utilizando redes de base radial. Master's thesis, UFRN.

DING, W., DERONG, L., e QINGLAI, W. (2011). Adaptive dynamic programming for finite-horizon optimal tracking control of a class of nonlinear systems. In *Control Conference (CCC), 2011 30th Chinese*, pages 2450–2455.

FAIRBANK, M., ALONSO, E., e PROKHOROV, D. (2013). An equivalence between adaptive dynamic programming with a critic and backpropagation through time. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(12):2088–2100.

HAYKIN, S. (1999). *Neural Network: A Compreensive Foundation*. Bookman.

HAYKIN, S. (2012). *Cognitive Dynamic Systems*. Cambridge University Press.

HU, T., ZHANG, C., YANG, L., e TAN, J. (2008). Engine based embedded control system design and implementation. In *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, pages 469–475.

IOANNOU, P. A. and SUN, J. (1996). *Robust Adaptive Control*. Prentice Hall.

JIANG, Y. e JIANG, Z.-P. (2012). Robust adaptive dynamic programming for nonlinear control design. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1896–1901.

JIANG, Y. e JIANG, Z.-P. (2013). Robust adaptive dynamic programming for optimal nonlinear control design. In *Control Conference (ASCC), 2013 9th Asian*, pages 1–6.

JIANG, Y. e JIANG, Z.-P. (2014). Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. *Neural Networks and Learning Systems, IEEE Transactions on*, PP(99):1–1.

KARAYIANNIS, N. (1998). Learning algorithms for reformulated radial basis neural networks. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 3, pages 2230–2235 vol.3.

KHAN, S. G., HERRMANN, G., LEWIS, F. L., PIPE, T., e MELHUISH, C. (2012). Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual Reviews in Control*, 36(1):42 – 59.

KIS, L., REGULA, G., e LANTOS, B. (2008). Design and hardware-in-the-loop test of the embedded control system of an indoor quadrotor helicopter. In *Intelligent Solutions in Embedded Systems, 2008 International Workshop on*, pages 1–10.

LANDELIUS, T. (1997). *Reinforcement Learning and Distributed Local Model Synthesis*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden. Dissertation No 469, ISBN 91-7871-892-9.

LENDARIS, G., COX, C., SAEKS, R., e MURRAY, J. (2002). A radial basis function implementation of the adaptive dynamic programming algorithm. In *Circuits and Systems,*

2002. MWSCAS-2002. The 2002 45th Midwest Symposium on, volume 2, pages II–338–II–341 vol.2.

LENDARIS, G., SHANNON, T., SCHULTZ, L., HUTSELL, S., e ROGERS, A. (2001). Dual heuristic programming for fuzzy control. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, volume 1, pages 551–556 vol.1.

Lewis, F. L. e LIU, D. (2013). *Reinforcement learning and approximate dynamic programming for feedback control*. John Wiley and Sons.

Lewis, F. L. e SYRMOS, V. L. (1995). *Optimal Control*. John Wiley and Sons, Inc., New York.

LIN, W.-S., TIEN, G., e TU, C.-H. (2006). Adaptive critic neuro-fuzzy control of two-wheel vehicle. In *Fuzzy Systems, 2006 IEEE International Conference on*, pages 445–450.

LIN, W.-S. e YANG, P.-C. (2007). Dhp adaptive critic motion control of autonomous wheeled mobile robot. In *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, pages 311–317.

LIU, D., WANG, D., e ZHAO, D. (2011). Adaptive dynamic programming for optimal control of unknown nonlinear discrete-time systems. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 242–249.

LIU, D., WANG, D., ZHAO, D., Wei, Q., e JIN, N. (2012). Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. *Automation Science and Engineering, IEEE Transactions on*, 9(3):628–634.

MARCIO EDUARDO G. Silva, Joao Viana da Fonseca Neto, F. d. C. d. S. (2014). Pnlms-based algorithm for online approximated solution of hjb equation in the context of discrete mimo optimal control and reinforcement learning. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation,Cambridge, United Kingdom*, pages 69–77.

MASHOR, M. (2004). Performance comparison between hmlp, mlp and rbf networks with application to on-line system identification. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 643–648 vol.1.

MITCHELL, T. M. (1997). *Machine Learning*. McGraw-Hill.

NISBET, R., IV, J. E., e MINER, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press.

OMIDVAR, M. e ELLIOTT, D. L. (1997). *Neural Systems for Control*. Academic Press.

PARK, J.-W., HARLEY, R., e VENAYAGAMOORTHY, G. (2002). Comparison of mlp and rbf neural networks using deviation signals for indirect adaptive control of a synchronous generator. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 1, pages 919–924.

POWELL, W. (2008). Approximate dynamic programming: Lessons from the field. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 205–214.

POWELL, W. B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley and Sons.

REGO, P. H. M., da FONSECA, J. V., e FERREIRA, E. M. (2013). 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*, 0(0):1–23.

SMITH, H. W. (1969). Dynamic controle of a two-stand cold mill. *Automatica*, 5:109–115.

STEVENS, B. e LEWIS, F. L. (1992). *Aircraft control and simulation*. Wiley.

SUTTON, R. S. e BARTO, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press.

WAN, J., Li, D., YE, F., ZHANG, C., e XIAO, S. (2007). Multitask schedulability simulation research of hydraulic embedded control system. In *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pages 352–355.

WANG, J., HU, H., WANG, J., Li, Z., e GONG, Z. (2009). Development of an embedded control system for magnetorheological fluid damper under impact load. In *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, pages 3–717–3–722.

WATKINS, C. (1989). *Learning from Delayed Rewards*. PhD thesis, Cambridge University.

WERBOS, P. (1992). *Handbook of Intelligent Control*, chapter Approximate Dynamic Programming for Real Time Control and Neural Modeling. Van Nostrand Reinhold.

WIERING, M. e VAN OTTERLO, M. (2011). *Reinforcement Learning: State-of-the-Art*. Springer.

WU, W., HAIFEI, G., e YAN, G. (2009). Embedded control system design for autonomous navigation mobile robot. In *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, volume 1, pages 200–203.

YU, H., XIE, T., PASZCZYNSKI, S., e WILAMOWSKI, B. (2011). Advantages of radial basis function networks for dynamic system design. *Industrial Electronics, IEEE Transactions on*, 58(12):5438–5450.

A

PROBLEMA DLQR

Conteúdo

A seguir são apresentadas as parametrizações para o regulador linear quadrático no tempo discreto (DLQR) abordados no Capitulo 2. Para sistemas discretos, a função de custo, ou indice de performance, é geralmente da forma

$$J_N = \sum_{k=0}^{N} L[x_k, u_k], \tag{A.1}$$

sendo k o instante amostrado a N o tamanho da amostra, x_k os estados do sistema e ainda u_k a entrada de controle do sistema.

Para um sistema físico, as entradas de controle sempre tem restrições referentes as limitações dos atuadores do mesmo, logo cada componente do vetor de controle é limitado tal que

$$\left|u_k^i\right| \leq U_i, \tag{A.2}$$

sendo U_i uma dada constante que limita a uma i-ésima componente do vetor de entrada. Para o caso da limitação da energia de controle tem-se

$$u_k^2 \leq M_i, \tag{A.3}$$

sendo M_i uma constante dada. A disponibilidade da energia de controle finita pode ser representada por meio da função da custo dada na Eq.(A.1) da seguinte forma

$$\sum_{k=0}^{N} \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k, \tag{A.4}$$

sendo $\mathbf{R}$ uma matriz de ponderação para para as entradas de controle. A Eq.(A.2) é chamada de forma quadrática. Uma política de controle que satisfaz tais restrições é chamada de *política admissível*. Para o projeto da política amissível com função de custo quadrática o projeto de controle consiste em determinar a seguinte lei de controle

$$\mathbf{u}_k = -\mathbf{K}_k \mathbf{x}_k. \tag{A.5}$$

A.1 Controle Linear Quadrático no Tempo Discreto (DLQR)

Para o sistema de controle e o modelo de função de custo mostrando anteriormente, para um sistema linear descrito pela seguinte equação

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{A.6}$$

sendo **A** e **B** matrizes de ponderação de estados e de entrada, respectivamente. O projeto consistem em determinar

$$\mathbf{u}_k^* = f(\mathbf{x}_k), \tag{A.7}$$

que minimiza a função de custo quadrática dada por

$$J_N = \sum_{k=0}^{N} \mathbf{x}_k^T \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T \mathbf{R}\mathbf{u}_k, \tag{A.8}$$

sendo **Q** a matriz de ponderação para o custo da transição de estados e ainda $\mathbf{Q} = \mathbf{Q}^T$ e $\mathbf{R} = \mathbf{R}^T$, e ainda $\mathbf{Q} \geq 0$ e $\mathbf{Q} \geq 0$.

A Eq.(A.8) é diferenciada em relação a **x** e rearranjada, obtendo-se a seguinte equação

$$\mathbf{P}_{k+1} = \mathbf{A}^\mathbf{T}\mathbf{P}_k\mathbf{A} + \mathbf{Q} - \mathbf{A}^\mathbf{T}\mathbf{P}_k\mathbf{B}\Big(\mathbf{B}^\mathbf{T}\mathbf{P}_k\mathbf{B} + \mathbf{R}\Big)^{-1}\mathbf{B}^\mathbf{T}\mathbf{P}_k\mathbf{A}, \tag{A.9}$$

sendo **P** positiva definida e simétrica. A Eq.(A.9) é denominada Equação Algébrica de Riccati no tempo Discreto (DARE) e J_N expresso da seguinte forma

$$J_N = \mathbf{x}_0\mathbf{P}_k\mathbf{x}_0. \tag{A.10}$$

Portando a lei de controle que implementa a lei de controle ótimo é dada por

$$\mathbf{K_k} = \Big(\mathbf{B}^\mathbf{T}\mathbf{P}_k\mathbf{B} + \mathbf{R}\Big)^{-1}\mathbf{B}^\mathbf{T}\mathbf{P}_k\mathbf{A} \tag{A.11}$$

que é a lei de controle que implementa a política ótima na Eq.(A.5).

B

PROPRIEDADES DO PRODUTO DE KRONECKER

Conteúdo

O produto de kronecker é uma operação entre duas matrizes de tamanho arbitrário que resulta em uma matriz bloco. Os conceitos e propriedade produto de kronecker são explorados em (BREWER, 1978), para a solução de problemas de diferenciação de vetores e matrizes.

B.1 Definição

Seja **A** uma matriz *nxm* e **B** uma matriz *pxq* tem-se

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}, \tag{B.1}$$

expandindo os elementos tem-se

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}. \tag{B.2}$$

Logo se **A** e **B** representam transformações lineares o produto $\mathbf{A} \otimes \mathbf{B}$ é produto tensor entre dois mapas.

B.2 Propriedades

B.2.1 Bilinearidade e Associatividade

$$\begin{aligned}
&\mathbf{A}\otimes(\mathbf{B}+\mathbf{C}) = \mathbf{A}\otimes\mathbf{B}+\mathbf{A}\otimes\mathbf{C},\\
&(\mathbf{A}+\mathbf{B})\otimes\mathbf{C} = \mathbf{A}\otimes\mathbf{C}+\mathbf{B}\otimes\mathbf{C},\\
&(k\mathbf{A})\otimes\mathbf{B} = \mathbf{A}\otimes(k\mathbf{B}) = k(\mathbf{A}\otimes\mathbf{B}),\\
&(\mathbf{A}\otimes\mathbf{B})\otimes\mathbf{C} = \mathbf{A}\otimes(\mathbf{B}\otimes\mathbf{C}),
\end{aligned} \tag{B.3}$$

sendo **A** , **B** e **C** matries e ainda um escalar k

B.2.2 Não Comutativa

Em geral a operação, por envolver matrizes, é não comutativa, excluindo-se os casos:

$$\mathbf{A}\otimes\mathbf{B} = \mathbf{P}(\mathbf{B}\otimes\mathbf{A})\mathbf{Q}, \tag{B.4}$$

aonde $\mathbf{P} = \mathbf{Q}^{\mathrm{T}}$ são as matrizes de permutação da operação.

B.2.3 Propriedade do Produto Misto

Sejam **A**, **B**, **C** e **D** matrizes de dimensões tal que os produtos **AC** e **BD** são admissíveis então

$$(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = \mathbf{AC}\otimes\mathbf{BD}. \tag{B.5}$$

B.2.4 Transposição

A transposta e a transposta conjugadas são distributivas

$$(\mathbf{A} \otimes \mathbf{B})^{\mathbf{T}} = \mathbf{A}^{\mathbf{T}} \otimes \mathbf{B}^{\mathbf{T}}, \tag{B.6}$$

e ainda

$$(\mathbf{A} \otimes \mathbf{B})^{*} = \mathbf{A}^{*} \otimes \mathbf{B}^{*}. \tag{B.7}$$

B.3 Equações Matriciais

Seja a equação matricial com **A**, **B**, **C** e incógnita **X** logo

$$(\mathbf{B}^{\mathbf{T}} \otimes \mathbf{A})\text{vec}(\mathbf{X}) = \text{vec}(\mathbf{AXB}) = \text{vec}(\mathbf{C}). \tag{B.8}$$

sendo vec$(\mathbf{X})$ a vetorização da matriz de solução **X** obtida empilhando se em ordem crescente as colunas de **X** . Observa-se ainda que tais tal equação tem solução somente se as matrizes **A** e **B** são não singulares.

C

CONVERGÊNCIA PARA O ALGORITMO DHP

Abaixo segue a prova de convergência para o esquema DHP para controle linear quadrático discreto no tempo (DLQR) conforme tratado em (LANDELIUS, 1997).

Teorema C.0.1. ***(Convergência para o Esquema DHP)*** *Assumindo que* $\alpha \in (0,1]$, $\mathbf{R} > 0$, $\mathbf{Q} \geq 0$ *e o par* $(\mathbf{A},\mathbf{B})$ *é controlável. Definindo a sequencia* $\{\mathbf{P}_k\}_{k=0}^{\infty}$ *de acordo com a Eq.(C.1) e considerando ainda* $\mathbf{P}_0 = 0$. *Então existe um solução única* $\mathbf{P}^* \geqslant 0$ *tal que* $\mathbf{P}_k \to \mathbf{P}^*$ *quando* $k \to \infty$ *na forma recursiva*

$$\mathbf{P}_{k+1} = f(\mathbf{P}_k, \mathbf{K}_k) = \mathbf{P}_k + \alpha\left[\mathbf{Q} + \mathbf{K}_k^{\mathbf{T}}\mathbf{R}\mathbf{K}_k + (\mathbf{A}+\mathbf{B}\mathbf{K}_k)^{\mathbf{T}}\mathbf{P}_k(\mathbf{A}+\mathbf{B}\mathbf{K}_k) - \mathbf{P}_k\right], \tag{C.1}$$

sendo o ganho de realimentação de estados $\mathbf{K}_k$ *definido por*

$$\mathbf{K}_k = \mathbf{K}(\mathbf{P}_k) = -\left(\mathbf{R} + \mathbf{B}^{\mathbf{T}}\mathbf{P}_k\mathbf{B}\right)^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{P}_k\mathbf{A} \tag{C.2}$$

O limite $\mathbf{P}^*$ *é também a solução única positiva semidefinida para e Equação Algébrica de Riccati no tempo Discreto (DARE)*

$$\mathbf{P}^* = \mathbf{Q} + \mathbf{A}^{\mathbf{T}}\left(\mathbf{P}^* - \mathbf{P}^*\mathbf{B}\left(\mathbf{R} + \mathbf{B}^{\mathbf{T}}\mathbf{P}^*\mathbf{B}\right)^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{P}^*\right)\mathbf{A} \tag{C.3}$$

Lema C.0.1. *Dada uma sequencia arbitraria* $\{\mathbf{H}_k\}_{k=0}^{\infty} \subset \Re^{nxn}$ *e ainda um matriz* $\mathbf{Z}_0 \geq 0$. *Usando um matriz como um ponto de partida e gerando a sequência* $\{\mathbf{Z}_k\}_{k=0}^{\infty}$ *usando a Eq.(C.1), isto é* $\mathbf{Z}_{k+1} = f(\mathbf{Z}_k, \mathbf{H}_k)$. *Sendo* $\mathbf{P}_0$ *uma matriz tal que* $0 \leq \mathbf{P}_0 \leq \mathbf{Z}_0$ *e gere uma nova sequencia de acordo com as Eqs. (C.1) e (C.2), isto é* $\mathbf{P}_{k+1} = f(\mathbf{P}_k, \mathbf{K}(\mathbf{P}_k))$ *então* $0 \leq \mathbf{P}_k \leq \mathbf{Z}_k$ *para* $k = 1,2,3\ldots.$

D

ASPECTOS DE IMPLEMENTAÇÃO

Para se analisar os aspectos relativos a implementação dos algoritmos foram levados em consideração principalmente a capacidade de operações matemáticas de dispositivos de circuitos integrados de aplicação específica (ASIC's), tendo que o projeto de sistemas embarcados consiste basicamente na concepção de estruturas de processamento lógico para uma única tarefa. Desta forma foram investigados as aspectos de cálculos de pontos flutuantes no segundo (FLOPs') que mede a capacidade de cálculo do processador e o seu CPU time, ou seja o tempo de execução, para fins de testes, em determinadas CPUs'(DANIEL D. GAJSKI, 2009).

Primeiramente tem-se os resultados para o estudo de caso I, ou seja, o controle longitudinal de um avião de caça F-16 apresentado na Tabela D.1, o sistema foi simulado e as operações o e tempo das mesmas foram calculados para cada algoritmo. Nestes primeiros resultados é importante observar que o número de cálculos por segundos, neste caso, de milhões de cálculos por segundo (Mflops) é constante para os Algoritmos 1, 2 e 3 nas três plantas, pois a estrutura dos mesmo não se altera para uma determinada planta. Este valor corresponde ao número de operações de ponto flutuante que o processador terá que realizar para executar o algoritmo levando em conta as operações de soma, subtração, multiplicação, transposta e inversa de matrizes presentes nos algoritmos. Portanto em relação aos algoritmos somente o CPU time será alterado de uma planta para outra.

Tabela D.1: Complexidade Computacional para o Estudo de Caso I

Algoritmo	MFlops	CPU Time(milissegundos)
I	6,6	20
II	6,5	21
III	7,1	22

Nota-se ainda na Tabela D.1 que todos os algoritmos executam entre 20 e 22 milissegundos, aonde o Algoritmo 3 apresenta o maior número de cálculos e por sua vez um maior tempo de duração. Já na Tabela D.2, para os resultados do sistema do Helicóptero 3DOF pode-se observar que o tempo de execução dos cálculos sofre pouca variação, mesmo levando em conta uma planta de sexta ordem aonde a convergência apresenta resultados entre 21 e 22 milissegundos de execução, para o ciclo de aprendizagem.

Tabela D.2: Complexidade Computacional para o Estudo de Caso II

Algoritmo	MFlops	CPU Time (milissegundos)
I	6,1	21
II	6,5	21,4
III	7,1	22

Por fim a D.3 apresenta os resultados para o sistema de laminação a frio (CRM), de décima ordem. Os detalhes a se observar estão relacionados com o exatamente a ordem do sistema que faz com que os cálculos de x_{k+1}, $\mathbf{K_u}$ e $\mathbf{K_w}$ e consequentemente d_{target} se torne mais custoso do ponto de vista computacional.

Tabela D.3: Complexidade Computacional para o Estudo de Caso III

Algoritmo	MFlops	CPU Time(milissegundos)
I	6,1	24
II	6,5	24,5
III	7,1	26

Alguns pontos ser ressaltados sobre os aspectos de implementação e resultados apresentados nas Tabelas D.1, D.2 e D.3 tais como:

- Para os testes realizados na plataforma **MATLAB** deve-se considerar que os sistemas são simulados dentro de ambientes multitarefas de de escalonamento preemptivo, ou seja, no surgimento de uma tarefa de maior prioridade para o Sistema Operacional (SO) o processador tem seus recursos desviados para a tarefa;

- Desta forma a variável *CPU time* pode apresentar pequenas variações em relação ao ambiente que a plataforma é executada .

- Para sistemas embarcados e ASIC's, os relógios (clock) dos processadores são geralmente de frequências mais baixas (Ex: μP tem em média 2.6 GHz e μC entre 1Mhz e 100Mhz), o que torna mais árdua ainda a tarefa de executar os algoritmos em tempo real para uma fatia de tempo kT_s e ainda levando em conta os ciclos de clock necessários para leitura, tratamento das variáveis medidas e da variável de atuação;

- Um ultimo aspecto é a que tem que ser observado para a implementação dos algoritmos é a memória de programa, pois o sistema irá manipular vetores de grande escala, portando o dispositivo microprocessado deve suportar uma memoria de variáveis relativamente alta.