UNIVERSIDADE ESTADUAL DO MARANHÃO

CENTRO DE CIÊNCIAS TECNOLÓGICAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO E SISTEMAS

MESTRADO PROFISSIONAL EM ENGENHARIA DE COMPUTAÇÃO E SISTEMAS

JANES VALDO RODRIGUES LIMA

PROJETO DE CONTROLADOR ÓTIMO ADAPTATIVO PARA VEÍCULO AÉREO NÃO TRIPULADO – MODELO QUADRICÓPTERO – VIA PROGRAMAÇÃO DINÂMICA APROXIMADA

SÃO LUÍS

2019

JANES VALDO RODRIGUES LIMA

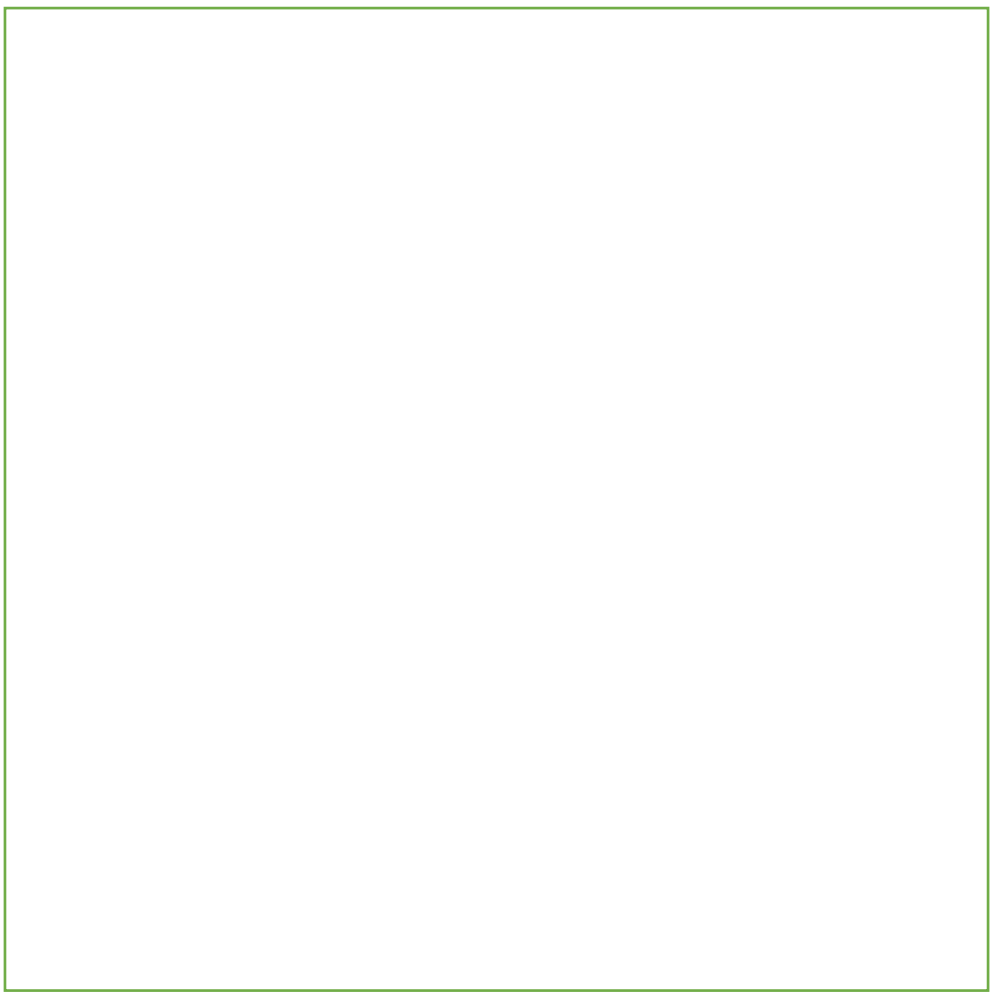
PROJETO DE CONTROLADOR ÓTIMO ADAPTATIVO PARA VEÍCULO AÉREO NÃO TRIPULADO – MODELO QUADRICÓPTERO – VIA PROGRAMAÇÃO DINÂMICA APROXIMADA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão como parte de requisitos exigidos para obtenção do título de mestre em Engenharia da Computação.

Orientação da Profa. Dra. Patrícia Helena Moraes Rêgo.
Coorientador: Dr. Alain Giacobini de Souza

SÃO LUÍS

2019



JANES VALDO RODRIGUES LIMA

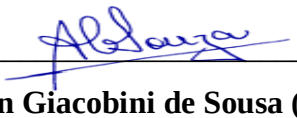
PROJETO DE CONTROLADOR ÓTIMO ADAPTATIVO PARA VEÍCULO AÉREO NÃO TRIPULADO – MODELO QUADRICÓPTERO – VIA PROGRAMAÇÃO DINÂMICA APROXIMADA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão como parte de requisitos exigidos para obtenção do título de mestre em Engenharia da Computação.

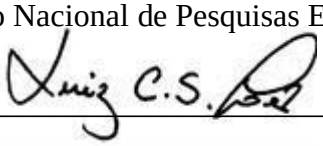
Orientação da Profa. Dra. Patrícia Helena Moraes Rêgo.
Coorientador: Dr. Alain Giacobini de Souza

Aprovado em: / / 2019

Drª Patrícia Helena Moraes Rêgo (*Orientadora*)
Doutora em Engenharia de Eletricidade
Universidade Estadual do Maranhão



PhD. Alain Giacobini de Sousa (Coorientador)
Doutor em Engenharia e Tecnologias Espaciais
Instituto Nacional de Pesquisas Espaciais



PhD. Luiz Carlos Sandoval Góes
Doutor em Engenharia de Eletricidade
Instituto Tecnológico de Aeronáutica

Prof. Dr. Mauro Sérgio Silva Pinto
Doutor em Engenharia de Eletricidade
Universidade Estadual do Maranhão

*"*O homem tem dois objetivos principais no estudo científico de seu ambiente: ele quer entender e controlar. Os dois objetivos se reforçam mutuamente, uma vez que um entendimento mais profundo permite um controle mais firme e, por outro lado, a aplicação sistemática de teorias científicas gera inevitavelmente novos problemas que requerem mais investigação e assim por diante*"*.

Richard E. Bellman e Robert E. Kalaba

A Deus, o Criador, o Feitor de toda a história.
Ao Pai de Luz. Aquele que era, Aquele que é,
Aquele que há de vir.

AGRADECIMENTOS

A Deus, o qual é o motivo da dedicação e o motivo principal dos agradecimentos. Dele provêm a sabedoria e o entendimento, o conhecimento e as oportunidades.

A minha mãe, que sempre mostrou que através da luta e da completa dedicação, que podemos conseguir alcançar nossos objetivos e proporcionou os valores que deram base para o meu futuro.

Aos meus familiares, minha esposa e amigos, os quais sempre me motivaram, apoiaram e veem valor em mim.

Um agradecimento muito especial à minha orientadora, Profª. Dra. Patrícia Helena Moraes Rêgo, pela inspiração, orientação preciosa, incansável dedicação e parceira constante durante a realização desse trabalho. Quando muitas vezes não via saída, ela estava encorajando e mostrando os caminhos para esse sucesso, desde a escolha do projeto de estudo até a finalização da dissertação. Que essa parceria dure por muito tempo.

Ao Prof. Dr. Alain Giacobini de Sousa que sempre foi solícito no processo de produção desse trabalho.

Ao Prof. Mse. Guilherme Bonfim, que se tornou um amigo nessa jornada de aprendizado e muito trabalho. O qual possibilitou muitas portas nesse processo. Muito obrigado!

Aos amigos da turma PECS/ITA, pelo acompanhamento e pela leveza, descontração e aprendizado nesses anos.

À Universidade Estadual do Maranhão - UEMA, ao Programa de Pós Graduação em Engenharia da Computação e Sistemas, e à Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão - FAPEMA pelos recursos materiais e financeiros destinados a essa pesquisa.

RESUMO

Apresenta-se nesta dissertação o projeto de um controlador ótimo adaptativo para um quadricótpero baseado em estratégias de crítico adaptativo, usando métodos de iteração de política e iteração de valor. Esta abordagem fornece algoritmos *online* de controle que são integrados nas estruturas de aprendizado por reforço ator-crítico. Estas estruturas de Aprendizado por Reforço (RL), que estão inseridas no quadro de Programação Dinâmica Aproximada (ADP), são usadas para resolver problemas de decisão ótima *online*, sem requerer o conhecimento completo do modelo da dinâmica do sistema a ser controlado. O objetivo é implementar um controlador ótimo que atue no controle do quadricóptero, utilizando somente os sinais de entrada de controle, estados, bem como os custos instantâneos, observados ao longo das trajetórias do sistema. Esta técnica de projeto de controle de realimentação é capaz de sintonizar *online* os parâmetros do controlador na presença de variações na dinâmica da planta e perturbações externas. A avaliação do algoritmo de controle será realizada no modelo simulado em MATLAB.

Palavras-Chave: Controle Ótimo Adaptativo. Aprendizagem por Reforço. Programação Dinâmica Aproximada. Controle de Quadricóptero.

ABSTRACT

The design of an adaptive optimal controller for a quadcopter based on adaptive critic strategies, using policy iteration and value iteration methods is presented in this dissertation. This approach provides online control algorithms that are integrated into actor-critic reinforcement learning structures. These Reinforcement Learning (RL) structures, which are inserted in the Approximate Dynamic Programming (ADP) structures, are used to solve optimal decision-making problems online, without requiring a complete knownledge of the system dynamics model to be controlled. The goal is to implement an optimal controller that acts on the quadcopter control, using only the control input signals, states, as well as the instantaneous costs observed along the system trajectories. This feedback control design technique is able to tune controller parameters online in the presence of variations in plant dynamics and external disturbances. The evaluation of the control algorithm will be performed in the simulated MATLAB model.

Keywords: Adaptive Optimal Control. Reinforcement Learning. Approximate Dynamic Programming. Quadcopter Control.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AC	*Adaptive Critic* (Crítico Adaptativo)
AD	*Action Dependent* (Dependente de Ação)
AD-DHP	*Action Dependent Dual Heuristic Programming* (Programação Heurística Dual Dependente de Ação)
AD-HDP	*Action Dependent Heuristic Dynamic Programming* (Programação Dinâmica Heurística Dependente de Ação
ADP	*Approximate/Adaptive Dynamic Programming* (Programação Dinâmica Aproximada/Adaptativa)
CLA	Centro de Lançamento em Alcântara
DHP	*Dual Heuristic Programming* (Programação Heurística Dual)
DOF	*Degree of Freedom* (Graus de Liberdade)
DT	*Discrete Time* (Tempo Discreto)
DT LQR	*Discrete Time Linear Quadratic Regulator* (Regulador Linear Quadrático de Tempo Discreto)
GPI	*Generalized Policy Algorithm* (Iteração de Política Generalizada)
HDP	*Heuristic Dynamic Programming* (Programação Dinâmica Heurística)
HJB	*Hamilton-Jacobi-Bellman*
IP	Iteração de Política (*Policy Iteration*)
IV	Iteração de Valor (*Value Iteration*)
LQR	*Linear Quadratic Regulador* (Regulador Linear Quadrático)
LS	*Least-squares* (Mínimos Quadrados)
RL	*Reinforcement Learning* (Aprendizagem por Reforço)
RLS	*Recursive Least-Squares* (Mínimos Quadrados Recursivos)
TD	*Temporal Difference* (Diferença Temporal)
UAV	*Unmanned Aerial Vehicle* (Veículo Aéreo Não – Tripulado)
VANT	Veículo Aéreo Não – Tripulado
VTOL	*Vertical Take-off and Landing* (Decolagem e Aterrisagem Vertical)
VFA	*Value Function Approximation* (Aproximação da Função Valor)

SUMÁRIO

1. INTRODUÇÃO

Os sistemas dinâmicos ligados aos mais diferentes setores, tais como a indústria, ciência, tecnologia, etc., devido aos requisitos cada vez mais exigentes, possuem uma complexidade cada vez mais elevada, exigindo dos controladores um desempenho ainda mais robusto e com estratégias de controle adequadas para execução das tarefas programadas, os quais são controladores de malha fechada de alto desempenho; isso é visto em aeronaves, veículos não tripulados e robôs móveis que requerem uma alta precisão e robustez do sistema de controle para operarem em ambientes desconhecidos. Uma abordagem para atender aos requisitos desses projetos é encontrada na teoria de controle ótimo, que lida com a operação de um sistema dinâmico complexo com um custo mínimo (LEWIS e VRABIE D., 2009 KIRK, 2004).

Especificamente os Veículos Aéreos Não Tripulados (VANTs - do inglês *Unmanned Aerial Vehicle (UAV)*), popularmente chamados de drones, tem alcançado notoriedade tanto no ramo industrial, quanto em aplicações civis com possibilidades de tomadas fotográficas e filmagens de alta resolução e, também, na engenharia aeroespacial, no envio de informações captadas no ar (VALAVANIS e VACHTSEVANOS, 2015), (DE CASTRO JORGE e INAMASU, 2014). São veículos inerentemente instáveis e requerem algoritmos de controle bem projetados para adaptar-se a ambientes não controlados e condições inesperadas. É desejável que estes sistemas satisfaçam não só as especificações de projeto, tais como figuras de mérito, mas também operem com um custo mínimo.

O objeto de estudo deste trabalho será um VANT, tipo quadricóptero, e com a temática a ser abordada a otimização do esforço de controle no processo de decolagem e aterrissagem vertical – controle de altitude –, que está sujeito a variações na dinâmica, perturbações externas e *Sense and Avoid* (este termo é usado para descrever a capacidade de um VANT de detectar o tráfego aéreo e responder com manobras apropriadas de evitar colisões, a fim de manter distâncias mínimas de separação).

Existem muitas técnicas de otimização e controle ótimo presentes na literatura para sistemas dinâmicos lineares e não-lineares (LEWIS e VRABIE D., 2009) (BRYSON e HO, 1975). No entanto, o projeto de um controlador ótimo com características de adaptabilidade tem um peso relevante, principalmente para o controle de trajetórias de voo, devido ao seu regulador auto ajustável. Recentemente, vários pesquisadores tentaram explorar as técnicas computacionais inteligentes para o projeto de controle adaptativo e ótimo (BHUVANESWARI, UMA e RANGASWAMY, 2009), (KAR e BEHERA, 2009).

Controladores ótimos têm sido obtidos por uma abordagem de programação dinâmica. A equação de Hamilton-Jacobi-Bellman (HJB) é a base para implementação de algoritmos de programação dinâmica (BERTSEKAS, 2012.). No entanto, a principal desvantagem da programação dinâmica clássica é: os métodos não são viáveis para aplicações em tempo real, devido à complexidade computacional associada à solução da equação HJB, que são de alto custo para problemas de controle ótimo de múltiplos estágios em grande escala (LENDARIS, 2009). De uma maneira geral, os métodos são *offline* e exigem o conhecimento completo do modelo do sistema dinâmico.

De acordo com (STINGU e LEWIS, 2011) existe atualmente uma dicotomia entre controle ótimo e controle adaptativo. Algoritmos de Controle Adaptativo aprendem *online* e fornecem controladores com garantia de desempenho para sistemas desconhecidos. Contudo, controladores adaptativos, geralmente, não são projetados com a qualidade de serem ótimos no sentido de minimizarem funções de custo, como definido no quadro de controle ótimo.

A Programação Dinâmica Aproximada (*Approximate Dynamic Programming – ADP*), em contrapartida, permite projetar controladores adaptativos que aprendem *online*, em tempo real, as soluções para problemas de controle ótimo, sem que haja conhecimento, a priori, do modelo da dinâmica do sistema a ser controlado (LEWIS e VRABIE D., 2009). Diferente da programação dinâmica tradicional, que requer um procedimento "para trás no tempo" (*backward in time*) para resolver a equação HJB, a ADP resolve os problemas de maneira "para frente no tempo" (*forward in time*). Este método é baseado em Aprendizagem por Reforço (*Reinforcement Learning - RL*) para aproximar a solução ótima de uma função de custo (função valor) que garanta otimalidade ao longo do tempo.

Aplicações de sucesso de ADP para controle de realimentação de sistemas dinâmicos são apresentadas em (FERRARI e STENGEL, 2004) (WEI e LIU, 2015). Em (HWANGBO e HUTTER, 2017) e (LEVINE e ZHANG, 2016), a utilização de aprendizagem por reforço mostrou-se promissora para uma rede neural visando à estabilização e controle de veículos aéreos. Uma visão geral das técnicas de ADP e seus avanços focados em controle ótimo adaptativo são apresentados em (KHAN e LEWIS, 2012), onde os autores destacam as aplicações de ADP em robótica. Pesquisas importantes nessa area são dadas em (LEWIS e LIU, 2012) e (WANG, ZHANG e LIU, 2009).

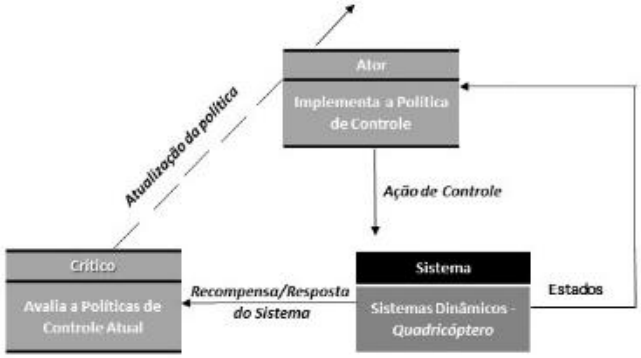
Segundo (SOUSA, 2018) os métodos de controle ótimo baseados em programação dinâmica aproximada e aprendizagem por reforço têm se destacado como uma ferramenta muito útil para melhorar o desempenho de sistemas do mundo real e contribuído para viabilizar as

implementações em tempo real de controladores ótimos. Mas alguns requisitos de desempenho ainda necessitam ser melhorados na implementação de métodos de *RL* e *ADP* para o controle de sistemas não-lineares, tais como a estabilidade numérica e a convergência dos algoritmos, dentre outros.

Entre os algoritmos iterativos propostos para estimar os parâmetros da função valor, em metodologia de projeto de controle baseada em ADP, os mínimos quadrados recursivos (*Recursive Least-Squares – RLS*) é um dos mais bem promissores. A eficiência de métodos RLS é devido a sua característica de robustez e adaptabilidade para lidar com as variações no tempo dos parâmetros de regressão, e a rápida convergência quando comparado com os métodos de gradiente estocástico. Inúmeras pesquisas e aplicações dos métodos RLS podem ser encontradas na literatura, alguns dos quais lidam com problemas de treinamento de redes neurais (YEH, SU e RUDAS, 2011) (YEH e SU, 2012.). Para melhorar a eficiência dos métodos heurísticos críticos adaptativos convencionais, estruturas baseadas em RLS têm sido propostas (XU, HE e HU, 2002). Além do mais, autores exploram métodos RLS para resolver problemas de aprendizagem por reforço ator-crítico (SOUSA, 2018).

A presente pesquisa se propõe a mostrar a aplicabilidade da Programação Dinâmica Aproximada e Aprendizagem por Reforço para o projeto de um controlador ótimo adaptativo de um quadricóptero que voa em um ambiente sujeito a perturbações e incertezas. A estrutura de controle que será usada neste trabalho é ilustrada na Figura 1.1, em que o agente de aprendizagem (controlador) interage com o ambiente (quadricóptero) inicialmente desconhecido. O componente ator aplica uma ação ou política de controle ao ambiente, e o componente crítico avalia o valor daquela ação. Baseado nesta avaliação, vários esquemas podem então ser usados para modificar ou melhorar a ação no sentido que a nova política produz um valor que é melhorado sobre o valor anterior. Desta forma, o agente RL é suposto a aprender a política ótima a partir de suas experiências sem conhecer os parâmetros do sistema dinâmico.

Figura 1.1 Estrutura de Controle RL Ator-Crítico



Em (STINGU e LEWIS, 2011) são destacados vários projetos de melhoria de estabilidade para quadricópteros, que são capazes de realizar voos autônomos usando apenas sensores a bordo para estimação da atitude, altitude, posição horizontal e voos translacionais; todavia, não conseguem atingir o desempenho ótimo. Neste ponto, a ADP *online* tem um grande potencial ao manter também a adaptabilidade e a robustez de outros algoritmos. A maior parte das pesquisas em ADP tem sido conduzida por conceitos de aprendizagem por reforço para sistemas que operam em tempo discreto (*Discrete Time - DT*). Embora o modelo do quadricóptero seja de tempo contínuo, todos os outros sinais externos ao sistema dinâmico são amostrados em tempo discreto. Portanto, neste trabalho, serão desenvolvidos algoritmos de controle de ADP de tempo discreto.

1.1 Motivação e Relevância

Na natureza, a maioria dos organismos atua sobre seu ambiente de maneira ótima para conservar os recursos enquanto alcança seus objetivos. Agentes buscam aprender a colaborar para melhorar suas chances de sobrevivência e crescimento. A premissa de que existe uma relação de causa e efeito entre ações e recompensas é inerente à aprendizagem animal. Tal princípio, caracterizado por fortes capacidades de autoaprendizagem e adaptação, substancia a Programação Dinâmica Aproximada (ADP), proposta por Werbos (WERBOS, 1990). A aprendizagem adaptativa em tempo real de controladores ótimos para sistemas complexos desconhecidos têm sido solucionados por métodos de ADP (LEWIS e LIU, 2012) e (LIU e WEI, 2014).

A complexidade dos sistemas dinâmicos, devido aos requisitos de desempenho do projeto, exige controladores de malha fechada de alto desempenho para executar as tarefas programadas, tais como, aeronaves de alto desempenho, veículos não tripulados e robôs móveis que requerem uma alta precisão de deslocamentos e robustez (FONSECA, FERREIRA e REGO, 2013) e (RÊGO, 2014). A ADP fornece soluções para sistemas dinâmicos que demandam controladores mais complexos. Neste trabalho, o objeto de estudo é um VANT do tipo quadricóptero, veículo aéreo com mais de um rotor de giro rápido responsáveis por empurrar o ar para baixo, criando assim uma força de empuxo, mantendo o veículo no ar. É apresentado o projeto de um controlador ótimo adaptativo baseado em ADP para esses veículos.

Os quadricópteros, com quatro rotores independentes e seis graus de liberdade, os movimentos translacional e rotacional são acoplados, e apenas quatro entradas independentes (as velocidades das hélices – rotores). A dinâmica resultante é não linear, especialmente depois de contabilizar os efeitos aerodinâmicos. Além disso, aeronaves de asas rotativas necessitam de

um controle permanentemente atuante para manter sua estabilidade e para executar manobras, ou seja, eles devem fornecer seu próprio amortecimento para parar de se mover e permanecer estáveis. Devido a esses fatores, há um problema de controle a ser explorado (GIBIANSKY, 2014).

Um controlador ótimo visa minimizar uma métrica (índice de desempenho) do sistema dinâmico no tempo de modo a obter um sinal de controle que seja capaz de executar seu objetivo, respeitando as restrições e requisitos (VARGAS e PAGLIONE, 2015). Tal classe de controladores podem ser obtidos através do princípio do mínimo de Pontryagin e a equação de Euler-Lagrange (condição necessária), ou por meio dos métodos de Bellman da Programação Dinâmica com a subsequente equação HJB (RÊGO, 2014).

A teoria de otimização contribui de forma significativa para a sistematização e resolução de problemas de controle ótimo. A Programação Dinâmica é um método de otimização matemática utilizada para resolver problemas de decisão multiestágios. A Programação Dinâmica Aproximada (ADP) e a Aprendizagem por Reforço (RL) são paradigmas para o aprendizado online de controle ótimo. A aprendizagem por reforço moderna é uma síntese de ideias da teoria de controle ótimo, programação dinâmica, métodos de diferença temporal e aproximação de função (FONSECA, FERREIRA e REGO, 2013). A programação dinâmica é amplamente considerada como uma excelente forma de resolver problemas de controle ótimo quando um modelo do sistema está disponível. Todavia, a ideia de encontrar uma solução alternativa na ausência de um modelo foi explorada e experimentou um desenvolvimento no campo da Aprendizagem por Reforço (TEIXEIRA, 2016).

Os projetos de controle ótimo são tradicionalmente realizados *offline* e, tendo por base um modelo consistente da dinâmica da planta. No entanto, é importante projetar controladores ótimos adaptativos, que são controladores que aprendem online as soluções para problemas de controle ótimo sem ter conhecimento completo da dinâmica da planta. Controladores ótimos adaptativos aprendem a controlar sistemas com dinâmicas desconhecidas (ou parcialmente desconhecidas), usando dados, em tempo real, medidos ao longo da trajetória do sistema, com o fim de otimizar o sistema e ganhos dos controladores. O controle ótimo adaptativo toma uma abordagem similar à de uma estrutura neural a qual se baseia inteiramente em aprendizagem em tempo real (FONSECA, FERREIRA e REGO, 2013).

Nessa perspectiva, pesquisadores têm proposto na literatura abordagens que agregam conceitos e métodos de controle ótimo e controle adaptativo tendo por base paradigmas de Aprendizagem por Reforço e Programação Dinâmica Aproximada (ADP) (WERBOS, 1992) e

(SI, 2004). De acordo com (WERBOS, 2012), a ADP tanto pode ser vista como uma extensão de controle adaptativo que, devido ao lookahead (refere-se a olhar para frente no tempo) implícito, alcança estabilidade sob condições muito mais fracas que as formas bem conhecidas de controle adaptativo direto e indireto (ASTROM e WITTENMARK, 1994), e também ADP pode ser formulada como uma parte de controle ótimo que busca métodos gerais computacionalmente viáveis para o caso não-linear estocástico. Desta forma, ADP pode ser vista como um campo de pesquisa promissor para desenvolver controladores inteligentes inspirados em estruturas neurais que fornecem habilidades para lidar de forma eficiente com o grau de complexidade de sistemas não-lineares incertos e parcialmente observáveis e obter uma redução no esforço de controle durante o movimento de voo pairado do veículo, então pode-se desenvolver algoritmos de controles adaptativos e ótimos baseados em paradigmas de ADP. Isso permite mais aplicações em tempo real de controladores ótimos baseados em HDP e, consequentemente, AD-HDP.

A partir dessas informações, pode-se aplicar essas metodologias ao Centro de Lançamento em Alcântara (CLA), utilizando os quadricópteros para fins que beneficiem o controle de verificação e busca no espaço com muitos benefícios que serão citados adiante.

Das aplicações mais relevantes e ajustáveis ao Centro, pode-se utilizar os quadricópteros para uso na segurança. Observar a rampa de lançamento, verificando se o espaço encontra-se em normais padrões de segurança, se antecipando a possíveis colisões e salvamento de pessoas. Ainda no uso desse equipamento, pode-se utilizar para o resgaste da carga útil de um foguete, baseado nos dados de trajetória normal e informado pelo rastreio.

Todavia, outras aplicações dos drones ao CLA, sendo que o objetivo desse trabalho é otimizar o controle desse sistema em tempo real, se dão por testar os algoritmos de rastreio, diminuindo gastos com foguetes de treinamento. Sendo mais leve e mais lento, poderá ser útil como plataforma de treinamento no uso dos equipamentos/algoritmos de rastreio. E, também, utilizar sua capacidade de voo e tamanho para filmagens no momento do lançamento e análise visual do comportamento do foguete, etc.

1.2 Objetivos

O objetivo geral desse trabalho é aplicar formulações de métodos de Programação Dinâmica Aproximada para a solução *online* de sistemas de controle ótimo multivariáveis baseadas em projetos críticos adaptativos e controle de aprendizagem em um modelo de quadricóp-

tero, mais popularmente chamado drone. No contexto de ADP, é verificável os custos computacionais dos processos IV e IP, processos e iteração dos algoritmos propostos para obter políticas de decisão ótimas baseadas na Equação HJB-Riccati discreta para realização de projeto *online* de sistema de controle.

Neste contexto, propõe-se desenvolver uma metodologia de controle e algoritmos para o projeto e análise de quadricópteros sob a ação de controle no processo de voo pairado, ou melhor, estável em relação a um ponto Z do espaço, em que o principal enfoque é buscar a redução do esforço de controle, otimizando a capacidade de mantê-lo no ar por mais tempo, através da equação HJB-Ricatti associada ao problema de controle ótimo DT LQR, usando dados medidos ao longo das trajetórias do sistema. Isso será resolvido através do Erro da Diferença Temporal (TD) e da Aproximação da Função Valor (VFA). O Erro da Diferença Temporal está relacionado à Controle Adaptativo em que ajusta valores e ações *online* ao longo da trajetória e atualiza o valor em cada passo de tempo à medida em que as observações de dados são tomadas ao longo de uma trajetória. E a Aproximação da Função Valor resolve a equação TD por aproximar a função valor V_h usando um aproximador paramétrico.

Para alcançar o objetivo principal, que é a redução no esforço de controle durante o movimento de voo pairado do veículo, o qual estará sujeito a variações nos parâmetros da planta, é necessário, como objetivos secundários, desenvolver algoritmos de controles ótimos adaptativos baseados em paradigmas de Programação Dinâmica Aproximada, e realizar diversas simulações a fim de analisar o desempenho dos controladores. O desenvolvimento do projeto de pesquisa abrange várias áreas do conhecimento de controle, dentre elas estão: Controle Adaptativo, Controle Ótimo, Aprendizagem por Reforço e o Estudo do Modelo de Quadricóptero.

1.3 Organização do Trabalho

Esta dissertação está dividida em capítulos que descrevem os fundamentos e estado da arte dos quadricópteros, a modelagem do veículo, bem como a metodologia e algoritmos de ADP que foram aplicados ao sistema de controle do quadricóptero. Formulações de controle ótimo são expostas desde abordagens de Bellman da Programação Dinâmica Clássica até as abordagens que empregam Programação Dinâmica Aproximada e Aprendizagem por Reforço na estrutura ator-crítico, que substanciam o desenvolvimento do projeto *online* de controle ótimo adaptativo.

Inicialmente, no capítulo 2, são apresentados os fundamentos e o estado da arte dos quadricópteros, uma proposta da cosmovisão do veículo e seu desenvolvimento científico, e também seu uso nos dias atuais, tanto para fins civis quanto para uso militar, os projetos de desenvolvimentos desses veículos aéreos não tripulados e a otimização para comandos autônomos.

Os modelos cinemático e dinâmico de um VANT são fornecidos no capítulo 3. Para isso utilizam-se das formulações do modelo de Newton-Euler descritas em (GIBIANSKY, 2014), (VARGAS e PAGLIONE, 2015), (QUAN, 2017) e (TOMMASO, 2008), que aqui são deduzidas passo a passo. Neste trabalho, tais modelos serão necessários para obter um simulador do processo para avaliar a metodologia de projeto de controle ótimo que será utilizada para fornecer os resultados de simulação de modo *offline*.

No capítulo 4, apresentam-se as formulações matemáticas da metodologia de ADP, que englobam desde as formulações que necessitam do conhecimento completo da dinâmica da planta até as formulações *online*, que não dependem do modelo da planta para otimizar o processo. A equação de Hamilton-Jacobi-Bellman (HJB) é descrita e também esquemas de iteração de política, iteração de valor, diferenças temporais e aproximação da função valor. Esses elementos básicos contribuem para a formulação do problema e sua associação com a equação de Bellman para desenvolver os esquemas de iteração de política aproximada e parametrização do processo.

Após a descrição matemática do veículo e a abordagem de controle, o capítulo 5 descreve os resultados baseados nas formulações matemáticas com resposta *offline*, na primeira seção. Tais resultados do sistema, são encontrados aplicando os métodos de aprendizagem por reforço, Iterações de Política e Iterações Valor com a dinâmica da planta conhecida, utilizando o modelo linearizado deduzido no capítulo 3. Após essas ações aplicam-se os métodos do processo de ADP para o sistema *online*, sem considerar a dinâmica da planta, na segunda seção do capítulo. Utilizando a aproximação da função valor e o erro da diferença temporal para resolver a equação de Bellman, usando dados capturados, medidos ao longo da trajetória do sistema.

E, por fim, no capítulo 6, apresentam-se as conclusões e avaliação sobre as metodologias abordadas do projeto de controlador ótimo adaptativo para o quadricóptero via programação dinâmica aproximada para aproximação da solução da equação HJB-Riccati e sugestões de trabalhos futuros.

2. DESCRIÇÃO DO VEÍCULO E ESTADO DA ARTE

2.1 Fundamentos

Veículos Aéreos Não Tripulados (do inglês *Unmanned Aerial Vehicle* – UAV), isto é, uma aeronave sem pilotos a bordo. O voo dos UAVs pode ser controlado autonomamente por computador de bordo ou por controle remoto de um piloto no solo ou em outro veículo.

O importante progresso nos últimos anos em tecnologia de sensoriamento, armazenamento de energia de alta densidade e processamento de dados tornou possível o desenvolvimento de veículos aéreos não tripulados. Na classe, tem-se o multicóptero também chamado de multirotor. Pode ser considerado como um tipo de helicóptero, que possui três ou mais hélices. Também tem a capacidade de decolagem e aterrisagem vertical. O multicóptero mais popular é o quadricóptero, descrito nesse trabalho. Um quadricóptero tem quatro entradas de controle, que são as quatro velocidades angulares da hélice. Ao contrário do helicóptero de rotor único, o ajuste de elevação rápida é realizado pelo controle das velocidades angulares da hélice. Devido à estrutura de múltiplos rotores, seus momentos anti-torque podem ser cancelados uns pelos outros. Graças à estrutura simples, um multicóptero é fácil de usar e apresenta alta confiabilidade e baixo custo de manutenção. (QUAN, 2017).

Como citado também em (GIBIANSKY, 2014), um quadricoptero é um veículo voador que usa rotores de giro rápido para empurrar o ar para baixo, então cria uma força de propulsão mantendo o veículo no ar. Estes podem ser organizados em rotores coplanares, ambos fornecendo impulso para cima, mas com pares girando para direções opostas, com a intenção de equilibrar os torques exercidos sobre o corpo do veículo.

O controle de um quadricóptero tem sua complexidade no fato dele possuir os movimentos rotacional e translacional acoplados, o que gera seis graus de liberdade (três translacionais e três rotacionais) e apenas quatro entradas independentes (velocidades dos rotores). As dinâmicas resultantes são não lineares, e ainda contabilizam os efeitos aerodinâmicos. Devido a isso, aeronaves de asas rotativas necessitam de um controle permanentemente atuante para manter sua estabilidade e para executar manobras.

2.2 Estado da Arte

Nos últimos anos, o estado da arte em Veículos Aéreos Não Tripulados (UAV) de decolagem e aterrisagem vertical (*Vertical Take-off and Landing* – VTOL) tem recebido diversas contribuições, avanços em pesquisas e desenvolvimentos (pode-se consultar a Seção 2.2.1).

A pesquisa em VTOL-UAV tem se focado bastante na estrutura dos quadricópteros, estabilidade, configurações híbridas, tais como estruturas com direções rotacionais não simétricas e/ou com dois rotores direcionais, derivações de modelagens e tarefa com múltiplos agentes (TOMMASO, 2008). Os quadricópteros se tornaram interessantes para o ramo da pesquisa a partir do século XX, ao mesmo tempo que outros veículos aéreos, devido ao fato da capacidade destes de decolarem e pousarem verticalmente, e conseguirem realizar o movimento de pairarem durante a execução do voo, isto é, maior manobrabilidade e versatilidade quando comparados a aviões, que necessitam de velocidade para manter a força de empuxo entre as asas com o propósito de gerarem sustentação.

Segundo (TOMMASO, 2008), grande parte das publicações científicas acerca de quadricópteros tem se concentrado na solução de algoritmos de controle e, com técnicas mais usuais, tais como: 1. A Teoria de Lyapunov (MURRIERI, BOUABDALLAH e SIEGWART., 2004), a qual garante, sob certas condições, a estabilidade assintótica do quadricóptero; 2. A estrutura de realimentação PD^2 (*Proportional-Derivative*) (NOTH, BOUABDALLAH e SIEGWART, 2005), com propriedade de convergência exponencial devido à compensação dos termos Coriolis e giroscópicos, e a estrutura PID (*Proportional-Integral-Derivative*) (HOFFMAN, HUANG, *et al.*, 2007) (RAWASHDEH, YANG, *et al.*, 2009), a qual não requer alguns parâmetros específicos do modelo e a lei de controle é muito mais fácil de implementar; 3. O Regulador Linear Quadrático (*Linear Quadratic Regulator* – LQR) (LOZANO, CASTILLO e DZUL, 2005), cuja vantagem é que o sinal de entrada ótimo é obtido a partir da realimentação completa dos estados (resolvendo a equação de Ricatti). No entanto, a solução analítica para a equação de Ricatti é difícil de ser obtida; 4. Técnicas Adaptativas (FRADKOV, ANDRIEVSKY e PEAUCELLE, 2005), (MOREL e LEONESSA, 2006), fornecem bom desempenho com parâmetros incertos e dinâmicas não modeladas.

Existem outros algoritmos de controle para melhoramento do desempenho do quadricóptero, tais como Técnicas Fuzzy (COZA e MACNAB, 2006), Rede Neurais (TARBOUCHI, DUNFIED e LABONTE, 2004), Controle "*backstepping*" (DIKMEN, ARISOY e TEMELTAS, 2009), Controle baseado em realimentação visual (HAMEL, GUENARD e MAHONY, 2007), e Aprendizagem por Reforço ((JANG, WASLANDER, *et al.*, 2006), (LEWIS e VRABIE D., 2009), (LEWIS e LIU, 2012), a qual será desenvolvida nesse trabalho.

As contribuições do presente trabalho focam-se principalmente na modelagem dinâmica precisa e revisada, formulações de metodologias de ADP e algoritmos online para o projeto e análise de sistemas de quadricópteros, sob ação de decolagem, buscando redução no esforço de controle.

2.2.1 Histórico de Revisão

Nesta seção, apresentam-se as fases e o desenvolvimento dos multicópteros e sacia-se um questionamento: por que os multicópteros são mais populares agora do que em 2005? A resposta pode ser extraída da história da tecnologia dos multicópteros. Geralmente, a história pode ser dividida em cinco períodos: período de dormência (antes de 1990), período de crescimento (1990-2005), período de desenvolvimento (2005-2010), período de atividade (2010-2013) e período de expansão (2013 – hoje) (QUAN, 2017).

2.2.1.1 A primeira fase (Antes de 1990): Período de Dormência.

Já em 1907, na França, os irmãos Breguet construíram seu primeiro helicóptero de transporte humano que era um quadricóptero. Etienne Oemichen, engenheira, também começou a experimentar projetos de asas rotativas em 1920. Seu primeiro modelo não conseguiu levantar do chão. No entanto, depois de alguns cálculos e reprojetos, sua segunda aeronave, a Oemichen No. 2, figura 2.1, estabeleceu um recorde mundial para helicópteros em 1923, permanecendo no ar por até 14 minutos. Em 1921, George De Bothezat e Ivan Jerome foram contratados para desenvolver um quadricóptero para o Corpo Aéreo do Exército dos EUA. Somente em meados da década de 1950 que o primeiro quadricóptero real voou, o qual foi projetado por Marc Adman Kaplan. O Modelo Convertawings “A”, figura 2.2, foi lançado em 1956. Em 1957, o Exército contratou a Curtiss-Wright para desenvolver o VZ-7, figura 2.3, como um protótipo de jipe voador, para transportar pequenas quantidades de homens e máquinas em terrenos acidentados. No entanto, eles não conseguiram atender aos requisitos de altitude e velocidade especificados no contrato do Exército. Os quadricópteros não tinham vantagens sobre os helicópteros de rotor único em escala completa. É por isso que o Exército dos EUA cancelou esses projetos. Desde então, os multicópteros foram quase abandonados. Durante os 30 anos seguintes, não houve muita melhora e esse tipo de aeronave atraiu pouca atenção.

Figura 2.1 Oemichen No. 2 (1923)

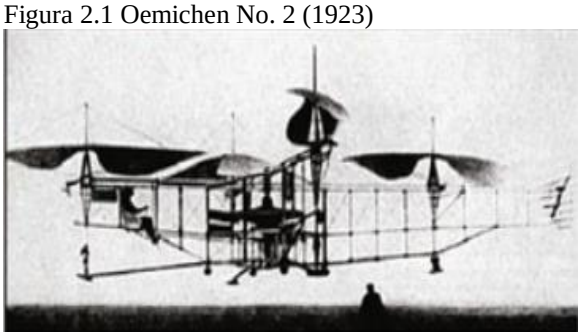


Figura 2.2 Convertawings “A” (1956)

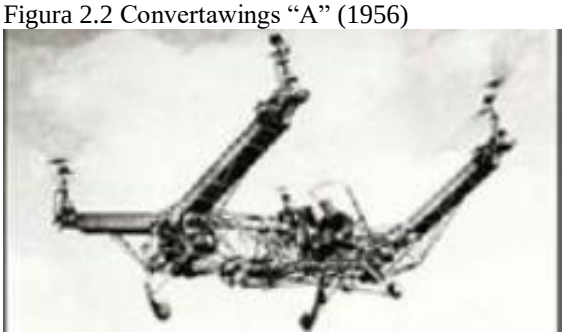
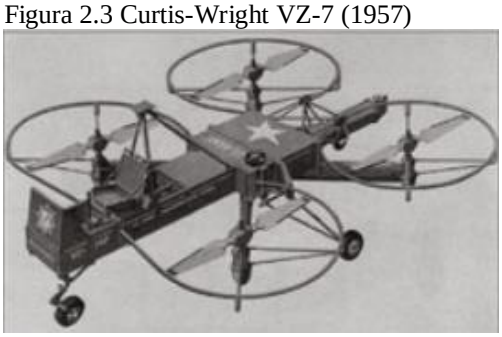


Figura 2.3 Curtis-Wright VZ-7 (1957)



2.2.1.2 A segunda fase (1990-2005): Período de Crescimento

Esta fase caracteriza-se por duas etapas: 1) A Pesquisa. Com o desenvolvimento da velocidade de computação e o tamanho de microcomputadores, tais como o Microcomputador de Chip Único (*Single Chip Microcomputers* - SCMs) e Processadores de Sinais Digitais (Digital Signal Processor-DSP), teve uma melhoria significativa, diminuindo os ruídos provocados e o projeto de pequenos quadricópteros. Nessa etapa, pesquisadores começaram a construir modelos e algoritmos de controle do projeto; 2) O Produto. Já que a aplicação do uso dos quadricópteros se afastou da aplicação militar, passou a ser utilizado por consumidores civis, a princípio no mercado de brinquedos. Na década de 1990, um mini quadricóptero, chamado Keyence Gyrosaucer, foi comercializado no Japão, figura 2.4. Nesse mesmo ano, o engenheiro Mike Dammar desenvolveu seu próprio quadricóptero movido a bateria, nos EUA. Em 2002, figura 2.5, um quadricóptero foi inventado e desenvolvido na competição Jovens Pesquisadores na Alemanha. Este modelo foi chamado posteriormente de Silverlit X-UFO.

Figura 2.4 Keyence Gyrosaucer (1990)

Figura 2.5 Quadricóptero Silverlit X-UFO (2002)

2.2.1.3 A terceira fase (2005-2010): Período de Desenvolvimento

Neste período a pesquisa aumentou significativamente, levando a inúmeras publicações de artigos. Muitos pesquisadores construíram seus próprios multicópteros para testar os algoritmos, especialmente os de controle de atitude. No entanto, construir um multicóptero não era tarefa simples, pois o comércio de equipamentos eletrônicos ainda não estava amadurecido e popular. Com isso, eram utilizados quadricópteros comerciais e o sistema óptico de captura e movimento para construir o ambiente de testes. A primeira geração de produtos foi lançada em abril de 2006, pela Microdrones GmbH, o MD4-200, figura 2.6. Em outubro de 2006, uma grande comunidade de pilotos automáticos MikroKopter foi criada por Holger Buss e Ingo Busker. Já em meados de 2007, a aeronave equipada com MikroKopter poderia pairar de forma constante. Em pouco tempo, outros componentes foram adicionados. Portanto, tornou-se possível realizar voos semi-autônomos. A Ascending Technologies GmbH, na Alemanha, projetou multicópteros para fins profissionais, civis e de pesquisa, em 2007. Em 2008, a Draganflyer projeta o Draganflyer X6, figura 2.7, ele apresenta uma construção de fibra de carbono, uma estrutura dobrável e tecnologia de piloto automático. Nessa época não era fácil de controlar os veículos, porque não tinham receptores GPS instalados. Mais importante ainda, os dispositivos inteligentes não foram inventados, dificilmente poderiam ser usados para fins de entretenimento ou para tirar fotos.

Figura 2.6 MD4-200 (2006)

Figura 2.7 Draganflyer (2008)

2.2.1.4 A quarta fase (2010-2013): Período de Atividade.

Nesse novo período, é visível a presença de inúmeras frotas de minúsculos robôs voadores realizando séries de manobras complexas, trabalhando juntas em tarefas. Em 2012, a Revista IEEE Robotics and Automation publicou uma edição especial sobre robótica aérea e a plataforma de quadricópteros, que resumiu e demonstrou algumas tecnologias de ponta. Nesta fase, muitos pilotos automáticos de código aberto sobre multicópteros surgiram, o que reduziu o limiar de construção de multicópteros para iniciantes. O desenvolvimento de dispositivos inteligentes, naquela época, de tecnologias relacionadas, impulsionou o desenvolvimento de multicópteros. Como exemplo, tem-se o quadricóptero chamado AR Drone, figura 2.8. Sua tecnologia e ideologia também eram muito avançadas. Inicialmente, tinha uma câmera voltada para baixo para medir o fluxo óptico e dois localizadores de alcance ultrassônico para medir a altitude, contribuindo para que sua velocidade pudesse ser obtida usando algoritmos de estimação. O AR Drone oferecia um Kit de Desenvolvimento de Software (Software Development Kit - SDK) para que os pesquisadores pudessem desenvolver suas próprias aplicações, o que significava que ele estava equipado com potenciais de pesquisa. Como resultado, o produto se espalhou rapidamente na academia. No final de 2012, a DJI (Da-Jiang Innovations Science and Technology Co., Ltd) lançou uma solução *all-in-one*, figura 2.9, quadricóptero "Phantom" pronto para voar.

Figura 2.8 AR Drone

Figura 2.9 Phanton

2.2.1.5 A quinta fase (2013-hoje): Período de Expansão

Nesta fase, a pesquisa em multicópteros tendia a torná-los mais autônomos e cooperativos. Em junho de 2013, Raffaello D'Andrea na ETH Zurich fez um discurso no TED Global 2013 sobre "Atletismo de Máquina", ou seja, a capacidade das máquinas de realizarem proezas dinâmicas que exploram plenamente suas capacidades físicas, incluindo captura de jogo, equilíbrio e tomada de decisões em conjunto e uma demonstração de um quadricóptero controlado

por Kinect. Em junho de 2015, uma seção especial sobre Inteligência de Máquina da revista Nature publicou o artigo “Science, technology and the future of small autonomous drones”. Este artigo de revisão sumarizou os desafios em projeto e fabricação, detecção e controle e tendências futuras de pesquisa no campo de pequenos drones. Em agosto de 2013, o Projeto PX4 e a 3D Robotics anunciaram o Pixhawk - um avançado projeto de piloto automático de hardware aberto para multicópteros, aeronaves de asa fixa, veículos terrestres e veículos anfíbios. A Pixhawk foi projetada para melhorar a facilidade de uso e a confiabilidade, oferecendo recursos de segurança sem precedentes em comparação com as soluções existentes. No final de 2013, um vídeo divulgado pela Amazon que prometia entregar pequenos pacotes nas casas em apenas 30 minutos pela "Prime Air" - um sistema de entrega futuro, consistia de quadricópteros, dentro de 5 anos. Essa ideia diminuiu ainda mais a distância entre os multicópteros e os clientes de massa. Mais e mais multicópteros foram projetados nesta fase.

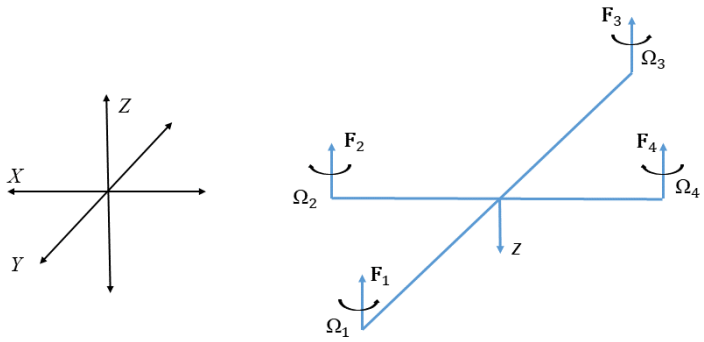
3. DESCRIÇÃO MATEMÁTICA E MODELAGEM DO VEÍCULO

3.1 Conceitos Básicos

Neste capítulo, serão apresentadas as informações específicas da arquitetura e do modelo do quadricóptero, tendo por base a equação genérica do corpo rígido de seis graus de liberdade. Mas antes de ser definido os sistemas de coordenadas, é vital relembrar a regra da mão direita. Visto que o polegar da mão direita apontará para a direção positiva de um determinado eixo, indicando a posição do corpo, oz. O dedo indicador aponta para a direção positiva longitudinal, ox, e o dedo do meio para a direção positiva transversal, seu movimento lateral, oy (QUAN, 2017).

Além disso, com o propósito de determinar a direção positiva de uma rotação, o dedo polegar aponta para a direção positiva do eixo de rotação, produto vetorial dos eixos $ox \times oy$, e os dedos curvados, indicador sobre o médio, apontam para a direção positiva da rotação (QUAN, 2017). Os sistemas de coordenadas usados nesse capítulo e a direção positiva dos ângulos obedecem a regra da mão direita, com a ressalva que a gravidade aponta no sentido negativo da direção oz, conforme mostra a figura 3.1 (GIBIANSKY, 2014).

Figura 3.1: Modelo genérico de um quadricóptero.



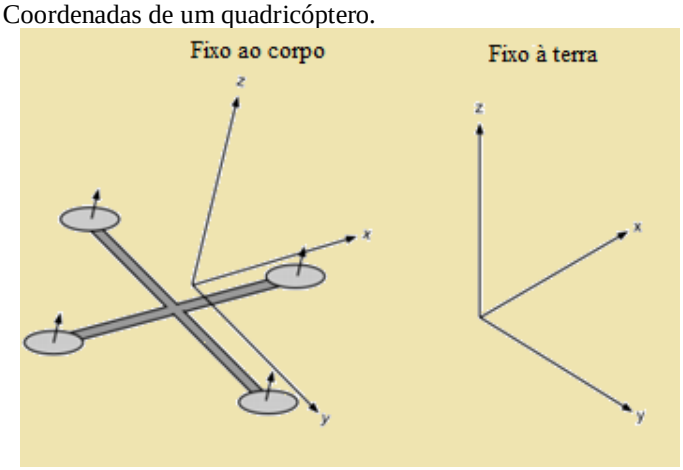
Os quadricópteros são caracterizados por possuírem seis graus de liberdade. Três graus de liberdade nas posições lineares – translacionais – $X, Y\ e\ Z$ (ox, oy, oz, com o, o ponto de origem), e três nas posições angulares – rotacionais – definidos pelos ângulos do sistema de coordenadas do corpo no sistema fixo da terra, representado pelos ângulos de Euler, $\Theta^E = [\emptyset\ \ \theta\ \psi]^T$, que são conhecidos como rolagem (*Roll*), Arfagem (*Pitch*) e guinada (*Yaw*), respectivamente e descrevem a orientação do veículo (VARGAS e PAGLIONE, 2015).

A dinâmica do veículo pode ser descrita pela sua posição, orientação, velocidade linear e angular e suas respectivas variações no tempo. O sistema de coordenadas $\Gamma^{\boldsymbol{E}} = [X\ \ Y\ \ Z]^T$ é usado para estudar os estados dinâmicos do quadricóptero em relação à superfície da Terra e

determinar a posição tridimensional, considerada plana (QUAN, 2017). $\Gamma^{\boldsymbol{E}}$ é o vetor de posição linear, $\boldsymbol{V^B} = [u \ v \ w]^T$ é o vetor de velocidade linear e $\boldsymbol{\omega^B} = [p \ q \ r]^{\mathrm{T}}$ o vetor de velocidade angular; sendo as componentes as taxas de variações dos ângulos de Euler em cada instante de tempo (VARGAS e PAGLIONE, 2015).

Na figura 3.2, pode-se visualizar os dois sistemas de coordenadas, *E-frame*, Sistema de Coordenadas Fixo à Terra (*Earth Inertial Reference*), e, *B-frame*, Sistema de Coordenadas Fixo ao Corpo (*Body-fixed Reference*). O *B-frame* é definido pela orientação do quadricóptero, com os eixos dos rotores apontando na direção positiva de z e os braços nas direções x e y. Este sistema está acoplado ao corpo do veículo, coincidindo com o centro da estrutura do quadricóptero. As velocidades linear ($\boldsymbol{V^B}$ [m.s-1]) e angular ($\boldsymbol{\omega^B}$ [rad.s-1]), as forças ($\boldsymbol{F^B}$ [N]) e os torques ($\boldsymbol{\tau^B}$ [N.m]) são definidos neste sistema, como mostrado na figura 3.1. Enquanto as posições linear ($\Gamma^{\boldsymbol{E}}$ [m]) e angular ($\boldsymbol{\Theta^E}$ [rad]) no sistema *E-frame*.

Figura 3.2: Sistemas de Coordenadas de um quadricóptero.



3.2 Descrição do Modelo Matemático

3.2.1 Análise Cinemática

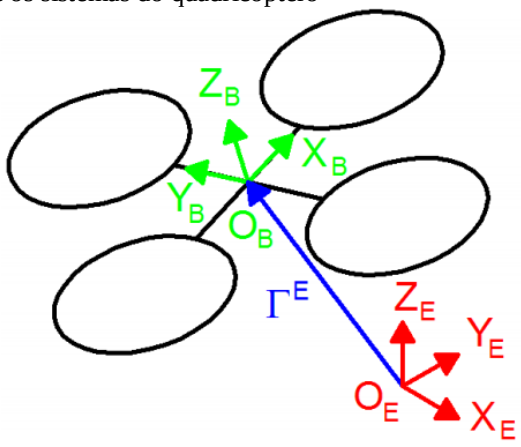
Pode-se descrever, segundo (TOMMASO, 2008), o comportamento dos sistemas pelas seguintes análises:

A posição linear, $\Gamma^{\boldsymbol{E}}$, de qualquer veículo aéreo, quadricóptero, etc; é determinada pelas coordenadas do vetor entre o centro de gravidade, *cg*, do *B-frame* e do *E-frame* em relação a *E-frame* de acordo com a equação (3.1) e da figura 3.3, sendo o cg o ponto de origem do sistema

$$\Gamma^{\boldsymbol{E}} = [X \ Y \ Z]^T \tag{3.1}$$

A figura 3.3 mostra os sistemas e suas relações:

Figura 3.3: Transformação entre os sistemas do quadricóptero



A atitude do veículo, ou posição angular $\boldsymbol{\Theta^E}$, de um quadricóptero é definida pela orientação do sistema *B-frame* em relação ao sistema *E-frame*. Isto pode ser obtido executando três consecutivas rotações dos ângulos de Euler (ângulo de guinada ψ, ângulo de arfagem θ e ângulo *de rolagem* ϕ); sobre os principais eixos que transformam de *E-frame* para o *B-frame*, ortogonais entre si.

A equação (3.2) mostra o vetor de atitude

$$\boldsymbol{\Theta^E} = [\phi \;\; \theta \;\; \psi]^{\mathrm{T}} \tag{3.2}$$

Como dito anteriormente, as velocidades linear $\boldsymbol{V^B}$ e angular $\boldsymbol{\omega^B}$ são expressas no sistema fixo ao corpo *B-frame*, e seus vetores de velocidades são definidos pelas equações (3.3) e (3.4)

$$\boldsymbol{V^B} = [u \;\; v \;\; w]^T \tag{3.3}$$

$$\boldsymbol{\omega^B} = [p \;\; q \;\; r]^T \tag{3.4}$$

Um procedimento interessante e possível é combinar as quantidades linear e angular dos sistemas para dar uma completa representação do corpo no espaço. Dois novos *vetores* são formados e definidos, sendo: vetor de *posição generalizado* $\boldsymbol{\xi}[+]$, a composição dos vetores posição linear e angular do quadricóptero, e o vetor de *velocidade generalizada* $\boldsymbol{\nu}[+]$, a união dos vetores velocidades linear e angular do quadricóptero, como demonstrado nas equações (3.5) e (3.6)

$$\boldsymbol{\xi} = \begin{bmatrix} \Gamma^{\boldsymbol{E}} \\ \boldsymbol{\Theta}^{\boldsymbol{E}} \end{bmatrix} = [X \;\; Y \;\; Z \;\; \phi \;\; \theta \;\; \psi]^T \tag{3.5}$$

$$\boldsymbol{\nu} = [\boldsymbol{V}^{\boldsymbol{B}} \; \boldsymbol{\omega}^{\boldsymbol{B}}]^T = [u \;\; v \;\; w \;\; p \;\; q \;\; r]^T \tag{3.6}$$

A equação (3.7) descreve a cinemática de um corpo rígido de seis graus de liberdade

$$\dot{\xi} = J_{\boldsymbol{\Theta}} \boldsymbol{\nu} \tag{3.7}$$

em que $\dot{\boldsymbol{\xi}}[+]$ é o vetor de velocidade generalizada com relação ao sistema *E-frame*, $\boldsymbol{\nu}[+]$ o vetor de velocidade generalizada em relação ao sistema *B-frame* e $\boldsymbol{J_{\Theta}}[-]$ a matriz generalizada.

A matriz generalizada $\boldsymbol{J_{\Theta}}$ é composta de quatro sub-matrizes de acordo com a equação (3.8)

$$\boldsymbol{J_{\Theta}} = \begin{bmatrix} \boldsymbol{R_{\Theta}} & 0_{3\times3} \\ 0_{3\times3} & \boldsymbol{T_{\Theta}} \end{bmatrix} \tag{3.8}$$

A matriz de rotação $\boldsymbol{R_{\Theta}}[-]$ é obtida pelas rotações sucessivas dos três ângulos de Euler da seguinte forma:

- Rotação em torno do eixo z do ângulo ψ (guinada) utilizando a matriz $\boldsymbol{R}(\psi, z)$;
- Rotação em torno do novo eixo y, gerado da rotação de z, do ângulo θ(arfagem) utilizando a matriz $\boldsymbol{R}(\theta, y)$;
- Rotação sobre o novo eixo x, gerado das rotações de z e o novo y, do angulo ϕ (rolamento) utilizando a matriz $\boldsymbol{R}(\phi, x)$;

Com isso, a matriz $\boldsymbol{R_{\Theta}}[-]$ é definida pela equação (3.9)

$\boldsymbol{R_{\Theta}} = \boldsymbol{R}(\psi, z)\, \boldsymbol{R}(\theta, y)\, \boldsymbol{R}(\phi, x) =$

$$\begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3.9}$$

A matriz de transferência $\boldsymbol{T_\theta}$ pode ser determinada por meio da resolução das taxas de variações dos ângulos de Euler $\dot{\boldsymbol{\Theta}}^{\boldsymbol{E}}$ no sistema fixo ao corpo, conforme é mostrado nas equações (3.10), (3.11) e (3.12)

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{R}(\phi, x)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \boldsymbol{R}(\phi, x)^{-1} \boldsymbol{R}(\theta, y)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \boldsymbol{T_\theta}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.10}$$

$$\boldsymbol{T_\theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3.11}$$

$$\boldsymbol{T_\theta} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \tag{3.12}$$

A relação entre a velocidade linear do sistema fixo ao corpo $\boldsymbol{V^B}$ e aquela do sistema da terra $\boldsymbol{V^E}$ (ou $\dot{\Gamma}^{\boldsymbol{E}}$) [$m\ s^{-1}$] envolve a matriz de rotação $\boldsymbol{R_\theta}$, como mostrado na equação (3.13)

$$\boldsymbol{V^E} = \dot{\Gamma}^{\boldsymbol{E}} = \boldsymbol{R_\theta V^B} \tag{3.13}$$

As equações (3.14) e (3.15) mostram que é possível também relacionar a velocidade angular no sistema da terra (ou variações de Euler) $\dot{\boldsymbol{\Theta}}^{\boldsymbol{E}}$[rad^{-1}] para aquela no sistema fixo ao corpo $\boldsymbol{\omega^B}$ por meio da matriz de transferência $\boldsymbol{T_\theta}[-]$, da mesma forma que para a velocidade linear

$$\boldsymbol{\omega^B} = \boldsymbol{T_\theta}^{-1} \dot{\boldsymbol{\Theta}}^{\boldsymbol{E}} \tag{3.14}$$

$$\dot{\boldsymbol{\Theta}}^{\boldsymbol{E}} = \boldsymbol{T_\theta}\, \boldsymbol{\omega^B} \tag{3.15}$$

3.2.2 Análise Dinâmica

As equações de movimentos são mais convenientemente formuladas no sistema fixo ao corpo pelas seguintes razões (VARGAS e PAGLIONE, 2015) e (TOMMASO, 2008)

- A matriz de inércia é invariante no tempo;
- A simetria do corpo do veículo pode ser usada para simplificar as equações, com $\boldsymbol{I_{yz}}$ e $\boldsymbol{I_{xy}}$ nulas;
- As forças de controle são geralmente vindas do sistema fixo ao corpo;
- As medições tomadas em voo são convertidas para o sistema fixo ao corpo.

A dinâmica de um corpo rígido de seis graus de liberdade leva em consideração a massa do corpo m[kg] e a matriz de Inércia $\boldsymbol{I}$[N m^2]. Pode-se observar como as equações da dinâmica do quadricóptero se formam a partir

- do *Primeiro Axioma de Euler*, da *Segunda Lei de Newton* seguem as derivações dos componentes lineares do movimento do corpo, de acordo com a equação (3.16)

$$\begin{aligned} m\ddot{\Gamma}^E &= F^E \\ m\dot{\overbrace{\boldsymbol{R_\theta} V^B}} &= \boldsymbol{R_\theta} F^B \\ m(\boldsymbol{R_\theta}\dot{V^B} + \dot{\boldsymbol{R_\theta}} V^B) &= \boldsymbol{R_\theta} F^B \\ m\boldsymbol{R_\theta}(\dot{V^B} + \omega^B \times V^B) &= \boldsymbol{R_\theta} F^B \\ m(\dot{V^B} + \omega^B \times V^B) &= F^B \end{aligned} \tag{3.16}$$

- e dos componentes angulares, extraídos do *Segundo Axioma de Euler* e da *Segunda Lei de Newton*, conforme a equação (3.17)

$$\begin{aligned} I\ddot{\Theta}^E &= \boldsymbol{\tau}^E \\ I\dot{\overbrace{\boldsymbol{T_\theta}\omega^B}} &= \boldsymbol{T_\theta}\boldsymbol{\tau}^B \\ &\ldots \\ I\dot{\boldsymbol{\omega}^B} + \boldsymbol{\omega}^B \times (I\boldsymbol{\omega}^B) &= \boldsymbol{T_\theta}\boldsymbol{\tau}^B \end{aligned} \tag{3.17}$$

Relacionando as equações (3.16) e (3.17), pode-se descrever o movimento de um quadricóptero, movimento de um corpo rígido de seis graus de liberdade em forma matricial pela equação (3.18)

$$\begin{bmatrix} m\boldsymbol{I_{3\times3}} & \boldsymbol{O_{3\times3}} \\ \boldsymbol{O_{3\times3}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}}^{\boldsymbol{B}} \\ \dot{\boldsymbol{\omega}}^{\boldsymbol{B}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega} \times (m\boldsymbol{V}^{\boldsymbol{B}}) \\ \boldsymbol{\omega}^{\boldsymbol{B}} \times (\boldsymbol{I}\boldsymbol{\omega}^{\boldsymbol{B}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^{\boldsymbol{B}} \\ \boldsymbol{\tau}^{\boldsymbol{B}} \end{bmatrix} \tag{3.18}$$

Para a equação (3.18) considera-se que o ponto central do quadricóptero coincide com seu centro de massa e que a suas estruturas em cruz coincide com os eixos principais de inércia. Assim, um vetor generalizado de força pode ser definido pela equação (3.19)

$$\Lambda = [F^B \quad \tau^B]^T = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \tag{3.19}$$

Reescrevendo a equação (3.18) do movimento do quadricóptero, resulta na seguinte equação

$$M_B \dot{v} + C_B(v).v = \Lambda \tag{3.20}$$

$\dot{v}$ é *o vetor generalizado de aceleração no B-frame*;
M_B é *a matriz de Inercia − uma matriz diagonal e constante*;
$C_B(v)$ é *a matriz centrípeta de Coriolis*

A equação (3.20) é totalmente genérica e tem seu uso para representar o modelo do quadricóptero, então o último vetor Λ conterá informações sobre sua dinâmica. Essas informações estão divididas em três partes.

1) A primeira contribuição é o vetor gravitacional $\boldsymbol{G_B}(\boldsymbol{\xi})$ que é gerado a partir da aceleração da gravidade [m/s²]. A equação (3.21) mostra as transformações para obter $\boldsymbol{G_B}(\boldsymbol{\xi})$

$$\boldsymbol{G_B}(\boldsymbol{\xi}) = \begin{bmatrix} F_G^B \\ 0_{3\times1} \end{bmatrix} = \begin{bmatrix} R_\Theta^{-1} F_G^E \\ 0_{3\times1} \end{bmatrix} = \begin{bmatrix} R_\Theta^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ 0_{3\times1} \end{bmatrix} = \begin{bmatrix} mg.s\theta \\ -mg.c\theta s\phi \\ -mg.c\theta s\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.21}$$

2) Cada par de hélice gira em sentido oposto ao outro par, duas hélices giram no sentido horário e outras duas giram no anti-horário, ocorre um desequilíbrio geral toda vez que a soma algébrica das velocidades dos rotores não é igual a zero. Então, a segunda contribuição considera os efeitos giroscópicos produzidos pela rotação das hélices. Se, além disso, as taxas de inclinação também forem diferentes de zero, o quadrimotor experimenta um torque giroscópico de acordo com a equação (3.22)

$$O_B(v)\Omega = \begin{bmatrix} 0_{3x1} \\ -\sum_{k=1}^{4} J_{TP}\left(\omega^B \times \begin{bmatrix}0\\0\\1\end{bmatrix}\right)(-1)^k\Omega_k \end{bmatrix} = \begin{bmatrix} 0_{3\times1} \\ J_{TP}\begin{bmatrix}-q\\p\\0\end{bmatrix}\Omega \end{bmatrix} =$$

$$J_{TP}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix}\Omega \tag{3.22}$$

$\boldsymbol{O_B(v)}$ é *a matriz dos efeitos giroscópcos devido a rotação das hélices*;
J_{TP} é *a matriz de Inércia Polar em torno do eixo da hélice*.

A equação (3.23) define o vetor $\boldsymbol{\Omega}[\mathrm{rad}^{-1}]$ que contém a velocidade total, utilizada em (3.22), e a velocidade de cada hélice

$$\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \qquad \boldsymbol{\Omega} = \begin{bmatrix}\Omega_1\\\Omega_2\\\Omega_3\\\Omega_4\end{bmatrix} \tag{3.23}$$

3) A terceira contribuição, demonstrada na equação (3.24), leva em consideração as forças e torques produzidos diretamente pelas entradas do movimento principal. Segue-se que ambas, forças e torques, são proporcionais à velocidade das hélices ao quadrado. Portanto, a matriz de movimento $\boldsymbol{E_B}$ é multiplicada por $\boldsymbol{\Omega^2}$ para obter o vetor de movimento $\boldsymbol{U_B(\Omega)}$. Outros conceitos são resultantes desta alocação, por exemplo o coeficiente de empuxo 'b'[N.s²] e coeficiente de arrasto 'd'[N.m.s²]

$$U_B(\Omega) = E_B\Omega^2 = \begin{bmatrix}0\\0\\U_1\\U_2\\U_3\\U_4\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ bl(\Omega_4^2 - \Omega_2^2) \\ bl(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{bmatrix} \tag{3.24}$$

Pelo que foi demonstrado na equação (3.24) é possível identificar uma matriz constante $\boldsymbol{E_B}$ que multiplicada pela velocidade das hélices ao quadrado $\boldsymbol{\Omega^2}$ produz o vetor de movimento $\boldsymbol{U_B(\Omega)}$. A equação (3.25) mostra a matriz de movimento

$$E_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \tag{3.25}$$

A partir da equação (3.20) possível descrever a dinâmica do quadricóptero, considerando estas três últimas contribuições de acordo com a equação (3.26)

$$M_B \cdot \dot{\nu} + C_B(\nu).\nu = \boldsymbol{G}_B(\boldsymbol{\xi}) + \boldsymbol{O}_B(\boldsymbol{\nu}).\boldsymbol{\Omega} + \boldsymbol{E}_B\boldsymbol{\Omega}^2 \tag{3.26}$$

$$\Lambda = Força\ gravitacional + Força\ giroscopca + Forças\ dos\ motores$$

Isolando a derivada do vetor de velocidade generalizada com relação ao sistema referencial do corpo $\dot{\boldsymbol{\nu}}$ na equação (3.26), obtém-se a equação (3.27)

$$\dot{\nu} = M_B^{-1}(-C_B(\nu).\nu + \boldsymbol{G}_B(\boldsymbol{\xi}) + \boldsymbol{O}_B(\boldsymbol{\nu}).\boldsymbol{\Omega} + \boldsymbol{E}_B\boldsymbol{\Omega}^2) \tag{3.27}$$

A equação (3.27) é expressa na forma de sistema de equações em (3.28)

$$\begin{cases} \dot{u} = (ur - wq) + g.s\theta \\ \dot{v} = (wp - ur) - g.c\theta s\phi \\ \dot{w} = (uq - vp) - g.c\theta s\phi + \dfrac{U_1}{m} \\ \dot{p} = \dfrac{I_{YY} - I_{ZZ}}{I_{XX}}qr - \dfrac{J_{TP}}{I_{XX}}q\Omega + \dfrac{U_2}{I_{XX}} \\ \dot{q} = \dfrac{I_{ZZ} - I_{XX}}{I_{YY}}pr - \dfrac{J_{TP}}{I_{YY}}q\Omega + \dfrac{U_3}{I_{YY}} \\ \dot{r} = \dfrac{I_{XX} - I_{YY}}{I_{ZZ}}pq + \dfrac{U_4}{I_{ZZ}} \end{cases} \tag{3.28}$$

As entradas de controle do sistema, que são as velocidades das hélices, são dadas na equação (3.29)

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = bl(\Omega_4^2 - \Omega_2^2) \\ U_3 = bl(\Omega_3^2 - \Omega_1^2) \\ U_4 = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \tag{3.29}$$

O sistema dinâmico do quadricóptero, na equação (3.28), está descrito no sistema de fixo ao corpo, mas pode-se representar em um sistema composto, com as equações lineares pelo sistema fixo à terra e as equações angulares fixo ao corpo. Este novo referencial será chamado *H-frame*, sistema híbrido. Esta nova referência é adotada porque é fácil expressar a dinâmica combinada com o controle (em particular para a posição vertical no referencial inercial da Terra). A equação (3.30) mostra o vetor de velocidade generalizada de quadricóptero com relação ao *H-frame* ($\boldsymbol{\zeta}$[+]).

$$\zeta = [\dot{\Gamma}^E \quad \omega^B]^T = [\dot{X} \ \dot{Y} \ \dot{Z} \ p \ q \ r]^T \tag{3.30}$$

A equação (3.31) é a representação matricial da dinâmica do quadricóptero no *H-frame*

$$M_H\dot{\zeta} + C_H(\zeta)\zeta = G_H + O_H(\zeta)\Omega + \boldsymbol{E}_H(\boldsymbol{\xi})\boldsymbol{\Omega}^2 \tag{3.31}$$

Resolve-se a equação (3.31) isolando a derivada do vetor de velocidade generalizada, obtendo-se, desta forma, a equação (3.32)

$$\dot{\zeta} = M_H^{-1}(-C_H(\zeta)\zeta + G_H + O_H(\zeta)\Omega + \boldsymbol{E}_H(\boldsymbol{\xi})\boldsymbol{\Omega}^2) \tag{3.32}$$

E como uma forma de sistema de equações do movimento para os seis graus de liberdade (6-DOF), tem-se a equação anterior em (3.33)

$$\begin{cases} \ddot{X} = (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)\dfrac{U_1}{m} \\ \ddot{Y} = (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi)\dfrac{U_1}{m} \\ \ddot{Z} = -g + (\cos\theta \cos\phi)\dfrac{U_1}{m} \\ \dot{p} = \dfrac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \dfrac{J_{TP}}{I_{XX}} q\Omega + \dfrac{U_2}{I_{XX}} \\ \dot{q} = \dfrac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \dfrac{J_{TP}}{I_{YY}} q\Omega + \dfrac{U_3}{I_{YY}} \\ \dot{r} = \dfrac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \dfrac{U_4}{I_{ZZ}} \end{cases} \tag{3.33}$$

Nesse ínterim, pode-se reescrever a equação (3.33) com os valores das variáveis de estado, e demonstrar na equação (3.34)

$$\dot{x} = f(x,u) = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ (\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)\dfrac{U_1}{m} \\ (-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi)\dfrac{U_1}{m} \\ -g + (\cos\theta \cos\phi)\dfrac{U_1}{m} \\ \dfrac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \dfrac{J_{TP}}{I_{XX}} q\Omega + \dfrac{U_2}{I_{XX}} \\ \dfrac{I_{ZZ} - I_{XX}}{I_{YY}} pr - \dfrac{J_{TP}}{I_{YY}} q\Omega + \dfrac{U_3}{I_{YY}} \\ \dfrac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \dfrac{U_4}{I_{ZZ}} \end{pmatrix} \tag{3.34}$$

Para projetar um controlador linear, o modelo não linear da Equação (3.34) deve ser linearizado em torno de um ponto de operação. No controle de VTOLs, geralmente é escolhido para ficar em voo pairado. Se for feita a suposição de que o veículo está em condições de voo pairado, pode-se considerar as seguintes aproximações

$$\begin{cases} U_1 \approx mg \\ p \cong q \cong r \cong 0 \\ sen(\psi) \cong 0 \\ sen(\theta) \cong \theta \\ sen(\phi) \cong \phi \end{cases} \tag{3.35}$$

Utilizando-se dessas aproximações, pode-se reescrever a equação (3.33) linearizada na seguinte forma

$$\begin{cases} \ddot{X} = +g\theta \\ \ddot{Y} = -g\phi \\ \ddot{Z} = -g + \dfrac{U_1}{m} \\ \ddot{\phi} = \dot{p} = \dfrac{U_2}{I_{xx}} \\ \ddot{\theta} = \dot{q} = \dfrac{U_3}{I_{yy}} \\ \ddot{\psi} = \dot{r} = \dfrac{U_4}{I_{zz}} \end{cases} \tag{3.36}$$

e escrever as variáveis de estado e as saídas do sistema na forma

$$x = [X \;\; Y \;\; Z \;\; \phi \;\; \theta \;\; \psi \;\; \dot{X} \;\; \dot{Y} \;\; \dot{Z} \;\; \dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}]^T$$

$$y = [X \;\; Y \;\; Z \;\; \psi]^T \tag{3.37}$$

A equação (3.36) pode ser escrita na forma de espaço de estado

$$A = \begin{bmatrix} 0_{3\times4} & 0_{3\times1} & 0_{3\times1} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times4} & 0_{3\times1} & 0_{3\times1} & 0_{3\times3} & I_{3\times3} \\ 0_{1\times4} & 0 & g & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & -g & 0 & 0_{1\times4} & 0_{1\times3} \\ 0_{4\times4} & 0_{4\times1} & 0_{4\times1} & 0_{4\times3} & 0_{4\times3} \end{bmatrix} \tag{3.38}$$

$$B = \begin{bmatrix} 0_{8\times1} & 0_{8\times1} & 0_{8\times1} & 0_{8\times1} \\ 1/m & 0 & 0 & 0 \\ 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \tag{3.39}$$

$$\mathrm{C} = \begin{bmatrix} \mathrm{I}_{3\times3} & \mathrm{O}_{3\times2} & \mathrm{O}_{3\times1} & \mathrm{O}_{3\times6} \\ \mathrm{O}_{1\times3} & \mathrm{O}_{1\times2} & 1 & \mathrm{O}_{1\times6} \end{bmatrix} \tag{3.40}$$

4. ABORDAGEM DE CONTROLE: De Aprendizagem por Reforço à Projeto Online de Controle Ótimo DT LQR

4.1 Contextualização

A maioria dos organismos na natureza age em seu habitat para conservar os recursos enquanto alcança seus objetivos de maneira ótima. Estes agentes estruturam-se a aprender a colaborar para a melhoria de suas chances de sobrevivência e crescimento, gerando assim uma premissa de relação entre causa e efeito, ações e recompensas inerente a tal aprendizado. Este cenário caracterizado por fortes capacidades de autoaprendizagem e adaptação, devido as ações baseadas nas interações, substancia a Aprendizagem por Reforço e a Programação Dinâmica Aproximada.

Em (LEWIS e LIU, 2012) pode-se observar que muitos processos de aprendizagem são baseados nas observações e nos comportamentos dos indivíduos ou grupos estudados. Como foi no caso do estudo de Charles Darwin, Adam Smith, Ivan Pavlov etc. A Aprendizagem por Reforço refere-se ao problema de um agente interagindo com um ambiente, tendo por base os organismos vivos que interagem com seu ambiente e usam estas interações para melhorar suas ações e sobreviverem. Tal cenário, denomina-se modificação das ações baseadas na interação com o ambiente de Aprendizagem por Reforço (*Reinforcement Learning* - RL). Esta se refere a um ator ou agente que interage com o ambiente e modifica suas ações, ou políticas de controle, baseadas em estímulos recebidos em resposta às suas ações.

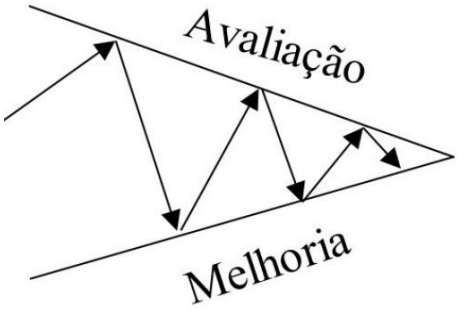
O objetivo da aprendizagem é maximizar uma recompensa escalar a longo prazo, ao invés de simplesmente uma recompensa imediata. Isso ocorre pela percepção do estado do ambiente e tomada de decisões que afetarão este estado. A RL recebe uma resposta sobre o desempenho da sua ação, permitindo melhorar tal desempenho em ações posteriores, ou seja, implica em uma relação de causa e efeito entre ações e recompensa/punição. Isto é um comportamento orientado à objetivos, pelo menos, na medida em que o agente tem o entendimento de recompensa versus falta de recompensa/punição.

Conforme estudado em (HAYKIN, 2008), o sistema é então projetado para aprender por reforço atrasado, isto é, o sistema observa uma sequência temporal de estímulos, correspondentes a vetores de estado, advindos também do ambiente, que resultaram num sinal de reforço heurístico. É relevante destacar que certas ações tomadas em momentos anteriores, numa sequência de passos de tempo, são os melhores determinantes do comportamento global do sistema. Em (LEWIS e LIU, 2012) os algoritmos de RL são construídos sob a premissa de que decisões de controle de sucesso devem ser lembradas por meio do sinal de reforço, de forma

a serem utilizadas uma segunda vez. A ideia se origina da aprendizagem experimental de animais. No entanto, a RL é fortemente conectada do ponto de vista teórico, com métodos de controle ótimo adaptativo, direto e indireto.

Diversos métodos têm sido desenvolvidos e aplicados para Aprendizagem por Reforço. Os algoritmos são construídos com a ideia de que decisões sucessivas de controle são tomadas com a intenção de maximizar o reforço ao longo do tempo, mantendo a estabilidade do sistema. Uma classe de métodos de Aprendizagem por Reforço é baseada na estrutura do Ator-Crítico, onde um componente ator aplica uma ação ou política de controle ao ambiente e um componente crítico avalia o valor dessa ação. Conforme a figura 4.1, a estrutura do ator-crítico implica duas etapas: avaliação da política pelo crítico, e seguida da melhoria de política. A etapa de avaliação da política é executada observando do ambiente os resultados das ações atuais. Ações ótimas podem ser baseadas em esforços mínimos, riscos mínimos, aprendizagem máxima e assim por diante.

Figura 4.1 Estrutura Ator-Crítico



No caso dos algoritmos de Aprendizagem por Reforço ótimo o processo de aprendizagem é direcionado para um nível mais elevado, não tendo mais como objeto de interesse os detalhes da dinâmica do sistema, mas um índice de desempenho que quantifica o quão próximo da otimilidade o sistema de controle de malha fechada opera. Em tal esquema, a RL é um meio de aprender comportamentos ótimos observando a resposta do ambiente a políticas de controle não ótimas. E não há conhecimento da saída correta, o agente apenas recebe alguma informação do ambiente por um reforço, uma punição ou recompensa, baseada nos estados e ações de controle. Com isso, a tarefa é determinar uma política para a seleção de ação que maximize sua recompensa a longo prazo.

Neste interim, teorias de otimização tem contribuído para sistematização e resolução de problemas de controle ótimo. E tem-se na literatura abordagens que agregam conceitos e métodos de controle ótimo e controle adaptativos tendo por base paradigmas de Aprendizagem

por Reforço e Programação Dinâmica Aproximada (WERBOS, 1992) e (SI, 2004). A Programação Dinâmica é um método de otimização matemática utilizada para resolver problemas de decisão multiestágios. A Programação Dinâmica Aproximada (*Approximate Dynamic Programming*- ADP) e a Aprendizagem por Reforço (RL) são paradigmas para o aprendizado online de controle ótimo. A Aprendizagem por Reforço moderna é uma síntese de ideias da teoria de controle ótimo, programação dinâmica, métodos de diferença temporal e aproximação de função (FONSECA, FERREIRA e REGO, 2013).

A programação dinâmica é amplamente considerada como uma excelente forma de resolver problemas de controle ótimo quando um modelo do sistema está disponível. Todavia, a ideia de obter uma solução alternativa na ausência de um modelo foi explorada e experimentou um desenvolvimento no campo da Aprendizagem por Reforço (TEIXEIRA, 2016). A Programação Dinâmica Aproximada é uma família de técnicas de Aprendizagem por Reforço para o controle de realimentação de sistemas projetados pelo homem. Uma vez que o método de RL envolve a modificação da política de controle baseada em respostas do ambiente, tem-se a percepção inicial de que a RL está intrinsecamente relacionada com controle adaptativo, uma família de técnicas de controle bem-sucedidas e que são altamente estudadas na Comunidade de Sistemas de Controle. Esta técnica pode ser vista como um campo de pesquisa promissor para desenvolver controladores inteligentes inspirados em estruturas neurais que fornecem habilidades para lidar de forma eficiente com o grau de complexidade de sistemas não-lineares incertos e parcialmente observáveis.

4.2 Sistemas Dinâmicos e Controle Ótimo de Realimentação

Existem métodos padrões para amostragem ou discretização das EDOs de espaço de estados de tempo contínuo não linear para obter formas de dados amostrados convenientes para controle baseado em computador. Os sistemas resultantes se desdobram em tempo discreto e são geralmente da forma de espaço de estado $x_{k+1} = F(x_k, u_k)$, sendo k o índice de tempo discreto. Estes sistemas satisfazem a propriedade de Markov, uma vez que seus estados no instante $k + 1$ dependem apenas do estado e da entrada no instante anterior k.

Neste caso, algoritmos para controle ótimo adaptativo solucionam a equação *Hamilton-Jacobi-Bellman* de maneira *online* no tempo sem o conhecimento da dinâmica do sistema. No caso do regulador linear quadrático (do inglês *Linear Quadratic Regulator* - LQR), isso corresponde à solução da equação de *Riccati* pelo algoritmo, sem o conhecimento da matriz do sistema.

Para facilitar a análise, considera-se uma classe de sistemas de tempo discreto descrita por dinâmica não linear na forma de equação diferencial de espaço de estado

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{4.1}$$

com o estado $x_k \in R^n$ e entradas de controle $u_k \in R^m$. A análise de tais formas é conveniente e pode ser generalizada para uma forma geral de dados amostrados

$$x_{k+1} = F(x_k, u_k) \tag{4.2}$$

Uma política de controle é definida como uma função do espaço de estado para o espaço de controle $h(\cdot)$: $R^n \rightarrow R^m$. Ou seja, para cada estado x_k, a política define uma ação de controle

$$u_k = h(x_k) \tag{4.3}$$

Tais mapeamentos são também conhecidos como controladores de realimentação. Um exemplo de política de controle é a realimentação linear da variável de estado. Na abordagem RL, a política de controle é aprendida em tempo real baseada em estímulos recebidos do ambiente. Esse tipo de aprendizagem de projeto de controlador está relacionado a noções de controle adaptativo.

Em Aprendizagem por Reforço, o ator é o agente que gera a política de controle. Ou seja, o ator é matematicamente descrito pela política (4.3), que tem o estado $x(t)$ como entrada e o controle $u(t)$ como saída. Tudo fora do ator é considerado o ambiente. Então, o sistema (4.1) é considerado como parte do ambiente, como todas as perturbações e efeitos externos. De fato, em aplicações padrões de RL, a dinâmica do sistema não é sequer considerada, e como parte do ambiente, nenhum modelo explicito da dinâmica, como em (4.1), é usado. A aprendizagem por reforço teve sucesso bastante notável em sistemas complexos com dinâmicas desconhecidas. No entanto, não considerar especificamente a dinâmica também torna impossível fornecer provas explícitas de estabilidade e desempenho.

4.3 Desempenho Ótimo Orientado à Objetivos

A ideia de comportamento ótimo orientado à objetivos é extraída da definição de Medida de Desempenho ou Função de Custo Determinística, a qual é dada em (4.4). Na sua forma

primária, o algoritmo de programação dinâmica trata de um problema de horizonte finito. Com o objetivo de estender o uso deste algoritmo para tratar do problema de horizonte infinito descrito pela função de custo, formula-se este método em termos matemáticos,

$$V_h(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k}\, r(x_i, u_i) \tag{4.4}$$

sendo $0 < \gamma \leq 1$ um fator de desconto e $u_k = h(x_k)$ uma política de controle de realimentação prescrita. A noção de fator de desconto é o fato de que os custos adquiridos no futuro são menos relevantes na determinação do comportamento ótimo. Na função de custo, custo de transição de estágio ou custo imediato, define-se também a função de utilidade $r(x_k, u_k)$, que é a medida de custo de controle à um passo. Uma forma típica de função de utilidade é a função energia quadrática $r(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$ ou a forma mais genérica

$$r(x_k, u_k) = Q(x_k) + u_k^T R u_k \tag{4.5}$$

sendo $Q(x_k) > 0$ e $R > 0$ definidas positivas.

Toma-se o custo de estágio quadrático em u_k para simplificar o desenvolvimento. Supõe-se que o sistema é estável em algum conjunto $\Omega \subset R^n$, ou seja, existe uma política de controle $u_k = h(x_k)$ de modo que o sistema de malha fechada em (4.1) é assintoticamente estável neste conjunto. Uma política de controle, $u_k = h(x_k)$, é dita ser admissível se esta for estabilizadora, para o sistema de malha fechada, e gera um custo finito $V_h(x_k)$.

Para qualquer política admissível $u_k = h(x_k)$, é definido seu custo ou valor $V_h(x_k)$. É importante observar que, dada qualquer política admissível, seu valor pode ser determinado pela avaliação da soma infinita (4.4).

O objetivo da teoria de controle ótimo é selecionar a política que minimize o custo, ou seja,

$$V^*(x_k) = \min_{h(\cdot)} \left(\sum_{i=k}^{\infty} \gamma^{i-k}\, r(x_i, h(x_i)) \right) \tag{4.6}$$

que é conhecido como o custo ótimo, ou Valor Ótimo. Consequentemente, a política de controle ótimo é dada por

$$h^*(x_k) = \arg\min_{h(\cdot)} \left(\sum_{i=k}^{\infty} \gamma^{i-k}\, r(x_i, h(x_i)) \right) \tag{4.7}$$

Desta forma, o problema é minimizar não simplesmente o custo à um passo ou função de utilidade, mas a soma de todos os custos descontados ou "*cost-to-go*".

4.4 Equação de Bellman e Programação Dinâmica

O valor ótimo e a política ótima fundamentam-se no princípio da otimalidade de Bellman, nos quais a programação dinâmica consiste de um método *offline* para determina-los. Por outro lado, a metodologia da aprendizagem por reforço busca encontrar políticas ótimas baseadas em experiência, executando decisões sequenciais que melhoram as ações de controle com base nos resultados observados utilizando a política atual. Tal procedimento resulta na construção de métodos para encontrar valores ótimos e políticas ótimas que podem ser executados "para frente no tempo". Nesta construção da equação de Bellman, primeiro é necessário reescrever a equação (4.4) da seguinte forma

$$V_h(x_k) = r(x_k, u_k) + \gamma \sum_{i=k+1}^{\infty} \gamma^{i-(k+1)}\, r(x_i, u_i) \tag{4.8}$$

e pode-se obter uma equação de diferença equivalente dada por

$$V_h(x_k) = r(x_k, h(x_k)) + \gamma V_h(x_{k+1}) \tag{4.9}$$

com $V_h(0) = 0$. Ou seja, ao invés de avaliar a soma infinita, equação (4.4), pode-se resolver a equação de diferença, equação (4.9), para obter o valor de usar uma política atual $u_k = h(x_k)$.

A equação (4.9) é uma equação não linear de Lyapunov, conhecida como a equação de Bellman. Esta equação é o ponto de partida para a construção de uma família de algoritmos de aprendizagem por reforço, que buscam encontrar políticas ótimas online em tempo real para a solução de problemas de controle. Avaliar o valor de uma política atual usando a equação de Bellman é o primeiro conceito-chave no desenvolvimento de técnicas de aprendizado por reforço.

O Valor Ótimo dado na equação (4.6) pode ser reescrito usando a equação de Bellman. Desta forma,

$$V^*(x_k) = \min_{h(\cdot)}(r(x_k, h(x_k)) + \gamma\, V_h(x_{k+1})) \tag{4.10}$$

O princípio de Bellman (LEWIS e LIU, 2012) afirma que "Uma política ótima tem a propriedade de que, não importam quais tenham sido as decisões anteriores (ou seja, controles), as decisões restantes devem constituir uma política ótima em relação ao estado resultante daquelas decisões anteriores". Isso se traduz para a forma

$$V^*(x_k) = \min_{h(\cdot)}(r(x_k, h(x_k)) + \gamma V^*(x_{k+1})) \tag{4.11}$$

A equação (4.11) é conhecida como equação da otimalidade de Bellman ou equação de Hamilton-Jacobi-Bellman (HJB) em tempo discreto. Assim a política ótima é dada por

$$h^*(x_k) = \arg\min_{h(\cdot)}(r(x_k, h(x_k)) + \gamma V^*(x_{k+1})) \tag{4.12}$$

De acordo com a equação (4.11), para se determinar a política ótima no instante $k + 1$, é preciso conhecer a política ótima no instante k. Assim, o Princípio de Bellman produz um procedimento "para trás no tempo" (*backwards-in-time*) para resolver o problema de controle ótimo. Isto é a base para algoritmos de programação dinâmica em uso extensivo na teoria de sistemas de controle, pesquisa operacional e em outros lugares. Estes são, por natureza, métodos de planejamento *offline*. Um exemplo de tal procedimento no projeto de controle de realimentação é o projeto da equação de Riccati para o problema LQR, que envolve a solução *offline* da equação de Riccati, dada a dinâmica do sistema conhecida. Os métodos de Programação Dinâmica geralmente exigem o conhecimento completo das equações dinâmicas do sistema, ou seja, $f(x)$ e $g(x)$ devem ser conhecidas.

4.5 Equações do Ponto Fixo

Em contraste aos projetos *offline* de programação dinâmica, se busca métodos de aprendizagem por reforço para aprendizado *on-line*, em tempo real, sem conhecer a dinâmica do sistema, ou seja, $f(x)$ e $g(x)$. Portanto, pode-se explorar a noção de que a equação de Bellman (4.9) e a equação da otimalidade de Bellman (4.11) são equações do ponto fixo para desenvolver métodos "para frente no tempo" (*forward-in-time*) para resolver o problema de controle ótimo.

4.5.1 Iteração de Política (IP)

A iteração de política é um método iterativo para determinar o controle ótimo. Em cada iteração do processo, realiza-se uma etapa de obtenção da função valor para a política atual, *avaliação de política*, funcionando em *loop* interno para cada iteração de política. Logo depois, uma nova política é obtida de modo melhorado, igual ou melhor a política atual. Esta etapa é denominada de *melhoria de política*. Ou seja, no algoritmo IP, inicialmente, seleciona-se qualquer política de controle admissível $h_0(x_k)$. Em cada iteração j, determina-se o valor da política atual usando a equação de Bellman,

$$V_{j+1}(x_k) = r\big(x_k, h_j(x_k)\big) + \gamma V_{j+1}(x_{k+1}) \tag{4.13}$$

e, em seguida, estabelece-se uma política melhorada usando

$$h_{j+1}(x_k) = \arg\min_{h(\cdot)}\big(r\big(x_k, h_j(x_k)\big) + \gamma V_{j+1}(x_{k+1})\big) \tag{4.14}$$

Em particular, se a função de utilidade $r(x_k, u_k)$ tiver a forma dada na equação (4.5) e a dinâmica for descrita pela equação (4.1), a etapa de melhoria da política será determinada por

$$h_{j+1}(x_k) = -\frac{\gamma}{2} R^{-1} g^T(x_k) \nabla V_{j+1}(x_{k+1}) \tag{4.15}$$

sendo $\nabla V(x) = \partial V(x)/\partial x$ o gradiente da função valor, considerado aqui um vetor coluna. Esse algoritmo converge sob certas condições para o valor ótimo e para a política de controle (4.11) (4.12)

A avaliação do valor da política atual usando a Equação de Bellman (4.13) equivale a determinar o valor de usar a política $h_j(x_k)$ iniciando em todos os estados atuais x_k.

Pode-se ser mostrado que a equação de Bellman é uma equação do ponto fixo. Ou seja, dada uma política admissível $u_k = h\,(x_k)$, existe um único ponto fixo $V_h(x_k)$, e o seguinte mapeamento de contração

$$V^{i+1}(x_k) = r(x_k, h(x_i)) + \gamma V^i(x_{k+1}) \tag{4.16}$$

pode ser iterado iniciando com qualquer valor $V^0(x_k)$, e resulta no limite $V^i(x_k) \to V_h(x_k)$. Portanto, pode-se substituir a etapa de iteração política (4.13) pela seguinte equação

$$V^{i+1}(x_k) = r\big(x_k, h_j(x_k)\big) + \gamma V^i(x_{k+1}) \tag{4.17}$$

para $i = 1, 2, \ldots$, onde a iteração em i é realizada com a mesma política $h_j(\cdot)$ até a convergência. Então, $V^i(x) \to V_{j+1}(x)$ quando $i \to \infty$.

4.5.2 Iteração de Valor (IV)

A Iteração de Valor é o método de busca de políticas e funções valor ótimas em aprendizagem por reforço. Este método usa a equação (4.11), a equação da otimalidade de *Bellman*, para calcular iterativamente uma função valor ótima e após obter uma política ótima. No algoritmo de Iteração de Valor, incialmente, seleciona-se qualquer política de controle $h_0(x_k)$, não necessariamente admissível ou estabilizadora, iterando-se j até a convergência. Em cada iteração j, atualiza-se o valor usando a equação

$$V_{j+1}(x_k) = r\big(x_k, h_j(x_k)\big) + \gamma V_j(x_{k+1}) \tag{4.18}$$

e, em seguida, determina-se uma melhoria de política pela seguinte forma

$$h_{j+1}(x_k) = \arg\min_{h(\cdot)} \big(r\big(x_k, h_j(x_k)\big) + \gamma V_{j+1}(x_{k+1})\big) \tag{4.19}$$

É importante observar que agora, o valor anterior é usado no lado direito de (4.13) em contraste com a etapa de Avaliação de Política do algoritmo IP. Além disso, o algoritmo IV não exige uma política inicial estabilizadora. Pode-se demonstrar que o algoritmo IV converge para o valor ótimo e para a política de controle ótima sob determinadas condições.

A Iteração de Valor se baseia no fato de que a Equação da Otimalidade de Bellman (4.11) também é uma equação do ponto fixo. Os passos intercalados de atualização de valor e melhoria de política são os meios de iterar o mapeamento de contração associado a equação (4.11).

Note que a Iteração de Política requer em cada passo a solução de (4.13), que é uma equação não linear de Lyapunov. Esta solução é difícil para sistemas não lineares gerais. Por

outro lado, a Iteração de Valor depende da solução de (4.18), que é simplesmente uma equação de recursão.

Geralmente, equações do ponto fixo podem ser usadas, com formulação adequada, como base para algoritmos de aprendizado por reforço *online* que aprendem observando dados acumulados ao longo das trajetórias do sistema.

4.6 Regulador Linear Quadrático em Tempo Discreto

A otimalidade dos controladores está diretamente ligada à solução da equação de *Ricatti*. Por isso, deve-se mostrar que as noções de Aprendizagem por Reforço de Iteração de Política e Iteração de Valor estão de fato alinhadas com ideias familiares em sistemas de controle de realimentação. Por meio de uma discretização do sistema dinâmico, o *DT LQR* pode ser caracterizado como um problema de múltiplos estágios e estados, assim estabelecendo um caminho das trajetórias através da Programação Dinâmica. Outro ponto para fomentar é fornecer fórmulas explícitas para as construções acima.

Uma ampla classe de importantes sistemas de Tempo Discreto (do inglês *Discrete* Time – DT) pode ser descrita por uma equação de estado linear e invariante no tempo

$$x_{k+1} = Ax_k + Bu_k \tag{4.20}$$

As políticas de controle de interesse são, então, realimentações da variável de estado da forma

$$u_k = h(x_k) = -Kx_k \tag{4.21}$$

com a política de controle, uma matriz de ganhos de realimentação K constante a ser determinada. Dada uma política prescrita, a função de custo é a soma de funções quadráticas

$$\begin{aligned} V_h(x_k) &= \sum_{i=k}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \\ &= \sum_{i=k}^{\infty} x_i^T (Q + K^T R K)\, x_i \equiv V_k(x_k) \end{aligned} \tag{4.22}$$

que tem a função de utilidade $r(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$ com matrizes de ponderação $Q = Q^T \geq 0, R = R^T > 0$. Assume-se que (A, B) é estabilizável, isto é, existe uma matriz de ganho de realimentação K tal que o sistema de malha fechada é assintoticamente estável

$$x_{k+1} = (A - BK)x_k \equiv A_c x_k \tag{4.23}$$

4.6.1 Solução de Controle Ótimo para o DT LQR

O objetivo é selecionar o ganho K de realimentação do estado, isto é, a política de controle para minimizar o *cost-to-go* $V_h(x_k) = V_K(x_k)$ para todos os estados atuais. Isso é chamado de problema do Regulador Linear Quadrático (*Linear Quadratic Regulator - LQR*) (LEWIS, VRABIE e SYRMOS, 2012).

Pode ser demonstrado que o valor ótimo para o LQR é quadrático no estado atual, com

$$V^*(x_k) = x_k^T P x_k \tag{4.24}$$

para alguma matriz P definida positiva, que deve ser determinada. Portanto, a equação de Bellman para o LQR é dada por

$$x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1}. \tag{4.25}$$

Em termos de ganho de realimentação, a equação (4.25) pode ser escrita na forma

$$x_k^T P x_k = x_k^T \big(Q + K^T R K + (A - BK)^T P (A - BK)\big) x_k. \tag{4.26}$$

Uma vez que a equação (4.26) deve ser satisfeita para todos os estados atuais x_k, tem-se

$$(A - BK)^T P (A - BK) - P + Q + K^T R K = 0 \tag{4.27}$$

A equação (4.27) é linear em P e é conhecida como uma equação de Lyapunov quando K for fixado. Ou seja, a equação de *Bellman* para o DT LQR equivale a equação de Lyapunov. Note que a solução da equação de Lyapunov (4.27) está atrelada ao total conhecimento das dinâmicas do sistema (A, B). Assim algoritmos de Aprendizagem por Reforço para aprendizagem de soluções *online* podem ser planejados, isto é, a *RL* permite que a equação de Lyapunov seja resolvida *online* em o conhecimento das matrizes A e B do sistema.

Expressando a equação de Bellman na seguinte forma

$$x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + (A x_k + B u_k)^T P (A x_k + B u_k) \tag{4.28}$$

a minimização é realizada diferenciando-se em relação à u_k para obter

$$Ru_k + B^T P(Ax_k + Bu_k) = 0$$

ou

$$u_k = -(R + B^T PB)^{-1} B^T PAx_k \tag{4.29}$$

O ganho ótimo de realimentação é, então, dado por

$$K = (R + B^T PB)^{-1} B^T PA. \tag{4.30}$$

Substituindo a equação (4.30) na equação de Bellman (4.28), e simplificando a equação HJB em tempo discreto, obtém-se

$$A^T PA - P + Q - A^T PB(R + B^T PB)^{-1} B^T PA = 0 \tag{4.31}$$

Esta equação é quadrática em P e é conhecida como equação de Riccati. Para resolver o problema de controle ótimo do LQR em tempo discreto, primeiro se resolve a equação de *Riccati* para P, em seguida o valor ótimo é dado por $V^*(x_k) = x_k^T P x_k$ e a política ótima por (4.30).

É importante observar que ao passar da formulação (4.25) da equação de Bellman para a formulação (4.27), que é a equação de Lyapunov, duas etapas foram realizadas. Primeiro, a dinâmica do sistema foi substituída por x_{k+1} para produzir (4.26), depois o estado atual x_k foi cancelado para obter (4.27). Essas duas etapas tornam impossível a aplicação de métodos de aprendizagem por reforço *online*, em tempo real, para encontrar o controle ótimo, o que será feito mais adiante. Devido a estas duas etapas, o projeto de controle ótimo na Comunidade de Sistemas de Controle é quase universalmente um procedimento *offline* envolvendo soluções de equações Riccati, em que é necessário um conhecimento completo da dinâmica (A, B) do sistema.

4.6.2 Iteração de Política e Iteração de Valor para o DT LQR

Os métodos para resolver a equação de Bellman, a Iteração de política e a Iteração de Valor, agora são apresentados em relação ao regulador linear quadrático em tempo discreto (*DT LQR*). Neste ponto, para o LQR, a equação de Bellman (4.9) é escrita como (4.25) e, portanto, equivale à equação de Lyapunov (4.27). Implementar sistemas dinâmicos *online* em tempo real

por meio dos dados observados e medidos ao longo da trajetória desse sistema é a relevância da IP e IV para sistemas de controle de malha fechada.

Portanto, no Algoritmo de **Iteração de Política**, a etapa de avaliação de política para o LQR, a qual produz a equação de Lyapunov, é

$$\left(A - BK_j\right)^T P_{j+1}\left(A - BK_j\right) - P_{j+1} + Q + K_j^T R K_j = 0 \tag{4.32}$$

e a atualização da política é

$$K_{j+1} = (R + B^T P_{j+1} B)^{-1} B^T P_{j+1} A \tag{4.33}$$

Em (4.32) e (4.33), este algoritmo se baseia na repetição das soluções da equação de Lyapunov em cada passo, também chamado de algoritmo de Hewer, o qual converge para a equação de Ricatti (4.31). Este procedimento necessita do total conhecimento das dinâmicas do sistema, com sua implementação *offline*.

No Algoritmo de **Iteração de Valor**, a etapa de avaliação da política para o LQR, produz a recursão de Lyapunov, que é dada por

$$P_{j+1} = \left(A - BK_j\right)^T P_j\left(A - BK_j\right) + Q + K_j^T R K_j \tag{4.34}$$

e a atualização da política (4.19) será (4.33).

Assim como na IP para o DT LQR, este algoritmo converge para a solução da equação de *Ricatti*. Novamente requer o total conhecimento das dinâmicas do sistema, matrizes A e B, para sua implementação. Tal recursão de Lyapunov é um algoritmo *offline*. No enquanto, observe que a Iteração de Política envolve a solução completa de uma equação de Lyapunov (4.32) em cada passo j e requer um ganho estabilizador K_j em cada passo. Isso é chamado de *backup* completo em termos de aprendizagem por reforço. Por outro lado, Iteração de Valor envolve apenas uma recursão de Lyapunov (4.34) em cada passo j, que é muito fácil de calcular e não requer um ganho estabilizador. Isso é chamado de *backup* parcial em aprendizagem por reforço.

A recursão (4.34) pode ser executada mesmo se K_j não for estabilizador. Se K_j for de fato estabilizador, então iterar a recursão de Lyapunov (4.34), com um ganho de realimentação fixo K_j, até a convergência fornece a solução para a equação de Lyapunov (4.32). A Aprendizagem por Reforço sugere outro algoritmo para resolver a equação de Riccati, a saber, Iteração de Política Generalizada (*Generalized Policy Algorithm- GPI*), descrito a seguir.

4.6.3 Algoritmo de Política Generalizada para o DT LQR

Inicialmente seleciona-se qualquer política de controle K_0, não necessariamente admissível ou estabilizadora. Em cada passo j do processo iterativo, atualize o valor usando

$$P_j^{i+1} = \left(A - BK_j\right)^T P_j^i \left(A - BK_j\right) + Q + K_j^T R K_j \tag{4.35}$$

$i = 0, 1, \ldots, M-1$, para algum M finito, com $P_j^0 = P_j$ como condição inicial. Em seguida, estabelecendo $P_{j+1} = P_j^M$, realize a etapa de Melhoria de Política, determinando uma política melhorada por

$$K_{j+1} = (R + B^T P_{j+1} B)^{-1} B^T P_{j+1} A \tag{4.36}$$

Este algoritmo leva M passos para resolver a equação de Lyapunov em cada iteração *j*. Ou seja, a etapa de atualização de valor no GPI consiste em M passos da recursão (4.34) usando o mesmo ganho fixo. Estabelecer $M = 1$ produz a Iteração de Valor, (4.34), enquanto que definir $M = \infty$, isto é, executar (4.35) até convergência, produz Iteração de Política, que resolve a equação de Lyapunov (4.32).

4.7 Controle Adaptativo: Aprendizagem por Reforço e Programação Dinâmica Aproximada

A solução de controle ótimo usando programação dinâmica é um procedimento "para trás no tempo". Portanto, ele pode ser usado para controle *offline*, mas não para aprendizagem *online*. A equação de Bellman (4.9) conduz a vários métodos iterativos para aprender a solução da equação de controle ótimo sem resolver diretamente a equação HJB, incluindo Iteração de Política e Iteração de Valor. No entanto, será descrito como formulá-los como métodos de aprendizagem por reforço *online* em tempo real para resolver o problema de controle ótimo usando dados medidos ao longo das trajetórias do sistema. Esses métodos são definidos como Programação Dinâmica Aproximada (ADP) (WERBOS, 1992), (WERBOS, 2012). Existem dois sub-métodos principais: o Erro da Diferença Temporal (*Temporal Difference* -TD) e a Aproximação da Função Valor (*Value Function Approximation* - VFA).

4.7.1 Erro da Diferença Temporal (TD)

Métodos de diferenças temporais conduzem a uma família de controladores ótimos para solucionar a equação de Bellman. Estes controladores tem a capacidade de aprender *online*

a solução de problemas quando o modelo não estiver disponível. Esses métodos podem ser considerados como técnicas de Aproximação Estocástica onde a equação de Bellman são substituídas por avaliações ao longo de um caminho amostrado.

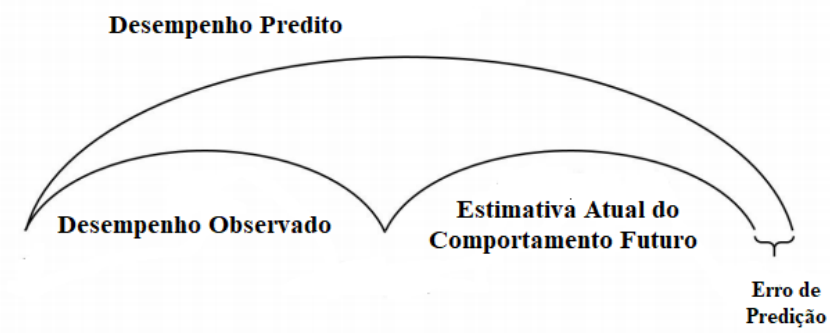
Programação Dinâmica Aproximada (ADP) é um método prático para determinar a solução de controle ótimo *online* "para frente no tempo" usando dados do sistema medidos ao longo das trajetórias do sistema. Ele é baseado em fornecer métodos para resolver o problema de programação dinâmica “para frente no tempo” em tempo real e para aproximar a função valor.

O método do Erro da Diferença Temporal (TD) é aplicável transformando esses conceitos em métodos de solução *online* “para frente no tempo”, com base na equação de Bellman, definindo um erro de equação residual variante no tempo que é dado por

$$e_k = r(x_k, h(x_k)) + \gamma V_h(x_{k+1}) - V_h(x_{k+1}) \tag{4.37}$$

A função e_k é conhecida como erro de diferença temporal. Se a equação de Bellman for válida, o erro de TD é zero. Portanto, para uma política de controle fixa, pode-se resolver a equação (4.37) com $e_k = 0$ a cada instante k para a função de valor $V_h(\cdot)$. Isso produz a melhor aproximação para o valor correspondente ao usar a política atual, ou seja, para a equação (4.4). O erro TD pode ser considerado como um erro de predição entre o desempenho predito e o desempenho observado em resposta a uma ação aplicada ao sistema, como observado nas figuras 4.2 e 4.3, em que a primeira destaca as ações da TD e a segunda mostra as parcelas desta equação em termos de Bellman

Figura 4.2 Componentes do Erro da Diferença Temporal

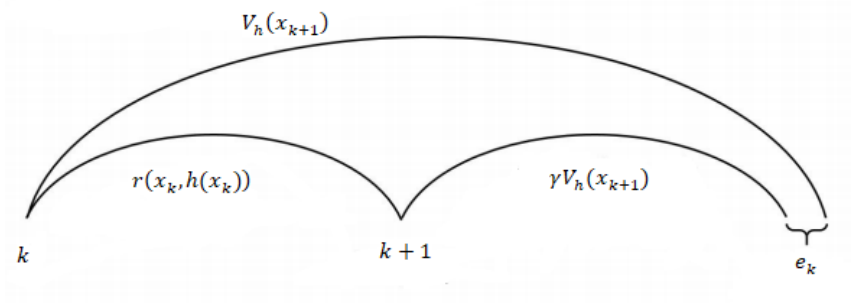


A figura 4.2 apresenta que a Aprendizagem por Reforço aplica um comando de ação e observa o comportamento ou o desempenho resultante. A diferença entre o desempenho predito e o desempenho observado mais a estimativa atual do comportamento futuro é usada para

modificar os comandos de ação para diminuir essa diferença. Isso é capturado formalmente na Equação de Bellman.

A figura 4.3 apresenta cada parcela dessa equação.

Figura 4.3 Parcelas do Erro da Diferença Temporal



4.7.2 Aproximação da Função Valor (VFA)

Já a Aproximação da Função Valor (VFA) fornece um meio prático para resolver a equação TD, pode-se aproximar a função valor V_h usando um aproximador paramétrico. Isso tem sido chamado de Programação Dinâmica Aproximada (ADP).

Agora, considera-se a VFA para o caso LQR. Em LQR, sabe-se que o valor de qualquer política de controle admissível $u_k = -Kx_k$ é quadrática no estado, isso é válido para alguma matriz P, ou seja, $V(x_k) = x_k^T P x_k$. Substituindo isso na equação (4.37), produz-se o erro LQR TD, que é dado por

$$e_k = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1} - x_k^T P x_k \tag{4.38}$$

A equação TD usa o produto de Kronecker para expressar função valor na forma

$$V_K(x_k) = x_k^T P x_k = \big(vec(P)\big)^T (x_k \otimes x_k) \equiv \bar{p}^T \bar{x}_k \tag{4.39}$$

sendo $\otimes$ o produto de Kronecker e $vec(P)$ o vetor formado pelo empilhamento das colunas da matriz P. Observe que $\bar{x}_k = x_k \otimes x_k$ é o vetor polinomial quadrático contendo todos os possíveis produtos das n componentes de x_k. Observando que P é simétrica e tem somente $n(n+1)/2$ elementos independentes, remove-se os termos redundantes em $x_k \otimes x_k$ para definir um conjunto de base quadrática $\bar{x}_k$ com $n(n+1)/2$ elementos independentes. O vetor de parâmetro desconhecido é $\bar{p}$, cujas componentes correspondem aos elementos da matriz P.

Usando estas construções, o erro TD é reescrito por

$$e_k = x_k^T Q x_k + u_k^T R u_k + \bar{p}^T \bar{x}_{k+1} - \bar{p}^T \bar{x}_k$$

$$= r(x_k, u_k) + \bar{p}^T \bar{x}_{k+1} - \bar{p}^T \bar{x}_k \tag{4.40}$$

No caso do LQR, um conjunto de base completa para a função de valor $V_h(x_k)$ é fornecido pelas funções quadráticas nas componentes de x_k. No caso não-linear, assume-se que o valor é suficientemente suave. Então, de acordo com o Teorema de aproximação de ordem mais elevada de Weierstrass, existe uma um conjunto de base denso $\{\phi_i(x)\}$ tal que

$$V_h(x) = \sum_{i=1}^{\infty} \omega_i \varphi_i(x) = \sum_{i=1}^{L} \omega_i \varphi_i(x) + \sum_{i=L+1}^{\infty} \omega_i \varphi_i(x)$$

$$= W^T \phi(x) + \varepsilon_L(x) \tag{4.41}$$

onde o vetor de base $\phi(x) = [\varphi_1(x) \quad \varphi_2(x) \quad \ldots \quad \varphi_L(x)]$: $R^n \rightarrow R^L$ e $\varepsilon_L(x)$ converge uniformemente para zero quando o número de termos L tende para o infinito.

4.7.3 Controle Ótimo Online por Aprendizagem por Reforço

Em controle ótimo baseado em Programação Dinâmica Aproximada, assume-se a aproximação

$$V_h(x_k) = W^T \phi(x), \tag{4.42}$$

e substituindo a equação (4.42) na equação TD de Bellman (4.37), obtém-se

$$e_k = r(x_k, h(x_k)) + \gamma\, W^T \phi(\, x_{k+1}) - W^T \phi(x_k) \tag{4.43}$$

A equação $e_k = 0$ é uma equação do ponto fixo. Esta equação é satisfeita em cada instante k para o valor $V_h(\cdot)$ correspondente à política atual $u = h(x)$. Assim, procedimentos iterativos para resolver a equação TD podem ser usados, incluindo Iteração de Política e Iteração de Valor.

Algoritmos de Iteração de Política e Iteração de Valor Online

No algoritmo de Iteração de Política *online*, inicialmente, seleciona-se uma política de controle admissível/estabilizadora $h_0(x_k)$. Em seguida, avalia-se essa política, determinando a solução de mínimos quadrados (LS) W_{j+1} para

$$W_{j+1}^T(\phi(x_k) - \gamma\phi(x_{k+1})) = r\big(x_k, h_j(x_k)\big) \tag{4.44}$$

com $\Phi(k) = \phi(x_k) - \gamma\phi(x_{k+1})$ um vetor de regressão, e, determina-se a etapa de melhoria de política por

$$h_{j+1}(x_k) = \arg\min_{h(\cdot)}\big(r(x_k, h(x_k)) + \gamma W_{j+1}^T\phi(x_{k+1})\big) \tag{4.45}$$

É imprescindível notar que as equações da forma (4.44) são exatamente as resolvidas pelas técnicas de mínimos quadrados recursivos (RLS). Portanto, pode-se executar as técnicas de RLS *online* até a convergência.

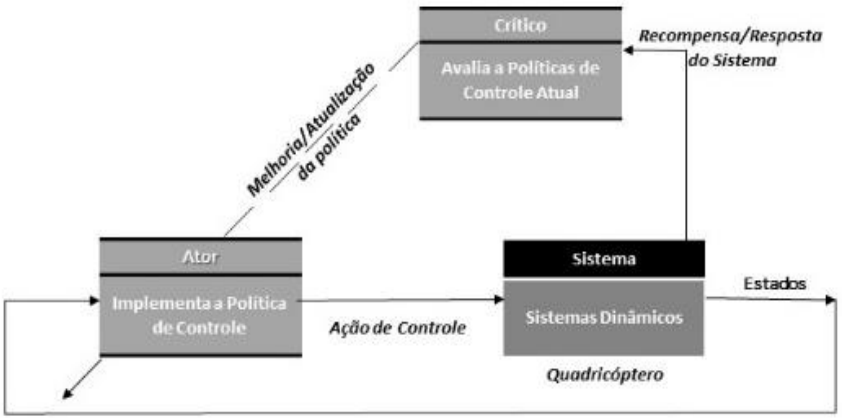
Isso fornece um algoritmo *online* de aprendizagem por reforço para resolver o problema de controle ótimo usando a Iteração de Política, baseado em dados medidos ao longo das trajetórias do sistema. De forma similar, um algoritmo *online* de aprendizagem por reforço pode ser dado com base na Iteração de Valor, onde, inicialmente, seleciona-se qualquer política de controle $h_0(x_k)$, não necessariamente admissível/estabilizadora. Após isso, determina-se a atualização de valor por meio da solução de mínimos quadrados para

$$W_{j+1}^T\phi(x_k) = r\big(x_k, h_j(x_k)\big) + W_j^T\gamma\phi(x_{k+1}) \tag{4.46}$$

e determina-se a política melhorada usando a equação (4.45).

Algoritmos *online* de Aprendizagem por Reforço são estruturas Ator-Crítico, conforme percebido na figura 4.4. O crítico e o ator são sintonizados sequencialmente em IP e IV. A atualização do valor no critico é desempenhada resolvendo (4.46) ou (4.44) usando técnicas padrões de Controle Adaptativo, isto é, RLS. Em seguida, o controle é atualizado utilizando a etapa de melhoria de política.

Figura 4.4 Aprendizagem por Reforço com uma Estrutura Ator-Crítico

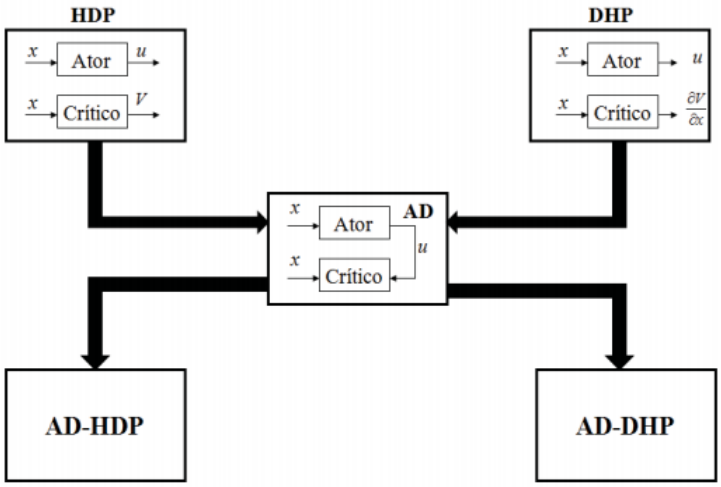


4.8 Aprendizagem Q

No aprendizado da função Valor pelos métodos de Aprendizagem por Reforço, o valor ótimo e o controle ótimo são armazenados como uma função do vetor de estado $x \in R^n$. Usando a Aproximação de Função Valor (do inglês *Value Function Approximation* – VFA), onde o crítico e, se desejar, o ator é parametrizado usando os aproximadores de função, isso é feito de maneira direta. Podendo ser implementado *online* usando-se um algoritmo de identificação de sistema de controle adaptativo, como exemplo os mínimos quadrados recursivos (RLS). Os pesos das redes neurais são sintonizados online para aprender o valor ótimo e a política de controle ótima para qualquer valor do estado x_k.

Werbos introduziu quatro métodos básicos de Programação Dinâmica Aproximada (ADP). Ele chamou a aprendizagem por reforço baseado no aprendizado da função valor escalar $V_h(x_k)$, de Programação Dinâmica Heurística (HDP). Nesse método, a função valor é estimada pelo crítico diretamente. HDP dependente de ação (AD HDP), introduzido como aprendizagem Q para MDP de estado discreto por Watkins, aprende a chamada função Q (também escalar) e permite realizar aprendizagem por reforço sem qualquer conhecimento da dinâmica do sistema, este é uma versão modificada da HDP, em que a rede do ator é conectada diretamente à rede do crítico. A programação heurística dual (DHP) usa a aprendizagem online da função custo $\lambda_k = \partial V_h(x_k)/\partial x_k$, que é o gradiente da função $V_h(x_k)$ e assim carrega mais informação do que o valor, aqui, a derivada da função valor em relação ao estado é estimada. AD DHP é baseado em aprender os gradientes da função Q, uma versão modificada da DHP, em que a rede do ator é conectada diretamente à rede do crítico.

Figura 4.5 Esquemas de ADP propostos por Werbos



Algumas características importantes devem ser notadas nas diferentes estruturas de ADP. A entrada do crítico recebe a informação do estado do sistema (e do modelo de referência

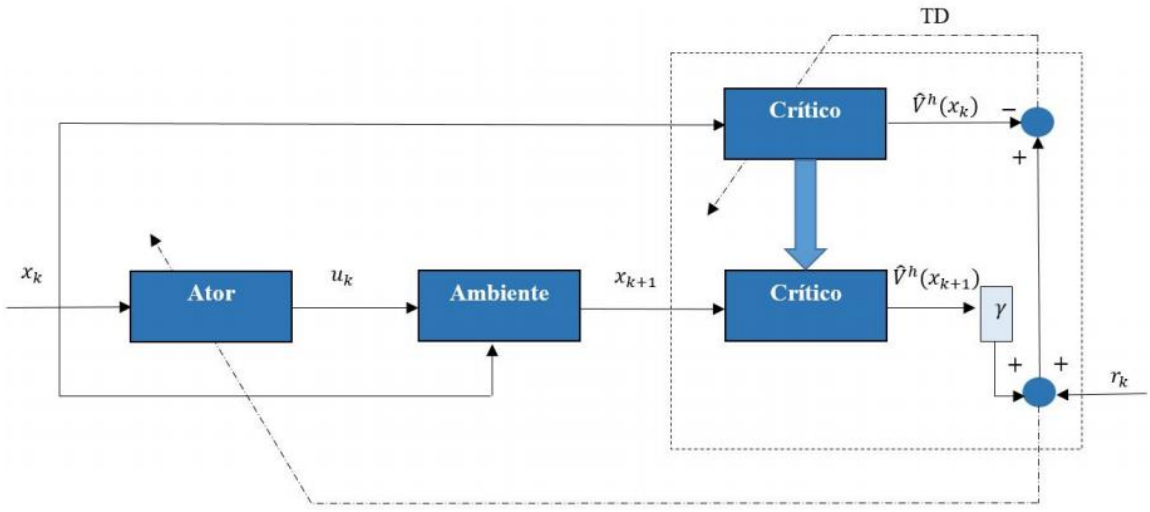
da planta, se for o caso). Na estrutura AD (*Action Dependent*) o crítico também promove uma aproximação da Função Valor $V_h(x_k)$. Na estrutura DHP, tem-se uma aproximação do gradiente de $V_h(x_h)$, denotado por $\nabla V_h(x_k) = \partial V_h(x_k)/\partial x_k$.

Para evitar conhecer qualquer uma das dinâmicas do sistema, deve-se fornecer um caminho alternativo para obter derivadas parciais em relação à entrada de controle que não passa pelo sistema. Werbos usou o conceito de retropropagação para conseguir isso usando HDP dependente de ação (AD-HDP). Watkins introduziu noções semelhantes para MDP de espaço discreto, que ele chamou de Aprendizagem Q.

O método de aprendizagem Q resulta em um algoritmo de controle adaptativo que converge *online* para a solução de controle ótimo para sistemas completamente desconhecidos. Ou seja, resolve a equação de Bellman e a equação de HJB *online* em tempo real usando dados medidos ao longo das trajetórias do sistema, sem qualquer conhecimento da dinâmica $f(x_k)$, $g(x_k)$.

A HDP é a estrutura de ADP mais básica e é ilustrada na figura 4.6. Nesta estrutura o crítico estima o aprendizado da função valor baseado diretamente na informação do estado x_k da planta, o crítico não requer o modelo da planta para o cálculo, o algoritmo de HDP utiliza o modelo da planta somente para a atualização do parâmetro do controlador (ator) (WANG, ZHANG e LIU, 2009).

Figura 4.6: Estrutura de HDP



No aprendizado da função valor $V_h(x_k)$ ou HDP, o conhecimento da dinâmica do sistema é exigido na atualização da política de controle $h(\cdot)$. No mínimo, é necessária a função de acoplamento de entrada $g(\cdot)$ ou a matriz B. Isso ocorre porque ao executar a minimização (sem restrições de controle)

$$h_{j+1}(x_k) = \arg\min_{h(\cdot)} \left(r(x_k, h(x_k)) + \gamma\, W_{j+1}^T \phi(x_{k+1}) \right) \tag{4.47}$$

deve-se derivá-la em relação à variável de controle para obter

$$0 = \frac{\partial}{\partial u_k}(Q(x_k) + u_k^T R u_k) + \frac{\partial}{\partial u_k} \gamma W^T{}_{j+1}\, \phi(x_{k+1})$$

$$= 2Ru_k + \left(\frac{\partial}{\partial u_k} \phi(x_{k+1}) \right)^T \gamma W_{j+1}$$

$$= 2Ru_k + \left(\frac{\partial x_{k+1}}{\partial u_k} \right)^T \nabla \phi^T(x_{k+1}) \gamma W_{j+1} \tag{4.48}$$

Entretanto, ao avaliar

$$\frac{\partial x_{k+1}}{\partial u_k} = g(x_k) \tag{4.49}$$

necessita-se da função de entrada $g(\cdot)$.

Para evitar tal conhecimento, Werbos propôs uma abordagem denominada Programação Dinâmica Heurística Dependente de Ação (*Action Dependent Heuristic Dynamic Programming* - ADHDP), também chamada Aprendizagem Q. Resolve a equação de *Bellman* e a equação de HJB de maneira *online* usando dados medidos ao longo da trajetória do sistema. Para isso, aprende a função Q usando métodos de diferença temporal através da execução de uma ação de controle u_k e medição à cada estágio de tempo.

Considere a equação de Bellman (4.9), que permite calcular o valor de usar qualquer política admissível prescrita $h(\cdot)$. O controle ótimo é determinado usando (4.10) ou (4.11). Portanto, define-se a função Q (qualidade) associada à política $u = h(x)$ como

$$Q_h(x_k, u_k) = r(x_k, u_k) + \gamma V_h(x_{k+1}) \tag{4.50}$$

Observa-se que a função Q é uma função tanto do estado x_k quanto do controle u_k no instante k. Ela tem sido denominada de função valor de ação. Define-se a função Q ótima como

$$Q^*(x_k, u_k) = r(x_k, u_k) + \gamma V^*(x_{k+1}) \tag{4.51}$$

Em termos de Q^*, pode-se escrever a equação da otimilidade de Bellman de forma mais simples

$$V^*(x_k) = \min_u(Q^*(x_k, u)) \tag{4.52}$$

e o controle ótimo como

$$h^*(x_k) = \arg\min_u(Q^*(x_k, u)) \tag{4.53}$$

Para empregar técnicas de aprendizado *online* para aprender a função Q, determina-se: 1. uma equação do ponto fixo para Q para usar TD, e 2. uma estrutura de aproximador paramétrico adequada para Q para usar VFA (Q *function approximation* - QFA).

Para obter uma equação do ponto fixo para a função Q, inicialmente deve-se observar que $Q_h(x_k, h(x_k) = V_h(x_k)$. Portanto, uma equação de Bellman para Q é dada por

$$Q_h(x_k, u_k) = r(x_k, u_k) + \gamma Q_h(x_{k+1}, h(x_{k+1})) \tag{4.54}$$

ou

$$Q_h(x_k, h(x_k)) = r(x_k, h(x_k)) + \gamma Q_h(x_{k+1}, h(x_{k+1})) \tag{4.55}$$

O valor ótimo Q satisfaz

$$Q^*(x_k, u_k) = r(x_k, u_k) + \gamma Q^*(x_{k+1}, h^*(x_{k+1})) \tag{4.56}$$

Esta equação é uma equação do ponto fixo ou equação de Bellman para Q. Compare-a com a equação (4.9). Agora, pode-se usar qualquer método de aprendizado por reforço *online* como base para a ADP, incluindo IP e IV.

4.9 Função Q para o DT LQR

De acordo com (4.50), a função Q para o caso LQR é dada por

$$Q_K(x_k, h(x_k)) = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1} \tag{4.57}$$

Em que P é a solução para a equação de Lyapunov para a política prescrita K. Portanto,

$$Q_K(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k + (A x_k + B u_k)^T P (A x_k + B u_k) \tag{4.58}$$

ou

$$Q_K(x_k, u_k) = \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q + A^T PA & A^T PB \\ B^T PA & R + B^T PB \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \equiv z_k^T H z_k \tag{4.59}$$

Isto é a função Q para o LQR, que é quadrática em (x_k, u_k).

Usando o produto de Kronecker, escreve-se

$$Q_K(x_{k,}\, u_k) = \bar{H}^T \bar{z}_k \tag{4.60}$$

com $\bar{H}$ = vec(H) e $\bar{z}_k = \ z_k \otimes z_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix} \otimes \begin{bmatrix} x_k \\ u_k \end{bmatrix}$ o conjunto de base quadrática composto pelos componentes do estado e pela entrada de controle. Então, a equação do ponto fixo é

$$\bar{H}^T \bar{z}_k = x_k^T Q x_k + u_k^T R u_k + \bar{H}^T \bar{z}_{k+1} \tag{4.61}$$

com $u_k = -K x_k$.

Assume-se um aproximador paramétrico da forma

$$Q_h(x, u) = W^T \phi\,(x, u) = W^T \phi\,(z) \tag{4.62}$$

com $\varphi\,(x, u)$ um conjunto de base de funções de ativação. Isso produz o erro TD

$$e_k = r(x_k, h(x_k)) + \gamma\, W^T \phi(\, x_{k+1}) - W^T \phi(x_k) \tag{4.63}$$

Os métodos de aprendizagem por reforço, incluindo IP ou IV, podem ser usados para aprender $\bar{H}$ = vec(H) *online*. A Iteração de Política é ilustrada abaixo. RLS ou descida do gradiente podem ser usado para identificar a função Q associada a uma dada política K como discutido no Algoritmo de Iteração de Política e Iteração de Valor *online*. E assim, considera-se que o problema de aproximação da função valor Q pode ser formulado por uma estrutura paramétrica similar a equação (4.62), pelo estimador RLS, da forma

$$Q(x, u) = \boldsymbol{\varphi}^T(z)\boldsymbol{\theta}, \tag{4.64}$$

para alguns parâmetros desconhecidos do vetor de pesos a serem estimados $\boldsymbol{\theta} = \left[\theta_1\ \theta_2\ \ldots\ \theta_{n_\theta}\right]^{\boldsymbol{T}}$ e conjunto do vetor de funções de base ou vetor de regressão $\boldsymbol{\varphi}(z) = \left[\phi_1(z)\ \phi_2(z)\ \ldots\ \phi_{n_\theta}(z)\right]^{\boldsymbol{T}}$ com $z_k = [x_k^T\ u_k^T]^T$. Esta aproximação leva em consideração o método RLS para o problema de estimação do parâmetro $\boldsymbol{\theta}$. Tal abordagem busca realizar a aprendizagem *online* para projetos de controle ótimo, via estimativas da função de custo de uma dada política, construída a partir dos dados $(z_k, z_{k+1}, r(x_k, u_k))$, que são observados ao longo da trajetória do sistema.

Para o caso LQR, define

$$Q_K(x_k, u_k) = z_k^T H z_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \tag{4.65}$$

Então produz

$$u_k = -(H_{uu})^{-1} H_{ux} x_k, \tag{4.66}$$

Uma vez que a matriz de Kernel quadrática H foi encontrada usando a aprendizagem por reforço $online$, a dinâmica do sistema não é necessária para esta etapa. Para o caso geral não-linear com a rede neural do crítico (4.62) obtém-se

$$\frac{\partial}{\partial u} Q_h(x_k, u) = \frac{\partial}{\partial u} W^T \phi(x_k, u) = 0 \tag{4.67}$$

No algoritmo de Iteração de Política para a função Q, inicialmente, seleciona-se qualquer política de controle $h_0(x_k)$, que seja admissível/estabilizadora. A etapa seguinte será avaliar essa política através da solução dos mínimos quadrados

$$W_{j+1}^T(\phi(z_k) - \gamma\phi(z_{k+1}) = r\big(x_k, h_j(x_k)\big) \tag{4.68}$$

e, por fim, aplica-se a melhoria de política

$$h_{j+1}(x_k) = \arg\min_{h(\cdot)} \big(W_{j+1}^T \phi(x_k, u)\big) \tag{4.69}$$

Como essa rede neural depende explicitamente da ação de controle u (HDP dependente da ação), as derivadas podem ser calculadas sem referência a detalhes adicionais, tal como o modelo da dinâmica do sistema. Várias idéias de Werbos são intrigantes sobre Q. Primeiro, um caminho alternativo foi obtido para retropropagar a derivada parcial $\partial/\partial u_k$, sem passar pela dinâmica do sistema. Em segundo lugar, a rede neural do crítico da função Q, (4.62), tem agora não apenas o estado x_k mas também a ação de controle u_k como suas entradas. Esta é a razão pela qual $\partial/\partial u_k$ pode ser avaliado sem passar através do sistema. Diz-se que a rede neural do crítico depende agora da ação; Werbos, denomina isso de HDP dependente de ação (AD-HDP).

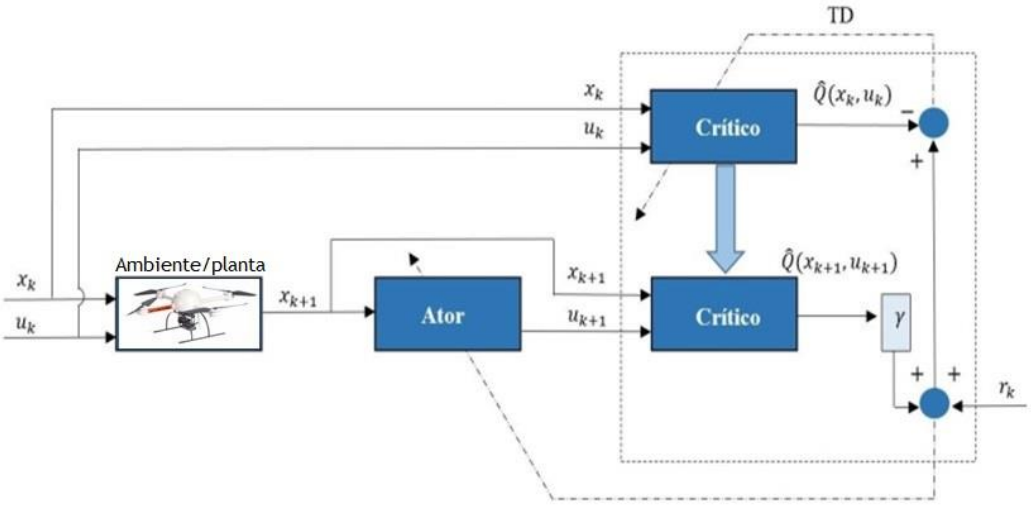
Os métodos de aprendizagem por reforço AD-HDP baseados na função Q podem ser determinados usando a equação de Bellman ou ponto fixo

$$Q_h(x_{k,}\, h(x_k)) = r(x_k, h(x_k)) + \gamma Q_h(x_{k+1}, h(x_{k+1})) \tag{4.70}$$

Tanto o Valor quanto a função Q são escalares para que o aprendizado seja avaliado com base em um estímulo de resposta bastante escasso do ambiente e a convergência pode ser lenta para sistemas com grande quantidades de estados.

A Estrutura de ADHDP é mostrada na Figura 4.7. Observa-se que tanto o estado x_k quanto o controle u_k são entradas para o crítico. O controle é obtido através da equação do gradiente, $\frac{\partial \hat{Q}(x_k,u_k)}{\partial u_k} = 0$. O método ADHDP não requer o conhecimento do modelo da planta para o treinamento do ator.

Figura 4.7: Estrutura de ADHDP



5. RESULTADOS DE SIMULAÇÃO

Neste capítulo são apresentados os resultados de simulação que ilustram o projeto de controle ótimo adaptativo via métodos de Programação Dinâmica Aproximada (ADP) para um sistema de quadricóptero, com seis graus de liberdade (6-DOF). Na Seção 5.1, realiza-se o projeto do controlador ótimo de forma *offline*, ou seja, primeiramente são fornecidas soluções pelos métodos de iteração de política e iteração de valor em suas versões *offline*, levando em consideração aspectos de convergência dos algoritmos IP e IV em relação a número de iterações e custo computacional. A avaliação de desempenho da resposta do sistema de controle proposto neste trabalho é apresentada na Seção 5.2, em que soluções *offline* de controle ótimo são aplicadas como políticas de referência para avaliar a exatidão das soluções de controle *online* obtidas pelos algoritmos de ADP. Uma das questões de interesse na análise dos resultados, é verificar, a partir de uma condição de operação anterior estável, a adaptabilidade dos controladores baseados em ADP perante variações nos parâmetros da planta (variação na massa do quadricóptero), durante o movimento de voo pairado do veículo. Os experimentos computacionais foram realizados em ambiente MATLAB.

5.1 Resultados de Controle Ótimo *Offline*

Nesta seção, são apresentados os resultados baseados nas formulações matemáticas deduzidas no capítulo 4 desse trabalho. Ressaltando que estes são para a resposta *offline* do sistema, utilizando o modelo linearizado do sistema de quadricóptero descrito no capítulo 3. Para fins de projeto de controle DT LQR, considera-se as matrizes A_d e B_d do modelo discretizado, expostas no Apêndice A. Nas descrições das figuras apresentadas abaixo, tem-se resultados para a Iteração de Valor e Iteração de Política. A Iteração de Valor requer apenas uma iteração do mapeamento de avaliação de política iterativa, sendo menos custoso computacionalmente, porém demanda um tempo maior para convergir. No entanto, a Iteração de Política executa o mapeamento até a convergência, tendo um processo computacional mais custoso, porém, converge bem mais rápido.

As figuras 5.1 (a)-(b) e 5.2 (a)-(b) descrevem a convergência dos coeficientes da matriz de parâmetros P de Riccati por IP e IV.

Figura 5.1 (a): Convergência dos coeficientes da matriz P de Ricatti por IV

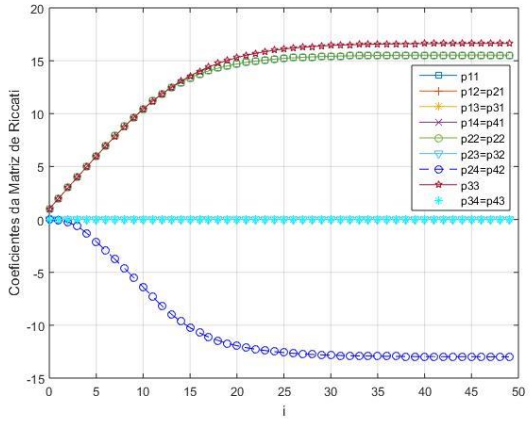


Figura 5.1 (b): Convergência dos coeficientes da matriz P de Ricatti por IP

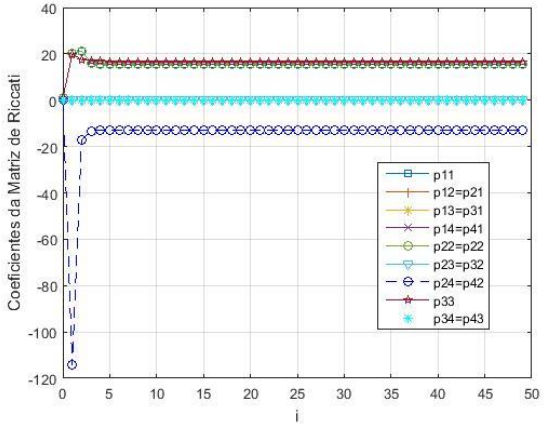


Figura 5.2 (a): Convergência dos coeficientes da matriz P de Ricatti por IV

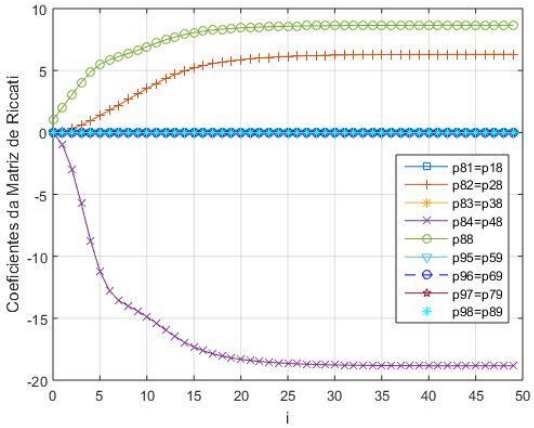
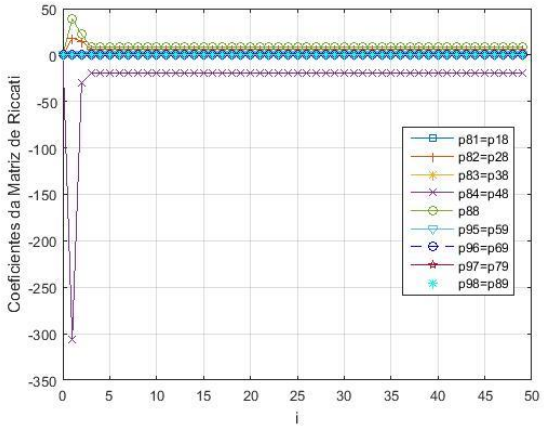


Figura 5.2 (b): Convergência dos coeficientes da matriz P de Ricatti por IP



O objetivo é selecionar o ganho K de realimentação do estado, ou seja, a política de controle para minimizar o *cost-to-go*, para todos os estados atuais, as iterações. Vê-se nas figuras 5.3 (a) e (b), 5.4 (a) e (b), 5.5 (a) e (b) e 5.6 (a) e (b) a resposta de ganho ótimo de realimentação de estados K para todos os elementos de ganhos.

Figura 5.3 (a): Convergência dos elementos da matriz de ganho Kde realimentação dos estados por IV. Elementos k_1 a k_{12}.

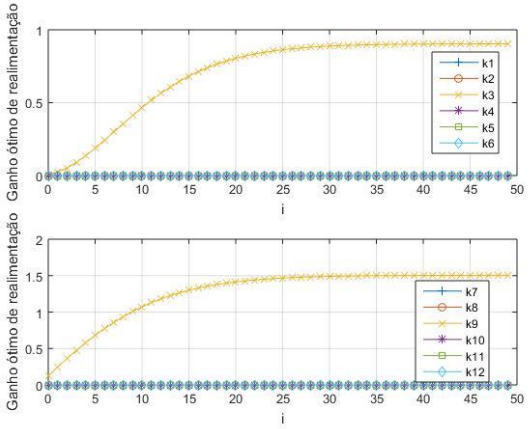


Figura 5.3 (b): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IP. Elementos k_1 a k_{12}.

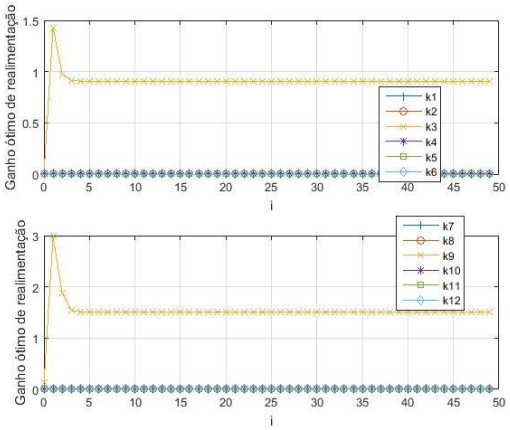


Figura 5.4 (a): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IV. Elementos k_{13} a k_{24}.

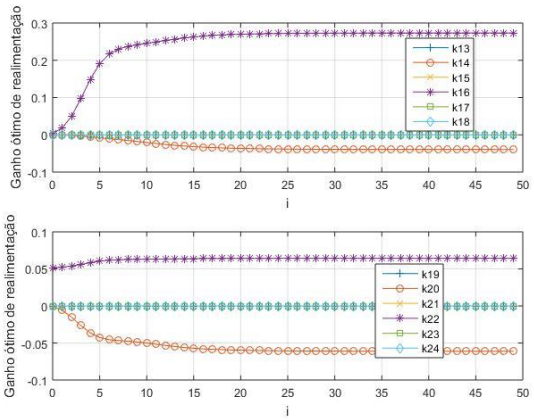


Figura 5.4 (b): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IP. Elementos k_{13} a k_{24}.

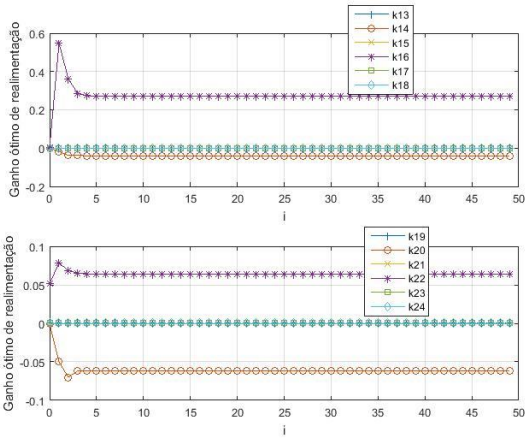


Figura 5.5 (b): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IV. Elementos k_{25} a k_{36}.

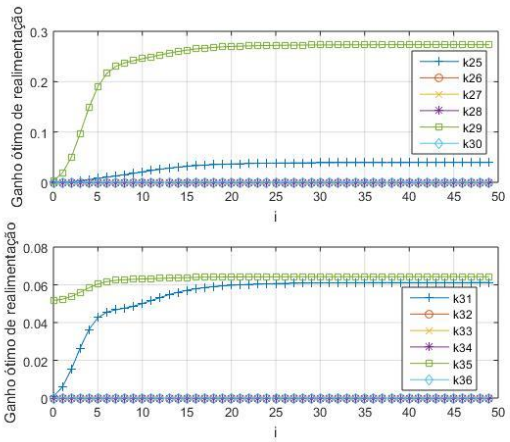


Figura 5.5 (b): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IP. Elementos k_{25} a k_{36}.

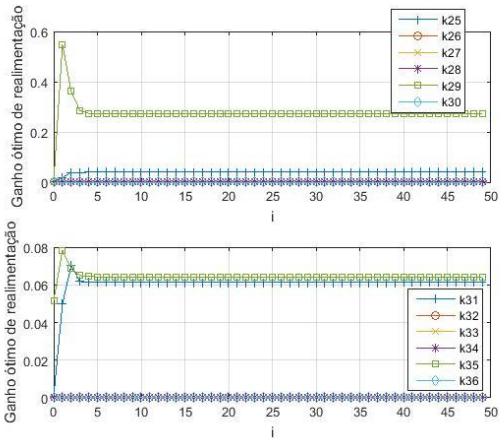


Figura 5.6 (a): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IV. Elementos k_{37} a k_{48}.

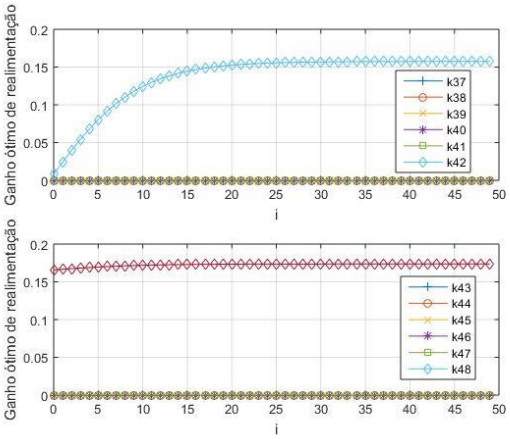
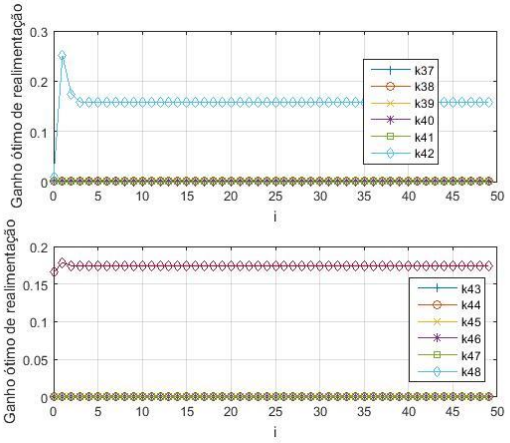


Figura 5.6 (b): Convergência dos elementos da matriz de ganho K de realimentação dos estados por IP. Elementos k_{37} a k_{48}



A trajetória ótima para cada variável de estado x_k pode ser observada nas figuras 5.7 (a)-(b), 5.8 (a)-(b), 5.9 (a)-(b) e 5.10 (a)-(b). Percebe-se que pelo método IP, os estados alcançam a condição de equilíbrio, $x_k = 0$, mais rápido, mantendo assim a premissa citada anteriormente, a qual estratégia IP requer um tempo de convergência menor em comparação com IV.

Figura 5.7 (a): Trajetória dos Estados por IV. Estados $x_1 = X$, $x_2 = Y$ e $x_3 = Z$

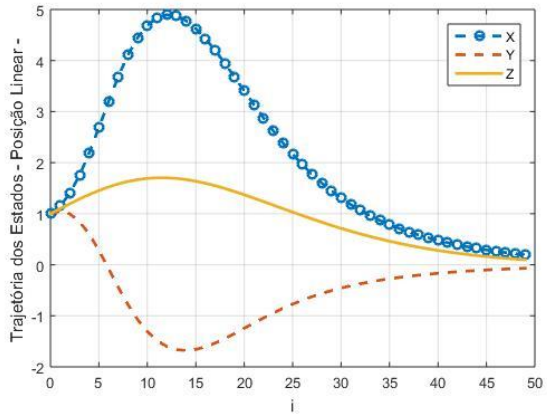


Figura 5.7 (b): Trajetória dos Estados por IP. Estados $x_1 = X$, $x_2 = Y$ e $x_3 = Z$

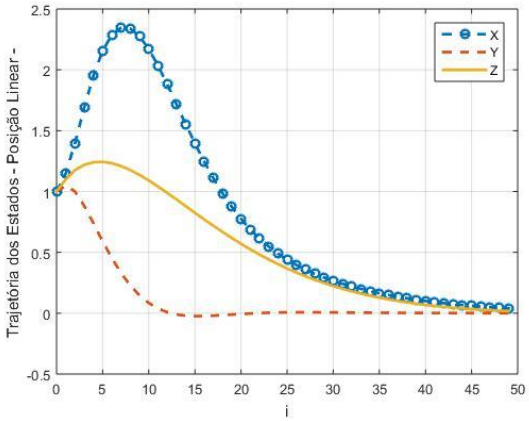


Figura 5.8 (a): Trajetória dos Estados por IV. Estados $x_4 = \phi$, $x_5 = \theta$ e $x_6 = \psi$

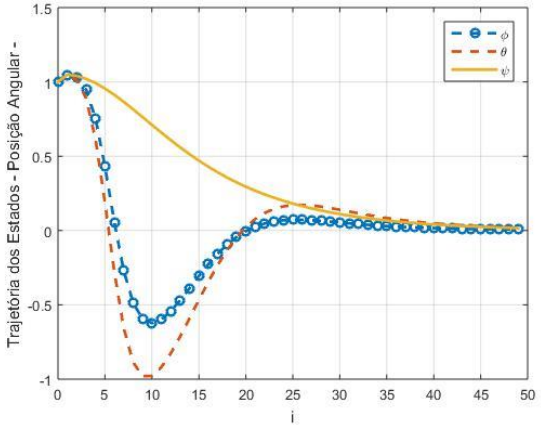


Figura 5.8 (b): Trajetória dos Estados por IP. Estados $x_4 = \phi$, $x_5 = \theta$ e $x_6 = \psi$

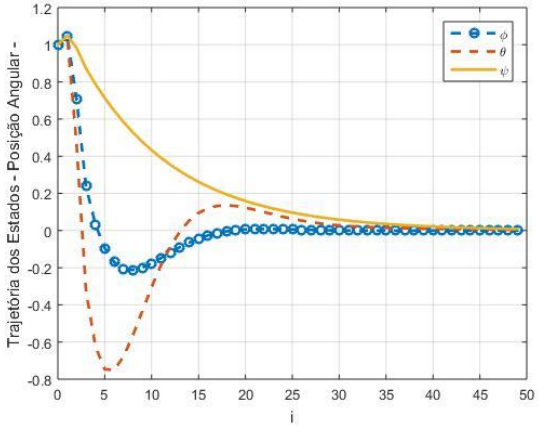


Figura 5.9 (a): Trajetória dos Estados por IV. Estados $x_7 = u$, $x_8 = v$ e $x_9 = w$

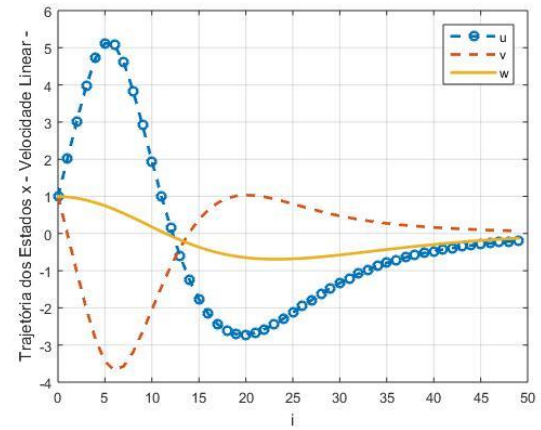


Figura 5.9 (b): Trajetória dos Estados por IP. Estados $x_7 = u$, $x_8 = v$ e $x_9 = w$

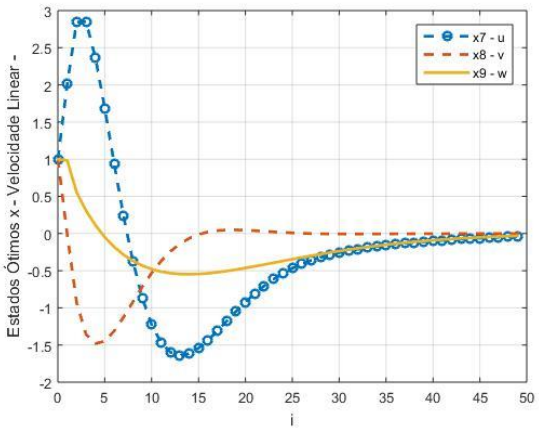


Figura 5.10 (a): Trajetória dos Estados por IV. Estados $x_{10} = p$, $x_{11} = q$ e $x_{12} = r$.

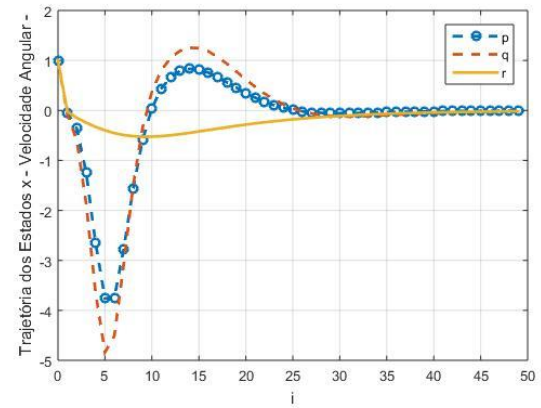
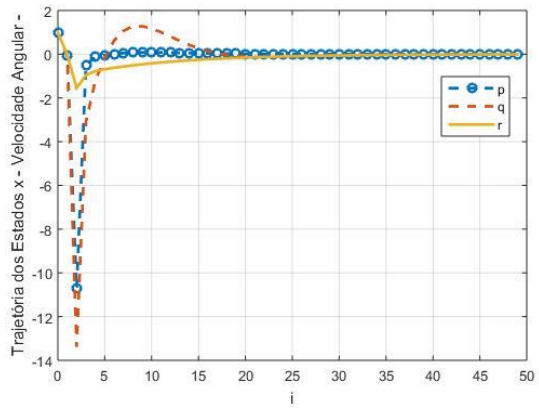


Figura 5.10 (b): Trajetória dos Estados por IP. Estados $x_{10} = p$, $x_{11} = q$ e $x_{12} = r$.



Nas figuras 5.11 (a)-(b), a função de custo fornece o ponto de melhoria no sistema, com ganhos de desempenho ótimo, duração e economia. E, as figuras 5.12 (a)-(b) e 5.13 (a)-(b), ilustram as trajetórias das ações de controle durante os processos iterativos de IP e IV.

Novamente, percebe-se que a Iteração de Política gera mais rápido os custos e ações de controle ótimos, apesar de ser mais custoso no ponto de vista computacional ao sistema.

Figura 5.11 (a): Custo Mínimo por IV.

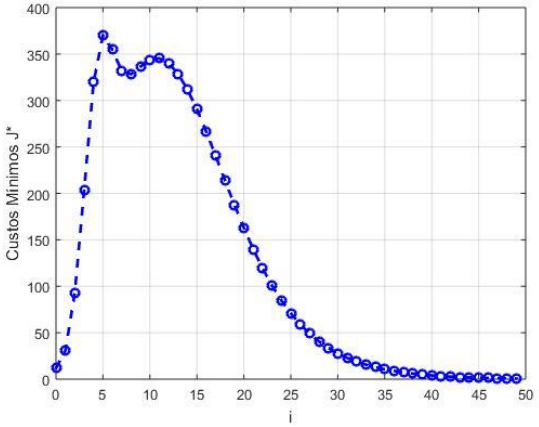


Figura 5.11 (b): Custo Mínimo por IP.

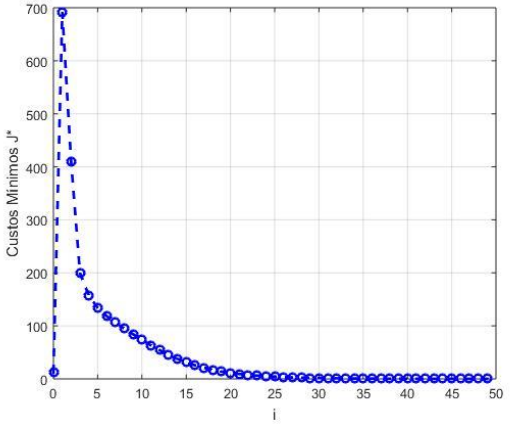


Figura 5.12 (a): Ação de Controle por IV. Entradas de Controle u_1 e u_2

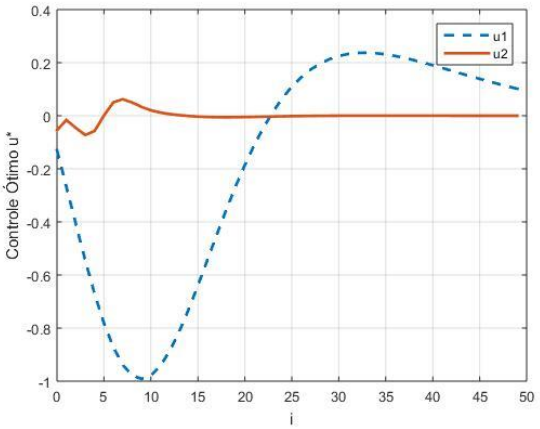


Figura 5.12 (b): Ação de Controle por IP. Entradas de Controle u_1 e u_2

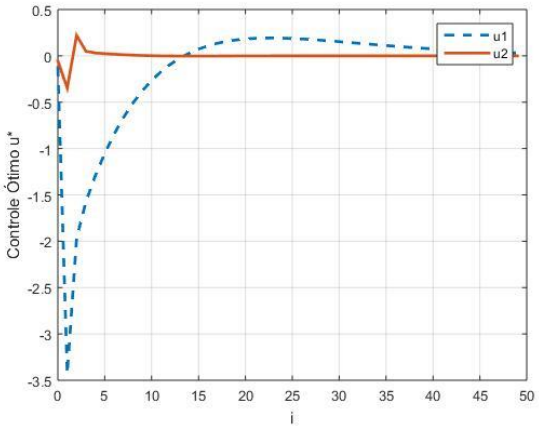


Figura 5.13 (a): Ação de Controle por IV. Entradas de Controle u_3 e u_4

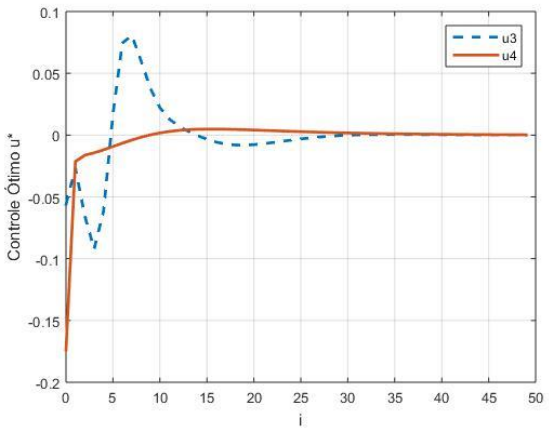
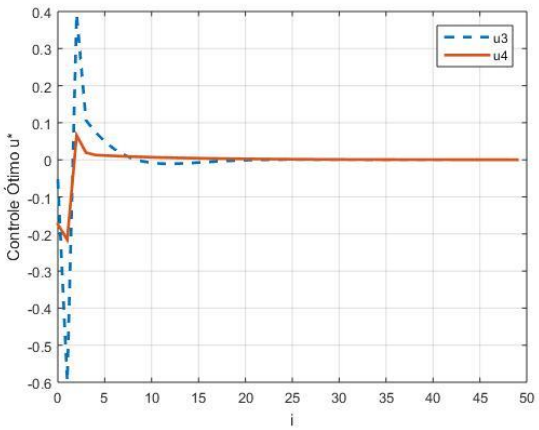


Figura 5.13 (b): Ação de Controle por IP. Entradas de Controle u_3 e u_4



Conclui-se que as soluções por IP e IV podem gerar ganhos ótimos de controle para um sistema onde se conheça a dinâmica da planta, o ambiente (ou parte dele, como é o caso do quadricóptero) e oferecem a escolha para o projetista de controle avaliar os recursos computacionais sob os quais o algoritmo necessitará para sua execução, tais como velocidade de execução e utilização de memória.

Todavia, vale ressaltar que os resultados são para o método *offline*, usando o modelo da dinâmica da planta (A_d, B_d). Na seção 5.2 as formulações de Programação Dinâmica Aproximada serão aplicadas para a verificação dos ganhos ótimos de controle que podem ser obtidos sem criteriosamente trabalhar com o modelo do sistema dinâmico. Apresentar-se-á os valores e condições iniciais para implementação e *setup* da simulação *online*.

5.2 Resultados de Controle Ótimo *Online* via ADP

Para tornar a Iteração de Política e a Iteração de Valor *online*, é indispensável utilizar a Aproximação da Função Valor (VFA) e o Erro da Diferença Temporal (TD), pois como já descritos no capítulo 4, são procedimentos para resolver a equação HJB e formular os métodos de Aprendizagem por Reforço em tempo real solucionando o problema de controle ótimo usando dados medidos ao longo das trajetórias do sistema. Esses passos são chamados de Programação Dinâmica Aproximada (ADP).

As etapas dos experimentos computacionais constituem o procedimento para avaliar o desempenho de algoritmos para o projeto de controladores via ADP. Nesse procedimento, são realizadas comparações para avaliar a precisão das soluções dos controladores propostos (políticas de decisão). Para tal propósito, a política de referência estabelecida é aquela fornecida pela solução IV *offline* da equação HJB. Tal política é usada para mostrar que os algoritmos de ADP desenvolvidos neste trabalho tem a capacidade de alcançar uma solução suficientemente próxima da solução exata.

O Setup do processo iterativo de ADP consiste em estabelecer quais são os melhores parâmetros para resolver uma dada aplicação, que nesse caso é otimizar a equação de custo, o índice de desempenho, tendo como referência o empirismo do projetista ou as relações entre as variáveis do sistema relacionados aos métodos. Para tal procedimento, usa-se parâmetros e condições iniciais que desempenham um papel relevante na convergência dos métodos.

Dentre as diversas simulações realizadas, selecionou-se diferentes valores para o fator de esquecimento μ, parâmetro relevante no processo de convergência e estabilidade numérica do estimador RLS, o qual é utilizado neste trabalho para determinar o parâmetro $vec(P)$. O parâmetro μ pondera dinamicamente a influência de dados amostrados na função de perda, descontando os dados mais antigos, de modo a tornar $vec(P)$ representativo para as propriedades atuais do sistema.

Neste trabalho, as matrizes de ponderação Q e R foram definidas pelo método de busca empírica de acordo com a expertise do usuário. Considerou-se Q e R com variações de testes em que $Q = 10^{qi} I_{12\times12}$ e $R = 10^{ri} I_{4\times4}$, onde qi e ri assumem valores de peso para avaliar os custos das políticas de controle e desvios dos estados durante o processo iterativo de ADP.

Neste projeto de controle, que é visando a adaptabilidade e redução de esforço de controle, se estipulou dados de variações para o peso do veículo estudado (quadricóptero) baseado na ideia de que o veículo pode ser usado para transportar um mecanismo de varredura óptica,

sensoriamento e/ou localização dentro do Centro de Lançamento de Alcântara (CLA) para checagem local, onde será feito o lançamento de satélites ou testes de lançamento, objetivando a fiscalização ou segurança do lançamento ou moradores da região etc.

Tal objetivo, a adaptabilidade, que é o cerne deste projeto de controlador ótimo via programação dinâmica aproximada, pode-se utilizar o quadricóptero com fins para a segurança; observar a rampa de lançamento; filmagens no momento do lançamento e análise visual do comportamento do foguete; sensoriamento e inspeção da região, principalmente a trajetória nominal do foguete, a qual hoje é inspecionada como o próprio veículo transportador de carga e pessoas do Centro (o avião do CLA), gerando custo elevados e ocupação desnecessária deste veículo, impossibilitando disponibilidade para socorro ou transporte humano no momento da checagem.

Os testes com variação de peso no quadricóptero foi idealizado para verificar a capacidade deste controlador estabilizar o mais rápido possível e manter-se qualificado para continuar a operação para o qual estava proposto, mantendo sua robustez e ratificando sua adaptabilidade na varredura, quer seja pelas perturbações externas, perdas de componentes de checagem (sensores, câmeras, etc.) ou até mesmo após liberar os balões meteorológicos para verificação da velocidade do vento nos instantes antecedentes ao lançamento.

Após todo o desenvolvimento do processo metodológico, espera-se como resultados finais os seguintes pontos, sendo aplicado a Programação Dinâmica Aproximada como base para alcançá-los:

- Resposta otimizada em relação aos processos convencionais de controle (sistemas *offline*), sob perturbações externas e incertezas;
- Formulações dos passos da metodologia atenderem as soluções *online* do sistema do controlador ótimo e para o sistema de controle adaptativo;
- Esperar que os resultados gerados possibilitem reduções de esforços de controle.

5.2.1 Setup das simulações

O setup do algoritmo de ADP consiste da informação que está relacionada aos parâmetros do sistema dinâmico (simulador do ambiente), ponderações de custo instantâneo, condições iniciais e parâmetros do processo iterativo. A informação é estabelecida tendo como referência a expertise do projetista ou as relações entre variáveis do sistema relacionadas aos métodos, e ela consiste dos seguintes elementos: matrizes do modelo discretizado (A_d, B_d) do

quadricóptero, matrizes de ponderação de estado e controle (Q, R), tempo de amostragem t_{samp}, fator de desconto γ, condições iniciais do ganho do controlador (política de controle inicial) e da estimativa da matriz P de Riccati (K_0, P_0) e número de iterações N. Quanto ao estimador RLS, tem-se: intervalo de revitalização de estado n_{revit} com respeito a condição de excitação persistente, condições iniciais da matriz de covariância, fator de esquecimento μ, e revitalização dos estados. Nas simulações, θ representa $vec(P)$.

5.2.2 Solução da equação HJB-Ricatti

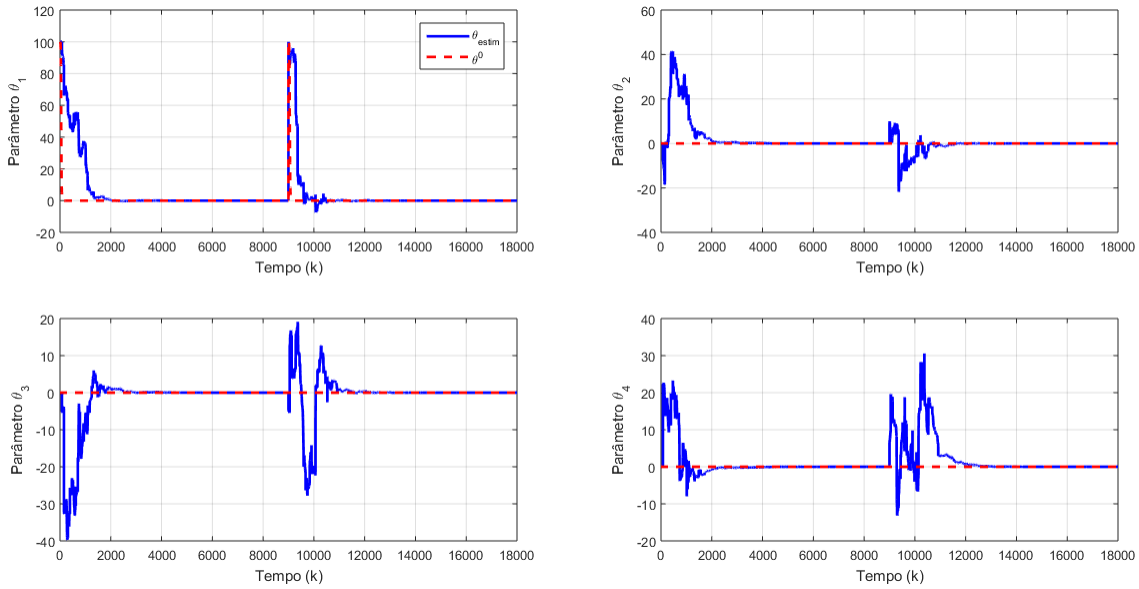
A estrutura paramétrica da função de custo DT LQR tem seu parâmetro $\boldsymbol{\theta}$ estimado utilizando o método RLS. A abordagem RLS considerada é para viabilizar a solução *online* da equação HJB-Riccati associada ao projeto de controle ótimo DT LQR. As componentes do parâmetro $\boldsymbol{\theta}$ estão associadas com os elementos da matriz P de Ricatti. Na Seção 4.7.1, nota-se os elementos independentes dessa matriz que serão formadas, a partir da expressão $n(n+1)/2$, onde n é o número de estados do veículo, 12. Então, serão 78 elementos considerados para a avaliação do comportamento de convergência da matriz P de Riccati.

Vetor de parâmetros $\boldsymbol{\theta}$

A evolução do processo iterativo para a solução da equação discreta de Ricatti é dada nas figuras 5.14–5.17 para um ciclo de 18000 iterações. Com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$ na figura 5.14; fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$ na figura 5.15; fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$ na figura 5.16; fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$ na figura 5.17.

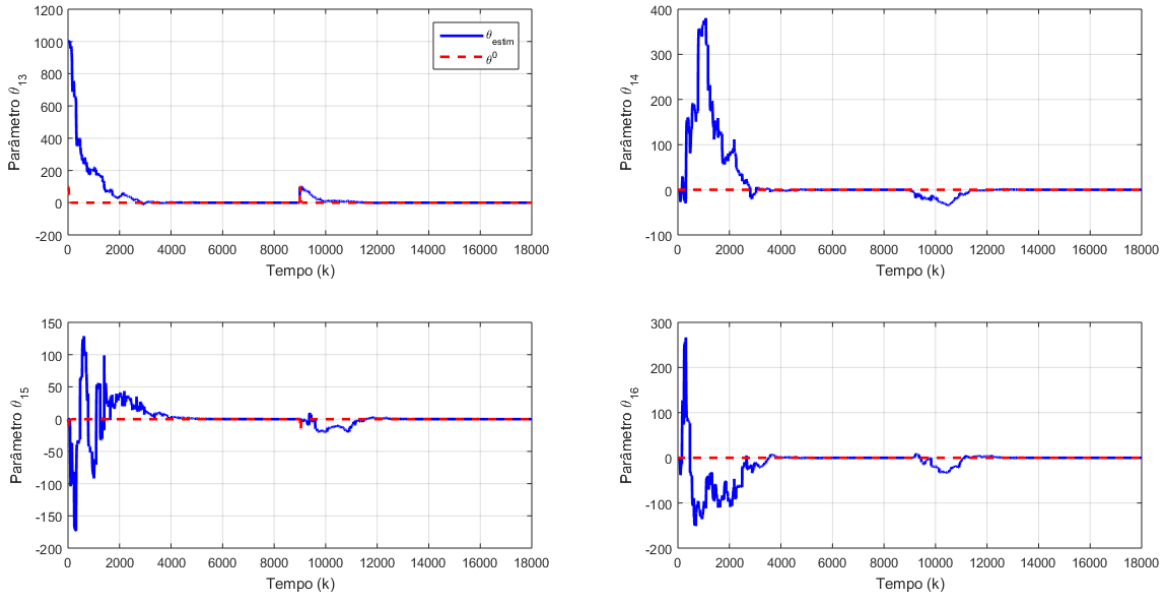
As informações apresentadas na figura 5.14 representam o comportamento de convergência dos elementos $p_{1,1}$, $p_{1,2}$, $p_{1,3}$ e $p_{1,4}$ da matriz P correspondente aos componentes do vetor de parâmetro $\boldsymbol{\theta}$: θ_1, θ_2, θ_3 e θ_4, respectivamente.

Figura 5.14: Evolução do processo iterativo dos parâmetros $p_{1,1}$, $p_{1,2}$, $p_{1,3}$ e $p_{1,4}$ para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$; o algoritmo padrão RLS–DTLQR



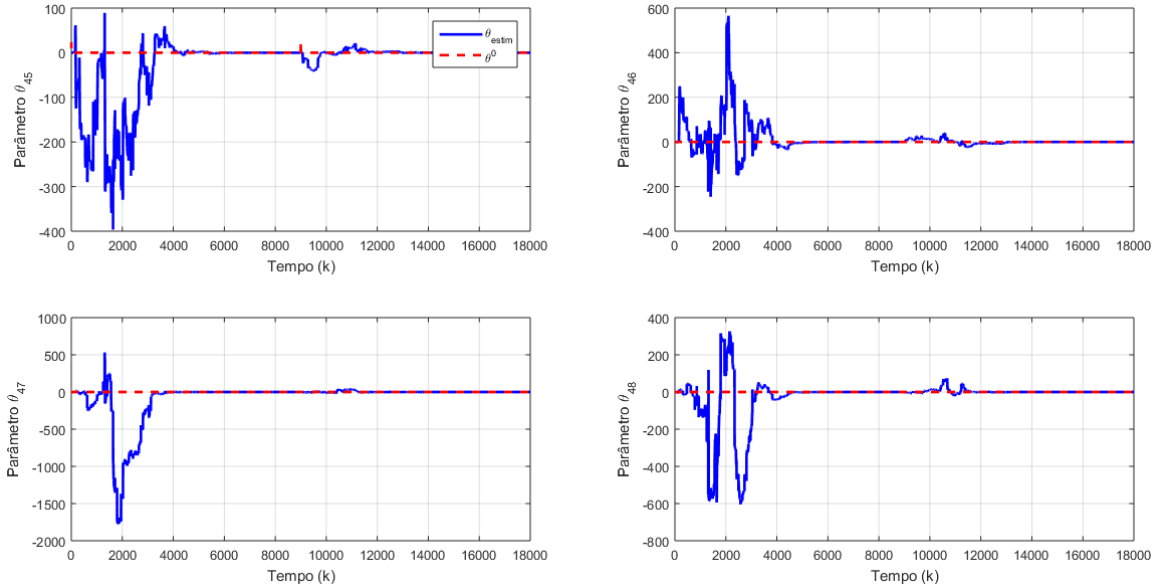
Na figura 5.15, mostra-se a evolução do comportamento de convergência dos elementos $p_{2,2}$, $p_{2,3}$, $p_{2,4}$ e $p_{2,5}$ da matriz P associada aos componentes do vetor de parâmetro $\boldsymbol{\theta}$: θ_{13}, θ_{14}, θ_{15} e θ_{16}, respectivamente.

Figura 5.15: Evolução do processo iterativo dos parâmetros $p_{2,2}$, $p_{2,3}$, $p_{2,4}$ e $p_{2,5}$ para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$; o algoritmo padrão RLS–DTLQR



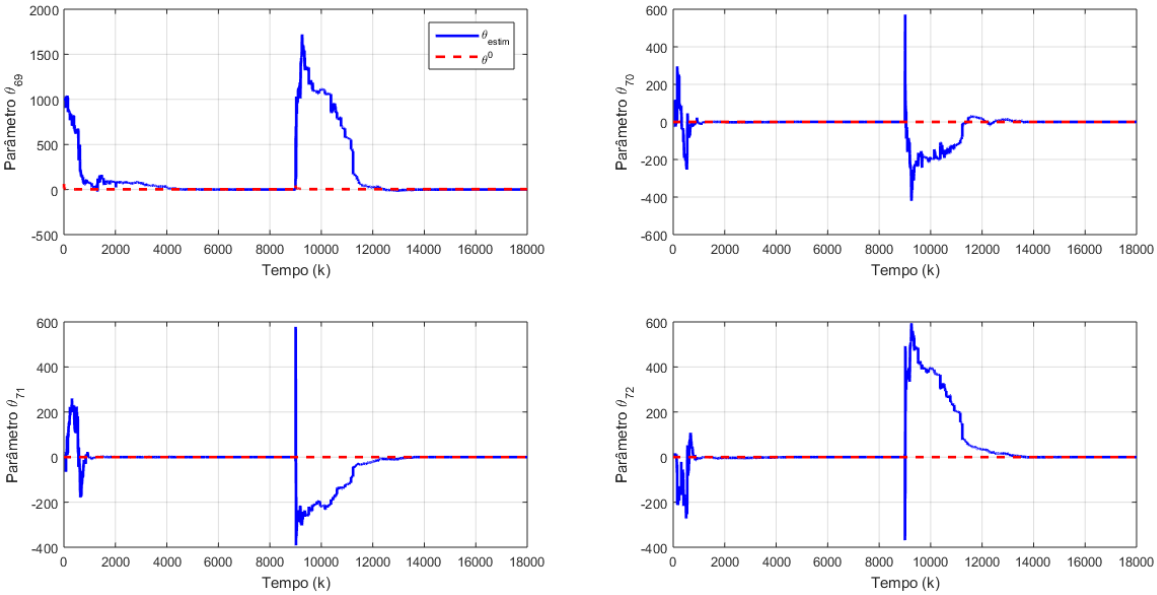
O comportamento de convergência dos elementos $p_{5,7}$, $p_{5,8}$, $p_{5,9}$ e $p_{5,10}$ da matriz P correspondente aos componentes do vetor de parâmetro $\boldsymbol{\theta}$: θ_{45}, θ_{46}, θ_{47} e θ_{48}, respectivamente, é mostrada na figura 5.16.

Figura 5.16: Evolução do processo iterativo dos parâmetros $p_{5,7}$, $p_{5,8}$, $p_{5,9}$ e $p_{5,10}$ para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$; o algoritmo padrão RLS–DTLQR



E, na figura 5.17, é ilustrado o comportamento de convergência dos elementos da matriz P: $p_{9,9}$, $p_{9,10}$, $p_{9,11}$ e $p_{9,12}$, correspondente aos componentes do vetor de parâmetro θ_{69}, θ_{70}, θ_{71} e θ_{72}, respectivamente.

Figura 5.17: Evolução do processo iterativo dos parâmetros $p_{9,9}$, $p_{9,10}$, $p_{9,11}$ e $p_{9,12}$ para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$; o algoritmo padrão RLS–DTLQR



Percebe-se claramente que os elementos convergem para o valor de referência com solução admissível em torno de 14000 iterações, apresentando estabilidade dos valores estimados em relação ao valor referencial, passando por uma variação abrupta no instante onde é

aplicada uma perturbação externa devido a mudança da massa do veículo para se observar o comportamento do sistema.

A ideia de se utilizar diferentes valores para o fator de esquecimento e fatores de peso nas matrizes de ponderação é a verificação do comportamento dos parâmetros da matriz P de Ricatti, ou seja, o comportamento do vetor de parâmetro $\boldsymbol{\theta}$ estimador RLS, pois obtendo comportamentos similares nos valores dos componentes θ_n, da quantidade $n(n+1)/2$ elementos independentes discutida na Seção 4.7.2, pode-se concluir que há estabilidade no sistema mantendo os valores estimados aos valores de referência, mesmo após a perturbação inserida no veículo e uma controlada sensibilidade dos efeitos dessa perturbação, apresentando severas oscilações somente em pontos específicos e pretendidos.

Após a variação na massa do quadricóptero, há um fator que deve ser verificado no processo iterativo do parâmetros $\boldsymbol{\theta}$, o qual também é esperado devido a mudança nos parâmetros iniciais do sistema do veículo, a política de referência IV *offline*. Espera-se que está sofra um determinado degrau, pois como houve mudança da massa do veículo, seu valor de referência sofre alteração.

Nas figuras 5.18 e 5.19, pode-se observar o comportamento da política de referência IV *offline* (linha tracejada), a qual é a linha de referência do sistema e como a mudança da massa do veículo leva a um salto de determinada ordem dessa linha.

Figura 5.18: Evolução do processo iterativo do parâmetro $p_{9,9}$ para um ciclo de 18000 iterações o algoritmo padrão RLS–DTLQR

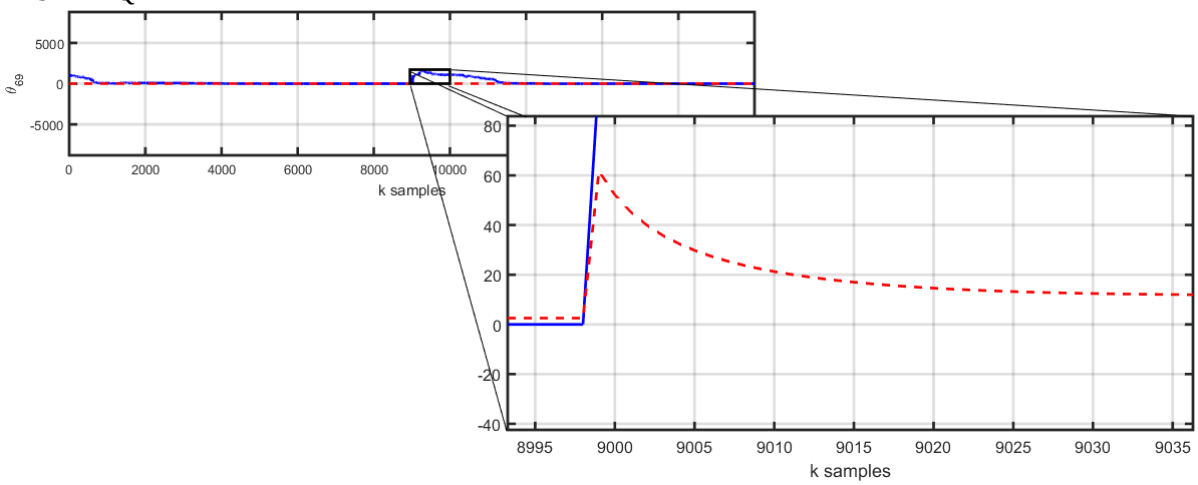
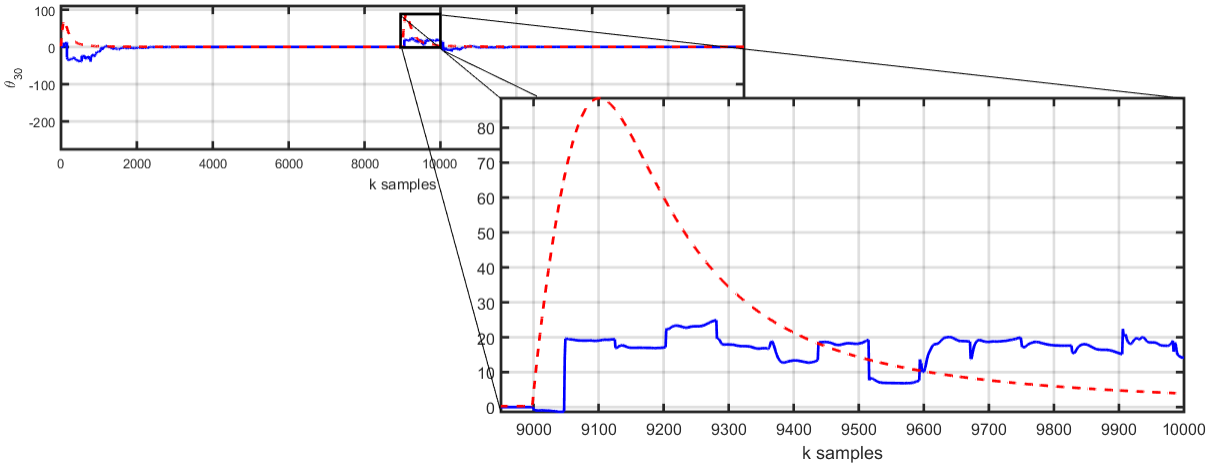


Figura 5.19: Evolução do processo iterativo do parâmetro $p_{3,9}$ para um ciclo de 18000 iterações o algoritmo padrão RLS–DTLQR



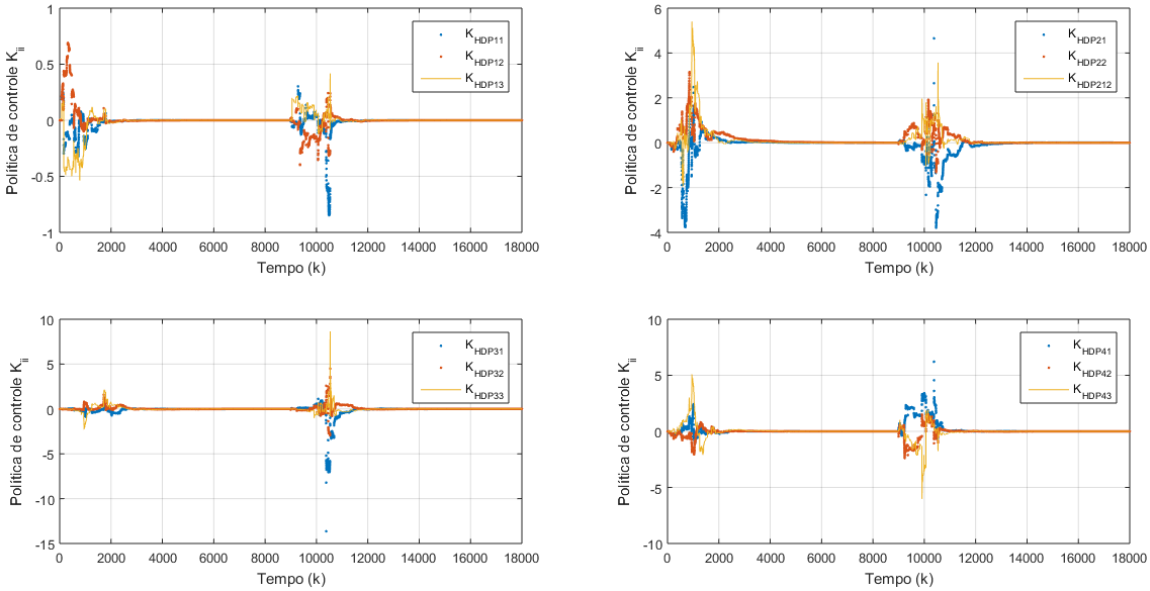
Após as simulações, tem-se sempre o resultado com a alteração da linha de referência, pois como citado, é esperado que assim aconteça, devido mudança da massa do veículo. No entanto, o que é importante aqui é a capacidade do controlador de estabilizar-se na presença dessa variação e a capacidade de voltar ao estado inicial de voo pairado, conforme desejado para este projeto.

Matriz de Ganho K

Para efeito de análise do desempenho do controlador baseado em ADP, mostra-se a seguir o comportamento do parâmetro K, oriunda da política de controle $u = -Kx$, equação (4.21), ou seja, a matriz de ganhos de realimentação. Percebe-se que após o sistema ser inicializado, há uma estabilidade em torno da iteração 4000 durando até o instante em que a perturbação é inserida na planta. A política de controle atua levando o sistema novamente a estabilizar.

Na figura 5.20, o comportamento da matriz de ganho é mostrado para os fatores de ponderação $qi = -5$ e $ri = -1$, quantificando que serão 48 elementos dessa matriz, no entanto, serão apresentados somente os mais relevantes.

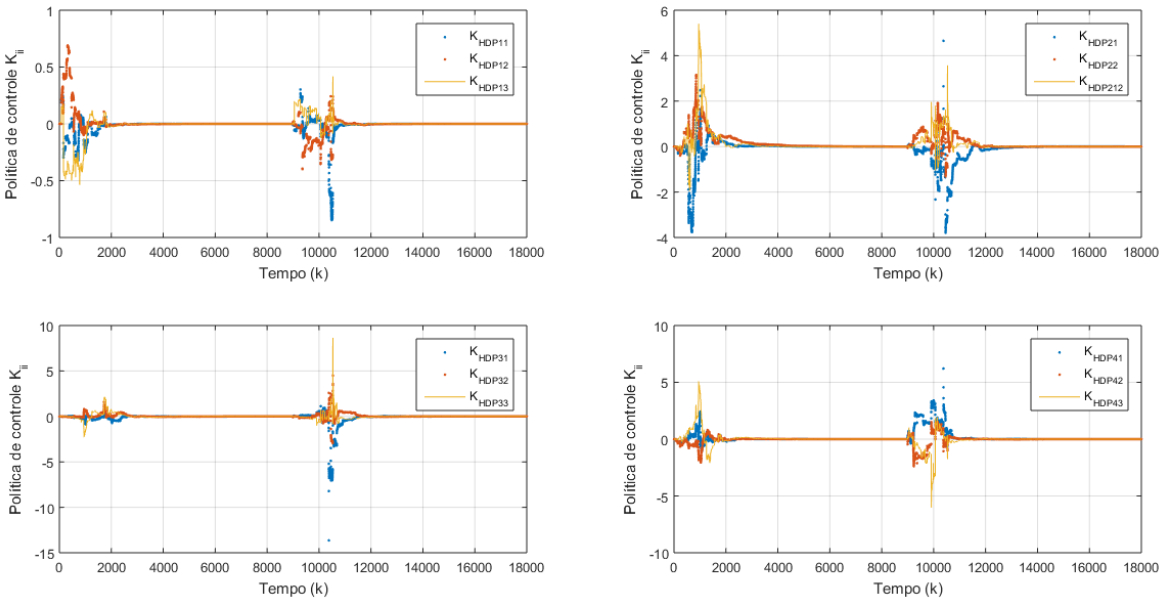
Figura 5.20: Convergência dos valores da Matriz de Ganho K para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$.



Comparar diferentes valores para o fator de esquecimento e fatores de peso nas matrizes de ponderação atribuem características mais relevantes para o projeto de controlador ótimo adaptativo e mais verificável para o projeto. Percebe-se claramente que os elementos convergem para o valor de referência com solução admissível em torno de 11000 iterações, apresentando estabilidade dos valores estimado.

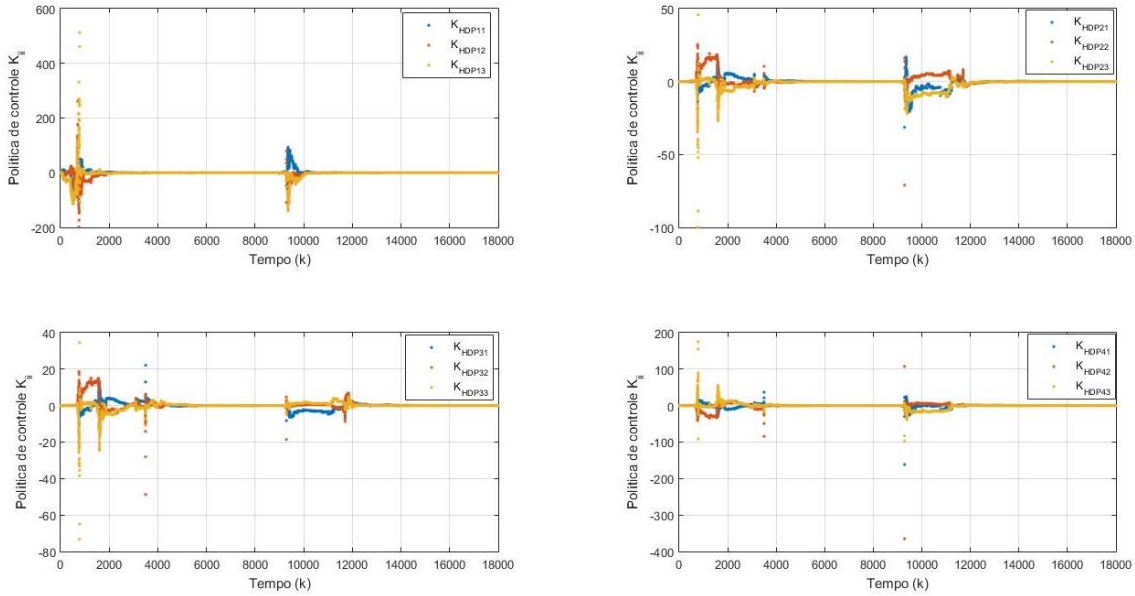
A figura 5.21 ilustra o comportamento da comportamento da matriz de ganho para os fatores de ponderação $qi = -4$ e $ri = -2$.

Figura 5.21: Convergência dos valores da Matriz de Ganho K para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$.



A figura 5.22 mostra o comportamento da política de controle para os fatores de ponderação $qi = -4$ e $ri = -4$.

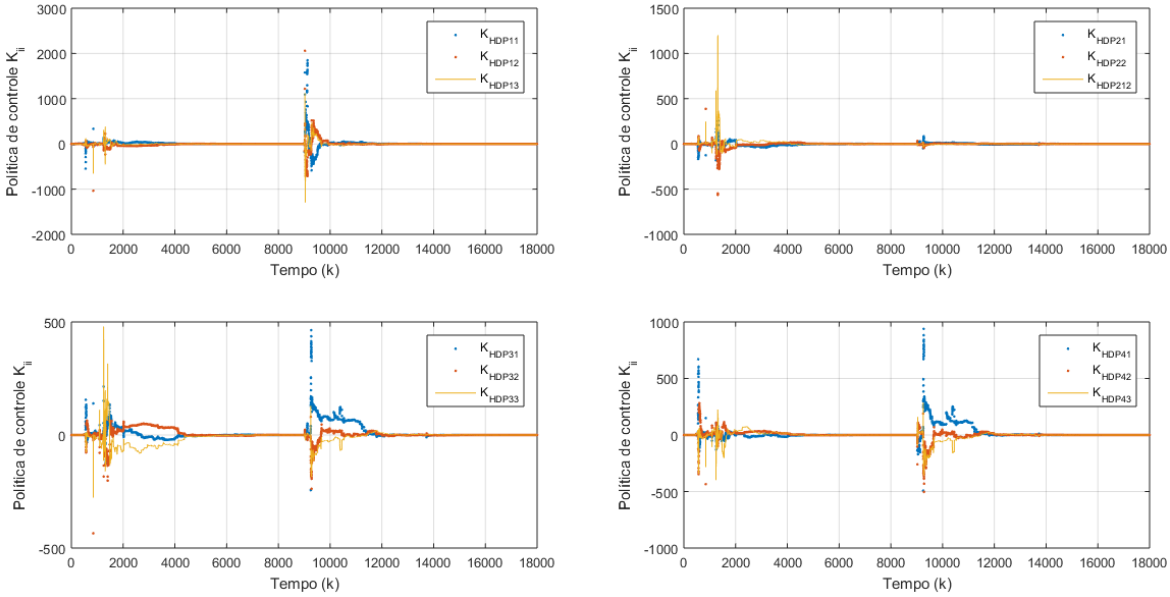
Figura 5.22: Convergência dos valores da Matriz de Ganho K para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$.



Percebe-se nesse comportamento da figura 5.22 a política de controle dos elementos da matriz de ganho, que converge para estabilidade durante todo o processo iterativo, ainda que apresentando oscilações dos valores de referência. Esse resultado é importante para o critério da escolha do fator de esquecimento e fatores de peso nas matrizes de ponderação que são determinantes para o projeto do controle ótimo adaptativo e, que as muitas simulações trazem mais confiabilidades para tal projeto.

Por fim, tem-se que a figura 5.23 ilustra o comportamento da política de controle para os fatores de ponderação $qi = -2$ e $ri = -2$.

Figura 5.23: Convergência dos valores da Matriz de Ganho K para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$.



Das figuras 5.20–5.23 percebe-se claramente que os elementos convergem para o valor de referência com solução admissível após 12000 iterações, apresentando estabilidade dos valores estimados em relação ao valor de referência. Os parâmetros passam por uma variação abrupta no instante onde é inserida uma perturbação externa devido a mudança da massa do veículo para se observar o comportamento da ação de política de controle e no instante inicial do processo iterativo, onde está aprendendo a política ótima pela trajetória do sistema.

Trajetória dos Estados x_k

Agora observa-se o comportamento da trajetória dos estados e seu comportamento na variação da massa do quadricóptero, conforme estipulado para o sistema. Os estados do quadricóptero são inicializados para serem x_0. As Entradas de Controle u_k e as Trajetórias dos Estados x_k para o veículo são mostradas em relação ao tempo. Para manter a condição de persistência de excitação necessária para a convergência do estimador RLS, isto é, evitar a singularidade da matriz de covariância do RLS, pode-se usar vários esquemas padrões, incluindo injeção de pequeno sinal de ruído à entrada de controle e reinicialização dos estados. Após esta etapa de inicialização, a dinâmica da aeronave avança no tempo e o ajuste das estruturas de parâmetros é realizado observando os estados *online*.

Nas figuras 5.24-5.27 pode-se analisar o comportamento dos estados x_1 a x_{12} na ação da ADP e com a variação de massa imposta ao sistema. Relembrando que os estados x_1 a x_3 e

x_4 a x_6 referem-se a posição linear e posição angular, respectivamente. Os estados x_7 a x_9 e x_{10} a x_{12} referem-se a velocidade linear e angular, respectivamente.

A figura 5.24 (a) e (b) retratam o comportamento dos estados na configuração do fator de esquecimento 0.9976 e dos pesos da matriz de ponderação $qi = -5$ e $ri = -1$.

Figura 5.24 (a): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$. Estados x_1 a x_6.

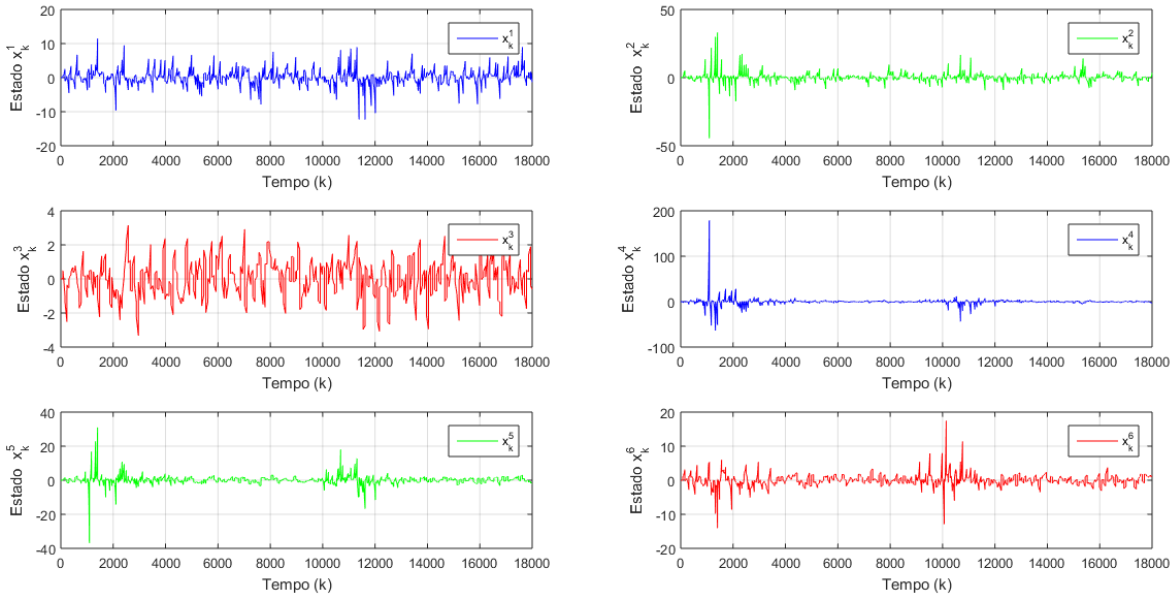
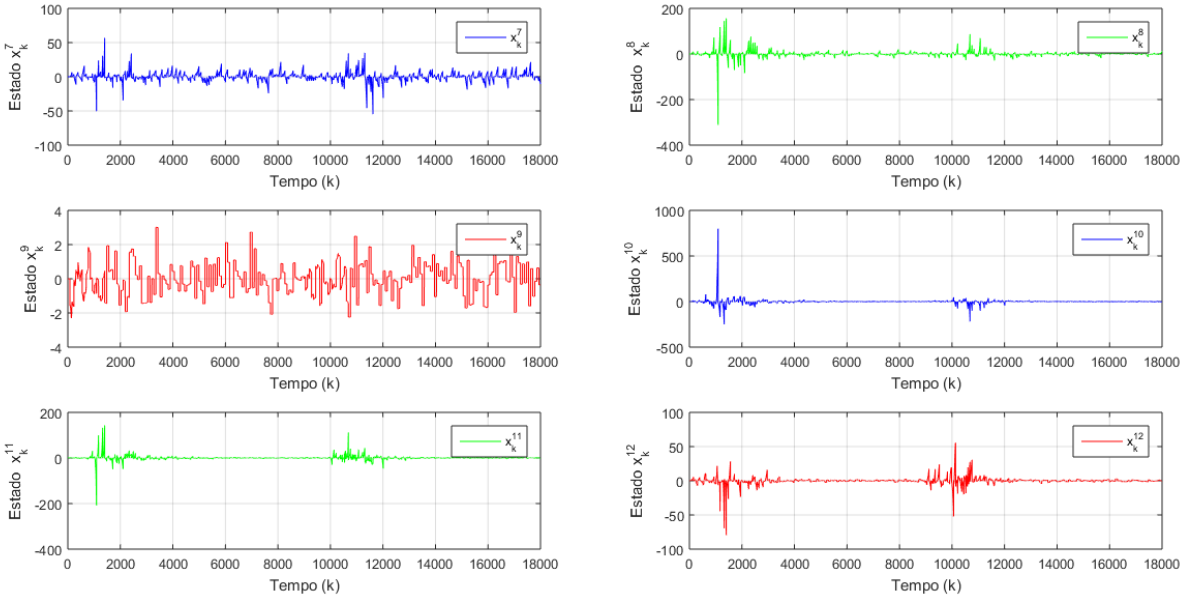


Figura 5.24 (b): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$. Estados x_7 a x_{12}.



É necessário observar a amplitude das respostas para ter a correta compreensão da estabilidade dos estados. Embora haja oscilações, elas tendem ao ponto de referência, ou seja, a estabilidade.

Em comparação, tem-se agora a figura 5.25 (a)-(b), na configuração do fator de esquecimento 0.9966 e dos pesos da matriz de ponderação $qi = -4$ e $ri = -2$.

Figura 5.25 (a): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$. Estados x_1 a x_6.

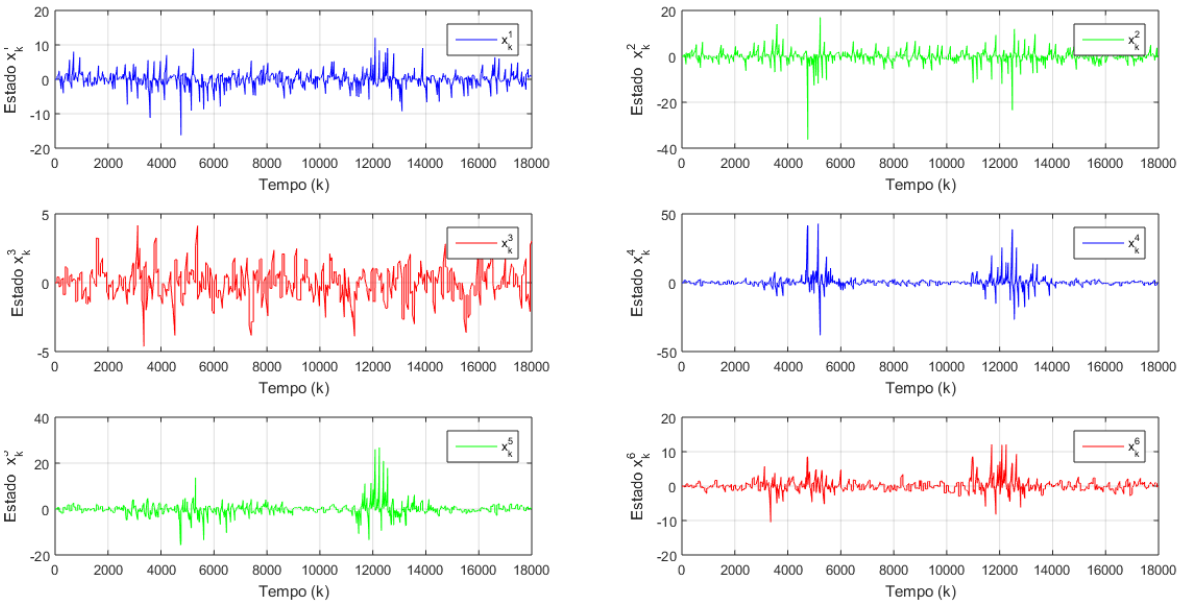
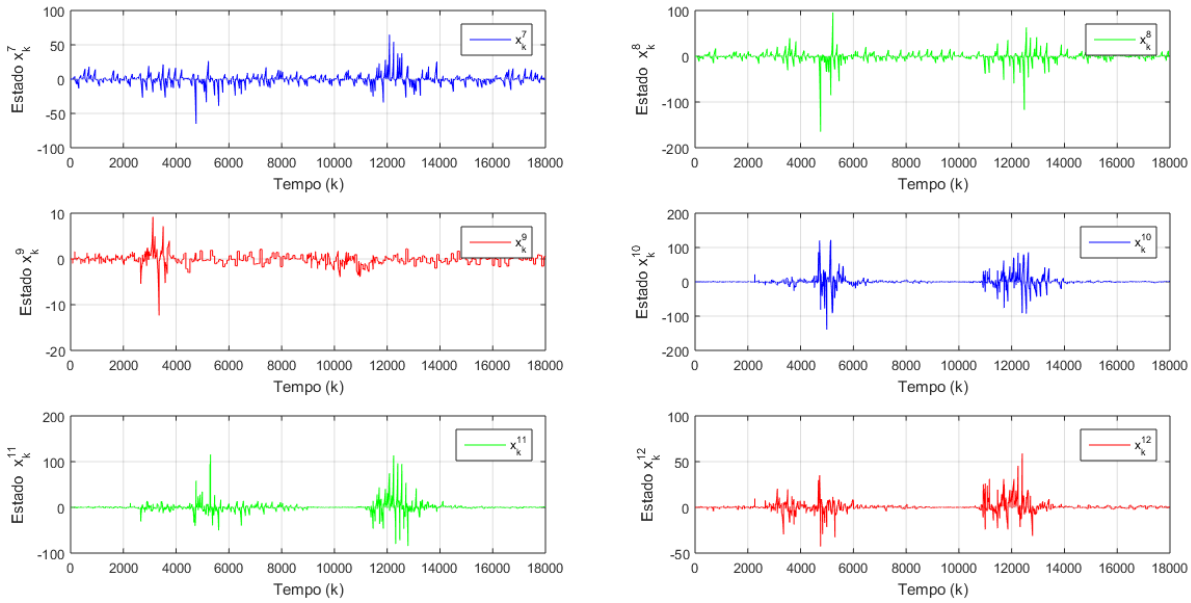


Figura 5.25 (b): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$. Estados x_7 a x_{12}.



Começar-se a perceber que há maiores picos de variação nas posições angulares (x_4 a x_6) e velocidades angulares (x_{10} a x_{12}), onde as amplitudes são relativamente maiores. Devido a mudança imposta à variação da massa do sistema, por ter que aprender a gerar uma nova matriz de ganho suficiente a se estabilizar e voltar a valores de referência.

Na configuração do fator de esquecimento 0.9946 e pesos da matriz de ponderação $qi = -4$ e $ri = -4$, obtêm-se as figuras 5.26 (a)-(b).

Figura 5.26 (a): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$. Estados x_1 a x_6.

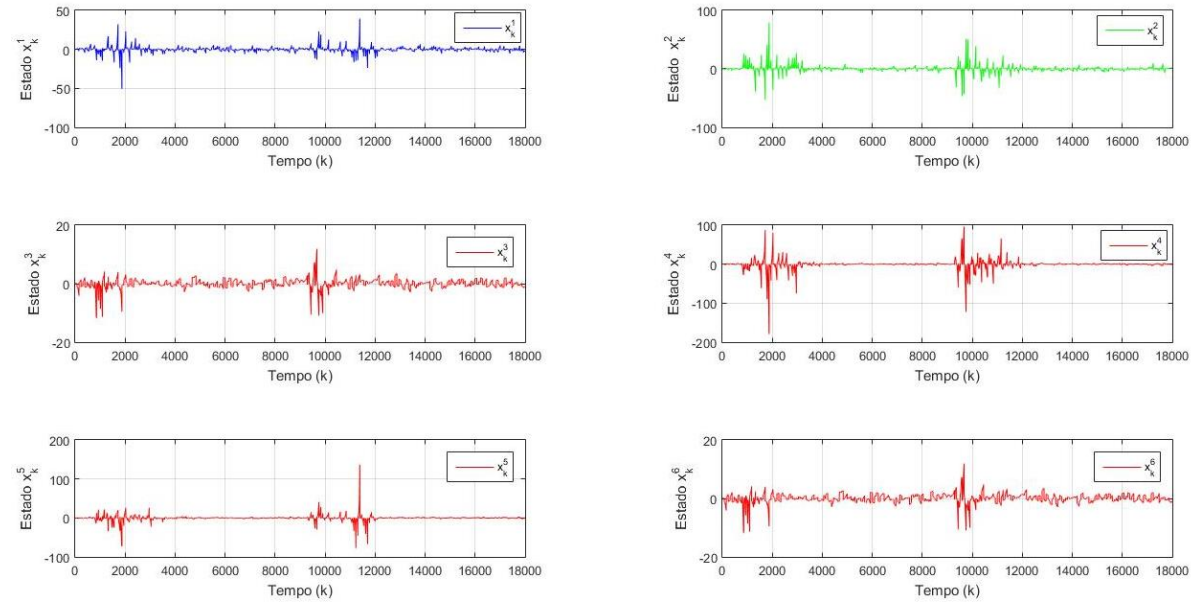
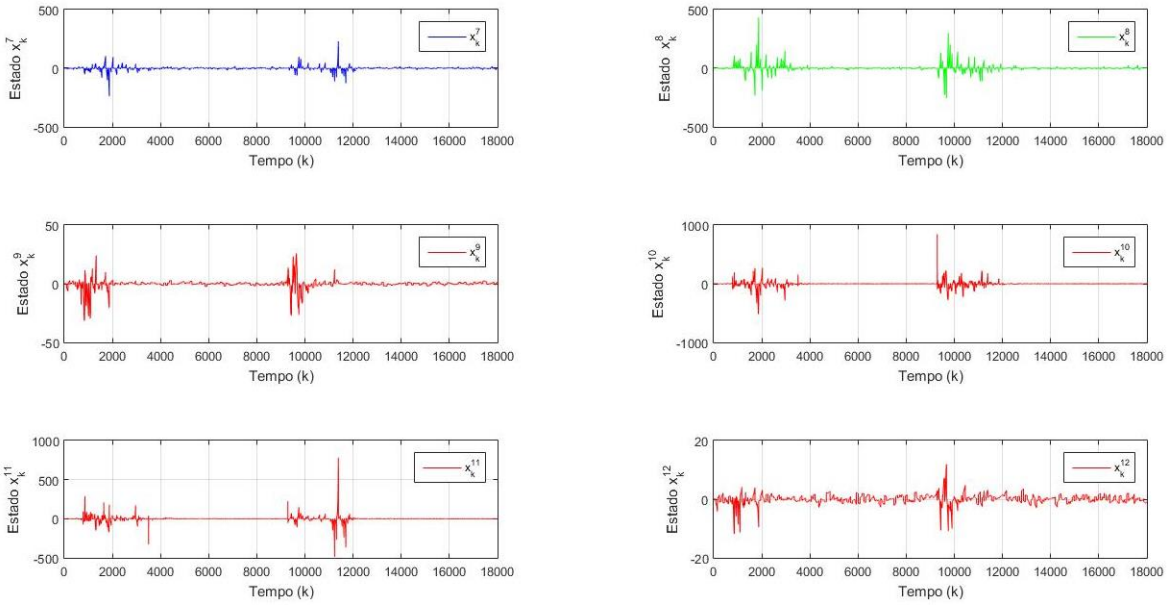


Figura 5.26 (b): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$. Estados x_7 a x_{12}.



E por fim, apresenta-se a resposta para a configuração do fator de esquecimento 0.9946 e pesos da matriz de ponderação $qi = -2$ e $ri = -2$, obtêm-se as figuras 5.27 (a) e (b), as quais mostram uma estabilidade mais acentuada em relação as outras configurações apresentadas. Com isso, percebe a importância de realizar diversas simulações, pois se tem um mapeamento maior de valores para uma resposta melhorada.

Figura 5.27 (a): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$. Estados x_1 a x_6.

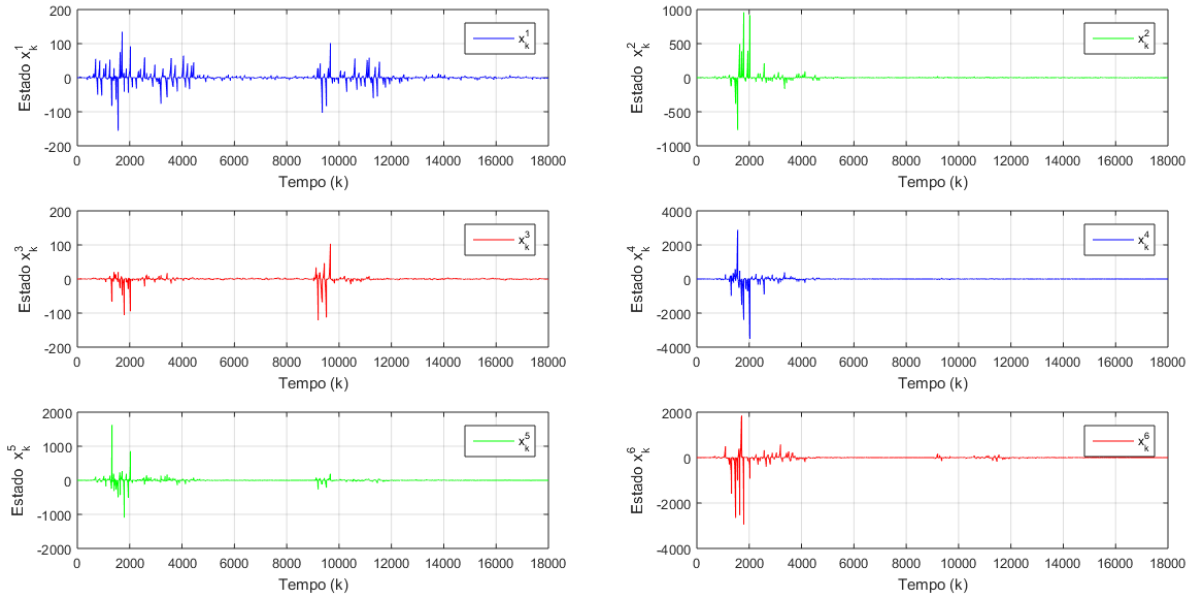
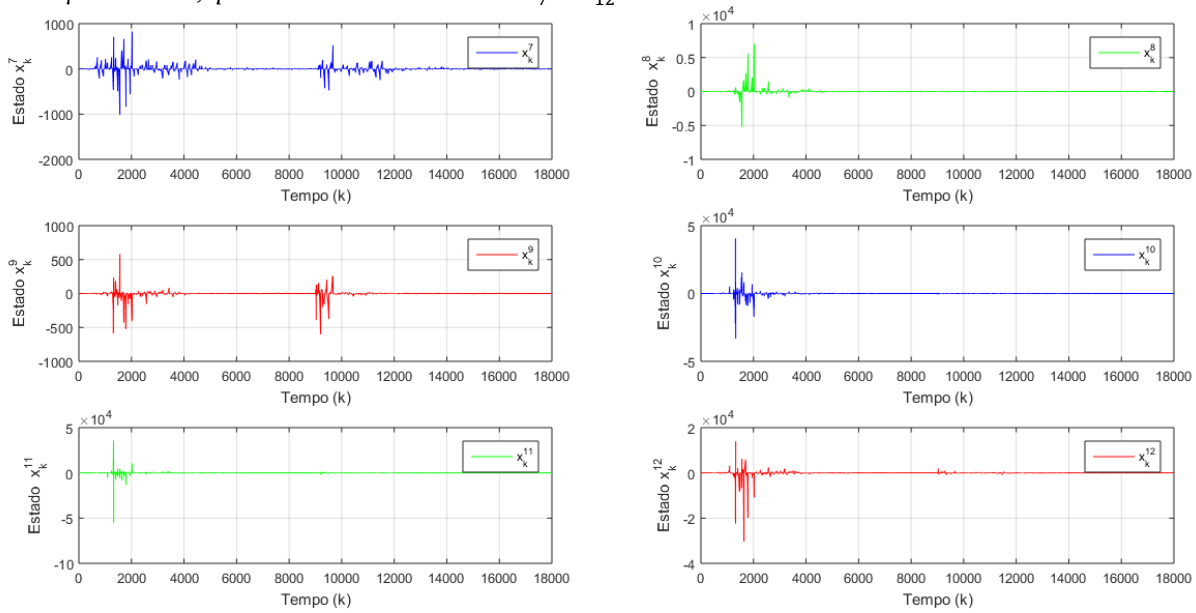


Figura 5.27 (b): Comportamento dos Estados do Sistema para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$. Estados x_7 a x_{12}.

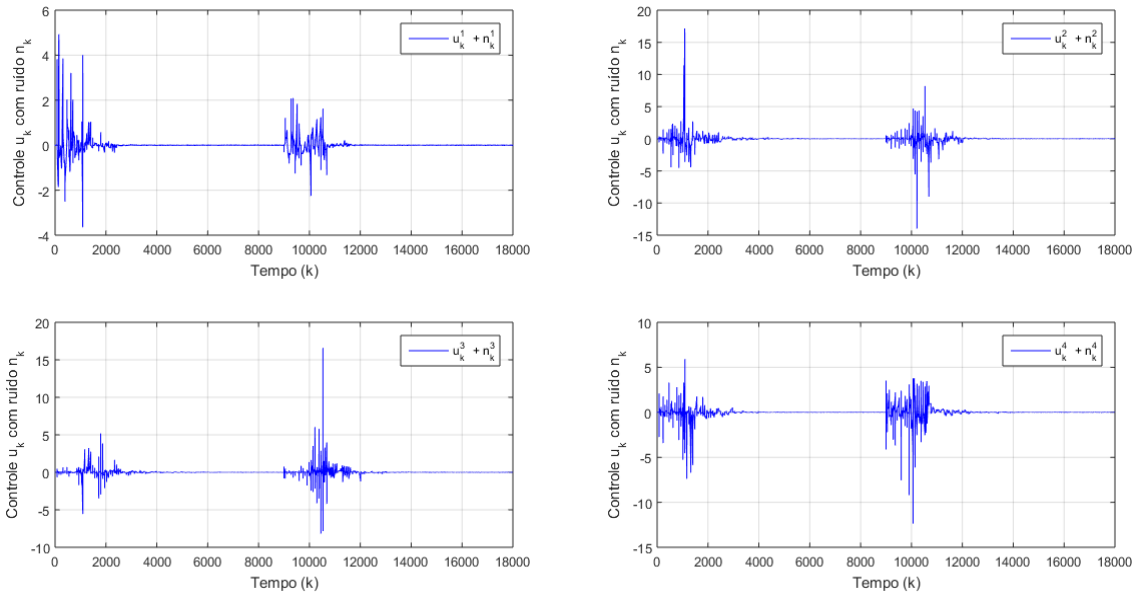


Ações de Controle u_k

Ações de Controle são procedimentos dentro da Lei de Controle (4.21) junto aos estados x, que geram valor necessário da matriz de ganho K. Juntamente com as Trajetórias dos Estados x_k, as Entradas de Controle u_k são mostradas em relação ao tempo para o veículo e incluem injeção de pequeno sinal de ruído. Abaixo serão apresentadas as respostas conforme a configuração estipulada para este projeto, nas figuras 5.28-5.30. A solução de controle ótimo é aprendida usando dados medidos ao longo das trajetórias do sistema.

Para a figura 5.28 tem-se a configuração do processo iterativo, o fator de esquecimento 0.9976 e pesos da matriz de ponderação $qi = -5$ e $ri = -1$.

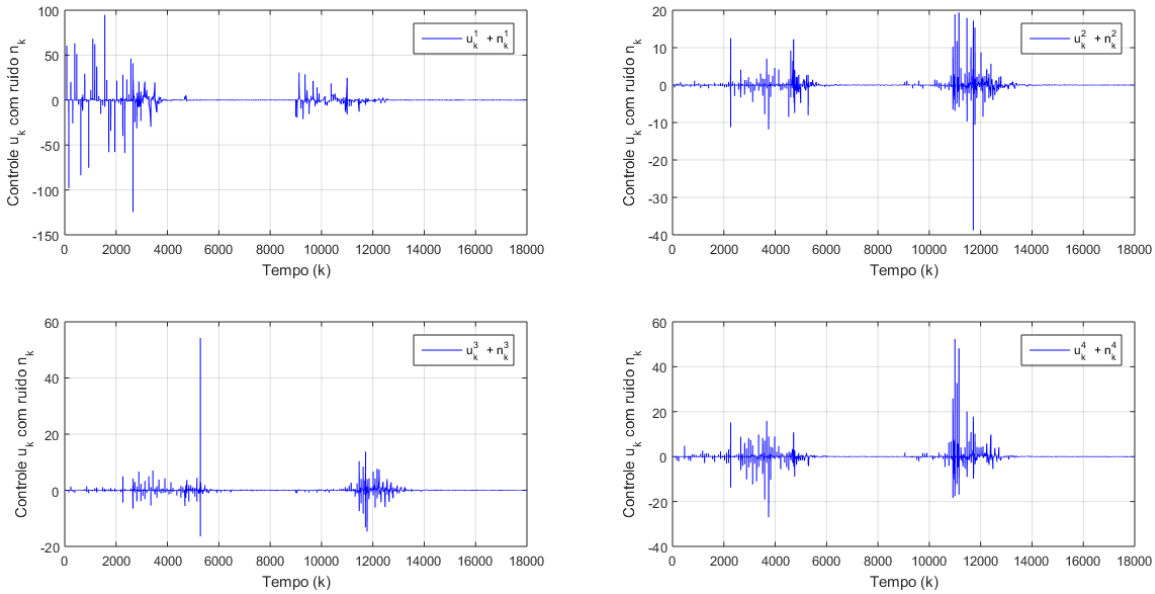
Figura 5.28: Ação de Controle para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$.



Conforme já esperado, as entradas de controle são requeridas no início do processo de estabilização no processo iterativo, consoante o objetivo do projeto que é estudo do quadricóptero em seu estado de voo pairado, definindo uma posição Z no espaço e, no intervalo das iterações onde há a perturbação, ou melhor, modificação da massa do sistema. Percebe-se que na ação da entrada de controle volta a estabilizar na região de 12000 iteração.

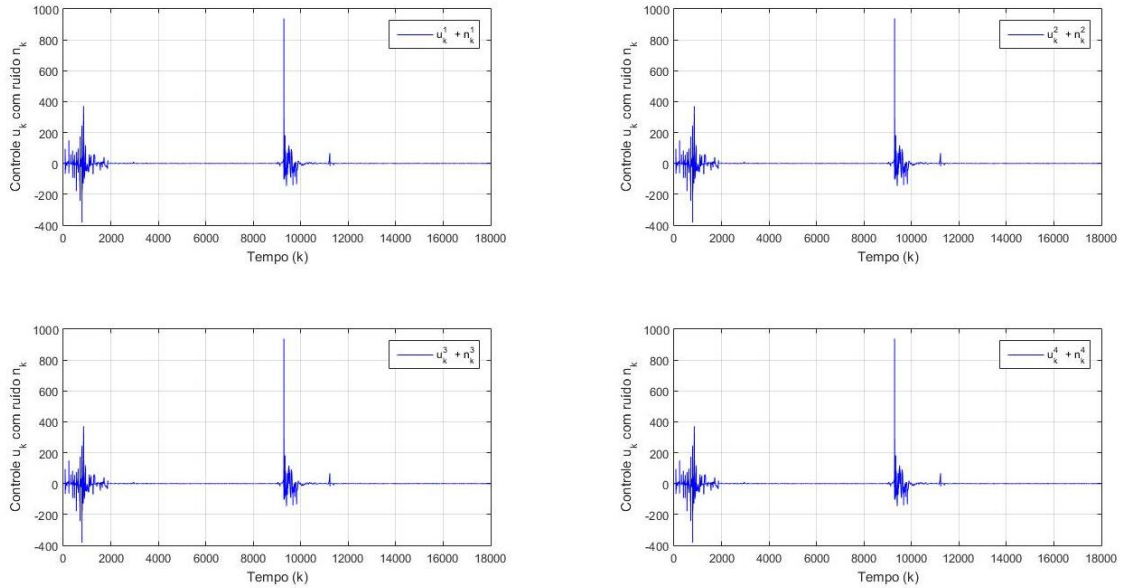
Dessa forma, o comparativo é apresentado na figura 5.29 para a configuração do fator de esquecimento 0.9966 e pesos da matriz de ponderação $qi = -4$ e $ri = -2$.

Figura 5.29: Ação de Controle para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$.



O ponto importante dessa comparação é perceber a região onde há entrada da ação de controle e até a amplitude de cada resposta. Mesmo com diferentes valores de parâmetros, a resposta é similar. Pode-se ver na figura 5.30.

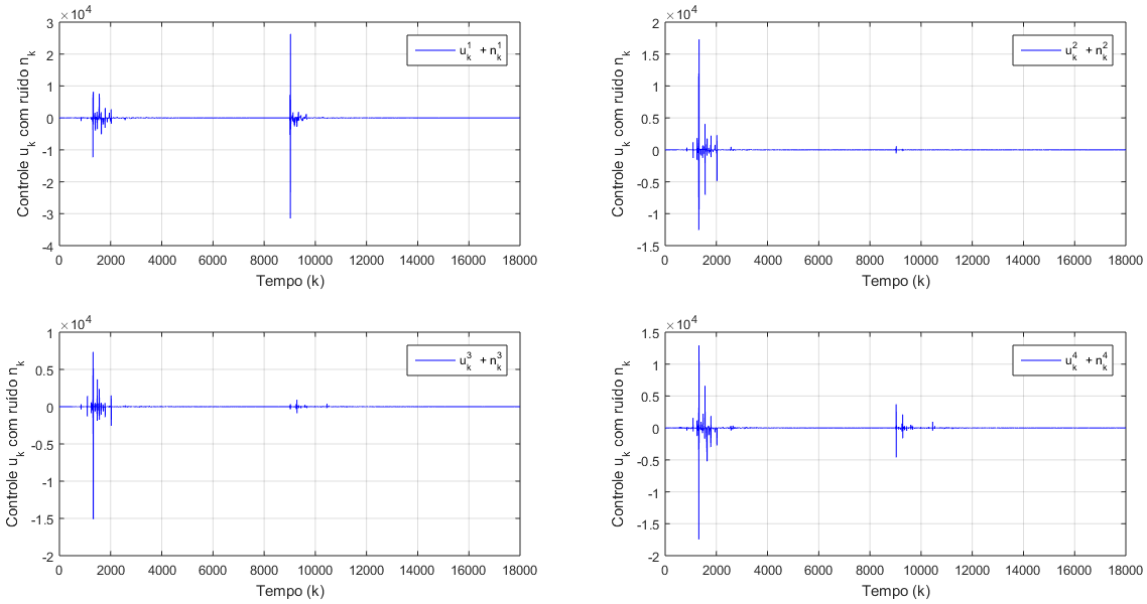
Figura 5.30: Ação de Controle para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$.



Relacionando o comportamento dessa entrada de controle, figura 5.30, pode-se examinar os estados referentes aos pesos da matriz de ponderação e fator de esquecimento, assim como a matriz de ganho apresentados nas figuras 5.26 e 5.22, respectivamente. Para melhor entendimento da resposta gerada aqui.

Por fim, se tem a resposta de controle referente a configuração do fator de esquecimento 0.9946 e pesos da matriz de ponderação $qi = -2$ e $ri = -2$ na figura 5.31.

Figura 5.31: Ação de Controle para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$.

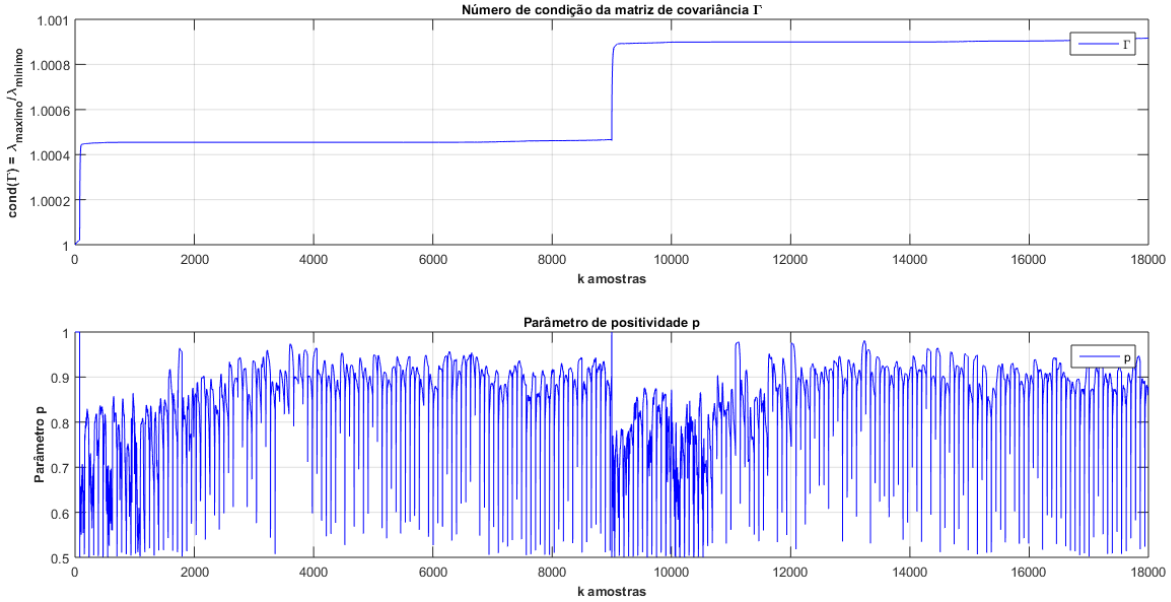


O interessante de simular diferentes valores para as matrizes de ponderação Q e R é devido a obter diferentes comportamentos para melhor escolha do projeto de controlador ótimo. Tal escolha ajuda a minimizar as perdas ou esforços de controle e manter por mais tempo o veículo em voo e com melhores condições de adaptabilidade.

Matriz de Covariância Γ_k e Parâmetro de Positividade p

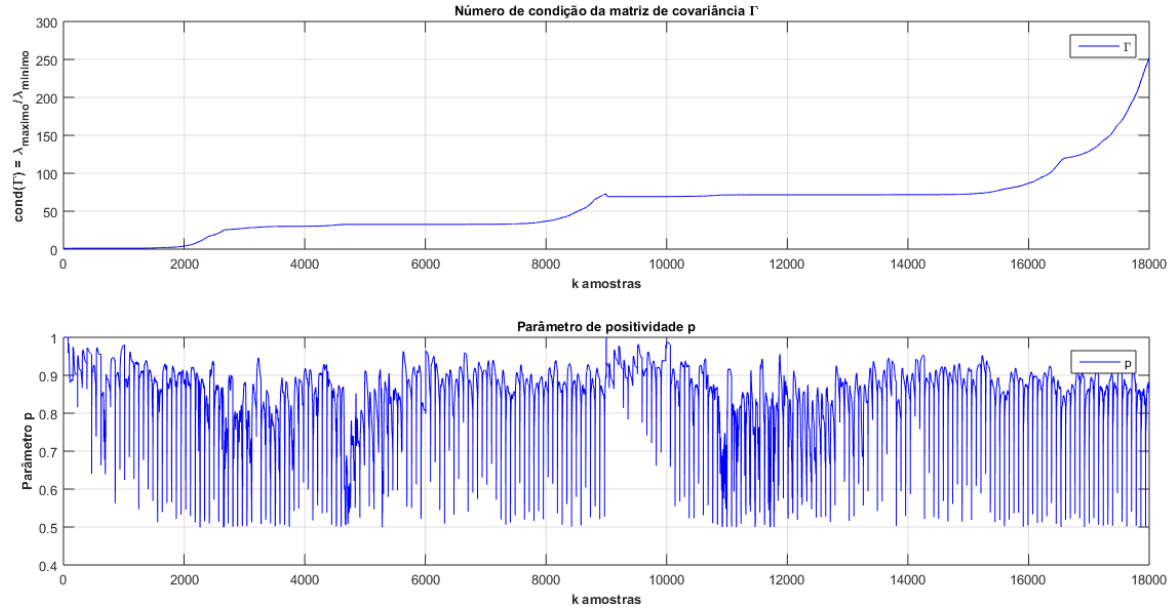
As figuras 5.32-5.35 mostram o comportamento do número de condição da matriz de covariância Γ_k e o parâmetro de positividade p do processo RLS. Na figura 5.32 observa-se que os valores de p se encontram dentro da faixa de referência $0 < p < 1$. Assim, o comportamento de p não mostra irregularidades no comportamento da matriz de covariância.

Figura 5.32: Número de condição da matriz de covariancia $\boldsymbol{\Gamma_k}$ e o parâmetro de positividade p para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$ do método RLS.



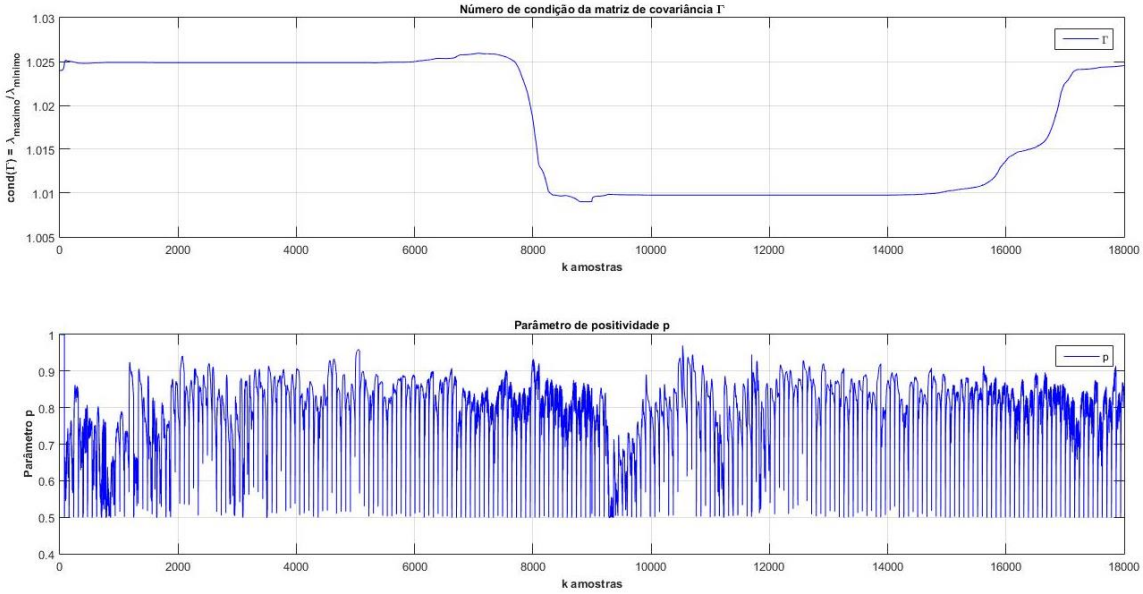
Para a figura 5.33 tem que os valores de p ainda se encontram dentro da faixa de validade $0 < p < 1$. Uma ordem de magnitude mais alta é observada no número de condição da matriz de covariância Γ_k.

Figura 5.33: Número de condição da matriz de covariancia $\boldsymbol{\Gamma_k}$ e o parâmetro de positividade p para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$ do método RLS.



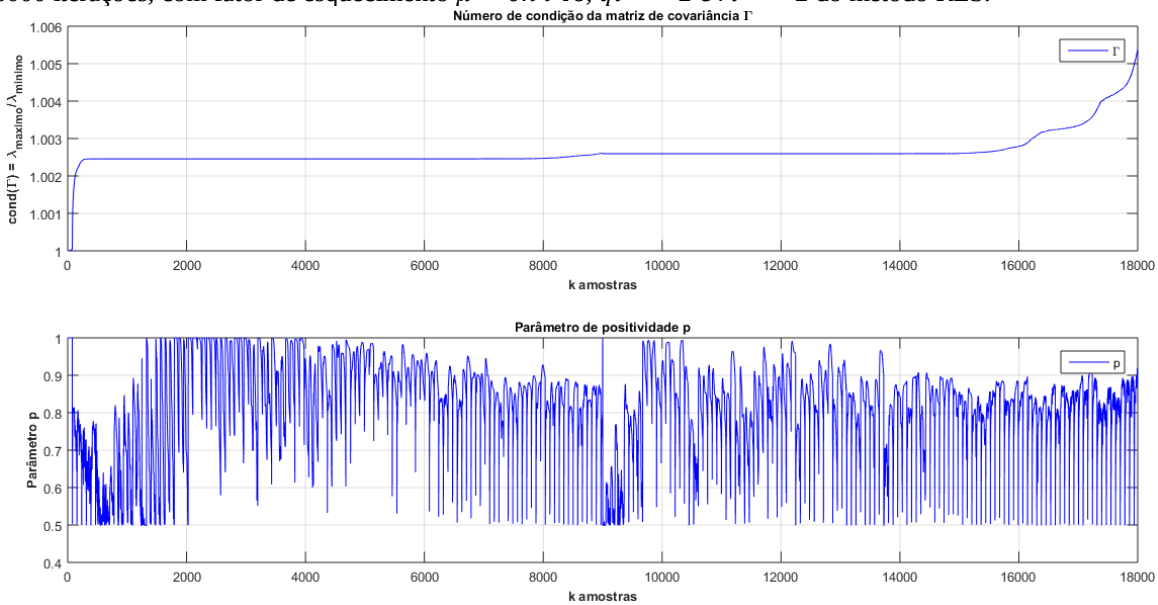
Os valores de p são encontrados dentro da faixa de referência $0 < p < 1$, na figura 5.34, e com a amplitude similar a figura 5.33, mesmo alterando os pesos da matriz de ponderação e o fator de esquecimento.

Figura 5.34: Número de condição da matriz de covariancia $\boldsymbol{\Gamma_k}$ e o parâmetro de positividade p para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$ do método RLS.



Ainda pode-se observar mais uma vez o comportamento do número de condição da matriz de covariância do método RLS e o parâmetro de positividade, os quais ainda são similares aos encontrados nas figuras anteriores, com valores de p encontrados dentro da faixa de referência. Na figura 5.35, utiliza-se o fator de esquecimento 0.9946 e com pesos $qi = -2$ e $ri = -2$ da matriz de ponderação.

Figura 5.35: Número de condição da matriz de covariancia $\boldsymbol{\Gamma_k}$ e o parâmetro de positividade p para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$ do método RLS.



Respostas ao Impulso

A resposta ao sinal impulso do sistema é uma maneira de analisar, ao longo do tempo, o comportamento do sistema dinâmico para o controlador obtido por ADP. Pode-se perceber que para cada sinal de entrada aplicada ao sistema, tem-se a saída correspondente a qual é sua resposta ao impulso baseada em cada configuração adotada, sendo mostrada as magnitudes de cada resposta e seu comportamento até estabilização nas figuras 5.36-5.39. A influência das matrizes de ponderação e dos pesos adotadas nestas matrizes dão características ao tempo de resposta do sistema, ao impulso, ao tempo de estabilização e, consideradas como parâmetros de projeto para fim de redução de controle.

Pode-se comparar os *frames* i, j das respostas abaixo, como em configuração matricial, e verificar o tempo de acomodação de cada resposta, percebendo que os valores dos pesos da matriz de ponderação interfere diretamente na resposta ao impulso.

Figura 5.36: Resposta ao impulso do sistema de malha fechada para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9976$, $qi = -5$ e $ri = -1$.

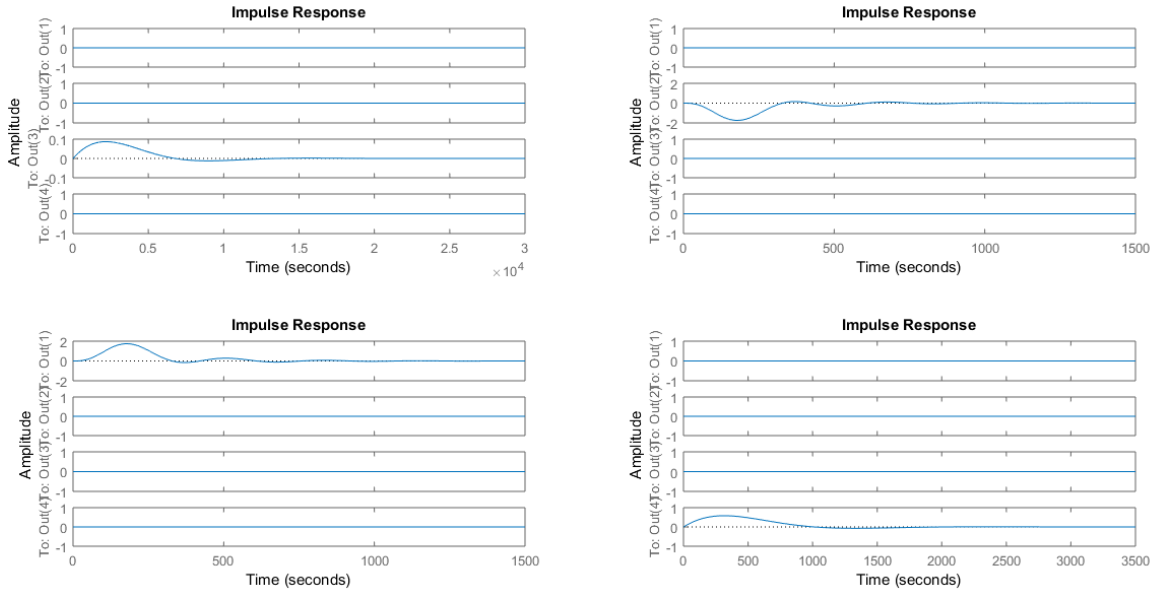


Figura 5.37: Resposta ao impulso do sistema de malha fechada para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9966$, $qi = -4$ e $ri = -2$.

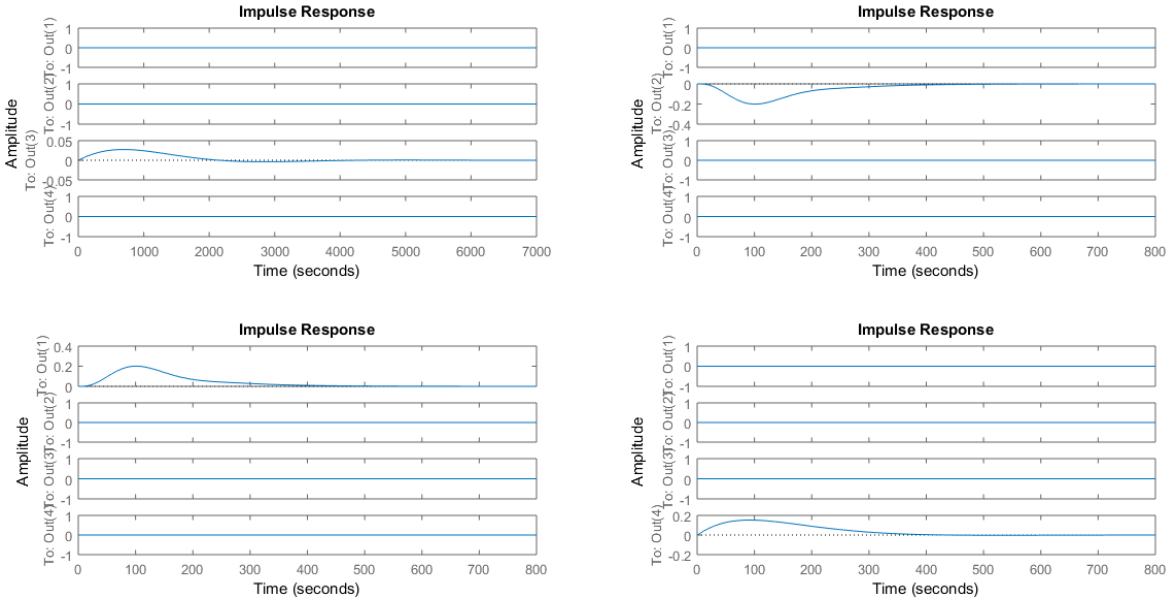


Figura 5.38: Resposta ao impulso do sistema de malha fechada para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -4$ e $ri = -4$.

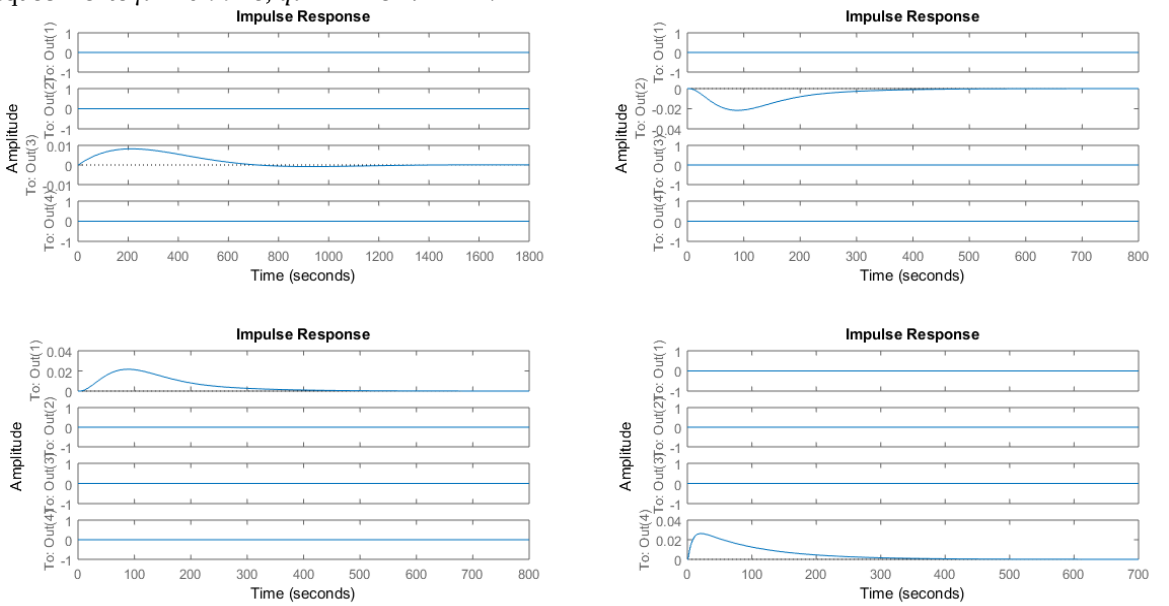
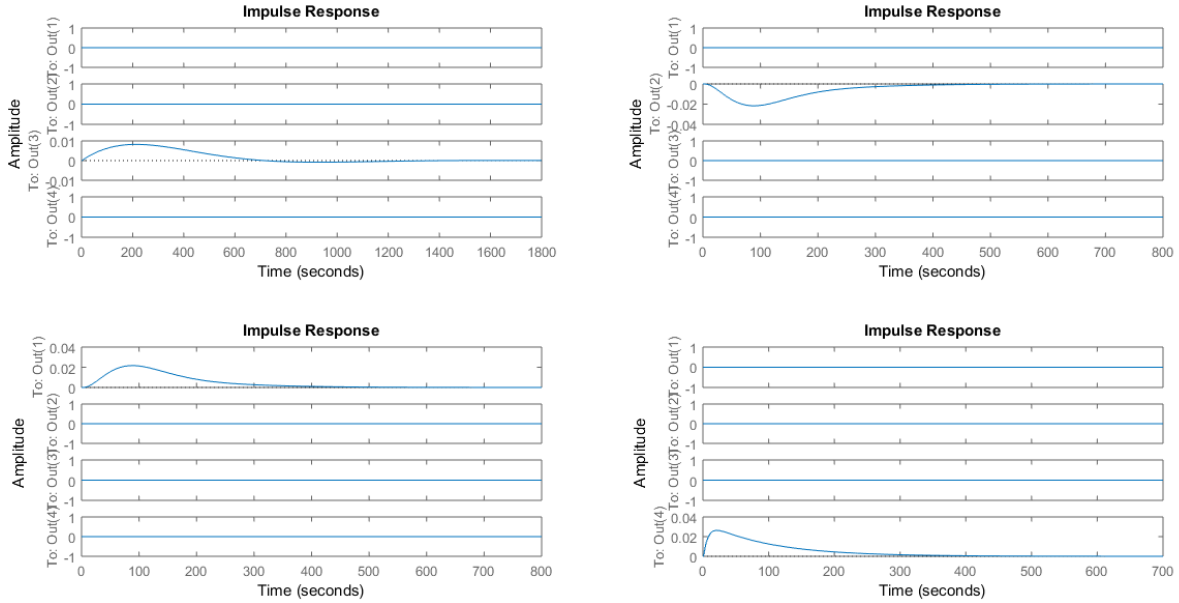


Figura 5.39: Resposta ao impulso do sistema de malha fechada para um ciclo de 18000 iterações, com fator de esquecimento $\mu = 0.9946$, $qi = -2$ e $ri = -2$.



É interessante perceber, após analisar cada elemento dos frames das figuras acima, quanto maior o peso aplicado as matrizes de ponderação, maior será o tempo para a resposta ao impulso estabilizar, ou acomodar-se. Dessa forma, determinar o peso ótimo corresponde a encontrar configurações para nosso objetivo de redução de esforço de controle.

6. CONCLUSÃO

Devido aos requisitos cada vez mais exigente, os sistemas de controle necessitam ser desenvolvidos para garantir melhores estabilidade, precisão e acúmulo de energia para a execução das tarefas pretendidas. Dessa forma, também se dá para o controle (autônomo ou não) do quadricóptero. Tendo em vista, que sua utilidade abrange setores de extensa competência. E ainda determinar o controle ótimo desse veículo, podendo esperar a minimização de erros e alcance dos objetivos com mais precisão.

Este trabalho que por objetivo é um controlador ótimo adaptativo *online* para veículo aéreo não tripulados via programação dinâmica aproximada, a partir de modelos do próprio veículo e até sem o conhecimento do modelo da planta, com base nos métodos propostos de crítico adaptativo procurou determinar a melhoria e/ou redução dos esforços de controle na presença de forças externas ou variações do próprio sistema, e no controle do veículo para resposta em tempo real. Aplicando em situações reais para verificação da rampa de lançamento no Centro de Lançamento de Alcântara ou outro fim de vistoria, conforme padrão antes do lançamento, com propósito de não pôr em risco vidas, não haver colisões com o ambiente e até uso de rastreio dos veículos lançadores, etc.

E em suma, o projeto de controlador ótimo adaptativo para veículo aéreo não tripulado, modelo quadricóptero, via programação dinâmica aproximada tem por ideia potencializar o ganho de controle do voo pairado do veículo, com o fim desde projeto auxiliar no mapeamento da região de lançamento de veículos lançadores de satélites, resgate de equipamentos e salvamento de vidas, redução de custos de operação e desgaste, através de controladores baseados em ADP, os quais mostraram, nos resultados encontrados “em voo”, terem sido adotáveis para a planta do veículo para todos os casos estudados no trabalho, por meio das simulações, com esforço de redução de sintonia adequada.

O objetivo de utilizar diferentes valores para o fator de esquecimento e fatores de peso nas matrizes de ponderação é a verificação do comportamento dos parâmetros do sistema. As ações de controle, a matriz de ganho, a resposta ao impulso, o comportamento dos parâmetros da matriz P de Ricatti e a trajetória dos estados. Dessa forma, se obteve controladores adotáveis na planta para todos os casos desenvolvidos, que aprendem usando dados medidos ao longo das trajetórias do sistema e se estabilizou o sistema mesmo em fase de perturbações externas, ou variações no modelo da planta.

6.1 Trabalhos Futuros

Para o desenvolvimento e direcionamento de futuros trabalhos, como extensão deste propõe-se:

1. Desenvolver um estudo sobre a estabilidade do sistema durante a trajetória de voo do veículo utilizando esquemas críticos adaptativos (DHP e AD-DHP);
2. Analisar o comportamento dos parâmetros de controle do sistema durante o processo de decolagem, alcançado determinada altitude e mantendo-se estável por instantes, e a aterrisagem vertical ao seu ponto de origem;
3. Utilizar outras formas de aproximação de função valor como redes neurais, lógica fuzzy e métodos de Kernel, possibilitando a aplicação dos algoritmos de ADP à dinâmicas não lineares do quadricóptero;
4. Estender os métodos desenvolvidos neste trabalho para processos dinâmicos parcialmente observáveis, nos quais nem todos os estados podem ser observados diretamente da planta, e testar a metodologia no sistema quadricóptero;
5. Implementar em veículos aéreos não – tripulados plataformas para desenvolver algoritmos de controle via programação dinâmica aproximada e estudos de controle embarcados.

APÊNDICE A

Parâmetros de Simulação

Aqui descreve-se os valores dos parâmetros utilizado na simulação do projeto de controle e as matrizes discretizadas da planta do sistema.

Tabela A.1: Descrição dos valores da etapa de simulação.

m	0.8	*Massa (**Kg**)*
l	*0.3*	*Distância do centro do quadricóptero ao rotor (**m**)*
d	$7*10^{-5}$	*Fator de arrasto (**N m s²**)*
b	$2*10^{-4}$	*Fator de empuxo (**N s²**)*
I_x	$5.17*10^{-5}$	*Momento de Inércia no eixo* x (**mr²**)
I_y	$5.17*10^{-3}$	*Momento de Inércia no eixo* y (**mr²**)
I_z	$1.17*10^{-2}$	*Momento de Inércia no eixo* x (**mr²**)
g	*9.81*	*Força da gravidade (**m/s²**)*
t_{samp}	*0.01*	*Tempo de Amostragem (**s**)*

Tabela A.2: Matriz *A* Discretizada

$A_d =$

1	0	0	0	0,0004905	0	0,01	0	0	0	1,64E-06	0
0	1	0	-0,000491	0	0	0	0,01	0	-1,64E-06	0	0
0	0	1	0	0	0	0	0	0,01	0	0	0
0	0	0	1	0	0	0	0	0	0,01	0	0
0	0	0	0	1	0	0	0	0	0	0,01	0
0	0	0	0	0	1	0	0	0	0	0	0,01
0	0	0	0	0,0981	0	1	0	0	0	0,000491	0
0	0	0	-0,0981	0	0	0	1	0	-0,0004905	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1

Tabela A.3: Matriz *B* Discretizada

$$\boldsymbol{B_d} = \begin{bmatrix} 0 & 0 & 7{,}91\text{E-}07 & 0 \\ 0 & -7{,}91\text{E-}07 & 0 & 0 \\ 6{,}25\text{E-}05 & 0 & 0 & 0 \\ 0 & 0{,}00967118 & 0 & 0 \\ 0 & 0 & 0{,}009671 & 0 \\ 0 & 0 & 0 & 0{,}002941 \\ 0 & 0 & 0{,}000316 & 0 \\ 0 & -0{,}00031625 & 0 & 0 \\ 0{,}0125 & 0 & 0 & 0 \\ 0 & 1{,}93423598 & 0 & 0 \\ 0 & 0 & 1{,}934236 & 0 \\ 0 & 0 & 0 & 0{,}588235 \end{bmatrix}$$

REFERÊNCIAS BIBLIOGRÁFICAS

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

BERTSEKAS, D. P. **Dynamic Programming and Optimal Control**. 4th. ed. Belmont, Massachusetts: Athena Scientific, 2012.

BHUVANESWARI, N. S.; UMA, G.; RANGASWAMY, T. R. **Adaptive and optimal control of a non-linear process using intelligent controllers.** [S.l.]: Applied Soft Computing, v. 9, 2009.

BRYSON, A. E.; HO, Y.-C. **Applied Optimal Control**. UK: Taylor and Francis, 1975.

COZA, C.; MACNAB, C. J. B. **A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization**. [S.l.]: [s.n.]. 2006.

DE CASTRO JORGE, L. A.; INAMASU, R. Y. **Uso de Veículo aereos não tripulados (Vant)**. [S.l.]: [s.n.], 2014.

DIKMEN, I. C.; ARISOY, A.; TEMELTAS, H. **ttitude control of a quadrotor, Recent Advances in Space Technologies**. 4th International Conference on, IEEE. [S.l.]: [s.n.]. 2009. p. 722-727.

FERRARI, S.; STENGEL, R. Online adaptive critic flight control. **Journal of Guidance**, v. 27, n. 5, p. 777-7786, Setembro-Outubro 2004.

FONSECA, J.; FERREIRA, E.; REGO, P. H. M. Online optimal dlqr-dfig control system design via recursive least-square approach and state heuristic dynamic programming for approximate solution of the hjb equation. **IEEE International Conference on**, p. 3174–3179, Oct 2013.

FRADKOV, A.; ANDRIEVSKY, B.; PEAUCELLE, D. **Adaptive control experiments for laas "helicopter" benchmark**. [S.l.]: [s.n.]. 2005. p. 760-765.

GIBIANSKY, A. **Quadcopter Dynamics, Simulation, and Control.** [S.l.]: [s.n.], 2014.

HAMEL, T.; GUENARD, N.; MAHONY, R. **A practical visual servo control for a unmanned aerial vehicle**. IEEE International Conference on Robotics and Automation. [S.l.]: [s.n.]. 2007. p. 1342 – 1348.

HAYKIN, S. S. Adaptive filter theory. **Pearson Education India**, India, 2008.

HOFFMAN, G. et al. **Quadrotor helicopter flight dynamics and control:** Theory and experiment. Proc. of the AIAA Guidance, Navigation, and Control Conference, pp. 1-20. [S.l.]: [s.n.]. 2007. p. 1-20.

HWANGBO, J.; HUTTER, M. Control of a Quadrotor with Reinforcement Learning. **IEEE ROBOTICS AND AUTOMATION LETTERS**, 2017.

JANG, J. S. et al. **Multi-agent quadrotor testbed control design:** Integral sliding mode vs. reinforcement learning. [S.l.]: [s.n.]. 2006.

KAR, I.; BEHERA, L. Direct adaptive neural control for affine nonlinear systems. . **Applied Soft Computing**, v. 9, p. 756-764, 2009.

KHAN, S.; LEWIS, F. Reinforcement learning and optimal adaptive control: An overview and implementation examples. **Annual Reviews in Control**, 2012.

KIRK, D. E. **Optimal Control Theory:** An Introduction. 1ª. ed. Mineola, New York: Dover Publications, Inc, v. Único, 2004.

LENDARIS, G. G. A retrospective on adaptive dynamic programming for control. **Neural Networks, IJCNN , International Joint Conference on**, Atlanta, Georgia, USA, june 2009. pp. 1750 –1757.

LEVINE, S.; ZHANG, T. **Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search.** IEEE International Conference on Robotics and Automation. [S.l.]: [s.n.]. 2016. p. 528-535.

LEWIS, F. L.; LIU, D. **Reinforcement learning and approximate dynamic programming for feedback control**. [S.l.]: John Wiley & Sons, 2012.

LEWIS, F. L.; VRABIE D. Adaptive dynamic programming for feedback control, 2009. 32-50.

LEWIS, F. L.; VRABIE, D.; SYRMOS, V. L. **Optimal Control**. 3nd. ed. USA: John Wiley and Sons, Inc., 2012.

LIU, D.; WEI, Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. **IEEE Transactions on Neural Networks and Learning Systems**, v. 25, n. 3, p. 621–634, 2014.

LOZANO, R.; CASTILLO, P.; DZUL, A. **Stabilization of a mini rotorcraft with four rotors**. IEEE Control Systems Magazine. [S.l.]: [s.n.]. 2005. p. 45-55.

MOREL, Y.; LEONESSA, A. **Direct adaptive tracking control of quadrotor aerial vehicles**. Florida Conference on Recent Advances in Robotics. Flórida : [s.n.]. 2006. p. 1-6.

MURRIERI, P.; BOUABDALLAH, S.; SIEGWART., R. Design and control of an indoor micro quadrotor. **IEEE International Conference on Robotics and Automation**, 2004.

NOTH, A.; BOUABDALLAH, S.; SIEGWART, R. Pid vs lq control techniques applied, 2005.

QUAN, Q. **Introduction to Multicopter Design and Control.** [S.l.]: [s.n.], 2017.

RAWASHDEH, O. et al. **Microraptor:** A low-cost autonomous quadrotor system. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. California, USA: Proceedings of the ASME. 2009.

RÊGO, P. H. M. **Tese de doutorado:** Aprendizagem por reforço e programação dinâmica aproximada para controle ótimo: uma abordagem para o projeto online do regulador linear. UFMA, MA.: [s.n.], 2014.

SI, J. **Handbook of learning and approximate dynamic programming**. [S.l.]: John Wiley & Sons, v. 2, 2004.

SOUSA, G. B. D. **Convergência e Estabilidade Numérica de Algoritmos de Programação Dinâmica Heurística Dependente de Ação Baseados na Abordagem RLS para Controle DLQR Online**. 1ª. ed. São Luís, MA: Universidade Estadual do Maranhão, v. Único, 2018.

STINGU, E.; LEWIS, F. L. **An Approximate Dynamic Programming Based Controller for an Underactuated 6DoF Quadrotor**. [S.l.]: [s.n.], 2011.

TARBOUCHI, M.; DUNFIED, J.; LABONTE, G. **Neural network based control of a four rotor helicopter**. IEEE Intrnational Conference on Industrial Technology. [S.l.]: [s.n.]. 2004. p. 1543-1548.

TEIXEIRA, H. T. **Aprendizado por reforço e programação dinâmica aproximada com máquinas kernel para controle de sistemas não lineares**. Campinas, SP.: [s.n.], 2016.

TOMMASO, B. **Tese de doutorado:** Modelling, Identification and Control of a Quadrotor Helicopter. Lund University: : Department of Automatic Control., 2008.

VALAVANIS, K.; VACHTSEVANOS, G. **Handbook of Unmanned Aerial Vehicles**. 1ª. ed. [S.l.]: Springer Reference, v. I, 2015.

VARGAS, F. J. T.; PAGLIONE, P. **Ferramentas de Algebra Computacional:** aplicações em modelagem, simulação e controle para engenharia.. 1. ed. ed. RJ: LTC, 2015.

WANG, F. Y.; ZHANG, H.; LIU, D. Adaptive dynamic programming: An introduction. **IEEE Computer Intelligence Magazine**, v. 4, n. 2, p. 39–47, May 2009.

WANG, F.-Y.; ZHANG, H.; LIU, D. Adaptive dynamic programming: An introduction. **IEEE Computational Intelligence Magazine**, v. v.4, n. n.2, 2009.

WEBER, E. M.; MAYER, C. A. N. M. Reinforcement Learning, Theory and Applications. **I-TECH Education and Publishing**, 2008.

WEI, Q.; LIU, D. . S. G. . L. Y. Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming. **IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS**, v. 62, n. 7, July 2015.

WERBOS, P. J. A menu of designs for reinforcement learning over time. **Neural networks for control**, 1990. 67–95.

WERBOS, P. J. Approximate dynamic programming for real-time control and neural modeling. **Handbook of intelligent control: Neural, fuzzy, and adaptive approaches**, p. 493-525, 1992.

WERBOS, P. J. **Reinforcement learning and approximate dynamic programming (rladp) |foundations, common misconceptions, and the challenges ahead, Reinforcement Learning and Approximate Dynamic Programming for Feedback Control**. [S.l.]: [s.n.], 2012.

XU, X.; HE, H.; HU, D. **Efficient reinforcement learning using recursive least-squares**. Journal of Artificial Intelligence Research. [S.l.]: [s.n.]. 2002. p. 259–292.

YEH, J.-W.; SU, S.-F. **Learning analysis for correlation of fuzzy rules in applying rls for neural**. 2012 IEEE International Conference. [S.l.]: IEEE. Granular Computing (GrC). 2012. p. 609–613.

YEH, J.-W.; SU, S.-F.; RUDAS, I. **Analysis of using rls in neural fuzzy systems**. IEEE. Systems, Man, and Cybernetics (SMC). [S.l.]: IEEE International Conference on 2011. 2011. p. 1831–1836.

ZHANG, T. et al. **Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search.** IEEE International Conference on Robotics and Automation (ICRA). [S.l.]: [s.n.]. 2016.