



Universidade Estadual de Montes Claros

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL E SISTEMAS

Reconfiguração de Sistemas de Controle de um Drone (Quadrimotor)

Dissertação submetida à banca avaliadora designada pelo Colegiado do Programa de Pós-Graduação em Modelagem Computacional e Sistemas da Universidade Estadual de Montes Claros, como parte dos requisitos exigidos para a obtenção do título de Mestre em Modelagem Computacional e Sistemas.

por

Rodrigo Baleeiro Silva

Orientador: Prof. Dr. Renato Dourado Maia

Agosto – 2017

Universidade Estadual de Montes Claros
Centro de Ciências Exatas e Tecnológicas
Programa de Pós-Graduação em Modelagem Computacional e Sistemas

Reconfiguração de Sistemas de Controle de um Drone (Quadrimotor)

Autor: Rodrigo Baleeiro Silva

Orientador: Prof. Dr. Renato Dourado Maia

Dissertação submetida à banca avaliadora, designada pelo Colegiado do Programa de Pós-Graduação em Modelagem Computacional e Sistemas da Universidade Estadual de Montes Claros, como parte dos requisitos exigidos para a obtenção do título de Mestre em Modelagem Computacional e Sistemas.

Banca Examinadora

Maurílio José Inácio, Dr.	DCC-UNIMONTES
Nilton Alves Maia, Dr.	DCC-UNIMONTES
Renato Dourado Maia, Dr. - Orientador	DCC-UNIMONTES

Montes Claros, MG

Agosto - 2017

S586r Silva, Rodrigo Baleeiro.
Reconfiguração de sistemas de controle de um Drone (Quadrimotor)
[manuscrito] / Rodrigo Baleeiro Silva. – 2017.
56 f. : il.

Bibliografia: f. 54-56.

Dissertação (mestrado) - Universidade Estadual de Montes Claros - Unimontes, Programa de Pós-Graduação em Modelagem Computacional e Sistemas/PPGMCS, 2017.

Orientador: Prof. Dr. Renato Dourado Maia.

1. Drone quadrimotor. 2. Estimação de falhas. 3. Reconfiguração do sistema de controle. I. Maia, Renato Dourado. II. Universidade Estadual de Montes Claros. III. Título.



RELATÓRIO DE DEFESA DE DISSERTAÇÃO

MESTRADO

Universidade Estadual de Montes Claros - Unimontes
 Pró-Reitoria de Ensino/Coordenadoria de Pós-Graduação
 Programa de Pós-Graduação em Modelagem Computacional e Sistemas
 - PPGMCS

1 - Identificação do Aluno

Nome: Rodrigo Baleeiro Silva

Matrícula: 010004334

Linha de Pesquisa: Inteligência Computacional, Otimização e suas Aplicações.

2 - Sessão de Qualificação

Título: "RECONFIGURAÇÃO DE SISTEMAS DE CONTROLE DE UM DRONE (QUADRIMOTOR)"

3 - Comissão Examinadora

Nome	Função	Assinatura
Prof. Dr. Renato Dourado Maia	Orientador (a)	<i>Renato Dourado Maia</i>
Prof. Dr. Nilton Alves Maia	Examinador(a)	<i>Nilton Alves Maia</i>
Prof. Dr. Maurílio José Inácio	Examinador(a)	<i>Maurílio José Inácio</i>

4 - Resultado

A comissão Examinadora, em 03/08/2017 após Defesa de Dissertação e arguição do(a) candidato(a), decidiu:

- pela aprovação da Dissertação
 pela reprovação da Dissertação
 pela revisão de forma, indicando o prazo de 30 dias para apresentação definitiva.
 pela reformulação da Dissertação, indicando o prazo de 45 dias para nova versão.

Preencher somente em caso de revisão de forma:

- O(a) aluno(a) apresentou a revisão de forma e a Dissertação foi aprovada.
 O(a) aluno(a) apresentou a revisão de forma e a Dissertação foi reprovada.
 O(a) aluno(a) não apresentou a revisão da forma.

Preencher somente em caso de revisão de reformulação:

- O(a) aluno(a) apresentou a reformulação e a Dissertação foi aprovada.
 O(a) aluno(a) apresentou a reformulação e a Dissertação foi reprovada.
 O(a) aluno(a) não apresentou a reformulação.

Autenticação

Orientador(a) Comissão Examinadora

03/08/2017 *Renato Dourado Maia*
 Data Assinatura

Autenticação

Coordenador

03/08/2017 *Nilton Alves Maia*
 Data Assinatura
 Prof. Nilton Alves Maia
 Coordenador do PPGMCS
 UNIMONTES
 Matr. 10406574

Resumo

As tarefas de estimar falhas e reconfiguração de controle em processos ou sistemas são uma área de pesquisa em crescente desenvolvimento e essa expansão pode ser associada á sua característica preventiva que, associada a um controle, previne perdas tanto materiais quanto humanas. Nesse contexto, este trabalho apresenta uma proposta de uma abordagem adaptativa para estimação de falhas e reconfiguração ativa de controle em Drone, baseada em duas etapas: na primeira etapa, a abordagem proposta utiliza métodos clássicos para estimar as falhas, na segunda etapa, propõem-se variações na reconfiguração de controle a partir de dados de estimação da falha. Para verificar a eficiência da abordagem, a reconfiguração de controle foi aplicada em um modelo de drone quadrimotor com falha em um dos motores, contextualizada como a queda de potência simulada nesse motor. Dessa forma, os resultados obtidos pelo sistema de reconfiguração são comparados com controladores utilizados apenas em operação normal, com uma ênfase na variável altura. De modo geral, a abordagem proposta obteve bons resultados, o que sugere que o sistema de reconfiguração é promissor para realizar as tarefas de estimação de falhas e reconfiguração de sistemas de controle de Quadrimotores.

Palavras-chave: Drone Quadrimotor, Estimação de falhas, Reconfiguração do sistema de controle.

Abstract

The tasks of estimate failures and reconfigurations of control in processes or systems are part of an area of growing research. This expansion can be associated with its preventive characteristic, that when associated with robust control, prevents losses in both materials and labor. In this context, this study proposes a approach adapted to the estimation of failures and the reconfiguration of active control of dynamic systems, based in two stages: in the first stage, we propose to use classic methods to estimate failures, in the second stage, we propose variations in the reconfiguration of dynamic systems based on data related to identifying the failure. In order to verify the efficiency of this approach, we applied this to a drone model with four engines with a failure in one of the motors, contextualized within the resultant loss of overall force. In this way, the results obtained from the system of reconfiguration are compared with controls that are used only under normal operating circumstances, with an emphasis on the variable of height. Generally, the proposed approach was more effective in addressing failures, suggesting that realizing tasks to estimate failures and reconfigure control in Quadrimotor is a promising field of study.

Keywords: Drone Quadrimotor, Estimation of Failures, Reconfiguration of Systems of Control.

Agradecimentos

A Deus, pois sempre concedeu vida, saúde, força e fé em todos os momentos de minha vida.

Ao meu orientador, Prof. Renato Dourado Maia, pelos ensinamentos, pelo exemplo de pesquisador e pela amizade.

Aos meus ex-alunos e amigos Daniel Neri e José Maia Neto pela contribuição neste trabalho.

Aos demais professores do PPGMCS, pelos ensinamentos e dedicação ao ministrar as disciplinas, que sem dúvida foram muito importantes para a realização dessa pesquisa.

Aos meus pais e irmãos que, de longe, sempre estiveram torcendo pelo meu sucesso.

Aos colegas do PPGMCS, pela amizade e apoio nesses quase dois anos de estudo.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Acrônimos	x
1 Introdução	1
1.1 Caracterização do Problema e Motivação	1
1.2 Objetivos	3
1.3 Metodologia	3
1.4 Artigos Publicados em Anais de Eventos	4
1.5 Trabalhos em Periódicos	5
1.6 Organização do Trabalho	5
2 Trabalhos Relacionados	6
2.1 Trabalhos com Drone de quatro motores e controlador PID	6
2.2 Trabalhos com Drone de quatro motores e controlador LQR	6
3 Revisão de Literatura	9
3.1 Fundamentos Importantes	9
3.1.1 Falhas	9
3.1.2 Detecção e Isolamento de Falhas	11
3.1.3 As Redes Neurais Artificiais	12
3.1.4 Classificação de Falhas	14
3.1.5 Reconfiguração de Controle	15
4 Abordagem proposta	18
4.1 O Quadrimotor	18
4.1.1 Modelagem do Quadrimotor Baseada em Newton-Euler	18
4.1.2 Dinâmica e Cinemática do Quadrimotor	24

4.1.3	Forças e Torques	29
4.1.4	Linearização do Sistema e Descrição no Espaço de Estados	32
4.2	O controlador Regulador Linear Quadrático - LQR	35
4.3	Projeto do controlador	36
4.3.1	Verificação de Controlabilidade e Observabilidade	36
4.3.2	Projeto do Controlador LQR	37
5	Resultados	42
5.1	Resultados do Estimador de Falhas	42
5.2	Resultados do Sistema de Reconfiguração de Controle	45
5.3	Resultados Numéricos do Sistema de Reconfiguração de Controle	49
6	Considerações Finais	52
	Referências Bibliográficas	54

Lista de Figuras

1.1	Esquemático do Sistema de Reconfiguração de Controle	4
2.1	Esquema de reconfiguração de controle com LQR e lógica quântica	7
2.2	Esquema de reconfiguração de controle com LQR	7
3.1	Funcionamento da detecção de falhas em um processo	10
3.2	Rede PMC com camadas escondidas	13
3.3	Princípio do Controle com SRCS	16
4.1	Movimento de Atitude do Quadrimotor	19
4.2	<i>Frame</i>	19
4.3	Rotação entre <i>frames</i> - 2D	20
4.4	Fixação de <i>Frame</i> em objetos	22
4.5	Fixação de <i>Frame</i> em objetos	22
4.6	Bases para o Momento de Inércia	27
4.7	Bases para o Momento de Inércia	29
4.8	Bases para o Momento de Inércia	30
4.9	Representação do Controlador no <i>Simulink</i>	40
4.10	Resposta da altura antes do projeto do controlador LQR	40
4.11	Resposta da altura após o projeto do controlador LQR	41
5.1	Simulação de falha	42
5.2	Configuração da rede mlp	43
5.3	Operação do estimador de falhas	44
5.4	Operação do estimador de falhas	45
5.5	Princípio do Controle proposto	47
5.6	Funções de pertinência para chaveamento de controlador por lógica <i>Fuzzy</i>	48
5.7	Operação do sistema de reconfiguração de controle	48
5.8	Comparativo entre o sistema normal e a proposta de reconfiguração	49

Lista de Tabelas

4.1	Parâmetros do Quadrimotor Real	34
4.2	Tabela de parâmetros dos estados	39
4.3	Tabela de parâmetros de entradas	39
5.1	Relação de tempo x falha	43
5.2	Controladores ativos	46
5.3	Cenário de falhas	50
5.4	Tabela de índice de desempenho dos sistemas de controle	51

Acrônimos

1. **FDI** – *Fault Detection and Isolation.*
2. **VANT** – *Veículo Aéreo Não Tripulado.*
3. **LQR** – *Linear Quadratic Regulator .*
4. **LQG** – *Linear Quadratic Gaussian Regulator.*
5. **CBM**– *Condition-Based Maintenance.*
6. **PHM** – *Prognostic and Health Managent.*
7. **PMC** – *Perceptron Multicamadas.*
8. **LM** – *Levenberg-Marquardt.*
9. **SRCS** – *Self-Repairing Control System.*
10. **ISE** – *Integral of the Square of the Error.*
11. **IAE** – *Integral of the Absolute Magnitude of the Error.*
12. **ITAE** – *Integral of Time Multiplied by Absolute of the Error.*
13. **ITSE** – *Integral of Time Multiplied by the Squared Error.*

Capítulo 1

Introdução

1.1 Caracterização do Problema e Motivação

O surgimento de novos processos cada vez mais complexos impõe grandes desafios para o seu gerenciamento e manutenção, pois toda essa complexidade também é descrita em termos de projeto, execução e demais passos tomados. Trabalhos como o de Ahmed-Zaid et al. (1991) demonstraram o uso de um sistema adaptativo para diagnosticar falhas em um caça F16 com um controle quadrático híbrido permitiu melhor desempenho do sistema de controle. Hajiyev e Caliskan (2013) inseriram um *feedback* em controladores LQR e LQG para minimizar as perturbações externas e aumentar a robustez, contornando um modelo preditivo. Exemplifica-se, assim, que, no controle de processos de voo, no qual o controle, detecção de falhas provenientes de erros físicos ou de *software* são feitos apenas pelo operador humano, a probabilidade de acidentes é alta, pois mesmo que a capacidade humana seja de alto nível, a sua precisão é falha, e processos repetitivos tornam essa característica ainda mais evidente.

O problema de diagnóstico de falha e reconfiguração de controle de sistemas é um ramo em desenvolvimento contínuo na engenharia, e sua importância torna-se cada vez maior devido à necessidade de sistemas de navegação cada vez mais sofisticados como em controles de veículos aeroespaciais (Silva et al., 2016). Sistemas de reconfiguração de controle como é tratado neste trabalho, em sua maioria, são conceituados como sistemas especialistas; sua aplicação pode ser constatada nos projetos de Reconfiguração em Tempo Real inseridos nos sistemas da *National Aeronautics and Space Administration* (NASA)(Souza e Silva, 2016).

Antes da reconfiguração há um aspecto importante que é o diagnóstico da falha. Tais concepções são comprovadas, nos casos de processos com alto grau de complexidade, pois

torna-se difícil para operadores humanos diagnosticar uma situação anormal em tempo efetivo, ocasionando tomadas de decisão erradas. Estudos estatísticos na área industrial mostram que aproximadamente setenta por cento dos acidentes são causados por erro humano, que acarreta não só perda econômica, mas diminuição de segurança e danos ambientais (Venkatasubramanian et al., 2003b). Assim, os sistemas de estimação de falhas e controladores robustos são fundamentais no conceito de Autorreparação de Controle de Voo em Tempo Real, o *Self-Repairing Real-time Flight Control Systems*, ou SRFC (Hajiyev e Caliskan, 2013).

O SRFC é um conceito baseado em detectar, isolar e recuperar falhas ou elementos em sistemas de voo. Dessa forma, os controles que utilizam o SRFC operam através de monitoramento, detecção, diagnóstico e reconfiguração (Rubertus, 1983). O SRFC se divide em duas partes distintas: a detecção e isolamento de falha (FDI) e a reconfiguração das leis de controle. O diagnóstico de falhas são uma tarefa importante no sistema, pois se as falhas não forem listadas corretamente podem ocasionar graves consequências na reconfiguração de controle. Durante as últimas décadas, foram propostas diversas técnicas de detecção de falhas, com métodos que partem dos clássicos baseados em modelos, até os mais modernos, baseados em dados do processo com emprego de inteligência computacional.

Assim, projetos que fazem uso da tecnologia de reconfiguração de controle com Drones são apresentados em Suresh e Kannan (2008) com um sistema neural funcionando com um controlador adaptativo para Drone *aircraft*, em Sharifi et al. (2010) utiliza-se um estimador de estados, baseando em observadores, para reconfigurar o sistema de controle do Quadrimotor e em Qian et al. (2016) um sistema *Fuzzy* em paralelo ao controlador padrão é utilizado para compensar possíveis falhas em atuadores. Além disso, outros trabalhos foram desenvolvimentos como em Efe (2011), que implementou um controle PID auxiliado de uma rede neural artificial (RNA) e contribui para a minimização dos efeitos causados pelos distúrbios. Já no trabalho de Neri (2016), utilizou-se a RNA para realizar chaveamentos entres os controladores H_2 e H_∞ em VANT (Veículo Aéreo Não tripulado) til rotor de acordo com limiares de velocidades. Apesar disso, os trabalhos baseados na reconfiguração de controle em VANT Quadrimotor são mais escassos, listando Chen et al. (2015), que faz uso de lógica quântica para produzir a reconfiguração de controle.

O presente trabalho, portanto, reconfigura o sistema de controle de um Drone Quadrimotor quando o VANT for submetido a uma falha em um dos motores. O sistema de reconfiguração é composto de uma RNA para estimar valores de falhas em um dos motores e baseando no valor da falha e nas funções de pertinências, um sistema *Fuzzy* gerencia uma combinação de controlador LQR tolerante ao nível da falha.

1.2 Objetivos

Este trabalho tem como objetivo principal propor uma abordagem para estimar falhas e reconfigurar sistema de controle de Drone. Para tanto, a abordagem proposta reconfigura o sistema de controle de forma adaptativa de um Drone Quadrimotor quando submetido a falhas em um dos motores.

Este trabalho está fundamentado nos seguintes objetivos específicos:

- Estudar e conhecer as metodologias utilizadas para a detecção e estimação de falhas existentes na literatura.
- Evidenciar o índice de desempenho gerado pela reconfiguração do sistema de controle para amenização do voo após possíveis falhas em um dos motores em um Drone(Quadrimotor) em relação ao sistema de controle padrão.
- Avaliar a metodologia proposta por meio de uma comparação de resultados gráficos entre o sistema de controle padrão e o sistema de reconfiguração proposto, com uma ênfase na variável altura antes e após a inserção de falha em um dos motores do Drone.

1.3 Metodologia

O desenvolvimento do projeto consiste em uma pesquisa quantitativa, pois se trata de um experimento desenvolvido no ambiente de simulação com a utilização de dados de um sistema real. Para alcançar os objetivos propostos nesta pesquisa, o trabalho realizado foi dividido em cinco etapas fundamentais que são descritas neste capítulo. A primeira parte aborda trabalhos relacionados com o tema do projeto proposto. Essa etapa é importante por proporcionar o acompanhamento do estado da arte sobre pesquisas relacionadas ao tema escolhido e será desenvolvido com mais intensidade no início do trabalho, ocorrendo, porém, durante quase todo o período de desenvolvimento da pesquisa. Já segunda parte do trabalho consiste em uma revisão bibliográfica de fontes de informações relacionadas à modelagem, estimação e classificação de falhas e sistema de controle para o quadrimotor. A terceira parte do trabalho trata-se da abordagem proposta com o entendimento da modelagem matemática do Drone Quadrimotor baseado em dados reais. Tal modelagem apresentada por equações matemáticas representará a planta conforme a figura 1.1, e permitirá a elaboração do sistema de controle do Quadrimotor, além disso, viabilizará a simulação de todo o sistema em operação normal e com falhas nos atuadores. Na quarta etapa apresenta-se os resultados, após a modelagem matemática do Quadrimotor e o projeto do sistema de estimação de falhas, baseando-se na entrada erro, entre os *setpoints* e as variáveis de saída (Altura, posição norte e posição leste), e a reconfiguração baseando na operação do sistemas de estimação e lógica

fuzzy para gerar a saída de controle (u), apresentada na figura 1.1. Já na ultima etapa é discutido todo o trabalho com as considerações finais.

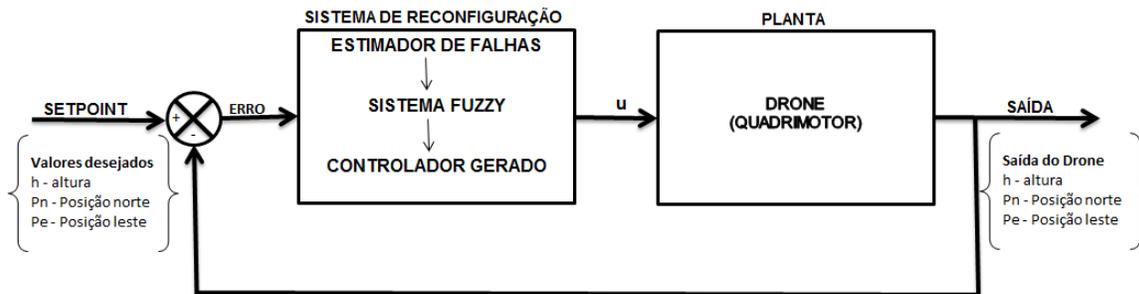


Figura 1.1: Equemático do Sistema de Reconfiguração de Controle

Para que o processo de reconfiguração possa ser aplicado e verificado foi desenvolvido a implementação do sistema no *software* Matlab[®] e assim analisado o sistema de reconfiguração do controle quanto a inserção de falhas e posição gerada em relação a variável altura.

1.4 Artigos Publicados em Anais de Eventos

1. R. B. Silva, R. D. Maia, M. F.S.V. D'Angelo e J. Souza. Reconfiguração de controle em um Drone Quadrimotor auxiliado por um classificador de falhas. *In: A Conferência Sul Em Modelagem Computacional - MCSUL*, 2016, Rio Grande-RS.
2. J. Souza, R. B. Silva e R. D. Maia. Sistema de Classificação de Falhas para Drone Quadrimotor. *In: A Conferência Sul Em Modelagem Computacional - MCSUL*, 2016, Rio Grande-RS.
3. R. B. Silva, H. A. M. Silva e R. D. Maia. Reconfiguração de Sistemas de Controle de um Drone (Quadrimotor). *In: Encontro Nacional de Modelagem Computacional e Engenharia de Materiais*, 2017, IPRJ-UERJ - RJ, aceito.

1.5 Trabalhos em Periódicos

1. R. B. Silva, R. D. Maia, M. F.S.V. D'Angelo e J. Souza. Reconfiguração de controle em um Drone Quadrimotor auxiliada por um classificador de falhas. *In: Revista de Informática Aplicada, Volume XX, Número X, 2017*, publicado¹.
2. J. Souza e R. B. Silva. Classificação de falhas em um Drone quadrimotor: uma aplicação neurofuzzy. *In: Revista de atividades de pesquisa e extensão da Faculdade de Ciência e Tecnologia*, 2017, Montes Claros - MG.

1.6 Organização do Trabalho

A parte restante deste trabalho é composta por mais 5 capítulos. No capítulo 2, mostra os trabalhos relacionados ao tema. No capítulo 3, é realizada a revisão de literatura e para fundamentá-la, é feita uma breve revisão sobre falhas, detecção e isolamento da falha, estimadores e classificadores de falha e redes neurais. No Capítulo 4, a abordagem proposta mostra a modelagem da planta e do sistema de controle. No Capítulo 5, o sistema de reconfiguração é aplicado em um modelo matemático de um Drone Quadrimotor com falha em um dos motores e, assim, gera-se e discute os resultados. Finalmente, no Capítulo 6, são realizadas as considerações finais.

¹Artigo selecionado do evento MCSUL2016 para publicação da versão estendida na Revista de Informática Aplicada (RIA), Volume XX, Número X, 2017

Capítulo 2

Trabalhos Relacionados

Neste capítulo será discutido os trabalhos com temática próxima da abordagem proposta.

2.1 Trabalhos com Drone de quatro motores e controlador PID

Pesquisas aplicadas a reconfiguração de sistemas de controle clássico baseados em falhas em atuadores de Drones de quatro motores são observadas nos trabalhos de Efe (2011) e Amoozgar et al. (2012) em que reconfigura-se um controlador PID (Proporcional - Integral - Derivativo). No trabalho elaborado por Amoozgar et al. (2012), um sistema de detecção de falhas em um dos atuadores manda informação para um reconfigurador *Fuzzy* sintonizar em tempo real um controlador PID e assim os resultados comprovam que a reconfiguração melhora o controle da aeronave.

2.2 Trabalhos com Drone de quatro motores e controlador LQR

O controlador LQR (Regulador linear quadrático) mesmo sendo muito eficiente quando há pequenas falhas e distúrbios é ainda melhorado ao adicionar o sistema de reconfiguração. No trabalho de Chen et al. (2015) utilizou-se o controlador LQR juntamente com a lógica

quântica para reconfiguração de controle. A figura 2.1 mostra como o esquemático da reconfiguração foi implementada, nota-se que o modelo de referência é utilizado para detectar a falha no quadrimotor. Já os ganhos K_x , K_u , K_e servem para ajustar o valor de referência para o sinal do controlador, d_i e d_o parâmetros de ajuste da lógica quântica e os Y_m , y_p e X_p são utilizados como saídas do sistema.

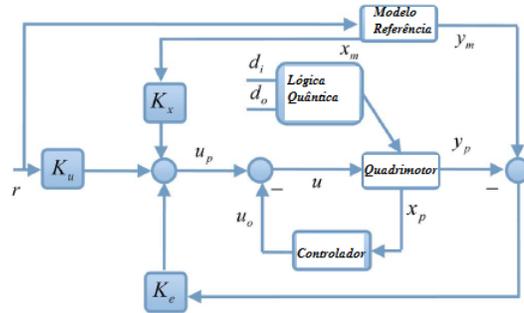


Figura 2.1: Esquema de reconfiguração de controle com LQR e lógica quântica

Fonte: Chen et al. (2015)

Outro trabalho que utiliza o LQR e sistema de reconfiguração após detecção de falha é o de (Büyükkabasakal et al., 2014) que utilizou o seguinte esquema mostrado na figura 2.2. Na figura 2.2 nota-se que após detectado a falha um sistema adaptativo, que recebe informações do sinal de entrada e da saída da planta, envia essa informação para o sistema de reconfiguração de controle.

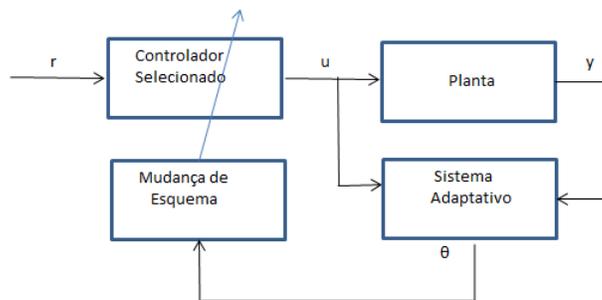


Figura 2.2: Esquema de reconfiguração de controle com LQR

Fonte: Büyükkabasakal et al. (2014)

Embora haja poucos trabalhos com temas próximo da proposta deste trabalho, é notável

que os sistemas de reconfiguração de controle sempre melhoram a performance dos controladores da aeronave. Baseando nesses últimos trabalhos com controlador LQR este trabalho propõe a reconfiguração do sistema de controle de um Drone quadrimotor quando estimado a falha em um dos motores.

Capítulo 3

Revisão de Literatura

Neste capítulo a proposta para reconfiguração de sistemas de controle para Drone é apresentada por uma breve revisão de literatura.

3.1 Fundamentos Importantes

Nas subseções a seguir serão abordados conceitos importantes para o entendimento deste trabalho. Serão discutidos assuntos como Falhas, diagnóstico de falha, redes neurais, estimadores e classificadores de falhas e reconfiguração de controle.

3.1.1 Falhas

O diagnóstico de falhas é um ramo em desenvolvimento na engenharia e o assunto está sendo difundido e bastante utilizado em sistemas de navegação cada vez mais precisos e sofisticados, como em controles de veículos aeroespaciais, atuadores automáticos e diversos sistemas que devem ser focados na garantia da segurança sem perder a viabilidade do projeto (Venkatasubramanian et al., 2003b). Com o surgimento de novos processos, que possuem maiores complexidades, impõem desafios para o seu gerenciamento e manutenção, pois como já é de conhecimento da área, um controle humano em sistemas complexos é considerado inviável e a falta de padronização pode ocasionar taxas de erro e probabilidades de falhas (Inacio, 2014).

O crescimento da detecção de falhas traz para diversos sistemas o poder de montar práticas de manutenção que não se restringem apenas às preventivas mas também a manutenções

predictivas. Esse aspecto fez com que a detecção de falhas se incorporasse ao gerenciamento e manutenção de sistemas, como o *Condition-Based Maintenance* (CBM) e o *Prognostic and Health Managent* (PHM). CMB se refere à manutenção baseada na condição e usa os dados da máquina, ou equipamento, adquiridos em tempo real para determinar a condição de operação, seja normal ou falha, permitindo aos operadores que existam manutenção ou reparos antes que o processo seja interrompido sem prévias (Souza e Silva, 2016). PHM se refere à técnica de tentar prever o comportamento da máquina ou equipamento em relação aos dados obtidos, o que permite estimar sua vida útil em função dos estados de operação atual e passados (George et al., 2006).

O sistema de detecção de falhas deve receber as informações de controle e as informações do estado atual do processo que são normalmente fornecidas pelo usuário e por sensores, sistema que pode ser visto na figura 3.1.

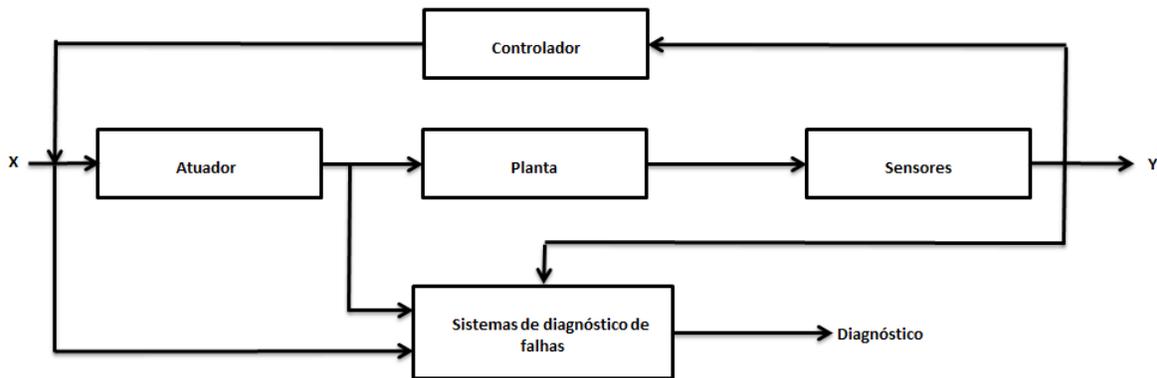


Figura 3.1: Funcionamento da detecção de falhas em um processo

Fonte: Inacio (2014)

A área de Detecção e Isolamento de Falhas, do inglês *Fault Detection and Isolation* (FDI), é vista no geral como a zona de estudo que engloba os sistemas que possam detectar, isolar e identificar a falha; tem seu crescimento expandido ao longo dos anos, assim como toda tecnologia que traz grandes melhorias para a sociedade. Dessa forma, essa área tem mudado constantemente, conforme novos métodos e técnicas são incorporados ao tema principal, iniciado com as técnicas CBM/PHM. Na literatura atual, não existe um consenso entre os termos usados no FDI. As definições mais aceitas e reproduzidas de acordo com Venkatasubramanian et al. (2003b) são:

1. Falha: o termo falha pode ser entendido como o afastamento da variável principal que está sendo observada, ou do parâmetro calculado e associado com o processo em

questão, em relação à faixa de valores aceitáveis para que o processo continue em pleno funcionamento.

2. Diagnóstico de falha: normalmente definido como o processo de detecção, isolamento e identificação de uma condição não aceitável que pode afetar o processo.

A falha denota a incapacidade de um elemento desempenhar a função para a qual foi projetada; é uma condição física que pode ser causada por diversos fatores que incluem erros no projeto, parametrização e implementações incorretas, além de agentes que estão no ambiente e podem gerar a falha, como interferência magnética, perturbação externa e duras condições ambientais. Esses efeitos têm ações drásticas em vários componentes e, especialmente na indústria aeroespacial, em que as aplicações estão sujeitas a fortes interferências magnéticas (Silva et al., 2016).

Uma falha é uma manifestação do sistema em que o estado lógico de um elemento difere do valor pretendido (Nelson, 1990). A falha na abordagem deste trabalho será aplicada no sinal de potência do motor do Drone Quadrimotor. Essa falha simulada, representa uma perda de potência em um dos motores que pode ser causada por desgaste do motor ou impedâncias geradas no circuito externo e interno que alimenta esse motor ocasionando a queda do VANT (Suresh e Kannan, 2008).

3.1.2 Detecção e Isolamento de Falhas

Há diferentes métodos para diagnosticar uma falha no sistema e que, ao longo das pesquisas, foram sendo aperfeiçoados, mas com uma diferença majoritária em relação a como as informações que serão transformadas em conhecimento do processo são utilizadas para o diagnóstico. Essas são normalmente coletadas por métodos baseados em modelos e por métodos baseados no histórico de processo. Nas duas formas, existe algum tipo de conhecimento prévio sobre o processo em questão, seja como modelagem ou banco de dados do histórico do processo (Venkatasubramanian et al., 2003a).

Os métodos baseados em modelos quantitativos utilizam um conjunto de relações matemáticas quantitativas para representar o processo físico, baseado em modelagem matemática. Esses métodos utilizam a redundância analítica para detectar e diagnosticar falhas (Silva et al., 2016). A redundância analítica consiste em comparar as variáveis do processo que chegam a partir dos sensores com os valores calculados analiticamente na modelagem do processo. Pode-se exemplificar com o método de observadores de estado de Takahashi e Peres (1999), os filtros de Kalman de Gao et al. (2008), e a estimação de parâmetros de Isermann (1984).

Os métodos baseados em modelos qualitativos usam um conjunto de informações da base de conhecimento feita em relação ao processo ou relações qualitativas para conseguir retirar

informações sobre o estado do sistema, concluindo se está em operação normal ou possui falhas (Venkatasubramanian et al., 2003a). Diversos métodos pertencem a essa categoria, métodos que empregam dígrafos como o de Maurya et al. (2006), métodos que usam árvores de falhas como o de Dutuit e Rauzy (2005), e sistemas *Fuzzy* como o de (Ragot e Maquin, 2006).

Em contraste com os métodos qualitativos e quantitativos que necessitam de conhecimento prévio do processo, os métodos FDI, baseados no histórico do processo, criam e acessam uma grande base de dados de informações para transformá-las em conhecimento sobre o processo. Esses dados são obtidos como variáveis de entradas e saídas por meio de sensores no início e final do processo, para que o banco de dados seja montado. Esse processo é conhecimento como extração de parâmetros ou características. Desse tipo de método, veem-se o emprego de sistemas em Laurentys et al. (2011), o uso de análise e tendência qualitativa em redes *neurofuzzy* em Caminhas e Takahashi (2001), redes Bayesianas em (D'Angelo et al., 2011).

Mesmo existindo diversos tipos de sistemas com emprego FDI, apenas um tipo de método pode não garantir um perfeito funcionamento e adequação a todos os requisitos propostos para a aplicação, pois a qualidade de qualquer um dos métodos supracitados depende da qualidade da informação (Silva et al., 2016).

3.1.3 As Redes Neurais Artificiais

As redes neurais artificiais são modelos computacionais que têm sua base oriunda do funcionamento do sistema nervoso de seres vivos, possuindo a capacidade de aquisição e manutenção do conhecimento, com a definição das unidades de processamento caracterizados pelos neurônios artificiais que interligam-se por um grande número de interconexões, que são conhecidas como sinapses (Júnior, 2016).

Desenvolveram-no a partir do conhecimento existente sobre o sistema nervoso biológico; os neurônios artificiais ou elementos computacionais são modelos simplórios dos neurônios biológicos. O poder de uma rede neural é extraído da sua estrutura paralela distribuída e a habilidade de aprender e generalizar (Silva et al., 2016).

Estimação de falhas com *Perceptron* Multicamadas - PMC

O *Perceptron* é basicamente uma rede neural de camadas simples e que possui a limitação de classificação em padrões linearmente separáveis; essa limitação pode ser superada com a aplicação da técnica de *Perceptron* Multicamadas (Haykin, 2001). Na técnica PMC, os neurônios artificiais são dispostos em camadas, pelo menos uma dessas camadas de neurônios é escondida e responsável pelo aprendizado e relacionada a alguma aplicação específica. As

redes PMC possuem características que tornam sua arquitetura versátil em relação a sua aplicação (Abdul Rahim et al., 2015).

A rede PMC simples possui treinamento supervisionado, o fluxo de informações tem, normalmente, uma direção, das entradas, percorrendo as camadas intermediária e sendo finalizada nas camadas de saída (Haykin, 2001). A figura 3.2 ilustra uma rede PMC.

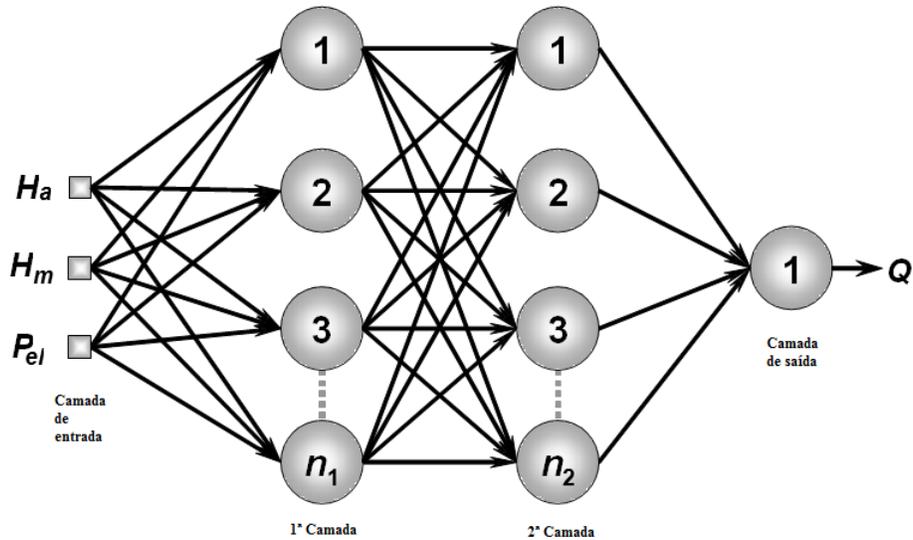


Figura 3.2: Rede PMC com camadas escondidas

Fonte: Abdul Rahim et al. (2015)

Na rede PMC de acordo com a figura 3.2 onde, n_1 , n_2 são a quantidade de neurônios nas camadas intermediárias, e o parâmetro Q representando a quantidade de neurônios na saída. O Algoritmo primário de treinamento utilizado nesse tipo de rede foi o *backpropagation*, que é baseado no método do gradiente descendente, em que a aplicação em relação a um vetor de pesos w tem o objetivo de buscar o valor ótimo para o erro quadrático médio entre a saída desejada e a resposta do combinador linear (Haykin, 2001). Proposta, que foi evoluindo ao longo dos anos de pesquisa e que na atualidade já se utiliza métodos como Resilient Propagation, Conjugate Gradient Based Methods, LM - Levenberg-Marquardt que são métodos em que se aplica à estimação baseados em dados numéricos (Abdul Rahim et al., 2015). Neste trabalho foi utilizado o MLP como estimador de falhas, para isso utilizou como entrada os erros de três variáveis para o treinamento. A falha estimada varia de 0 a 1, sendo o valor “1” operação normal, ou seja, 100% do sinal de potência do motor, já o valor

“0” representa 0%. Além disso, o algoritmo escolhido para treinamento do MLP foi o de Levenberg-Marquardt, visto que o sistema apresentou características não lineares.

Algoritmo de treinamento Levenberg-Marquardt(LM)

O algoritmo de Levenberg-Marquardt é mais poderoso que o método do gradiente descendente, pois consiste na otimização e aceleração do algoritmo *backpropagation*. Para RNAs muito grandes não é recomendável a aplicação do Algoritmo LM devido ao aumento do seu custo computacional (Liu, 2010). A técnica de LM propõe uma solução interconectada entre o algoritmo do gradiente descendente e o método iterativo de Gauss-Newton, incorporando a velocidade de convergência de uma RNA de segunda ordem (Liu, 2010). Tal detalhe faz com que o algoritmo de treinamento LM funcione melhor para sistemas não lineares como é o caso modelo matemático do Quadrimotor. Recomenda-se leitura em Bishop (1995) para melhor entendimento do equacionamento do algoritmo.

3.1.4 Classificação de Falhas

Em qualquer tipo de gerenciamento de processos, máquinas e equipamentos em diversas áreas, é fundamental a chamada tarefa de detecção e classificação de falhas. Pode-se diferenciá-las, pois a detecção baseia-se simplesmente em possuir a informação se existe falha ou não; e a classificação, além de mostrar a falha existente, deve informar também as classes ou tipos de falha, ou, até mesmo, a indicação do local da falha (Silva et al., 2016).

Os estimadores de falhas se diferenciam de classificadores, pois ao estimar a falha, no lugar de classes, a estimação tem como saída normalmente valores referêntes ao nível da falha do sistema (Suresh e Kannan, 2008). Assim, de modo geral, um sistema de classificação de falhas é um sistema capaz de detectar, isolar e identificar a falha; este classificador deve, então, receber tanto as informações de controle quanto as informações do estado atual do processo em questão. Os sistemas também devem lidar com falhas que estão fora do escopo do diagnóstico, como incertezas estruturais e ruídos de medição George et al. (2006).

Em um projeto de um sistema de classificação de falhas, são desejadas características como a rapidez de diagnosticar uma falha, a robustez, a *outliers* e ruídos, a isolabilidade de diferentes falhas e a adaptabilidade de mudanças no processos. E, em alguns processos, há a necessidade da identificação de novidade, que em caso de falhas deve afirmar se é uma falha conhecida ou uma falha nova, além de outros ligados aos requisitos computacionais e de modelagem (Venkatasubramanian et al., 2003b).

Diferentes tipos de diagnósticos de falhas são encontrados na literatura, e a principal diferença entre eles está na forma em que os dados do processo são utilizados para a classificação, em que basicamente são divididos em métodos baseados em modelos e baseados

no histórico do processo. Métodos baseados em modelos utilizam um conjunto de relações matemáticas que representam o processo físico além do conhecimento prévio do processo, seja qualitativo ou quantitativo. Em contraste a esses métodos têm-se os métodos de diagnósticos de falhas que utilizam grandes quantidades de dados do processo para extrair o conhecimento sobre esse. Tais dados são obtidos a partir de variáveis de entrada e de saída medidas por sensores. Esse tipo de método, que tende ao uso de inteligência computacional, possui ramos nas redes neurais artificiais, em redes *neurofuzzy*, redes Bayesianas e sistemas imunes artificiais (Inacio, 2014).

3.1.5 Reconfiguração de Controle

Em caso de falha de um elemento da planta, a reconfiguração de controle é proposta como automática, que se caracteriza como reconfiguração ou reestruturação do sistema, de tal maneira que o sistema se recupere em uma condição segura e posteriormente possa ser controlado manualmente para uma parada segura. Esse tipo de reconfiguração é demonstrado por Suresh e Kannan (2008), Sharifi et al. (2010) e Qian et al. (2016) em que é proposto um erro em elementos de uma aeronave, o controle automático deve tomar as melhores decisões inseridas anteriormente para que a aeronave se recupere à condição inicial de queda até que possa ser novamente pilotada e, em caso de falhas graves e sem condições de permanecer em voo, possa tornar a aterrissagem possível e de forma segura.

Um sistema que possua reparação automática de controle, ou autorreparação, SRCS do termo em inglês, *Self-Repairing Control System*, consiste na detecção das falhas, diagnóstico e etapas de reconfiguração do sistema. Isso implica na detecção e tentativa de classificação, isolamento e quantificação da falha, para medir o impacto mediante o processo. Na figura 3.3 é ilustrado o princípio do SRCS em um sistema proposto neste trabalho, levando em conta a dinâmica de um voo, seu controle, as falhas e a detecção proposta, onde o \mathbf{w} representa o *setpoint*, o \mathbf{u} a variável manipulada, o \mathbf{z} a saída de atuação no processo e o \mathbf{y} a saída para retroalimentação do controlador.

Portanto, após a detecção e diagnóstico da falha e de seu impacto no sistema dinâmico, a reconfiguração do sistema de controle é iniciada para que a falha seja atenuada e assim a condição inicial de queda seja superada e o voo ou aterrissagem sejam seguros. A característica primária de qualquer sistema de reconfiguração é a detecção de variações bruscas no sistema, seja nos sensores, atuadores, na superfície de controle ou qualquer parte do processo que possa ser considerado uma falha, por isso, a detecção da falha é o primeiro passo para ação corretiva. Desse modo, é requerido que seja rápido, com atraso mínimo de detecção e, portanto sensíveis à falha, garantindo que alarmes falsos sejam minimizados, mas que também sejam robustos a erros de modelagem e à existência de redundância.

Partindo do pressuposto de que um observador ou filtro seja criado para que seja insensível

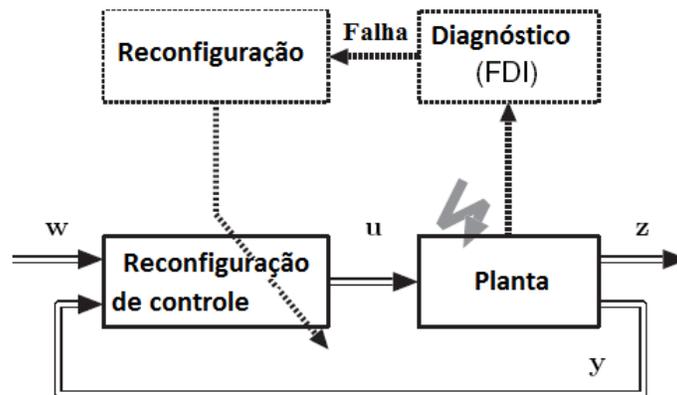


Figura 3.3: Princípio do Controle com SRCS

a alguma falha hipotética, pode-se assim explorar a falha para estabelecer como o algoritmo de decisão atuará, pois a detecção e decisão baseiam-se na sensibilidade do observador e do filtro. Se a falha for detectada e permanecer nesse estado, o sistema então pode passar para o próximo passo e ser capaz de reconfigurar os seus componentes, substituindo ou isolando esse sistema ou subsistema do restante do processo. Essa reconfiguração pode ser feita através de *backup* de configurações normais, ou reconfiguração das entradas de controle (Hajiyev e Caliskan, 2013).

As leis que regem as técnicas de reconfiguração de controle formam a parte mais importante de um sistema que utiliza a autorreparação. Os controladores têm de possuir autoridade, que pode ser definida como a capacidade de substituir o piloto quando o sistema considera que é necessário, para que o processo continue normal. Nas técnicas de reconfiguração, é de uso o termo Fly-By-Wire (FBW) que significa plena autoridade; é usado em processos que o controlador automático pode assumir todo o controle para recuperar o sistema de falhas impostas, além de fazer isso em um tempo aceitável (Chen et al., 2015).

Existem várias técnicas de reconfiguração para sistemas na literatura, das mais conhecidas pode-se exemplificar a técnica *Pseudo-Inverse*, que se baseia na reconstrução do comando quando as forças e movimentos são classificados como falhos, assim utilizando a matriz de eficácia e é feito o inverso para tentar voltar à operação normal, mas depende da matriz de entrada que deve ser conhecida perfeitamente. Há também as técnicas baseadas no controle adaptativo que inclui *self-repairing*, modelo de referência, controle adaptativo e controle preditivo. Usa-se o *feedback* incluso a controladores robustos como LQR e LQG, maximizando a robustez e minimizando as perturbações externas (Hajiyev e Caliskan, 2013). Existem trabalhos com uso de regras simbólicas e conjuntos fuzzy para realização da reconfiguração de controle. Já Ahmed-Zaid et al. (1991) utilizou do Sistema adaptativo para diagnosti-

car falhas em um caça baseado em Gain Sheduling, com o adição de um controle quadrático híbrido adaptativo. Neste trabalho é proposto um sistema de reconfiguração de controle ativo baseado na realimentação das detecções de falhas nos atuadores do drone.

Capítulo 4

Abordagem proposta

Para testar a eficiência da abordagem proposta, o sistema adotado foi um modelo matemático de um VANT Quadrimotor por meio da metodologia adotada por (Beard, 2008). Nesse modelo, foram inseridas falhas em um dos motores. Assim, ao adicionar as falhas, a rede neural realiza a estimação dessas imperfeições e após tal processo ocorre a reconfiguração do controlador de modo a manter o sistema tolerante a essas eventuais falhas.

4.1 O Quadrimotor

O Drone é um dispositivo voador não tripulado, rotulado como VANT, e tem seu uso crescente, decorrente de ter se tornado uma tecnologia barata, de modo que ficou acessível para aplicações civis. A configuração mais comum é composta de quatro motores (Quadrimotor), embora existam outras configurações com um número superior ou inferior de motores. Segundo Chen et al. (2015), a estimação de falhas que viabilize a reconfiguração de controle de um Quadrimotor é importante, pois, ao analisar as principais causas de defeitos, é notável que as falhas nos atuadores tenham os maiores índices, e essas falhas normalmente são as principais causadoras de quedas, por estarem diretamente ligadas à estabilidade e ao voo do quadrimotor.

4.1.1 Modelagem do Quadrimotor Baseada em Newton-Euler

Utilizou-se neste trabalho a modelagem obtida em Beard (2008), que permitiu a elaboração do sistema de controle do drone, além disso, viabilizou a simulação de todo o sistema.

A figura 4.1 demonstra os ângulos que são importantes para a implementação das técnicas de controle, pois ao exercer variações nessas atitudes tem-se uma projeção da propulsão de cada hélice para uma direção, assim viabilizando o deslocamento do drone nos eixos x e y .

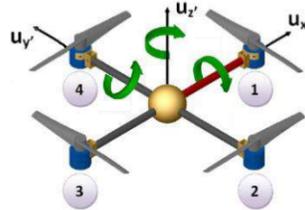


Figura 4.1: Movimento de Atitude do Quadrimotor

Fonte: Beard (2008)

Possuindo o conhecimento das angulações e movimentos os quais o drone exerce, pode-se fazer uso do método de Newton-Euler, que descreve a dinâmica de um corpo rígido de seis graus de liberdade, para realizar a modelagem do VANT, obtendo equações que representam a dinâmica do sistema com respeito ao referencial inercial.

Frames de referência

Nesta seção foram definidos os *frames* de referência para modelagem do Quadrimotor. Um *frame* é um sistema de referência, o qual expressa a posição e orientação de um corpo (Craig, 2012). A figura 4.2 ilustra um *frame* de referência.

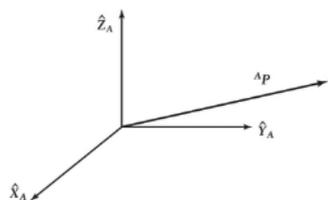


Figura 4.2: *Frame*

Fonte: Craig (2012)

Matriz de rotação

Observando a figura 4.3, o vetor \mathbf{p} pode ser expresso por \mathcal{F}^0 (especificado por $(\hat{i}^0, \hat{j}^0, \hat{k}^0)$), ou por \mathcal{F}^1 (especificado por $(\hat{i}^1, \hat{j}^1, \hat{k}^1)$).

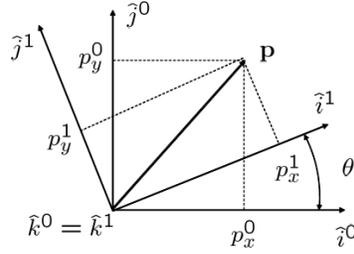


Figura 4.3: Rotação entre *frames* - 2D

Tem-se no frame \mathcal{F}^0 :

$$\mathbf{p} = p_x^1 \hat{i}^0 + p_y^0 \hat{j}^0 + p_z^0 \hat{k}^0 \quad (4.1)$$

Já no *frame* \mathcal{F}^1 , tem-se:

$$\mathbf{p} = p_x^1 \hat{i}^1 + p_y^1 \hat{j}^1 + p_z^1 \hat{k}^1 \quad (4.2)$$

Por definição, essas duas expressões são equivalentes:

$$p_x^1 \hat{i}^1 + p_y^1 \hat{j}^1 + p_z^1 \hat{k}^1 = p_x^1 \hat{i}^0 + p_y^0 \hat{j}^0 + p_z^0 \hat{k}^0 \quad (4.3)$$

Isolando o \mathbf{p}^1 e multiplicando os elementos \hat{i}^1, \hat{j}^1 , e \hat{k}^1 encontra-se a seguinte matriz multiplicando o \mathbf{p}^0 .

$$\mathbf{p}^1 \triangleq \begin{pmatrix} p_x^1 \\ p_y^1 \\ p_z^1 \end{pmatrix} = \begin{pmatrix} \hat{i}^1 \cdot \hat{i}^0 & \hat{i}^1 \cdot \hat{j}^0 & \hat{i}^1 \cdot \hat{k}^0 \\ \hat{j}^1 \cdot \hat{i}^0 & \hat{j}^1 \cdot \hat{j}^0 & \hat{j}^1 \cdot \hat{k}^0 \\ \hat{k}^1 \cdot \hat{i}^0 & \hat{k}^1 \cdot \hat{j}^0 & \hat{k}^1 \cdot \hat{k}^0 \end{pmatrix} \quad (4.4)$$

De acordo com a geometria apresentada na figura 4.1, nota-se que a rotação foi em torno do eixo \mathbf{z} , assim encontra-se:

$$\mathbf{p}^1 = R_0^1 \mathbf{p}^0 \quad (4.5)$$

em que , ao substituir o produto escalar por senos e cossenos, obtém-se a seguinte matriz:

$$R_0^1 \triangleq \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.6)$$

A notação R_0^1 é usada para representar a matriz de rotação do *frame* \mathcal{F}^0 em relação ao *frame* \mathcal{F}^1 . O mesmo procedimento de mapeamento do movimento realizado, porém, com rotação em relação ao eixo y gera-se a matriz de rotação:

$$R_0^1 \triangleq \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (4.7)$$

Já com a rotação em relação ao eixo x resulta-se a matriz de rotação em:

$$R_0^1 \triangleq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix} \quad (4.8)$$

A Matriz R_0^1 possui as seguintes propriedades:

$$\mathbf{P.1.} \quad (R_a^b)^{-1} = (R_a^b)^T = R_b^a \quad (4.9)$$

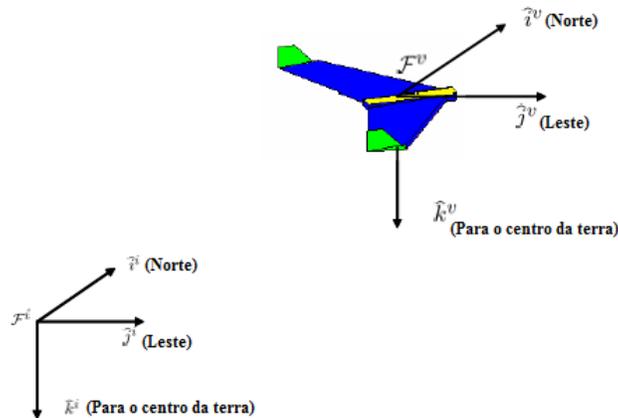
$$\mathbf{P.2.} \quad R_b^c R_a^b = R_a^c \quad (4.10)$$

$$\mathbf{P.3.} \quad \det R_a^b = 1 \quad (4.11)$$

Utilizou-se a equação (4.5) para mapear o movimento de rotação do Quadrimotor para um sistema de coordenadas inercial. Já para a translação, basta realizar uma soma vetorial com o sistema inercial.

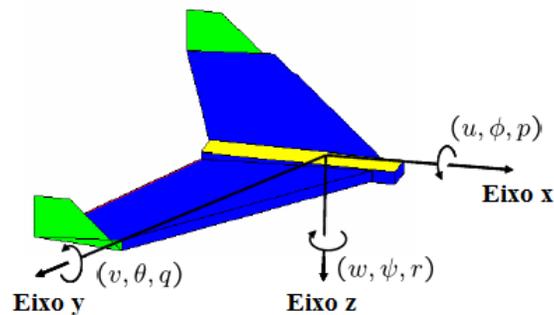
Fixação de Frames de Coordenadas

Para modelar o Quadrimotor, foram fixados *frames*, como mostra a figura 4.4. Todo movimento do veículo é mapeado por meio da comparação entre o *frame* fixado no seu centro de massa em relação ao referencial inercial.

Figura 4.4: Fixação de *Frame* em objetos

Fonte: Beard (2008)

O Quadrimotor pode gerar os movimentos de rotação nos eixos Roll(x), Pitch(y) e Yaw(z) que envolvem, respectivamente, os seguintes ângulos θ , ϕ e ψ conforme figura 4.5.

Figura 4.5: Fixação de *Frame* em objetos

Fonte: Beard (2008)

Os movimentos de rotação descritos na figura 4.5 podem ocorrer de forma separada ou ao mesmo tempo. Assim, para mapear todos esses movimentos, foi realizado o mapeamento individual e, posteriormente, encontrou a equação geral de todas as rotações.

O Movimento do veículo em relação ao eixo Yaw: *frame* \mathcal{F}^{v_1}

A transformação da rotação do frame \mathcal{F}^v em relação ao frame \mathcal{F}^{v_1} permite encontrar o vetor posição:

$$p^{v_1} = R_v^{v_1}(\psi)p^v \quad (4.12)$$

Em que:

$$R_v^{v_1}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.13)$$

O Movimento do veículo em relação ao eixo Pitch: \mathcal{F}^{v_2}

Já a rotação do frame \mathcal{F}^{v_1} em relação ao frame \mathcal{F}^{v_2} resulta em:

$$p^{v_2} = R_{v_2}^{v_1}(\theta)p^{v_1} \quad (4.14)$$

Em que:

$$R_{v_2}^{v_1}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \quad (4.15)$$

O Movimento do veículo em relação ao eixo Roll: *Frame* \mathcal{F}^b

A transformação da rotação do *frame* \mathcal{F}^{v_2} em relação ao *frame* \mathcal{F}^b resulta em:

$$p^b = R_v^{v_2}(\psi)p^{v_2} \quad (4.16)$$

Em que:

$$R_v^{v_2}(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \quad (4.17)$$

A transformação completa

A transformação completa permite encontrar os três movimentos de rotação através da seguinte equação:

$$R_v^b(\phi, \theta, \psi) = R_{v_2}^b(\phi) R_{v_1}^{v_2}(\theta) R_v^{v_1}(\psi) \quad (4.18)$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.19)$$

$$= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix} \quad (4.20)$$

Em que $c\phi \triangleq \cos\phi$ and $s\phi \triangleq \sin\phi$.

4.1.2 Dinâmica e Cinemática do Quadrimotor

Nesta seção são obtidas as equações de dinâmica e cinemática de um corpo rígido. Embora as equações que serão mostradas sejam gerais para qualquer corpo, foram utilizados a notação do sistema de coordenadas e frames abordados na literatura aeronáutica. Para melhor entendimento da modelagem do Quadrimotor, são definidas nas próximas subseções as variáveis de estado do Quadrimotor, a cinemática e a dinâmica.

Variáveis de estado do Quadrimotor

As variáveis de estado do Quadrimotor são divididas em doze:

p_n = Posição inercial sentido norte: posição linear em X em relação ao *frame* da Terra: X;

p_e = Posição inercial sentido leste: posição linear em Y em relação ao *frame* da Terra: Y;

h = Altitude do drone: posição Linear em Z em relação ao *frame* da Terra: Z;

u = Velocidade Linear em X expressa no *frame* do corpo;

v = Velocidade Linear em Y expressa no *frame* do Corpo;

w = Velocidade Linear em Z expressa no *frame* do Corpo;

ϕ = Ângulo de *roll* (Rolagem) em relação ao *frame* da Terra;

θ = Ângulo de *pitch* (Arfagem) em relação ao *frame* da Terra;

Ψ = Ângulo de *yaw* (Guinada) em relação ao *frame* da Terra;

p = Velocidade angular de *pitch* expressa no *frame* do Corpo;

q = Velocidade angular de *roll* expressa no *frame* do Corpo;

r = Velocidade angular de *yaw* expressa no *frame* do Corpo.

As variáveis de estados podem ser organizadas num vetor de estados:

$$x = \begin{bmatrix} p_n \\ p_e \\ h \\ u \\ v \\ w \\ \phi \\ \theta \\ \Psi \\ p \\ q \\ r \end{bmatrix} \quad (4.21)$$

As doze variáveis são divididas em quatro grupos. Esses grupos são os de variáveis relacionadas às posições inercial(p_n, p_e, h) e angular(ϕ, θ, Ψ) e os responsáveis pelas velocidades Linear(u, v, w) e angular(p, q, r).

Cinemática do Quadrimotor

As variáveis de estado $p_n, p_e, e -h$ refere-se ao frame inercial; enquanto as velocidades $u, v, e w$ são do frame fixado ao corpo (Beard, 2008). Portanto, a relação entre Posição e velocidades é dada por:

$$\frac{d}{dt} \begin{pmatrix} p_n \\ p_e \\ -h \end{pmatrix} = R_b^v \begin{pmatrix} u \\ v \\ w \end{pmatrix} = (R_v^b)^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (4.22)$$

A relação entre os ângulos $\phi, \theta, e \psi$ e as velocidades angulares $p, q, e r$ é avaliada através de frames diferentes, mas em Beard (2008), é definida pelas equações a seguir .

Ao relacionar $p, q, e r$ com $\dot{\phi}, \dot{\theta}$ e $\dot{\psi}$ para valores numéricos pequenos (valores em decimais). Nota-se que as seguintes matrizes de rotação derivadas dos ângulos $\dot{\phi}, \dot{\theta}, \dot{\psi}$ geram:

$$R_{v2}^b(\dot{\phi}) = R_{v1}^{v2}(\dot{\theta}) = R_v^{v1}(\dot{\psi}) = I, \quad (4.23)$$

Assim:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = R_{v_2}^b(\dot{\phi}) \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_{v_2}^b(\phi) R_{v_2}^{v_1}(\dot{\theta}) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_{v_2}^b(\phi) R_{v_1}^{v_2} \theta R_{v \rightarrow v_1}(\dot{\psi}) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \quad (4.24)$$

$$= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \quad (4.25)$$

Invertendo:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) \sec(\theta) & \cos(\phi) \sec(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (4.26)$$

Dinâmica do Quadrimotor

Seja \mathbf{v} o vetor de velocidade do Quadrimotor e observando que as leis de Newton aplicadas ao movimento de translação, obtém-se a seguinte equação:

$$m \frac{d\mathbf{v}}{dt_i} = \mathbf{f}, \quad (4.27)$$

em que m é a massa do Quadrimotor, \mathbf{f} é o vetor de forças aplicadas ao Quadrimotor, e $\frac{d}{dt_i}$ A derivada temporal. Da equação de Coriolis, tem-se:

$$m \frac{d\mathbf{v}}{dt_i} = m \left(\frac{d\mathbf{v}}{dt_i} + \omega_{b/i} \times \mathbf{v} \right) = \mathbf{f} \quad (4.28)$$

Em que $\omega_{b/i}$ é a velocidade angular do frame do corpo em relação à frame inercial. Como a força de controle é calculada e aplicada no sistema de coordenadas do corpo, e uma vez que $\omega_{b/i}$ é medida, expressa-se a equação 4.28. Em coordenadas do corpo, em que $v^b \triangleq (u, v, w)^T$, e $\omega_{b/i}^b \triangleq (p, q, r)^T$. Portanto, nas coordenadas do corpo a equação 4.28, torna-se:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad (4.29)$$

Em que $f^b \triangleq (f_x, f_y, f_z)^T$ e os pontos em u, v e w representam as derivadas das variáveis.

Para o movimento de rotação, aplica-se a segunda lei de Newton:

$$\frac{d\mathbf{h}^b}{dt_i} = \mathbf{m} \quad (4.30)$$

em que \mathbf{h} é o momento angular e \mathbf{m} é o torque aplicado. Usando a equação de Coriolis, tem-se:

$$\frac{d\mathbf{h}}{dt_i} = \frac{d\mathbf{h}}{dt_b} + \omega_{b/i} \times \mathbf{h} = \mathbf{m} \quad (4.31)$$

Novamente, a equação 4.31 é resolvida em coordenadas do corpo em que $\mathbf{h}^b = \mathbf{J}\omega_{b/i}$ em que \mathbf{J} é a matriz de inércia constante dada por:

$$\mathbf{J} = \begin{pmatrix} \int (y^2 + z^2) dm & -\int xy dm & -\int xz dm \\ \int xy dm & \int (x^2 + z^2) dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (x^2 + y^2) dm \end{pmatrix} \quad (4.32)$$

$$\triangleq \begin{pmatrix} -J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{pmatrix} \quad (4.33)$$

Sabendo que os momentos de inércia para o drone são calculados assumindo um centro de massa esférico, com massa \mathbf{M} e raio \mathbf{R} , e pontos de massa \mathbf{m} localizados a uma distância de \mathbf{L} a partir do centro, como na figura 4.6.

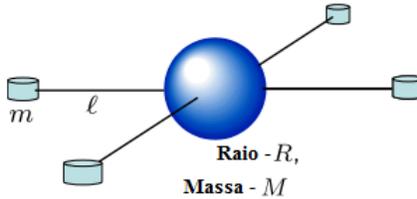


Figura 4.6: Bases para o Momento de Inércia

Fonte: Beard (2008)

Conforme mostrado na figura 4.6, o Quadrimotor é essencialmente simétrico em relação a todos os três eixos, portanto $J_{xy} = J_{xz} = J_{yz} = 0$ o que implica que:

$$\mathbf{J} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \quad (4.34)$$

Portanto:

$$\mathbf{J}^{-1} = \begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix} \quad (4.35)$$

A inércia para uma esfera sólida é dada por $J = 2MR^2/5$. Portanto,

$$J_x = \frac{2MR^2}{5} + 2l^2m \quad (4.36)$$

$$J_y = \frac{2MR^2}{5} + 2l^2m \quad (4.37)$$

$$J_z = \frac{2MR^2}{5} + 4l^2m \quad (4.38)$$

Definindo $\mathbf{m}^b \triangleq (\tau_\phi, \tau_\theta, \tau_\psi)^T$, pode-se escrever a equação 4.31 em coordenadas do corpo como:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix} \left[\begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \right] \quad (4.39)$$

$$= \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix} \quad (4.40)$$

O modelo de seis graus de liberdade para a cinemática e a dinâmica do Quadrimotor podem ser resumidos pelas seguintes equações (4.41) - (4.44):

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\phi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (4.41)$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} fx \\ fy \\ fz \end{pmatrix} \quad (4.42)$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (4.43)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix} \quad (4.44)$$

4.1.3 Forças e Torques

Esta seção descreve como ocorre a atuação das forças e torques de cada um dos motores. Com essa análise podem-se inserir falhas nos motores através da modelagem. Como não existem superfícies de elevação aerodinâmicas, as forças e os momentos aerodinâmicos são insignificantes. As forças e os momentos são principalmente devido à gravidade e à rotação das quatro hélices (figura 4.7).

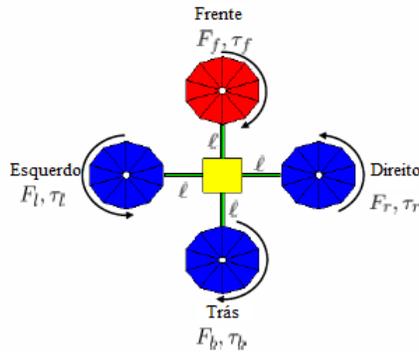


Figura 4.7: Bases para o Momento de Inércia

Fonte: Beard (2008)

A figura 4.7 mostra uma vista superior dos atuadores do Quadrimotor. Já, na figura 4.8, cada propulsor produz uma força F e um torque τ .

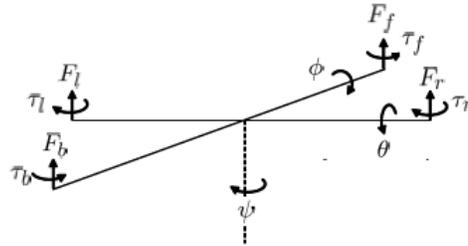


Figura 4.8: Bases para o Momento de Inércia

Fonte: Beard (2008)

A força total que atua no Quadrimotor é dada por:

$$F = F_f + F_r + F_b + F_l \quad (4.45)$$

O torque responsável pelo movimento Roll(ϕ) é produzido pelas forças dos motores direito(r) e esquerdo(l):

$$\tau_\phi = l(F_l - F_r) \quad (4.46)$$

Da mesma forma, o torque do movimento Pitch(θ) é produzido pelas forças dos motores da frente(f) e de trás(b):

$$\tau_\theta = l(F_f - F_b) \quad (4.47)$$

Devido à terceira lei de Newton, o arrasto das hélices produz um torque de guinada (ψ) no corpo do Quadrimotor. A direção do torque será oposta do movimento das hélices de mesmo eixo. Portanto, o torque total de guinada é dado por:

$$\tau_\psi = \tau_r + \tau_l + \tau_f + \tau_b \quad (4.48)$$

O decolagem e arrasto produzidos pelas hélices são proporcionais ao quadrado da velocidade angular (Beard, 2008). Assume-se que a velocidade angular é diretamente proporcional à recomendação de modulação de largura de pulso enviada ao motor. Portanto, a força e o torque de cada motor podem ser expressos como:

$$F_* = k_1 \delta_* \quad (4.49)$$

$$\tau_* = k_2 \delta_* \quad (4.50)$$

Em que k_1 e k_2 são constantes que precisam ser determinadas experimentalmente, δ_* é o sinal de comando do motor, e $*$ representa f, r, b , e l .

Por isso, as forças e torques no Quadrimotor podem ser escritos na seguinte forma de matriz:

$$\begin{pmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \begin{pmatrix} k_1 & k_1 & k_1 & k_1 \\ 0 & -lk_1 & 0 & lk_1 \\ lk_1 & 0 & lk_1 & 0 \\ -k_2 & k_2 & -k_2 & k_2 \end{pmatrix} \begin{pmatrix} \delta_f \\ \delta_r \\ \delta_b \\ \delta_l \end{pmatrix} \triangleq M \begin{pmatrix} \delta_f \\ \delta_r \\ \delta_b \\ \delta_l \end{pmatrix} \quad (4.51)$$

As estratégias de controle derivadas nas seções subsequentes especificarão forças e Torques. Os comandos reais dos motores podem ser expressos por:

$$\begin{pmatrix} \delta_f \\ \delta_r \\ \delta_b \\ \delta_l \end{pmatrix} = M^{-1} \begin{pmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \quad (4.52)$$

Observa-se que os comandos de modulação de largura de pulso precisam estar entre "zero" e "um". Assim, quando estão em "um" enviam a potência total nos motores e quando estão em "zero" a potência é nula. Com essa informação é possível adicionar falha nos motores.

Além da força exercida pelo motor, a gravidade também exerce uma força sobre o quadrimotor. Na estrutura do veículo \mathcal{F}^v , a força de gravidade que atua no centro de massa é dada por:

$$\mathbf{f}_g^v \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (4.53)$$

No entanto, uma vez que \mathbf{v} na segunda equação de dinâmica e cinemática é expresso em \mathcal{F}^b , deve-se transformar a estrutura do corpo para encontrar a seguinte representação:

$$\mathbf{f}_g^b = R_v^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{pmatrix} \quad (4.54)$$

Dessa forma, as equações são representadas a seguir:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (4.55)$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \quad (4.56)$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (4.57)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x} \tau_\phi \\ \frac{1}{J_y} \tau_\theta \\ \frac{1}{J_z} \tau_\psi \end{pmatrix} \quad (4.58)$$

As equações 4.55, 4.56, 4.57 e 4.58 representam a modelagem completa do sistema quadrimotor.

4.1.4 Linearização do Sistema e Descrição no Espaço de Estados

Após encontrar as equações 4.55, 4.56, 4.57 e 4.58 não lineares apresentadas na seção anterior, realizou-se a linearização para permitir o projeto do sistema de controle. A linearização do sistema é descrita por uma equação não linear, baseando no processo da expansão da série de Taylor. Esse método pode ser descrito como uma função $F(x)$ de uma única variável x , supondo que \bar{x} é um ponto de equilíbrio do sistema tal que $f(x) = 0$. O ponto \bar{x} é um ponto de equilíbrio do sistema $x = f(x)$, tem-se que $x = 0$ quando $x = \bar{x}$, isto é, o sistema atinge o equilíbrio. Assim os pontos próximos ao ponto de equilíbrio são aceitos como lineares. Com a aplicação da série de Taylor, é possível encontrar as matrizes que regem as ações do Quadrimotor e assim aplicar o controle. A fim de alcançar a forma do espaço de estados, representado como:

$$\begin{cases} \dot{x} = A.x + B.u \\ y = C.x + D.u' \end{cases} \quad (4.59)$$

E poder escrito da seguinte forma:

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases} \quad (4.60)$$

Em que x é o vetor de estados, previamente definido na equação 4.59, e u é o vetor de entradas:

$$u = \begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (4.61)$$

O vetor u representa as entradas de referência de velocidades para os motores. Já o $f(x, u)$ é um conjunto de funções que representa o comportamento do sistema. O conjunto citado foi definido pelas variáveis de estado definidas na equação 4.21. Assim, neste trabalho, as derivadas das variáveis de estado são definidas a partir da equação 4.58 em conjunto com as matrizes de transformação corpo-terra para as variáveis referenciadas no sistema terra.

Após definir os conjuntos de equações, pode-se então linearizar o modelo. Para linearizá-lo, utiliza-se o conceito da série de Taylor (Pfeifer, 2013). Observando a equação 4.60, pode-se desenvolvê-las com a série de Taylor de primeira ordem da seguinte forma:

$$\begin{cases} \dot{x} \approx f(\bar{x}, \bar{u}) + \left. \frac{\partial f(x,u)}{\partial x} \right|_{(\bar{x}, \bar{u})} \cdot (x - \bar{x}) + \left. \frac{\partial f(x,u)}{\partial u} \right|_{(\bar{x}, \bar{u})} \cdot (u - \bar{u}) \\ y \approx h(\bar{x}, \bar{u}) + \left. \frac{\partial h(x,u)}{\partial x} \right|_{(\bar{x}, \bar{u})} \cdot (x - \bar{x}) + \left. \frac{\partial h(x,u)}{\partial u} \right|_{(\bar{x}, \bar{u})} \cdot (u - \bar{u}) \end{cases} \quad (4.62)$$

Em que:

- \bar{x} é o vetor ponto de equilíbrio dos estados;
- \bar{u} é o vetor de entradas nominais.

Os pontos de equilíbrio das entradas foram considerados nulos, exceto para altura(h), sendo essa altitude nominal do Quadrimotor. Assim, para limitar os esforços de controle, o somatório dos empuxos individuais se igualam à força peso do corpo para mantê-lo em equilíbrio no ar.

De acordo com o método de linearização, pode-se expressar:

$$\begin{cases} \tilde{x} = (x - \bar{x}) \\ \tilde{u} = (u - \bar{u}) \end{cases} \quad (4.63)$$

Consideram-se \tilde{x} e \tilde{u} as flutuações em torno do ponto de equilíbrio de cada variável ou valor nominal de cada entrada. Ao substituir a equação 4.63 na equação 4.62, tem-se:

$$\begin{cases} \dot{\tilde{x}} \approx f(\bar{x}, \bar{u}) + \left. \frac{\partial f(x,u)}{\partial x} \right|_{(\bar{x}, \bar{u})} \cdot \bar{x} + \left. \frac{\partial f(x,u)}{\partial u} \right|_{(\bar{x}, \bar{u})} \cdot \bar{u} \\ \tilde{y} \approx h(\bar{x}, \bar{u}) + \left. \frac{\partial h(x,u)}{\partial x} \right|_{(\bar{x}, \bar{u})} \cdot \bar{x} + \left. \frac{\partial h(x,u)}{\partial u} \right|_{(\bar{x}, \bar{u})} \cdot \bar{u} \end{cases} \quad (4.64)$$

Através da semelhança entre as equações 4.64 e a 4.59 podem-se definir as matrizes de estados como:

$$\begin{cases} A_{lin} = \frac{\partial f(x,u)}{\partial x} \Big|_{(\bar{x},\bar{u})} \\ B_{lin} = \frac{\partial f(x,u)}{\partial u} \Big|_{(\bar{x},\bar{u})} \\ C_{lin} = \frac{\partial h(x,u)}{\partial x} \Big|_{(\bar{x},\bar{u})} \\ D_{lin} = \frac{\partial h(x,u)}{\partial u} \Big|_{(\bar{x},\bar{u})} \end{cases} \quad (4.65)$$

Modelo no espaço de estados são definidos a partir das matrizes encontradas pela equação 4.65, mas para isso foram substituídos os parâmetros de um Drone real, conforme tabela 4.1.

Tabela 4.1: Parâmetros do Quadrimotor Real

Descrição do parâmetro	Parâmetro	Valor
Massa do quadrotor	m	2,24 kg
Aceleração da gravidade	g	9,81 m/s^2
Momento de Inércia em torno do eixo x	Jx	0,0363 $Kg.m^2$
Momento de Inércia em torno do eixo y	Jy	0,0363 $Kg.m^2$
Momento de Inércia em torno do eixo z	Jz	0,0615 $Kg.m^2$

Tais parâmetros foram obtidos por meio da tese de (Raffo, 2011).

4.2 O controlador Regulador Linear Quadrático - LQR

O Regulador Linear Quadrático, ou somente LQR, é uma tecnologia de uso ótimo e robusto, de fácil implementação aliada a um bom desempenho. É usado amplamente em diversos tipos de processos, como o da robótica até os processos químicos. O controlador LQR foi escolhido para este trabalho por ser uma técnica de controle ótimo no domínio do tempo baseada na realimentação de estados que apresenta boas propriedades de estabilidade robusta. Outra característica do LQR é a capacidade de estabelecer uma harmonia entre a energia gasta e o sinal de controle, sendo capaz de garantir a estabilidade do sistema frente a pequenas perturbações ou falhas (Büyükkabasakal et al., 2014). Considerando um sistema invariante no tempo, descrito como mostra a equação 4.66 :

$$\dot{X} = Ax(t) + Bu(t)y = Cx(t) \quad (4.66)$$

Assim, a implementação de um controlador LQR é baseado em determinar uma lei de controle $u(t)$ que minimize a uma função quadrática $J = f(x, u)$ onde Ogata e Severo (2003) diz que a 4.67 é:

$$J \doteq \int_0^{\infty} (x(t)'Qx(t) + u(t)'Ru(t))dt \quad (4.67)$$

Em que as matrizes de ponderação são denominadas por $Q = Q' > 0$ e $R = R' > 0$. Assim as variáveis de controle podem estar presentes na constante de realimentação de estados, dadas por 4.68:

$$u(t) = -Kx(t) \quad (4.68)$$

A solução para $u(t)$ que torne o problema ótimo quadrático é expressa por 4.69:

$$u(t) = -Kx(t) = -(R)^{-1}B'Px(t) \quad (4.69)$$

Tanto que P satisfaça a equação de Ricatti 4.70:

$$A'P + PA - PB(R)^{-1}B'P + Q = 0 \quad (4.70)$$

Desse modo, a função custo para a condição inicial pode ser obtida através do cálculo da equação 4.67 é dada por 4.71:

$$J \doteq \int_0^{\infty} (x(t)'Q + K'RK)x(t)dt == x(\infty)Px(\infty) + x(0)'Px(0) \quad (4.71)$$

Portanto, considerando um sistema com realimentação estável, tem-se $x(\infty) \rightarrow 0$, concluindo que 4.72:

$$J = x(0)'Px(0) \quad (4.72)$$

Isso assumindo que o sistema em questão seja controlável e observável.

4.3 Projeto do controlador

Após a modelagem do sistema são inseridos dados reais de um Quadrimotor e linearizado para obtenção das variáveis de espaço de estados. Só assim inicia o projeto do controlador e, para isso, serão realizados os testes de controlabilidade e observabilidade (Franklin et al., 2006).

4.3.1 Verificação de Controlabilidade e Observabilidade

Para implementar o controlador LQR, o sistema deve ser controlável e observável (Pfeifer, 2013).

Um sistema é dito completamente “controlável” se existir um vetor $u(t)$ que leve o sistema de qualquer estado inicial $x(t_0)$ a qualquer estado final $x(t_1)$, em que $t_1 > t_0$.

Pelo método algébrico:

$$\mathcal{C} = [B \quad A.B \quad \dots \quad A^{n-1}.B] \quad (4.73)$$

e

$$Posto(\mathcal{C}) = n_{posto} \quad (4.74)$$

Em que \mathcal{C} é a matriz de controlabilidade, n é a dimensão do vetor de estados e n_{posto} é o resultado do posto. O sistema é dito completamente controlável quando n_{posto} é igual a n . A ordem do subsistema não controlável é:

$$Dim_{n\tilde{a}o \ cntr} = n - n_{posto} \quad (4.75)$$

Com o *software* Matlab[®], obtém-se o posto da matriz, por meio do seguinte comando:

$$rank(ctrb(A, B)) = 12$$

Como citado anteriormente na modelagem, o total de variáveis de estado é 12, portanto:

$$n = 12 \quad (4.77)$$

Ou seja, o sistema é controlável porque a dimensão do vetor de estados é igual ao posto.

Já para verificar se os estados são observáveis, utiliza-se a avaliação da matriz de observabilidade:

$$\mathcal{O} = \begin{bmatrix} C \\ C.A \\ \vdots \\ C.A^{n-1} \end{bmatrix} \quad (4.78)$$

Assim como na avaliação da matriz de controlabilidade, o posto da matriz \mathcal{O} indica se o sistema é completamente observável. E o comando no *software* Matlab[®] para realizar essa avaliação é:

$$\text{rank}(\text{obsv}(A, C)) = 12 \quad (4.79)$$

O sistema em questão é completamente observável, visto que o posto da matriz de observabilidade é igual a dimensão n do sistema.

Conforme análise, o sistema é controlável e observável o que permite a aplicação do controlador LQR. Dessa forma, a lei de controle é dada pela equação a seguir:

$$u = -k_{LQR} \cdot (x - x_{referências}) \quad (4.80)$$

Em que k_{LQR} é o ganho que será obtido de cada uma das técnicas que serão aplicadas nas subseções a seguir.

4.3.2 Projeto do Controlador LQR

O controlador LQR é um controlador ótimo, capaz de amenizar incertezas da modelagem de sistemas dinâmicos isso faz com que ele seja o mais adequado para este trabalho (Pfeifer, 2013).

Observando a equação 4.67, é importante encontrar os valores das matrizes Q e R baseando em dados reais de limitação do Quadrimotor.

Para definir o ganho K de retroação, utiliza-se:

$$k = R^{-1} \cdot B^T \cdot P \quad (4.81)$$

Em que P é calculado a partir da equação de Riccati 4.70.

A partir desse ponto, definem-se as matrizes de compromisso. Seleccionam-se inicialmente matrizes e diagonais. Os valores para essas matrizes serão definidos a partir do método de Bryson, em que cada elemento da matriz diagonal é:

$$Q = \frac{1}{x_i^2, max} \quad (4.82)$$

e

$$R = \frac{1}{u_i^2, max} \quad (4.83)$$

Observando que x_i^2, max é o valor máximo desejável do estado x_i , e similarmente u_i^2, max é o valor máximo desejável do controle u_i . Considera-se a utilização desta técnica apenas uma primeira iteração para gerar o controlador, adicionalmente, testes com o controlador e a planta irão gerar novos valores para os elementos da matriz.

Como anteriormente citado, para gerar as matrizes, é necessária a definição dos valores máximos dos estados e entradas do sistema, sendo eles definidos na Tabela 4.2 e Tabela 4.3. Ressalta-se que os valores para as velocidades angulares dos motores são incrementais aos seus valores nominais de operação.

Com esses valores definidos, utilizam-se da equação 4.82 e equação 4.84 para gerar as matrizes Q e R . A partir da definição dessas matrizes, utiliza-se o comando do *software* Matlab[®] para gerar o ganho do controlador LQR, sendo:

$$k = lqr(A, B, Q, R, Ts) \quad (4.84)$$

Esse comando já retorna o ganho para o controlador LQR no domínio do tempo discreto. Com o ganho do controlador, realizou-se a simulação do sistema através do *software* Matlab[®] no ambiente *simulink* conforme figura 4.9. A estrutura apresentada na figura 4.9 mostra o sistema de controle o qual é adicionado o sistema de reconfiguração, já o modelo do Quadrimotor contém as equações que regem as propriedades do Drone. Além disso, o bloco integrador é utilizado para converter as saída do sistema do modelo que é dado pela derivadas das variáveis, em variáveis úteis para realimentação do controlador.

Antes do projeto, realizou-se a simulação do sistema com um valor de ganho ajustado pelo método empírico e encontrou-se a resposta para variável altura de acordo a figura 4.10.

Já após o projeto do controlador, obteve-se a resposta apresentada figura 4.11 que apresentou um menor tempo para entrar em regime permanente comparado com a figura 4.10.

Observou-se, também, uma diminuição no tempo de subida da resposta do sistema (em relação à variável altura) conforme a comparação entre as figuras 4.10 e 4.11, após a realização do projeto do controlador.

Tabela 4.2: Tabela de parâmetros dos estados

Variáveis de Estado	Parâmetros Máximos
P_n	0,5 m
P_e	0,5 m
h	0,3 m
u	0,3 m/s
v	0,3 m/s
w	0,3 m/s
ϕ	pi/18 rad
θ	pi/18 rad
ψ	pi/18 rad
p	pi/20 rad/s
q	pi/20 rad/s
r	pi/20 rad/s

Tabela 4.3: Tabela de parâmetros de entradas

Entradas	Parâmetros máximos
F	0,7 N
τ_ϕ	0,2 N.m
τ_θ	0,2 N.m
τ_ψ	0,08 N.m

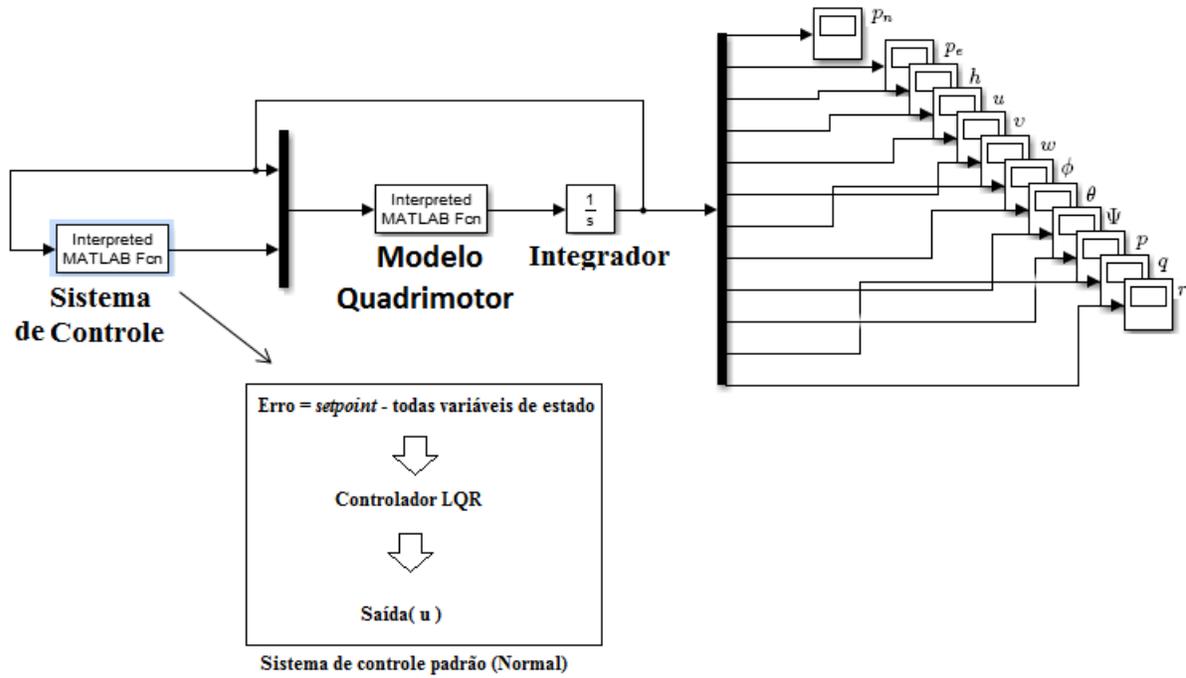


Figura 4.9: Representação do Controlador no *Simulink*

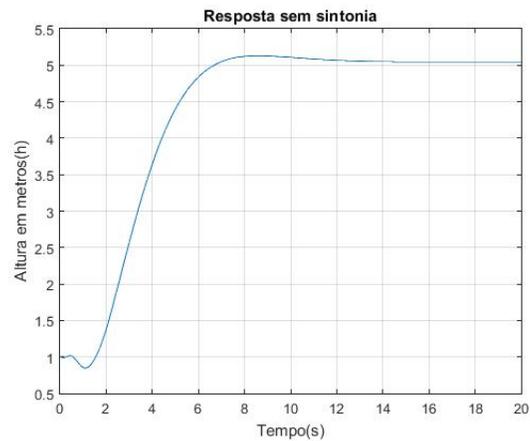


Figura 4.10: Resposta da altura antes do projeto do controlador LQR

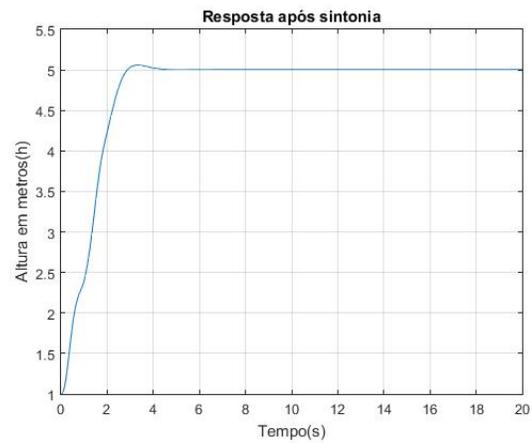


Figura 4.11: Resposta da altura após o projeto do controlador LQR

Capítulo 5

Resultados

Neste capítulo será abordado os resultados encontrados após simulação da proposta de reconfiguração de sistema de controle comparando com o sistema padrão de controle.

5.1 Resultados do Estimador de Falhas

Os experimentos para este trabalho foram realizados no *software* MatLab[®], principalmente no ambiente *simulink*. Inicialmente foi proposto o controle do sistema com a inserção de falhas; como resultados tem-se a figura 5.1 com as três variáveis altura (h), posição leste (Pe) e posição norte (Pn), mostrando o sistema com falha sem a adição do sistema de reconfiguração de controle. Nota-se que inicialmente as três variáveis estavam em “um” metro e foi inserido um novo *setpoint* com valor igual a “quatro” metros.

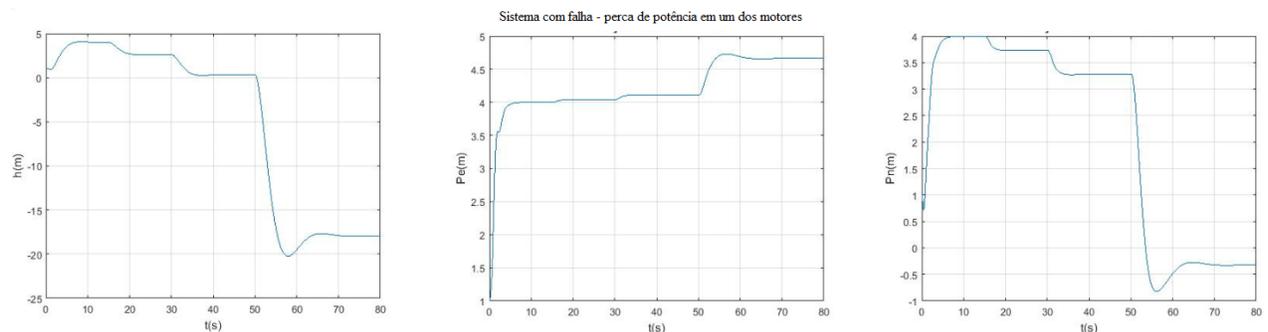


Figura 5.1: Simulação de falha

Observando os gráficos da figura 5.1, nota-se que as falhas inseridas possuem um valor de entrada temporal, os quais são mostrados na tabela 5.1. É notável, na figura 5.1, que a inserção de falha em 15s (falha 20%) o controlador LQR padrão consegue amenizar se mantendo no ar, já em 30s e falha de 40% o sistema não consegue recuperar mais a altura nem as posição norte e leste. Além disso, é notável no primeiro gráfico da figura 5.1 que o modelo não limita a altura ao patamar “0”, mas tal limitação é adicionada para as próximas simulações apresentadas.

Tabela 5.1: Relação de tempo x falha

Tempo	Falha em percentual de perda do sinal controle em um dos motores
Até 15s	0%
15 - 30s	20 %
30 -50s	40%
50 - 80s	80%

Após induzir essas falhas no modelo matemático, gerou-se o treinamento da rede neural (MLP), conforme a figura 5.2, para utilização na estimação de falhas que será auxiliadora na reconfiguração do sistema de controle. Para tal treinamento, utilizaram-se como entrada os erros entre o valor desejado e o valor real das variáveis h , P_n e P_e . Em mudanças de *setpoint* o sistema de reconfiguração e o estimador são desabilitados, pois tal mudança representa uma falsa falha no Quadrimotor.

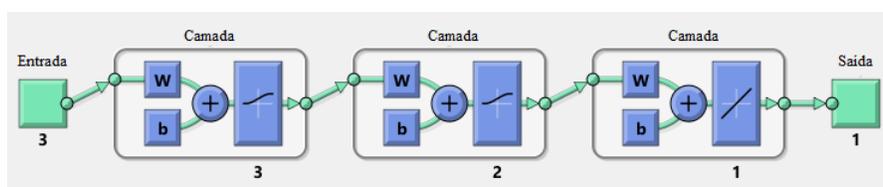


Figura 5.2: Configuração da rede mlp

A rede foi configurada, baseando-se em experimentos de simulação e gráficos gerados, com os seguintes parâmetros:

- Camadas: 3 Camadas com 3, 2, 1 neurônios respectivamente.
- Método de Treinamento: Levenberg-Marquardt (2ª Ordem).

- Performance: *Mean Squared Error*.
- Épocas: 5000 a 8000.

Dessa forma, considerando “1” para sinal máximo de controle e “0” para sinal mínimo, gerou-se o seguinte estimador de falhas, como visto na figura 5.3.

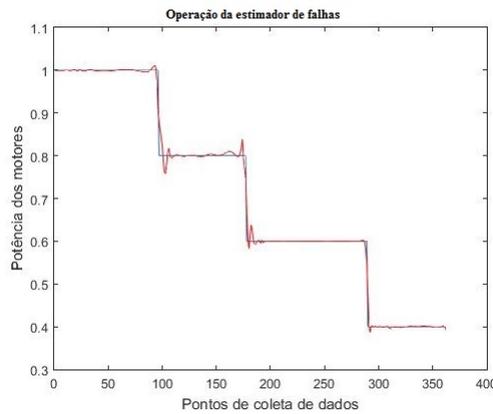


Figura 5.3: Operação do estimador de falhas

Na figura 5.3, a linha em azul refere-se à amostra que foi utilizada para treinamento, já na linha vermelha é a operação da rede neural para estimação da falha. Nota-se que para um limiar de erro aceitável, o estimador consegue separar as ocorrências de falha no motor do Quadrimotor. Observa-se, na figura 5.4, um novo teste do estimador com acréscimo do valor de 80% de falha, e comprova-se que a função aproxima bem do esperado.

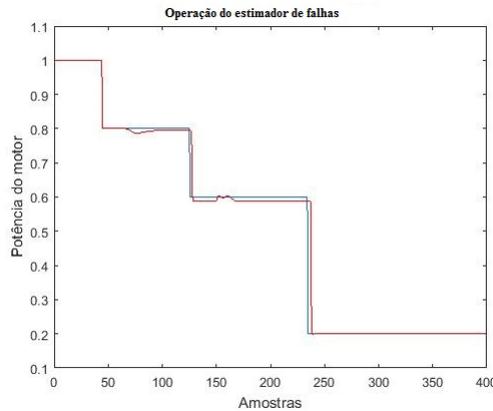


Figura 5.4: Operação do estimador de falhas

5.2 Resultados do Sistema de Reconfiguração de Controle

As implementações de todo o sistema iniciaram-se desde o entendimento da modelagem matemática do Quadrimotor baseada em Newton - Euler, seguido do processo de linearização e do projeto e sintonia do controlador LQR. Após a definição do controlador, foram gerados os experimentos para produzir uma base de dados para treinamento da RNA. Assim, após tal definição, geraram-se 10 configurações de controladores (tolerantes aos níveis de falhas de 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% e 100%), nas quais se observou que para o limiar de 20 % do nível da falha, o controlador conseguiu atuar amenizando-a, ou seja, para falhas de 1% a 20% o controlador de 20% de falha é tolerante. Como, com aumento da quantidade de controladores aumentava-se o tempo de processamento do sistema de controle, experimentalmente, definiram-se os cinco controladores com maior eficiência para a reconfiguração . Os cinco controladores escolhidos foram os tolerantes as falhas de 20%, 40%, 60%, 80% e o de operação normal (como são mostrados na tabela 5.2).

Assim, após tal etapa de estimação, geraram-se modelos de controladores LQR para as situações abordadas na tabela 5.1. Definiram-se tais modelos de controladores baseando em experimentos e análise das respostas simuladas. Assim modelaram-se os controladores conforme a tabela 5.2, em que para cada controlador obtém-se também um novo ponto de operação em um nova matriz de parâmetros para o sistema LQR.

A estrutura do sistema de reconfiguração, baseado na retroação desse, é dado de acordo com a figura 5.5.

Dentro do bloco sistema de controle apresentado na figura 5.5, há um sistema de recon-

Tabela 5.2: Controladores ativos

Controlador	Falha em percentual de perda de sinal de controle
1	(Operação normal)
2	(Falha de 20%)
3	(Falha de 40%)
4	(Falha de 60%)
5	(Falha de 80%)

figuração de controle que utiliza dados do estimador de falhas e através do sistema *Fuzzy* (Mamdani) gera a saída do controlador. O sistema de reconfiguração funciona da seguinte forma: Inicialmente o estimador de falhas envia um valor de 0 a 1 para uma variável x que faz parte das funções de pertinência 5.1, 5.2, 5.3, 5.4 e 5.5.

$$p1 = e^{-\frac{1}{2}\left(\frac{x-1}{\sigma}\right)^2} \quad (5.1)$$

$$p2 = e^{-\frac{1}{2}\left(\frac{x-0,8}{\sigma}\right)^2} \quad (5.2)$$

$$p3 = e^{-\frac{1}{2}\left(\frac{x-0,6}{\sigma}\right)^2} \quad (5.3)$$

$$p4 = e^{-\frac{1}{2}\left(\frac{x-0,4}{\sigma}\right)^2} \quad (5.4)$$

$$p5 = e^{-\frac{1}{2}\left(\frac{x-0,2}{\sigma}\right)^2} \quad (5.5)$$

Como pode-se observar as funções de pertinências 5.1, 5.2, 5.3, 5.4 e 5.5 são gauseanas e foram escolhidas baseando-se em experimentos o qual permitiu também inferir as regras apresentadas no gráfico da figura 5.6. Já o valor da saída do controlador da reconfiguração u é calculado de acordo com a equação 5.6, apresentada a seguir.

$$u = \frac{\sum u_i \cdot p_i}{\sum p_i} \quad (5.6)$$

Em que: u_i corresponde ao valor da saída máxima de pertinência de cada controlador projetado baseando-se em falhas parciais, e p_i corresponde à pertinência associada aos valores de entrada (valor gerado pelo estimador de falhas).

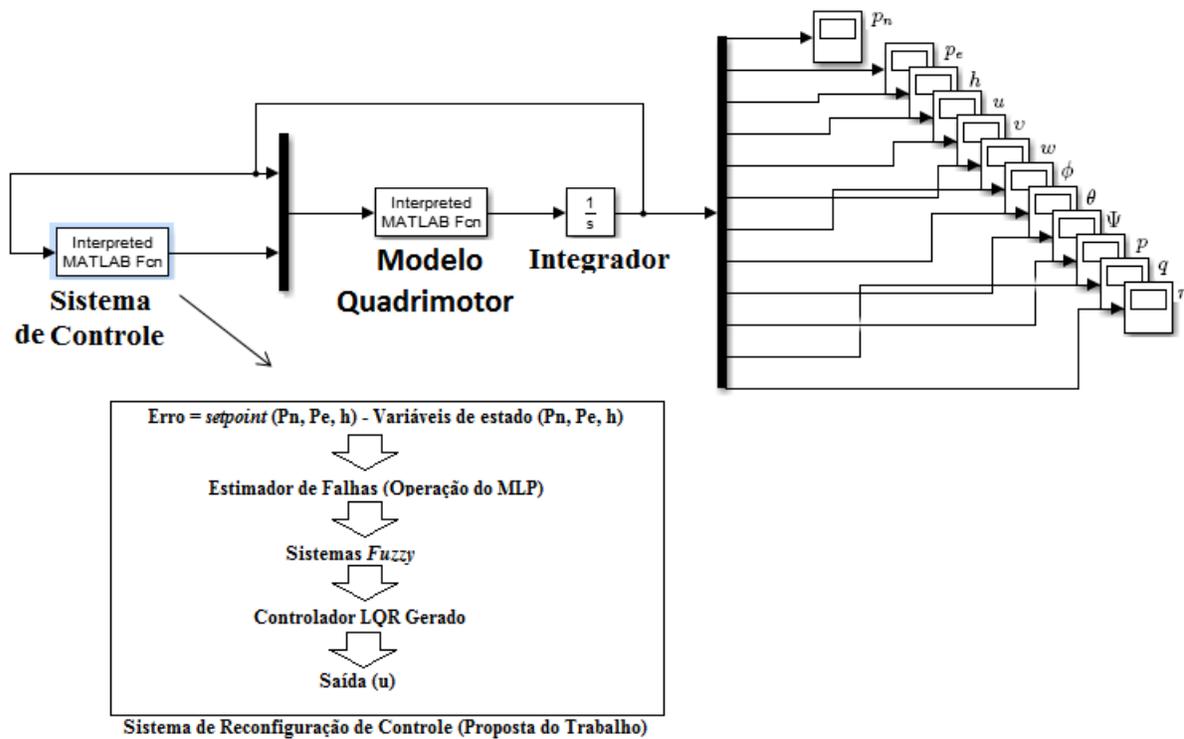


Figura 5.5: Princípio do Controle proposto

A figura 5.7 representa o sistema após a adição do sistema de reconfiguração de controle auxiliado pelo estimador de falhas.

Ao analisar os gráficos da figura 5.7, observa-se que a variável altura consegue ser controlada com maior eficiência em relação às demais. Apenas quando a falha é de maior intensidade no momento do tempo em cinquenta segundos, por características físicas, afeta de forma drástica o voo do Quadrimotor, visto que esse tem que permanecer no ar com basicamente três motores com eficiência máxima e um com apenas vinte por cento de eficiência.

Para uma melhor avaliação da variável altura e do sistema de reconfiguração observa-se na figura 5.8 o sistema em azul como sendo a metodologia proposta e em vermelho o sistema de controle sem reconfiguração de controle para um cenário em que a falha aumenta gradativamente conforme a tabela 5.3.

Observa-se que na tabela 5.3 que o percentual de falhas induzidas no sinal de controle de potência foi aumentado de forma gradativa de 10%.

Ao comparar os dois sistemas na figura 5.8 nota-se que o sistema sem reconfiguração (em vermelho) com a inserção da primeira falha no tempo de 15 segundos já começa perder altura enquanto o sistema com reconfiguração de controle ameniza a falha e mantém a altura setada.

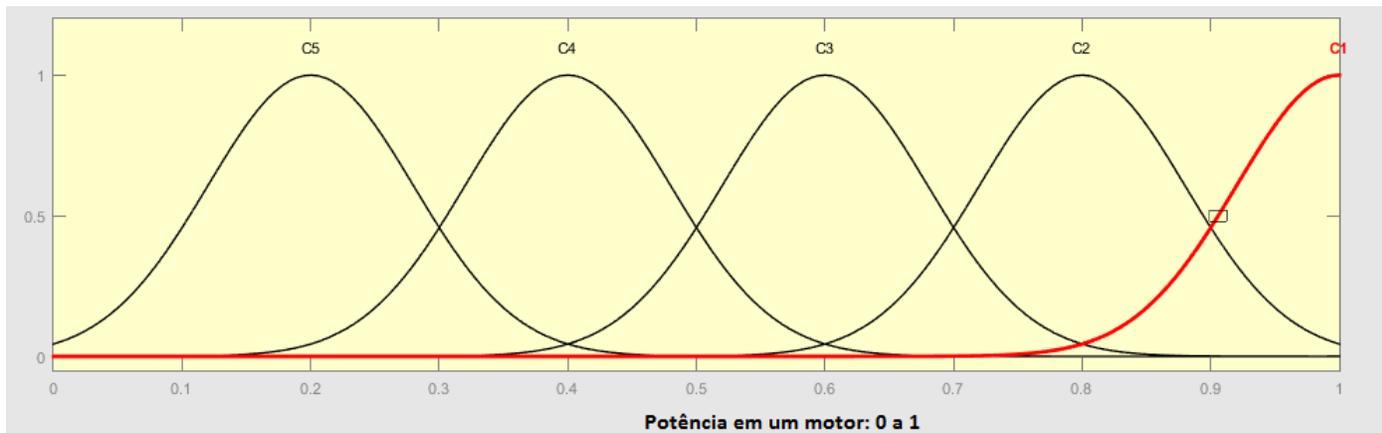


Figura 5.6: Funções de pertinência para chaveamento de controlador por lógica *Fuzzy*

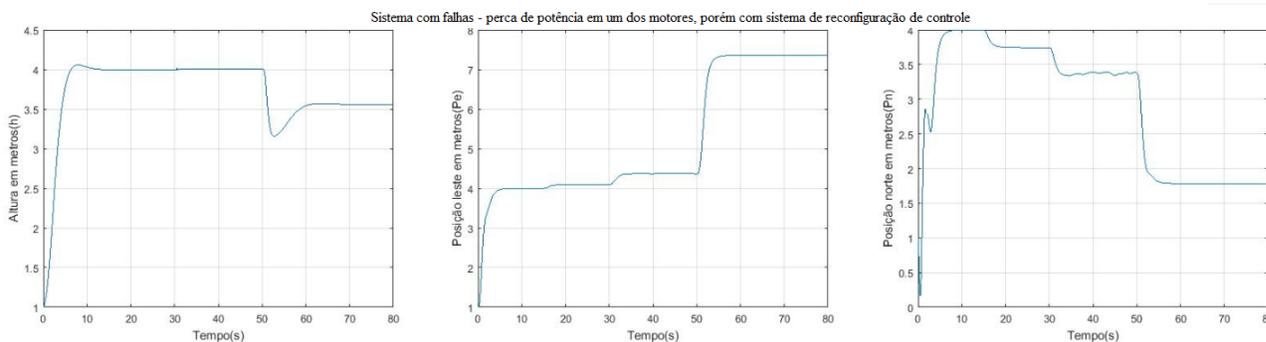


Figura 5.7: Operação do sistema de reconfiguração de controle

Apenas no tempo de aproximadamente de 135 segundos o sistema de reconfuração começa a perder altura visto que a intensidade da falha simulada já chega a 70 %. Portanto, a proposta de reconfiguração de controle graficamente fica evidente ser melhor do que sistema padrão de controle.

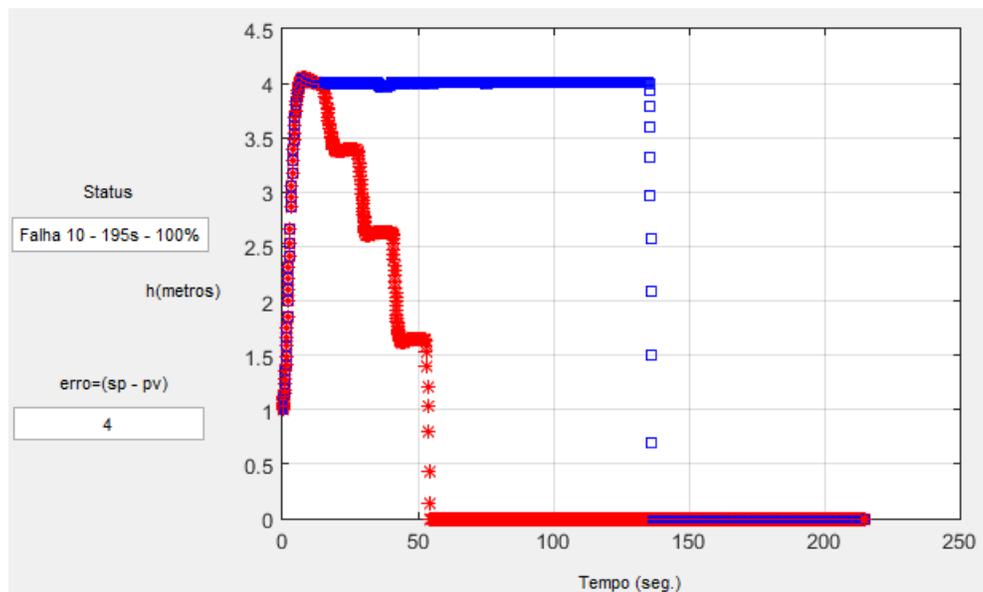


Figura 5.8: Comparativo entre o sistema normal e a proposta de reconfiguração

5.3 Resultados Numéricos do Sistema de Reconfiguração de Controle

Para comprovar numericamente a eficiência do Sistema de Reconfiguração de Controle utiliza-se a tabela 5.4 que mostra quatro índices de desempenho comparados entre a proposta de reconfiguração de controle e um controlador LQR padrão. É importante saber que o índice deve ser um número positivo ou nulo para ser útil (Dorf, 2009). Portanto, o sistema que minimiza esse índice é considerado o melhor. Os índices de desempenho mais usados são : ISE (*Integral of the Square of the Error*), IAE (*Integral of the Absolute magnitude of the Error*), ITAE (*Integral of Time multiplied by Absolute of the Error*) e o ITSE (*Integral of Time multiplied by the Squared Error*).

Tabela 5.3: Cenário de falhas

Tempo (segundos)	Perda potência de sinal (%)
0 - 15	0
15 - 35	10
35 - 55	20
55 - 75	30
75 - 95	40
95 - 115	50
115 - 135	60
135 - 155	70
155 - 175	80
175 - 195	90
195 - 215	100

Assim, tem-se:

- ISE - Integral do quadrado do Erro:

$$ISE = \int_0^T e^2(t) dt \quad (5.7)$$

- IAE - Integral do valor absoluto do erro:

$$IAE = \int_0^T |e(t)| dt \quad (5.8)$$

- ITAE - Integral do tempo multiplicado pelo valor absoluto do erro:

$$ITAE = \int_0^T t|e(t)| dt \quad (5.9)$$

- ITSE - Integral do tempo multiplicado pelo quadrado do erro:

$$ITSE = \int_0^T te^2(t) dt \quad (5.10)$$

Considerando e como o erro entre o valor desejado e o valor real das variáveis, o T como o tempo de cálculo do índice na simulação e t o tempo instantâneo no momento do erro.

Para todos os índices é notável que a proposta de reconfiguração de controle obteve melhor desempenho, visto que os índices resultam em um valor numérico menor.

Tabela 5.4: Tabela de índice de desempenho dos sistemas de controle

Índice	Controlador normal	Controlador com sistema de reconfiguração
ISE	$3,5662 \times 10^{14}$	$2,0235 \times 10^7$
IAE	$2,3861 \times 10^8$	$1,6336 \times 10^4$
ITAE	$5,1043 \times 10^{10}$	$3,4164 \times 10^6$
ITSE	$7,6303 \times 10^{16}$	$4,2874 \times 10^9$

Como a proposta deste trabalho é reconfigurar o sistema de controle do Quadrimotor através do estimador de falhas para evitar perdas com a queda do quadrimotor, pode-se considerar um bom resultado numérico, porém com algumas limitações que são geradas dos dados de um sistema real.

Capítulo 6

Considerações Finais

Este trabalho teve como objetivo principal propor uma abordagem para estimação de falhas e reconfiguração de controle em Drone de forma adaptativa, ou seja, a metodologia proposta reconfigura o sistema de controle de um Drone Quadrimotor após a detecção do nível de falha em perda de potência (simulada) em um dos motores. Assim, após estimar a falha por uma rede neural, o sistema realiza a modificação do controlador para a falha estimada através da lógica *fuzzy*.

Para tanto, este trabalho propôs um sistema de estimação de falhas baseado em RNA que utilizou como dados os erros entre as variáveis altura, posição norte e posição leste que foram retiradas da simulação de um modelo matemático do Quadrimotor, após inclusão de falhas em um dos motores. O sistema apresenta uma função que desabilita a reconfiguração na mudança de *setpoint*, pois nesse caso tal mudança implicaria em uma falsa falha em um dos atuadores.

Os resultados da operação da RNA demonstraram que o sistema teve um bom desempenho para estimar a queda de potência parcial em um dos motores do Quadrimotor. Além disso, como o sistema de reconfiguração necessitou de tais resultados do estimador de falhas para reagir e amenizar os efeitos das falhas, observa-se que os sistemas de estimação de falhas e o de reconfiguração, baseados em lógica *Fuzzy*, integrados, reagiram bem mantendo o Quadrimotor por um tempo maior de voo em relação à abordagem com apenas um controlador LQR padrão.

Ao finalizar o projeto, concluíram-se, com êxito, os objetivos específicos que foram estudar e conhecer as metodologias utilizadas para a detecção e estimação de falhas existentes na literatura e, através de uma análise numérica, evidenciou-se um maior desempenho da proposta de reconfiguração do sistema do Quadrimotor ao ser submetido a falhas em um dos motores quando comparado ao sistema normal, sem reconfiguração. Além disso, avaliou-se

a metodologia proposta através de uma comparação de resultados gráficos do sistema de controle padrão com um sistema de reconfiguração proposto, com uma ênfase na variável altura e na detecção de falha de um dos motores do Drone.

A contribuição deste trabalho foi propor uma abordagem de reconfiguração de controle adaptativa tolerante a algumas falhas estimadas por uma RNA e controlador modificado por meio da lógica *Fuzzy*. Dessa forma, a abordagem proposta é mais um opção de sistema de reconfiguração de controle basendo em estimação de falhas em atuadores de VANTs e reconfiguração baseada em lógica *Fuzzy*. Analisando os resultados da abordagem proposta na estimação de falhas e na reconfiguração, percebe-se que a abordagem proposta é promissora para estimação e reconfiguração de sistemas dinâmicos.

Futuramente pretendem-se aplicar e avaliar o comportamento e a eficiência da abordagem proposta em um sistema físico do Drone e, além disso, aspira-se à inserção de falhas em mais de um dos motores simultaneamente, pretende-se também abordar as falhas causadas por sensores e a aplicação de classificadores de falhas substituindo os estimadores.

Referências Bibliográficas

- Abdul Rahim, N., Paulraj, M., Adom, A. H., Shukor, S. A. A., e Masnan, M. J. (2015). Homogeneous multi-classifier system for moving vehicles noise classification based on multilayer perceptron. *Journal of Intelligent & Fuzzy Systems*, 29(1):149–157.
- Ahmed-Zaid, F., Ioannou, P., Gousman, K., e Rooney, R. (1991). Accommodation of failures in the f-16 aircraft using adaptive control. *IEEE Control Systems*, 11(1):73–78.
- Amoozgar, M. H., Chamseddine, A., e Zhang, Y. (2012). Fault-tolerant fuzzy gain-scheduled pid for a quadrotor helicopter testbed in the presence of actuator faults. *IFAC Proceedings Volumes*, 45(3):282–287.
- Beard, R. (2008). *Quadrotor Dynamics and Control*. Brigham Young University.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.
- Büyükkabasakal, K., Fidan, B., e Savran, A. (2014). Mixing adaptive fault tolerant control of quadrotor uav. *Asian Journal of Control*.
- Caminhas, W. M. e Takahashi, R. H. (2001). Dynamic system failure detection and diagnosis employing sliding mode observers and fuzzy neural networks. *Proceedings of the Joint 9th IFSA and 20th NAFIPS*, pgs. 304–309.
- Chen, F., Jiang, R., Wen, C., e Su, R. (2015). Self-repairing control of a helicopter with input time delay via adaptive global sliding mode control and quantum logic. *Information Sciences*, 316:123–131.
- Craig, J. J. (2012). *Robótica*. 3ª edição.
- D’Angelo, M. F., Palhares, R. M., Takahashi, R. H., e Loschi, R. H. (2011). Fuzzy/bayesian change point detection approach to incipient fault detection. *IET control theory & applications*, 5(4):539–551.

- Dorf, R. C. (2009). Sistemas de controle modernos. 11. *Ed. Rio de Janeiro: LTC*.
- Dutuit, Y. e Rauzy, A. (2005). Approximate estimation of system reliability via fault trees. *Reliability Engineering & System Safety*, 87(2):163–172.
- Efe, M. Ö. (2011). Neural network assisted computationally simple pi d control of a quadrotor uav. *IEEE Transactions on Industrial Informatics*, 7(2):354–361.
- Franklin, G. F., Powell, J. D., e Emami-Naeini, A. (2006). *Feedback control of dynamic systems*. Pearson Prentice Hall, Upper Saddle River, NJ.
- Gao, H., Chen, T., e Wang, L. (2008). Robust fault detection with missing measurements. *International Journal of Control*, 81(5):804–819.
- George, V., Frank, L., Michael, R., Andrew, H., e Biqing, W. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. John Wiley & Sons, Inc.
- Hajiyev, C. e Caliskan, F. (2013). *Fault diagnosis and reconfiguration in flight control systems*, volume 2. Springer Science & Business Media.
- Haykin, S. (2001). Redes neurais: Princípios e práticas; trad. *Paulo Martins Engel*.
- Inacio, M. J. (2014). Diagnóstico de falhas baseado em sistema inteligente evolutivo.
- Isermann, R. (1984). Process fault detection based on modeling and estimation methods—a survey. *Automatica*, 20(4):387–404.
- Júnior, E. M. d. S. (2016). Estimativa de volume de madeira de árvores de eucalipto utilizando redes neurais artificiais do tipo rbf.
- Laurentys, C., Bomfim, C., Menezes, B., e Caminhas, W. M. (2011). Design of a pipeline leakage detection using expert system: A novel approach. *Applied Soft Computing*, 11(1):1057–1066.
- Liu, H. (2010). On the levenberg-marquardt training method for feed-forward neural networks. Em *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 1, pgs. 456–460. IEEE.
- Maurya, M. R., Rengaswamy, R., e Venkatasubramanian, V. (2006). A signed directed graph-based systematic framework for steady-state malfunction diagnosis inside control loops. *Chemical Engineering Science*, 61(6):1790–1810.
- Nelson, V. P. (1990). Fault-tolerant computing: Fundamental concepts. *Computer*, 23(7):19–25.

- Neri, D. (2016). Adaptive control strategies for improved forward flight of a tilt-rotor uav.
- Ogata, K. e Severo, B. (2003). *Engenharia de controle moderno*. Prentice Hall do Brasil.
- Pfeifer, E. (2013). Projeto e controle de um uav quadrirotor.
- Qian, M., Xiong, K., Wang, L., e Qian, Z. (2016). Fault tolerant controller design for a faulty uav using fuzzy modeling approach. *Mathematical Problems in Engineering*, 2016.
- Raffo, G. V. (2011). *Robust Control Strategies for a Quadrotor Helicopter*. PhD thesis, Universidad de Sevilla Escuela Técnica Superior de Ingeniería.
- Ragot, J. e Maquin, D. (2006). Fault measurement detection in an urban water supply network. *Journal of Process Control*, 16(9):887–902.
- Rubertus, D. P. (1983). Self-repairing flight control systems overview. pgs. 1280–1286.
- Sharifi, F., Mirzaei, M., Gordon, B. W., e Zhang, Y. (2010). Fault tolerant control of a quadrotor uav using sliding mode control. Em *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pgs. 239–244. IEEE.
- Silva, R. B., Maia, R. D., D’Angelo, M. F. S. V., e Souza, J. (2016). Reconfiguração de controle em um drone quadrimotor auxiliado por um classificador de falhas. *A Conferência Sul Em Modelagem Computacional - MCSUL*, 7(86):760–769.
- Souza, J. e Silva, R. B. (2016). Sistema de classificação de falhas para drone quadrimotor. *A Conferência Sul Em Modelagem Computacional - MCSUL*, 7(76):497–504.
- Suresh, S. e Kannan, N. (2008). Direct adaptive neural flight control system for an unstable unmanned aircraft. *Applied Soft Computing*, 8(2):937–948.
- Takahashi, R. e Peres, P. (1999). Unknown input observers for uncertain systems: a unifying approach. *European Journal of Control*, 5(2):261–275.
- Venkatasubramanian, V., Rengaswamy, R., e Kavuri, S. N. (2003a). A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313–326.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., e Kavuri, S. N. (2003b). A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311.