

Universidade Federal de Juiz de Fora
Faculdade de Engenharia
Engenharia Elétrica com Habilitação em Robótica e Automação Industrial

Felipe Ferraz Martins

**Planejamento de Trajetória, Controle de Seguimento de Trajetória e
Interface Gráfica de Simulação para Veículos Aéreos Não-Tripulados**

Juiz de Fora

2017

Felipe Ferraz Martins

Planejamento de Trajetória, Controle de Seguimento de Trajetória e Interface Gráfica de Simulação para Veículos Aéreos Não-Tripulados

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Elétrica com Habilitação em Robótica e Automação Industrial da Universidade Federal de Juiz de Fora, como requisito para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Manuel Arturo Rendón Maldonado

Juiz de Fora

2017

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Martins, Felipe.

Planejamento de Trajetória, Controle de Seguimento de Trajetória e Interface Gráfica de Simulação para Veículos Aéreos Não-Tripulados / Felipe Ferraz Martins. – 2017.

85 f. : il.

Orientador: Prof. Manuel Arturo Rendón Maldonado

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Engenharia Elétrica com Habilitação em Robótica e Automação Industrial, 2017.

1. Quadricóptero. 2. Planejamento de Trajetória. 3. Seguimento de Trajetória. I. Maldonado, Manuel Arturo Rendon, orient. II. Título.

Felipe Ferraz Martins

Planejamento de Trajetória, Controle de Seguimento de Trajetória e Interface Gráfica de Simulação para Veículos Aéreos Não-Tripulados

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Elétrica com Habilitação em Robótica e Automação Industrial da Universidade Federal de Juiz de Fora, como requisito para obtenção do grau de Engenheiro Eletricista.

Aprovado em: 24 de novembro de 2017

BANCA EXAMINADORA

Prof. Dr. Manuel Arturo Rendón Maldonado -
Orientador
Universidade Federal de Juiz de Fora

Professor Dr. André Luis Marques Marcato
Universidade Federal de Juiz de Fora

Professor Dr. Alexandre Bessa dos Santos
Universidade Federal de Juiz de Fora

ATA DE DEFESA DE TRABALHO FINAL DE CURSO - TFC

DATA DA DEFESA: **24/11/2017**
CANDIDATO: **Felipe Ferraz Martins**
ORIENTADOR: **Prof. Manuel Arturo Rendón Maldonado**

BANCA EXAMINADORA: (*Nome Completo / Instituição*)

Prof. **Manuel Arturo Rendón Maldonado** (Presidente / UFJF)
Prof. **André Luis Marques Marcato** (UFJF)
Prof. **Alexandre Bessa dos Santos** (UFJF)

TÍTULO:

PLANEJAMENTO DE TRAJETÓRIA, CONTROLE DE SEGUIMENTO DE TRAJETÓRIA E INTERFACE GRÁFICA DE SIMULAÇÃO PARA VEÍCULOS AÉREOS NÃO-TRIPULADOS

LOCAL: **FACULDADE DE ENGENHARIA.**
HORA DO INÍCIO: 16:00 h

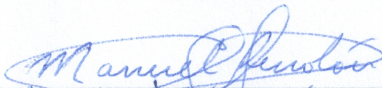
Em sessão pública, após exposição de cerca de 20 minutos, o candidato foi argüido oralmente pelos membros da banca tendo como resultado:

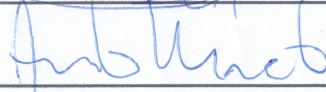
APROVAÇÃO COM NOTA 100;
 REPROVAÇÃO.


Na forma regulamentar foi lavrada a presente ata que é abaixo assinada pelos membros da banca na ordem acima determinada e pelo candidato.

Juiz de Fora, 11 de novembro de 2017.

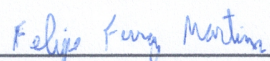
Presidente:







Candidato:



AGRADECIMENTOS

Muitas pessoas contribuíram direta e indiretamente para a execução deste trabalho, o qual representa a conclusão de uma etapa importante em minha vida.

Agradeço, primeiramente, à minha mãe, Fátima Ferraz, a qual é a principal responsável por minhas vitórias. Agradeço também ao meu pai Flávio, ao meu irmão Gabriel, aos meus primos Fábio e Fausto Júnior, aos meus tios Mary, Fausto (in memoriam), Gustavo e Marcela, aos meus avós Cícero (in memoriam) e Conceição (in memoriam), à madrasta da minha mãe Auxiliadora, e a toda minha família que, com muito apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Agradecimento especial aos meus amigos de longa data Bruno Caniato, Felipe Freesz, Larissa Freesz, Rodrigo Varandas e Victor Fabri, os quais sempre estiveram presentes em minha vida. Agradeço também a todos os integrantes do BDA (Allan Mattar, Bernardo Pedrosa, Hugo Tome, Matheus Oliveira, Rodrigo Daccacche), a todos os remanescentes do B5 (Alberto Janicke, Carlos Spoljarick, Gabriel Sanguinetti, Jackson Carvalho, Jefferson Carneiro, Matheus Zaglia, Nicholas Meschke, Rodrigo Quintas, Wissam Quemel, dentre outros) e aos inúmeros amigos que fiz durante o curso de engenharia (Ana Carolina, André Maia, Bruna Barbosa, Daniel Ramalho, Douglas Ferreira, Eduardo Simões, Fabiane Barbosa, Gabriel Bruno, Lucas Guida, Mariane Vilas Boas, Paulo Roberto, Pedro Alcântara, dentre outros).

Agradeço ao meu orientador, Manuel Rendón, com quem eu trabalhei em conjunto por mais de 5 anos desenvolvendo conhecimento. Agradeço também a todos os professores com quem tive contato durante o curso, os quais foram cruciais em diversas etapas do meu aprendizado.

Agradeço ao diretor da unidade ArcelorMittal Juiz de Fora, Waldenir Lima, e a todos os excelentes profissionais do Laboratório de Eletrônica (Antônio, Davi Lacerda, Fábio Pereira, José Fernando, José Sérgio, Leonardo Berbert, Odenir Pereira, Robson Delfino, Sebastião Paiva, Sidnei Rocha, Thaison Reis, dentre outros), onde tive uma oportunidade de estágio.

*“The saddest aspect of life right now is that science gathers knowledge faster
than society gathers wisdom.”*
(Isaac Asimov)

RESUMO

O presente trabalho visa apresentar e comparar diferentes tipos de controlador para um quadricóptero: Controle para Pequenos Ângulos, Controle do Vetor Empuxo Desejado e Controle por Rastreamento Geométrico. A principal diferença entre as abordagens está no cálculo do erro que será realimentado em cada controlador. Para o ajuste dos ganhos desses controladores, alguns métodos são discutidos: *Non-Dimensional Tuning* e *Particle Swarm Optimization*. Ainda, são apresentadas equações simplificadas do modelo dinâmico de um quadricóptero, além de uma explicação da base teórica sobre *frames* e matrizes de rotação. Por fim, desenvolve-se um algoritmo simples de planejamento de trajetória ótima chamado de Trajetória com Snap Mínimo. Todo o desenvolvimento do presente trabalho culmina na comparação entre os três controladores apresentados. Para realizar tal comparação, implementou-se um simulador em MatLab com diversas funcionalidades. Os resultados obtidos mostram que o Controlador por Rastreamento Geométrico possui uma melhor convergência entre os três controladores apresentados.

Palavras-chave: Quadricóptero. Planejamento de Trajetória. Seguimento de Trajetória.

ABSTRACT

The present work aims at presenting and comparing different types of controllers for a quadricopter: Small Angle Control, Desired Thrust Vector Control and Geometric Tracking Control. The main difference between the approaches is in the calculation of the error that will be fed back into each controller. For tuning the gains of these controllers, some methods are discussed: Non-Dimensional Tuning and Particle Swarm Optimization. Also, we present simplified equations of the dynamic model of a quadricopter, as well as an explanation of the theoretical basis on frames and rotation matrices. Finally, a simple algorithm for optimum trajectory planning is developed, called the Minimum Snap Trajectory. The whole development of the present work culminates in the comparison between the three controllers presented. To perform this comparison, a simulator was implemented in MatLab with several functionalities. The results shown that the Geometric Tracking Controller has the best convergence between the controllers presented.

Key-words: Quadrotor. Trajectory Planning. Trajectory Following.

LISTA DE ILUSTRAÇÕES

Figura 1 – Quadricóptero	17
Figura 2 – Principais frames em um quadricóptero	24
Figura 3 – Rotação em 2D	25
Figura 4 – Diagrama de Blocos do sistema	32
Figura 5 – Curvas obtidas seguindo restrições de continuidade e diferenciabilidade	47
Figura 6 – Ambiente de Simulação	49
Figura 7 – Modificar parâmetros do sistema	50
Figura 8 – Ambiente de Simulação em funcionamento	50
Figura 9 – Gráfico em três dimensões com <i>snapshots</i>	51
Figura 10 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 5$ segundos	53
Figura 11 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 5$ segundos	53
Figura 12 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 5$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.003659m^2	54
Figura 13 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 2$ segundos	54
Figura 14 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 2$ segundos	55
Figura 15 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 2$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.376825m^2	55
Figura 16 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 2	56
Figura 17 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 2	57
Figura 18 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 2 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.123305m^2	57
Figura 19 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 3	59
Figura 20 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 3	60

Figura 21 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 3 em comparação com a trajetória desejada. O Erro Quadrático Médio tendeu ao infinito . . .	60
Figura 22 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 5$ segundos . . .	61
Figura 23 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 5$ segundos	62
Figura 24 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 5$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.000072m^2	62
Figura 25 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 2$ segundos . . .	63
Figura 26 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 2$ segundos	63
Figura 27 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 2$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.044599m^2	64
Figura 28 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 2	65
Figura 29 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 2	66
Figura 30 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 2 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.023816m^2	66
Figura 31 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 3	67
Figura 32 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 3	68
Figura 33 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 3 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 1.745761m^2	68
Figura 34 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 5$ segundos	69
Figura 35 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 5$ segundos	70

Figura 36 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 5$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.000022m^2	70
Figura 37 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 2$ segundos	71
Figura 38 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 2$ segundos	71
Figura 39 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 2$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.001320m^2	72
Figura 40 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 2	73
Figura 41 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 2	74
Figura 42 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 2 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.001526m^2	74
Figura 43 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3	75
Figura 44 – Ilustração com <i>snapshots</i> da manobra de recuperação da trajetória executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3 entre $t = 1$ segundo e $t = 3,5$ segundos . . .	76
Figura 45 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3	76
Figura 46 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.022389m^2	77

LISTA DE TABELAS

Tabela 1 – Restrições para uma sequência de pontos pré-definidos	46
Tabela 2 – Pontos arbitrária para o cálculo de uma curva	46
Tabela 3 – Restrições para uma sequência de pontos pré-definidos para uma trajetória $\psi(t)$	48
Tabela 4 – Trajetória 2	56
Tabela 5 – Trajetória 3	58
Tabela 6 – Comparação do Erro Quadrático Médio entre os três controladores utilizados	77
Tabela 7 – Tempo de processamento de cada iteração dos controladores na placa Arduino Mega 2560	78

LISTA DE ABREVIATURAS E SIGLAS

VANT	Veículo Aéreo Não-Tripulado
DoF	Graus de Liberdade
IFAC	<i>International Federation of Automatic Control</i>
ROS	<i>Robot Operating System</i>
GPS	<i>Global Positioning System</i>
VTOL	<i>Vertical Take-Off and Landing</i>
PSO	<i>Particle Swarm Optimization</i>

LISTA DE SÍMBOLOS

ϕ	<i>Roll</i>
θ	<i>Pitch</i>
ψ	<i>Yaw</i>
$\dot{\phi}$	Taxa de variação de <i>Roll</i>
$\dot{\theta}$	Taxa de variação de <i>Pitch</i>
$\dot{\psi}$	Taxa de variação de <i>Yaw</i>
p	Velocidade Angular relacionada ao eixo x no <i>Frame</i> do Corpo Rígido
q	Velocidade Angular relacionada ao eixo y no <i>Frame</i> do Corpo Rígido
r	Velocidade Angular relacionada ao eixo z no <i>Frame</i> do Corpo Rígido
\dot{p}	Aceleração Angular relacionada ao eixo x no <i>Frame</i> do Corpo Rígido
\dot{q}	Aceleração Angular relacionada ao eixo y no <i>Frame</i> do Corpo Rígido
\dot{r}	Aceleração Angular relacionada ao eixo z no <i>Frame</i> do Corpo Rígido
$\hat{i}, \hat{j}, \hat{k}$	Vetores unitários de um <i>frame</i>
$s\phi, s\theta, s\psi$	Seno dos ângulos ϕ, θ e ψ
$c\phi, c\theta, c\psi$	Cosseno dos ângulos ϕ, θ e ψ
R_X^Y	Matriz de rotação de um vetor representado em um <i>frame</i> arbitrário X para um <i>frame</i> arbitrário Y

SUMÁRIO

1	INTRODUÇÃO	16
1.1	APRESENTAÇÃO DO SISTEMA	17
1.2	REVISÃO BIBLIOGRÁFICA	17
1.2.1	Classificação	17
1.2.2	Linhas de pesquisa	19
1.2.2.1	<i>Controle</i>	19
1.2.2.2	<i>Planejamento de Trajetória</i>	20
1.2.2.3	<i>Visão Computacional</i>	20
2	MODELAGEM DO SISTEMA	21
2.1	FRAMES	21
2.1.1	Principais <i>frames</i>	21
2.1.1.1	<i>Principais frames utilizados para quaisquer veículos aéreos</i>	22
2.1.1.2	<i>Principais frames utilizados para quadricópteros</i>	23
2.1.2	Matrizes de rotação e transformação	24
2.1.3	Equação de Coriolis	29
2.2	EQUAÇÕES DE NEWTON-EULER PARA MODELAGEM DE UM QUADRICÓPTERO	30
3	CONTROLE DO SISTEMA	32
3.1	CONTROLE PARA PEQUENOS ÂNGULOS	33
3.2	CONTROLE DO VETOR EMPUXO DESEJADO	35
3.3	CONTROLE POR RASTREAMENTO GEOMÉTRICO	36
3.4	AJUSTE DOS GANHOS DOS CONTROLADORES	39
3.4.1	Ganhos do Controlador para Pequenos Ângulos	39
3.4.2	Ganhos de quaisquer controladores	40
4	PLANEJAMENTO DE TRAJETÓRIA	42
5	AMBIENTE DE SIMULAÇÕES EM MATLAB	49
6	RESULTADOS	52
6.1	CONTROLE PARA PEQUENOS ÂNGULOS	52
6.1.1	Trajetória 1	52
6.1.2	Trajetória 2	56
6.1.3	Trajetória 3	58
6.2	CONTROLE DO VETOR EMPUXO DESEJADO	61

6.2.1	Trajectoria 1	61
6.2.2	Trajectoria 2	64
6.2.3	Trajectoria 3	67
6.3	CONTROLE POR RASTREAMENTO GEOMÉTRICO	69
6.3.1	Trajectoria 1	69
6.3.2	Trajectoria 2	72
6.3.3	Trajectoria 3	75
6.4	COMPARAÇÃO ENTRE OS CONTROLES UTILIZADOS	77
6.5	TEMPO DE PROCESSAMENTO	78
7	CONCLUSÕES	79
8	TRABALHOS FUTUROS	80
	REFERÊNCIAS	81
	APÊNDICE A – Planejamento de Trajetória para n Pontos .	84

1 INTRODUÇÃO

O rápido desenvolvimento de sensores, baterias e sistemas embarcados fazem possível o estudo de Veículos Aéreos Não-Tripulados (VANTs). De acordo com o Departamento de Controle do Espaço Aéreo, VANT é todo e qualquer tipo de aeronave que pode ser controlada nos 3 eixos e que não necessite de pilotos embarcados para ser guiada [1]. Nessa área, os pequenos sistemas de Decolagem e Aterrissagem Vertical (VTOL, em inglês) são bastante estudados por possuírem pequenas dimensões e serem versáteis, sendo ideais para tarefas de monitoramento e avaliação em locais de difícil acesso.

Atualmente, uma das questões mais pesquisadas na área de VANTs é o controle autônomo (capacidade de tomar decisões com relação a vários aspectos, como planejamento de rota, seguimento de rota e operação conjunta com vários VANTs, sem intervenção humana), pois podem-se desenvolver técnicas de otimização de tempo de percurso, consumo de energia e segurança de voo, fatores fundamentais para suas principais aplicações. Mas, para que se possa desenvolver controles de alto nível, devemos entender a fundo a modelagem do sistema e seus controles de baixo e alto nível: controle de orientação e controle de posição.

Para a modelagem matemática de aeronaves, existem duas abordagens [2]: desenvolver equações físicas que representam o sistema ou utilizar técnicas de identificação de sistemas. A última considera o sistema como uma caixa-preta, para estimar o modelo dinâmico com base nos dados de excitação e de resposta da planta. Ambas podem ser utilizadas em conjunto, para que se possa simplificar o problema.

O presente trabalho foca na modelagem por equações físicas de um quadricóptero, no desenvolvimento de estratégias de controle, além do desenvolvimento de uma plataforma de simulação gráfica em MATLAB. Escolheu-se tal sistema pelo seu baixo custo de produção, design simples e grande agilidade em manobras.

O sistema de controle de um quadricóptero deve controlar os seis graus de liberdade (DoF, em inglês) do sistema, apesar deste ser considerado um sistema sub-atuado. De acordo com [3], um sistema robótico é classificado como sub-atuado se um sistema de controle não pode produzir acelerações arbitrárias em algum dos seus graus de liberdade em um determinado instante no tempo. Como um quadricóptero possui seis graus de liberdade e apenas quatro entradas, o controle das variáveis não atuadas/passivas é dado pela dinâmica acoplada entre elas e as variáveis atuadas/ativas (que estão diretamente associadas ao atuador) [2].

O Capítulo 2 apresenta os conceitos necessários para a modelar as equações diferenciais da dinâmica da planta. O Capítulo 3 apresenta as equações para diversos tipos de controladores. O Capítulo 4 expõe um método de planejamento de trajetória ótima. O Capítulo 5 apresenta um ambiente de simulação gráfica desenvolvido em MatLab.

1.1 APRESENTAÇÃO DO SISTEMA

Quadricóptero é um veículo aéreo de asa rotativa que possui quatro conjuntos motor-hélice, também chamados de rotores, apontados para cima, os quais devem possuir a capacidade de criar um somatório de empuxos com módulo maior do que o módulo do vetor peso do sistema. Além disso, o empuxo gerado por esses conjuntos deve ser independente, para que se possa criar torque sobre todos os eixos do corpo do sistema. Na Figura 1 podemos ver um modelo de quadricóptero comercial.



Figura 1 – Quadricóptero

Para que se possa desenvolver o controle dessa planta, faz-se necessária uma malha de *feedback* contendo sinais de diferentes tipos de sensores, como acelerômetro, giroscópio, magnetômetro, barômetro e GPS. Cada sensor é crucial em diferentes etapas das malhas de controle. Como exemplo, para a malha de controle de orientação, pode-se utilizar os dados do acelerômetro e do giroscópio. Já para a malha de controle de posição, pode-se utilizar os dados do GPS e do barômetro.

1.2 REVISÃO BIBLIOGRÁFICA

A presente seção irá abordar uma breve introdução histórica e momentânea de VANTs, tal qual atuais linhas de pesquisa na área.

1.2.1 Classificação

A presente subseção será destinada a diferentes classificações de veículos aéreos.

Nos últimos anos o mercado de VANTs vem crescendo de forma exponencial, tanto no Brasil como no mundo [4]. Com toda essa exposição, as dúvidas e confusões envolvendo a nomenclatura dos equipamentos são compreensíveis.

Topologicamente, veículos aéreos podem ser classificados em diversos grupos. Duas das topologias mais comumente utilizadas são:

- Aeronaves com Asa Fixa (*Fixed-Wing Aircraft*): É capaz de voar usando asas que geram *lift* causada pela velocidade do avanço do veículo e pelo formato das asas. O *lift* é criado pelo fluxo de ar sobre um perfil aerodinâmico. A forma de um perfil aerodinâmico faz com que o ar flua mais rápido na parte de cima da asa do que embaixo, logo o rápido fluxo de ar diminui a pressão do ar circundante. Como a pressão do ar é maior abaixo do perfil aerodinâmico do que acima, é criada uma força de *lift* resultante. O principal exemplo para esse tipo de aeronave é o avião.
- Aeronaves com Asa Rotativa (*Rotary-Wing Aircraft*): É capaz de voar utilizando asas rotativas capazes de gerar *lift*. Tais asas giram em torno de um eixo, criando uma diferença de pressão no perfil aerodinâmico, assim como ocorre nas Aeronaves com Asa Fixa. Um eixo de motor em conjunto com asas rotativas é referido como rotor. Os principais exemplos para esse tipo de aeronave são o helicóptero e o quadricóptero.

Em termos legais, uma das principais dúvidas é relacionada ao uso do termo informal “drone”. Tal termo, cunhado nos Estados Unidos, designa todo e qualquer objeto voador não tripulado, seja ele de qualquer propósito, origem ou característica. Ou seja, é um termo genérico, sem amparo técnico ou definição na legislação [5].

No Brasil, segundo a legislação pertinente, caracteriza-se como VANT toda aeronave projetada para operar sem piloto a bordo. Porém, tal aeronave deve ter caráter não recreativo e possuir carga útil embarcada. Legalmente, nem todo “drone” pode ser considerado um VANT, uma vez que um Veículo Aéreo Não Tripulado utilizado como hobby ou esporte enquadra-se na legislação pertinente aos aeromodelos [5], e não na legislação pertinente a “Sistemas de Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro”. Além disso, a legislação citada distingue VANTs em diferentes categorias [1], dentre elas:

- Aeronave Remotamente Pilotada: Aeronave não tripulada pilotada a partir de uma estação de pilotagem remota, utilizada com propósitos não recreativos.
- Aeronave Não Tripulada Automática: Aeronave não tripulada que possibilita a intervenção do piloto, a qualquer momento, na condução e no gerenciamento do voo, mesmo tendo todos os parâmetros e os perfis de voos conduzidos por sistemas computacionais.
- Aeronave Não Tripulada Autônoma: Aeronave não tripulada que não possibilita a intervenção do piloto na condução e no gerenciamento do voo. Atualmente, o voo de tais aeronaves não é autorizado.

1.2.2 Linhas de pesquisa

A presente subseção será destinada a apresentar algumas linhas de pesquisa para o desenvolvimento de quadricópteros, e foi redigida com base em informações retiradas de [6].

O desenvolvimento de Veículos Aéreos Não Tripulados Autônomos capazes de realizar tarefas de forma confiável deve apresentar um controle de baixo nível robustos e serem capazes de perceber o ambiente e atuarem de acordo com a informação obtida. Como essas etapas não são triviais, existe bastante estudo relacionados a cada um desses temas. A Subseção 1.2.2.1 apresenta pesquisas relacionadas a diversos controladores de quadricópteros. A Subseção 1.2.2.2 apresenta algumas pesquisas relacionadas a Planejamento de Trajetórias. Por fim, A Subseção 1.2.2.3 apresenta algumas pesquisas relacionadas a Visão Computacional.

1.2.2.1 Controle

A presente subseção será destinada a apresentar algumas pesquisas na área de controle.

Boa parte do projeto de um Veículo Aéreo Não-Tripulado diz respeito ao seu sistema de controle. Estabilidade de voo, manobras, decolagem e pouso são dependentes de um bom projeto de controladores de voo.

A identificação da dinâmica de voo com fidelidade suficiente para ser utilizado no desenvolvimento do controlador é um dos principais desafios enfrentados. Os modelos matemáticos, os quais são apenas uma aproximação da dinâmica do veículo, acabam sendo utilizados para descrever uma dinâmica complexa do mundo real [7]. As características não lineares e multivariáveis fazem com que o quadricóptero seja um sistema difícil de se controlar [8].

Diversos controladores já foram propostos para esse sistema, dentre eles: técnicas de controle clássica como PD [9] e PID [10, 11], técnicas de controle ótimo como LQR [12] e LQG [13], técnicas de controle não linear como Lyapunov [14] e *Backstepping* [9, 14, 15, 16] e técnicas de controle inteligente e adaptativo como Fuzzy [8, 17], Redes Neurais [18] e Reinforcement Learning [19].

Na Universidade Federal de Juiz de Fora algumas pesquisas desse cunho já foram desenvolvidas, como [6, 20]. O presente trabalho utilizará como base principalmente o desenvolvimento descrito em [21, 22, 23]. Além disso, parte do presente trabalho foi apresentado no congresso *International Federation of Automatic Control 2017* (IFAC 2017) [24], que ocorreu na cidade de Toulouse, França.

1.2.2.2 *Planejamento de Trajetória*

A presente subseção será destinada a apresentar algumas pesquisas na área de planejamento de trajetória.

Uma das principais tarefas que permite a utilização de um veículo autônomo é o desenvolvimento de algoritmos para o planejamento para a movimentação do veículo. Dado um conjunto de pontos pré-definidos possíveis no espaço de navegação do sistema, o veículo deve conseguir determinar uma maneira de percorrer todos esses pontos, de modo que restrições do ambiente, como obstáculos, e outras restrições como tempo gasto, energia necessária e restrições ligadas ao movimento da aeronave, sejam devidamente atendidas. Existem diferentes técnicas na literatura para realizar o planejamento de trajetória de veículos aéreos. Algumas delas podem ser vistas em [25, 26]. O presente trabalho utilizará como base o desenvolvimento descrito em [21, 27].

1.2.2.3 *Visão Computacional*

A presente subseção será destinada a apresentar algumas pesquisas na área de visão computacional.

A visão computacional é uma ferramenta que está cada vez mais ganhando espaço na robótica. A captura e o processamento de imagens podem ser utilizados em diversas aplicações, como identificação de objetos, monitoramento e inspeção. Além disso, pode trabalhar em conjunto com algoritmos de mapeamento e de localização. Alguns trabalhos nesta área são [28, 29].

2 MODELAGEM DO SISTEMA

Neste capítulo, as equações matemáticas que regem a cinemática e a dinâmica de um quadricóptero são apresentadas na Subseção 2.2. Para o desenvolvimento de tais equações, o conceito de *frame* é apresentado na Seção 2.1. Os principais *frames* utilizados no desenvolvimento das equações presentes nesse trabalho são apresentados nas Subseções 2.1.1.1 e 2.1.1.2. Equações de mudança de *frames* são apresentadas na Subseção 2.1.2.

2.1 FRAMES

Essa seção descreverá os diferentes *frames* de referência e sistemas de coordenadas que são utilizados para descrever a posição e orientação da aeronave.

Em álgebra linear, um *frame* de um Espaço com Produto Interno é uma generalização do conjunto de vetores base de um espaço vetorial, de forma que se possa representar conjuntos de maneira linearmente dependentes. Logo, os *frames* se tornam bases de diferentes sistemas de coordenadas; no presente trabalho, tais bases são consideradas ortonormais (vetores base perpendiculares e normalizados). Entretanto, é importante frisar que nem todos os *frames* descritos na literatura são ortonormais, como mostrado em [30]. Como descrito em [31], tal abordagem sempre possuiu ampla utilização em processamento de sinais e imagens, séries não-harmônicas de Fourier, compressão de dados e na teoria de amostragem; porém atualmente passou a ser amplamente utilizada em matemática aplicada, física, engenharia e ciências da computação.

A representação do modelo do sistema a partir de diferentes *frames* é necessária, pois todas equações podem ser descritas no corpo rígido do quadricóptero; porém é usual medir seu deslocamento em relação a um *frame* inercial, fixado na superfície da Terra [32]. Além disso, diferentes sensores medem grandezas em diferentes *frames*. Enquanto o GPS fornece uma medida com relação à um *frame* inercial, o acelerômetro e o giroscópio fornecem medidas com relação ao *frame* do corpo rígido do quadricóptero. Tais *frames* serão apresentados formalmente na Subseção 2.1.1.2.

Existem duas possíveis operações básicas, as quais permitem transformar um conjunto de dados referidos a um sistema de coordenadas para outro: rotação e translação. Sendo parte crucial nas equações desse sistema, as matrizes de rotação e transformação de velocidades angulares (ambas relacionadas com rotações em corpos rígidos) serão apresentadas na Subseção 2.1.2.

2.1.1 Principais *frames*

A presente subseção apresentará os principais *frames* utilizados na modelagem e no controle de quadricópteros e de aeronaves em geral.

2.1.1.1 Principais frames utilizados para quaisquer veículos aéreos

A atual subseção apresentará os principais *frames* que são utilizados nas equações de modelagem e controle de quaisquer veículos aéreos.

Definir os *frames* com sabedoria é de extrema importância para que se possa desenvolver o restante do projeto. Como dito na Seção 2.1, uma das principais motivações para se definir diferentes *frames* em robótica é o fato de que diferentes equações são mais facilmente desenvolvidas a partir de referenciais específicos. Além disso, diferentes sensores podem efetuar medições a partir de diferentes referenciais [32].

São necessários alguns *frames* básicos para o desenvolvimento de equações da dinâmica de quaisquer veículos aéreos. De acordo com [33], os seguintes *frames* são os mais utilizados.

- *Earth-Centered Inertial (ECI) Frame*: é definido um *frame* inercial (Newtoniano), negligenciando o movimento orbital da Terra em volta do Sol, tal como sua rotação. É centrado no centro geométrico da Terra e possui seus vetores unitários: $\{\hat{i}\}^{ECI}$ alinhado com o equinócio de verão; $\{\hat{k}\}^{ECI}$ aponta para o pólo norte, alinhado com o eixo de rotação da Terra; e $\{\hat{j}\}^{ECI}$ completa a Regra da Mão Direita.
- *Earth-Centered-Earth-Fixed (ECEF) Frame*: fixado no centro geométrico da Terra e rotaciona com ela. Possui seus vetores unitários: $\{\hat{i}\}^{ECEF}$ aponta para a interseção do Meridiano de Greenwich com a Linha do Equador; $\{\hat{k}\}^{ECEF}$ aponta para o pólo norte, alinhado com o eixo de rotação da Terra; e $\{\hat{j}\}^{ECEF}$ completa a Regra da Mão Direita.
- *Vehicle Body-Fixed (B) Frame*: possui origem localizada no centro de massa instantâneo do veículo. Possui seus vetores unitários: $\{\hat{i}\}^B$ aponta em direção ao nariz do veículo; $\{\hat{j}\}^B$ aponta para lateral direita do veículo, quando visto por trás; e $\{\hat{k}\}^B$ aponta para baixo.
- *Vehicle Reference Coordinate System (R)*: devido à mudança do centro de massa do veículo, tal *frame* é desenvolvido para que se possa localizar diversos itens presentes no veículo, tais como localização acelerômetros, centro de referência do momento aerodinâmico e localização do centro de massa. Esse *frame* é usualmente centrado no nariz da aeronave, e seus vetores unitários $\{\hat{i}\}^R$ e $\{\hat{k}\}^R$ possuem sentidos opostos aos descritos pelo *Vehicle Body-Fixed Frame*.
- *Local Geocentric North-East-Down Coordinates (C)*: possui sua origem na superfície da Terra, ao longo da reta que atravessa o centro do planeta e a aeronave. Seus vetores unitários $\{\hat{i}\}^C$, $\{\hat{j}\}^C$ e $\{\hat{k}\}^C$ apontam para o norte, para o leste e para o centro da terra, respectivamente.

- *Local Geodetic North-East-Down Coordinates* (N): possui origem na superfície da Terra, ao longo da reta normal que passa pelo planeta e pela aeronave. Seus vetores unitários $\{\hat{i}\}^N$, $\{\hat{j}\}^N$ e $\{\hat{k}\}^N$ apontam para o norte, para o leste e para o centro da terra, respectivamente.
- *Wind* (W) *Frame*: possui origem coincidente com o centro de massa. Seu vetor unitário $\{\hat{i}\}^W$ aponta na direção do vetor velocidade entre o *Frame* B e o *Frame* W, e sendo referenciado no *Frame* ECEF.
- *Velocity* (V) *Frame* (ou *Flight Path Frame*): possui origem na superfície da Terra, ao longo da reta normal que passa pelo planeta e pela aeronave. Seu vetor unitário $\{\hat{i}\}^V$ aponta na direção do vetor velocidade entre o *Frame* B e o *Frame* ECEF, e sendo referenciado no *Frame* B.
- *Propulsion (engine) Coordinates* (P): possui origem no ponto de aplicação do empuxo (motor). Seu vetor unitário $\{\hat{i}\}^P$ aponta na direção do vetor empuxo de tal forma que todo o empuxo do motor é representado apenas no eixo x desse *frame*.

2.1.1.2 Principais frames utilizados para quadricópteros

Essa subseção apresentará os principais *frames* que serão utilizados nas equações de modelagem e controle de um quadricóptero apresentados no presente trabalho.

Uma grande variedade de *frames* foi apresentada na Subseção 2.1.1.1, porém vários efeitos que ocorrem em grandes aeronaves podem ser desconsiderados quando desenvolvemos a modelagem de pequenos quadricópteros, como por exemplo a necessidade de considerar *frame* referente ao centro da Terra como inercial. Devido ao pequeno deslocamento do sistema, pode-se considerar um ponto na superfície da Terra como inercial. Além disso, alguns *frames*, como o que acompanha todas as gerações de empuxo do sistema, podem ser ocultados, uma vez que, num quadricóptero, tais *frames* estão intrinsecamente acoplados ao *frame* do corpo rígido. Tais *frames* foram evidenciados por existirem diversas aeronaves capazes de modificar a orientação do vetor empuxo com relação ao corpo rígido. Foguetes, por exemplo, podem possuir *movable nozzles*, os quais fornecem um método de controle do vetor empuxo para o veículo [34].

Para um quadricóptero, é necessário considerar apenas três *frames* principais. Eles estão listados abaixo e ilustrados na Figura 2.

- *Frame* Inercial (I): usualmente centrado na superfície terrestre, com a origem definida no local de partida. Muitas vezes seus vetores unitários $\{\hat{i}\}^I$, $\{\hat{j}\}^I$ e $\{\hat{k}\}^I$ apontam para o norte, para o leste e para o centro da terra, respectivamente. Tal abordagem recebe o nome de Sistema *North-East-Down* (NED). Outra abordagem bastante utilizada é chamada de Sistema *East-North-Up* (ENU), o qual possui seus vetores

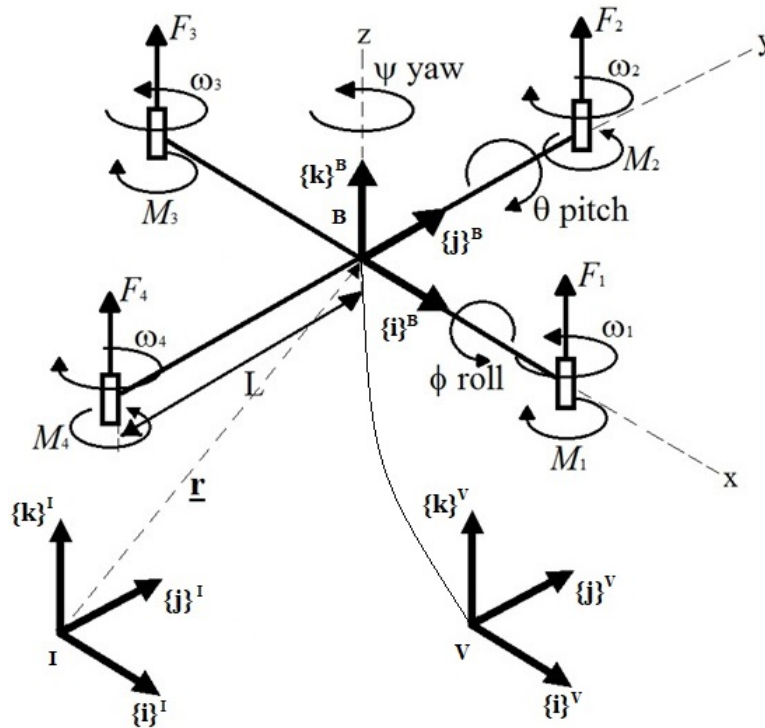


Figura 2 – Principais frames em um quadricóptero

unitários $\{\hat{i}\}^I$, $\{\hat{j}\}^I$ e $\{\hat{k}\}^I$ apontando para o leste, para o norte e em direção, porém com sentido oposto, ao centro da terra, respectivamente. O presente trabalho utiliza o sistema ENU.

- *Frame* do Veículo (V): referência com vetores unitários paralelos aos vetores unitários do *Frame* Inercial, porém com origem no centro geométrico do sistema.
- *Frame* do Corpo Rígido (B): referência acoplada ao sistema, com origem no seu centro geométrico. No presente trabalho definiu-se seus vetores unitários $\{\hat{i}\}^B$ e $\{\hat{j}\}^B$ na direção de dois braços do quadricóptero perpendiculares entre si, de forma que o produto vetorial $\{\hat{i}\}^B \times \{\hat{j}\}^B$, quando o sistema estiver em seu estado de flutuação, forme um vetor unitário $\{\hat{k}\}^B$ com direção, porém sentido oposto, ao centro da terra.

2.1.2 Matrizes de rotação e transformação

Na presente seção será apresentado o conceito de rotação de *frames*.

Rotação é uma operação de mudança de orientação dos eixos cartesianos de um *frame* com relação a outro. A matriz de rotação que representa a mudança de orientação do *frame* 0 ao *frame* 1 pode ser escrita como R_0^1 . Convenciona-se o sentido positivo de rotação seguindo a Regra da Mão Direita.

Dado dois *frames* F^0 e F^1 , pode-se derivar as matrizes de rotação nos três eixos. Considere a rotação entre ambos *frames* igual a θ em torno do eixo z , como mostrado na Figura 3.

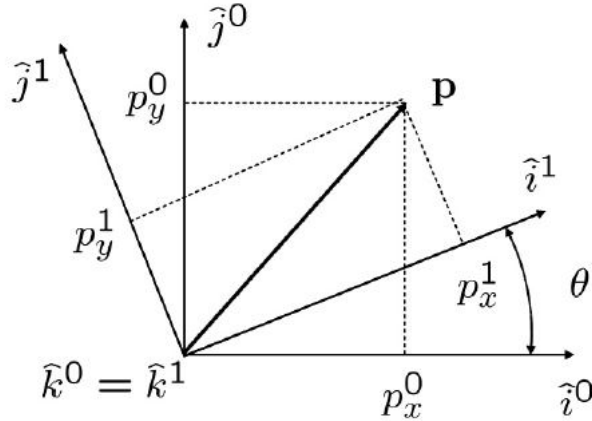


Figura 3 – Rotação em 2D

Fonte: [32]

O ponto \mathbf{p} pode ser expresso com relação aos eixos cartesianos de ambos *frames*.

$$\{p\}^0 = \{p_x\}^0 \hat{i}^0 + \{p_y\}^0 \hat{j}^0 + \{p_z\}^0 \hat{k}^0$$

$$\{p\}^1 = \{p_x\}^1 \hat{i}^1 + \{p_y\}^1 \hat{j}^1 + \{p_z\}^1 \hat{k}^1$$

Como são duas formas de representar o mesmo ponto, pode-se igualar as duas expressões. Efetuando tal operação, tem-se:

$$\{p_x\}^0 \hat{i}^0 + \{p_y\}^0 \hat{j}^0 + \{p_z\}^0 \hat{k}^0 = \{p_x\}^1 \hat{i}^1 + \{p_y\}^1 \hat{j}^1 + \{p_z\}^1 \hat{k}^1 \quad (2.1)$$

Aplicando produto escalar do vetor

$$\begin{bmatrix} \hat{i}^0 \\ \hat{j}^0 \\ \hat{k}^0 \end{bmatrix}$$

em ambos os lados da Equação 2.1 e reescrevendo o resultado na forma matricial, tem-se:

$$\{\vec{p}\}^0 = \begin{bmatrix} \hat{i}^0 \cdot \hat{i}^1 & \hat{i}^0 \cdot \hat{j}^1 & \hat{i}^0 \cdot \hat{k}^1 \\ \hat{j}^0 \cdot \hat{i}^1 & \hat{j}^0 \cdot \hat{j}^1 & \hat{j}^0 \cdot \hat{k}^1 \\ \hat{k}^0 \cdot \hat{i}^1 & \hat{k}^0 \cdot \hat{j}^1 & \hat{k}^0 \cdot \hat{k}^1 \end{bmatrix} \{\vec{p}\}^1 \quad (2.2)$$

Resolvendo todos os produtos internos da matriz encontrada na Equação 2.2 por meio da geometria da Figura 3, tem-se

$$\{\vec{p}\}^0 = R_1^0 \{\vec{p}\}^1 \quad (2.3)$$

onde

$$R_1^0 \triangleq \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

A matriz desenvolvida na Equação 2.4 é chamada de Matriz de Rotação, e está relacionada a uma rotação em torno do eixo z em θ radianos. De forma análoga, pode-se calcular Matrizes de Rotação relacionadas a rotações em torno dos dois eixos restantes. Para uma rotação em torno do eixo x , tem-se:

$$R_1^0 \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} \quad (2.5)$$

Enquanto que uma rotação em torno do eixo y resulta em:

$$R_1^0 \triangleq \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad (2.6)$$

Como descrito na Subseção 2.1.1.2, o quadricóptero possui três *frames* principais: o *Frame* Inercial, o *Frame* do Veículo e o *Frame* do Corpo Rígido. Para que se possa representar um vetor, o qual está referenciado no *Frame* do Corpo Rígido, no *Frame* do Veículo, deve-se desenvolver *frames* intermediários auxiliares. Tal desenvolvimento pode ser efetuado de diversas formas, as quais resultam em diferentes matrizes de rotação em torno de três eixos. Para o presente trabalho, escolheu-se a forma Z-X-Y.

- *Frame* Auxiliar 1 (A1): com origem no centro geométrico do sistema. Seus vetores unitários apresentam uma rotação em torno do eixo y de θ radianos em relação ao *frame* B.
- *Frame* Auxiliar 2 (A2): com origem no centro geométrico do sistema. Seus vetores unitários apresentam uma rotação em torno do eixo x de ϕ radianos em relação ao *frame* A1.

Além disso, considera-se que para a obtenção do *frame* V é necessária uma rotação em torno do eixo z de ψ radianos em relação ao *frame* A2. A partir do que foi descrito nesse capítulo, as matrizes de rotação que serão utilizadas no presente trabalho podem ser escritas. A rotação que transforma a representação de um vetor no *frame* A1 ao *frame* A2, a qual ocorre em torno do eixo x , pode ser escrita como:

$$R_{A1}^{A2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}.$$

Já a rotação que transforma a representação de um vetor no *frame* B ao *frame* A1, a qual ocorre em torno do eixo y , pode ser escrita como:

$$R_B^{A1} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}.$$

Finalmente, a rotação que transforma a representação de um vetor no *frame* A2 ao *frame* V, a qual ocorre em torno do eixo z , pode ser escrita como:

$$R_{A2}^V = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A matriz de rotação em torno de três eixos correspondente à forma Z-X-Y pode ser observada na Equação 2.7.

$$R_{Bz-x-y}^V = R_{A2}^V(\psi)R_{A1}^{A2}(\phi)R_B^{A1}(\theta)$$

$$R_{Bz-x-y}^V \triangleq \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\theta s\phi c\psi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \quad (2.7)$$

Os ângulos presentes na matriz de rotação da Equação 2.7 possuem nomes especiais. O ângulo de rotação em torno do eixo x do quadricóptero, representado na matriz R_{Bz-x-y}^V por meio da letra grega ϕ , também é chamado de *roll angle* (ângulo de rolagem). Já o ângulo de rotação em torno do eixo y , representado por meio da letra grega θ , é chamado de *pitch angle* (ângulo de arfagem). Por fim, o ângulo de rotação em torno do eixo z , representado por meio da letra grega ψ , é chamado de *yaw angle* (ângulo de guinada). Tal nomenclatura é vastamente utilizada na área.

Vale ressaltar que a matriz de rotação que relaciona o *Frame* Inercial com o *Frame* do Veículo

$$R_V^I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

uma vez que o *Frame* do Veículo foi considerado sempre paralelo ao *Frame* Inercial (Subseção 2.1.1.2).

Por fim, a relação entre a derivada dos ângulos ϕ , θ e ψ e as velocidades angulares medidas por um sensor fixo ao *Frame* do Corpo Rígido não é direta (Equação 2.8). Considerando p , q e r as velocidades angulares em torno dos eixos x , y e z do *Frame* do Corpo Rígido, respectivamente, pode-se estabelecer tal relação representando as taxas angulares $\dot{\phi}$, $\dot{\theta}$ e $\dot{\psi}$ no mesmo *frame* em que as velocidades angulares p , q e r estão representadas: *Frame* B. Essa operação está explicitada na Equação 2.9.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \neq \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.8)$$

$$\begin{Bmatrix} p \\ q \\ r \end{Bmatrix}^B = I \begin{Bmatrix} 0 \\ \dot{\theta} \\ 0 \end{Bmatrix}^B + R_{A_1}^B \begin{Bmatrix} \dot{\phi} \\ 0 \\ 0 \end{Bmatrix}^{A_1} + R_{A_1}^B R_{A_2}^{A_1} \begin{Bmatrix} 0 \\ 0 \\ \dot{\psi} \end{Bmatrix}^{A_2} \quad (2.9)$$

Resolvendo a Equação 2.9, tem-se:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = I \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + (R_B^{A_1})^{-1} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + (R_B^{A_1})^{-1} (R_{A_2}^{A_1})^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}^{-1} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} c\theta & s\phi s\theta & -c\phi s\theta \\ 0 & c\phi & s\phi \\ s\theta & -c\theta s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} c\theta \\ 0 \\ \dot{\phi} s\theta \end{bmatrix} + \begin{bmatrix} -\dot{\psi} c\phi s\theta \\ \dot{\psi} s\phi \\ \dot{\psi} c\phi c\theta \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.10)$$

A matriz T apresentada na Equação 2.10 transforma a taxa de variação dos ângulos *roll*, *pitch* e *yaw* em velocidades angulares no *Frame* do Corpo Rígido. Essa é uma importante propriedade para a simulação de um veículo aéreo e para relacionarmos grandezas nos controladores presentes nesse trabalho.

É importante notar que

$$T^{-1} = \begin{bmatrix} c\theta & 0 & s\theta \\ \frac{s\phi s\theta}{c\phi} & 1 & -\frac{c\theta s\phi}{c\phi} \\ -\frac{s\theta}{c\phi} & 0 & \frac{c\theta}{c\phi} \end{bmatrix}$$

possui descontinuidades. Tais descontinuidades devem ser evitadas para um bom funcionamento do modelo descrito. Uma possível solução para o problema apresentado é a utilização de *Quaternions* para a representação das rotações sobre três eixos, como desenvolvido em [35, 36].

2.1.3 Equação de Coriolis

A presente subseção apresentará a Equação de Coriolis (também chamada de *Transport Theorem* ou *Basic Kinematic Equation*), a qual possui uma grande importância na modelagem de um quadricóptero. Existem diversas demonstrações matemáticas dessa equação. Utilizando a Fórmula de Rodriguez [37], por exemplo, pode-se chegar numa demonstração como formulado em [38]. Porém, neste trabalho, optou-se apresentar um caminho mais simples, descrito em [39].

Considere dois *frames* distintos. O primeiro é fixo e possui vetores unitários $\{\hat{i}\}^I$, $\{\hat{j}\}^I$, $\{\hat{k}\}^I$; enquanto que o segundo rotaciona e possui vetores unitários $\{\hat{i}\}^B$, $\{\hat{j}\}^B$, $\{\hat{k}\}^B$. Primeiramente, deve-se estabelecer que dado um vetor $\vec{r} = (\hat{i}, \hat{j}, \hat{k})$, sua derivada pode ser escrita como

$$\frac{d\vec{r}}{dt} = \vec{\Omega} \times \vec{r}, \quad (2.11)$$

onde $\vec{\Omega}$ representa a velocidade angular. Escrevendo \vec{r} em termos de suas componentes x , y e z , tem-se que

$$\frac{d\vec{r}}{dt} \rightarrow \frac{d\hat{i}}{dt} = \vec{\Omega} \times \hat{i}, \quad \frac{d\hat{j}}{dt} = \vec{\Omega} \times \hat{j}, \quad \frac{d\hat{k}}{dt} = \vec{\Omega} \times \hat{k}. \quad (2.12)$$

Agora, considere um vetor \vec{a} , o qual está escrito referenciado no *frame* rotacional B como

$$\vec{a} = a_x \{\hat{i}\}^B + a_y \{\hat{j}\}^B + a_z \{\hat{k}\}^B. \quad (2.13)$$

Escrevendo a taxa de variação desse vetor com relação ao *frame* inercial,

$$\left\{ \frac{d\vec{a}}{dt} \right\}^I = \frac{d}{dt}(a_x \{\hat{i}\}^B) \frac{d}{dt}(a_y \{\hat{j}\}^B) + \frac{d}{dt}(a_z \{\hat{k}\}^B). \quad (2.14)$$

Aplicando a Regra da Derivada do Produto em todos os termos, tem-se que

$$\begin{aligned} \left\{ \frac{d\vec{a}}{dt} \right\}^I &= \frac{da_x}{dt} \{\hat{i}\}^B + \frac{da_y}{dt} \{\hat{j}\}^B + \frac{da_z}{dt} \{\hat{k}\}^B + a_x \frac{d\{\hat{i}\}^B}{dt} + a_y \frac{d\{\hat{j}\}^B}{dt} + a_z \frac{d\{\hat{k}\}^B}{dt} \\ \left\{ \frac{d\vec{a}}{dt} \right\}^I &= \left\{ \frac{d\vec{a}}{dt} \right\}^B + a_x \frac{d\{\hat{i}\}^B}{dt} + a_y \frac{d\{\hat{j}\}^B}{dt} + a_z \frac{d\{\hat{k}\}^B}{dt} \end{aligned} \quad (2.15)$$

Porém, pode-se utilizar o que foi descrito na Equação 2.12 para resolver o restante da Equação 2.15. O resultado é apresentado a seguir.

$$\begin{aligned} \left\{ \frac{d\vec{a}}{dt} \right\}^I &= \left\{ \frac{d\vec{a}}{dt} \right\}^B + a_x \vec{\Omega} \times \{\hat{i}\}^B + a_y \vec{\Omega} \times \{\hat{j}\}^B + a_z \vec{\Omega} \times \{\hat{k}\}^B \\ \left\{ \frac{d\vec{a}}{dt} \right\}^I &= \left\{ \frac{d\vec{a}}{dt} \right\}^B + \{\vec{\Omega}\}^B \times \{\vec{a}\}^B \end{aligned} \quad (2.16)$$

Onde $\{\vec{\Omega}\}^B$ representa a velocidade angular do corpo, referenciado no *frame* que está rotacionando.

2.2 EQUAÇÕES DE NEWTON-EULER PARA MODELAGEM DE UM QUADRICÓPTERO

Com base nos *frames* definidos na Subseção 2.1.1.2 e explicitados na Figura 2 e utilizando as matrizes de rotação R e de transformação T definidas na Sessão 2.1.2, pode-se obter as equações do modelo dinâmico do sistema. Considerando que o corpo é simétrico com relação aos planos x - z e y - z do *frame* do corpo e que as únicas forças atuantes sobre ele são o peso (\vec{P}) e os quatro empuxos provenientes dos conjuntos motor-hélice (\vec{E}), pode-se descrever sua aceleração linear resultante com relação ao *Frame* Inercial por meio da Segunda Lei de Newton:

$$m\{\ddot{\mathbf{r}}\}^I = \{\vec{P}\}^I + \{\vec{E}\}^I$$

$$m\{\ddot{\mathbf{r}}\}^I = \{\vec{P}\}^I + R_B^I \{\vec{E}\}^B \quad (2.17)$$

$$m\{\ddot{\mathbf{r}}\}^I = \left\{ \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \right\}^I + R_B^I \left\{ \begin{bmatrix} 0 \\ 0 \\ \sum F_i \end{bmatrix} \right\}^B \quad (2.18)$$

Considera-se L o tamanho do braço do quadricóptero (Figura 2) e I a matriz de inércia apresentada na Equação 2.19, considerando simetria com relação à massa do sistema, observado a partir do centro geométrico do corpo.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.19)$$

Em virtude das forças produzidas pelos rotores, são produzidos momentos no corpo rígido (LF_i), causando a rotação do sistema em torno dos eixos x e y . A rotação em torno do eixo z se dá pelo torque criado pelo giro dos motores, os quais estão fixados à planta. Com base na Equação de Coriolis apresentada na Seção 2.1.3, a equação que descreve a aceleração angular no *Frame* do Corpo Rígido pode ser escrita como mostrado na Equação 2.20.

$$I \left\{ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \right\}^B = \left\{ \begin{bmatrix} L(F_2 - F_4) \\ L(F_1 - F_3) \\ M_1 + M_2 + M_3 + M_4 \end{bmatrix} \right\}^B - \left\{ \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\}^B \times I \left\{ \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\}^B \quad (2.20)$$

Para que se possa encontrar a taxa de variação dos ângulos *roll*, *pitch* e *yaw* da matriz de rotação R descrita na Equação 2.7, pode-se utilizar a matriz de transformação T , apresentada na Equação 2.10.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\phi & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

Na modelagem desenvolvida no presente capítulo, a dinâmica dos rotores foi ignorada, porém poderia ser considerada como em [21, 40].

3 CONTROLE DO SISTEMA

O presente capítulo será destinado a diferentes controladores para um quadricóptero. Todos os controladores terão duas etapas: controle de (*position control*) e controle de orientação (*attitude control*). Como representado na Figura 4, o controle de posição irá informar ao controle de orientação quais ângulos *roll*, *pitch* e *yaw* o quadricóptero deve apresentar para que possa perseguir a trajetória desejada.

Para o controle de posição, é necessário um vetor $\mathbf{r}_T(t)$ (Equação 3.1), tal como sua primeira e segunda derivadas, os quais, juntos, representam a trajetória desejada nos três eixos cartesianos e no ângulo *yaw*. O cálculo da trajetória ótima será desenvolvido no Capítulo 4. Por ora, considera-se que tal trajetória é sabida.

$$\mathbf{r}_T(t) = \begin{bmatrix} x_T(t) \\ y_T(t) \\ z_T(t) \\ \psi_T(t) \end{bmatrix} \quad (3.1)$$

Para que se possa controlar qualquer sistema, deve-se observar as equações que o descrevem. As equações que descrevem o movimento de um quadricóptero (2.18 e 2.20) possuem uma estrutura em cascata: o movimento rotacional é desacoplado do movimento de translação, porém o segundo é dependente da matriz de rotação R_B^I gerada pelo primeiro [41]. Pode-se perceber isso pela Equação 2.17: o vetor empuxo total \vec{E}^B , por estar intrinsecamente acoplado ao eixo *z* do *Frame* do Corpo Rígido, se relaciona apenas com a magnitude do empuxo. Sua direção é determinada pela matriz de rotação R_B^I , a qual transporta a representação do vetor \vec{E}^B para o *Frame* Inercial, gerando \vec{E}^I . Logo, percebe-se que a magnitude do vetor \vec{E}^B é diretamente controlável, porém não sua direção. Então, para que se altere a direção do empuxo total, os três ângulos que compõe a orientação do corpo rígido precisam ser alterados adequadamente.

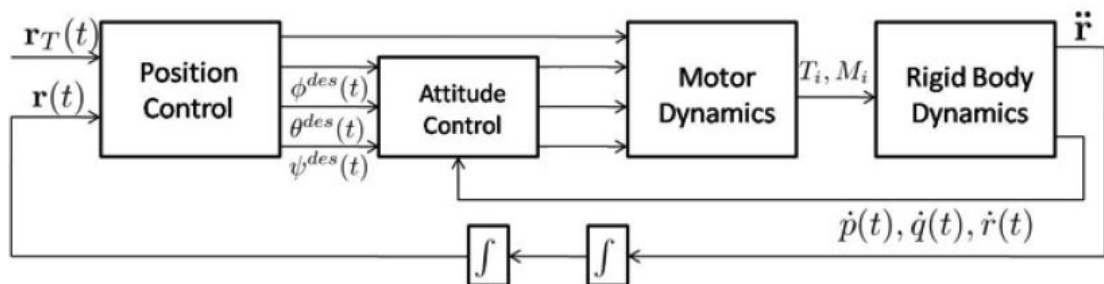


Figura 4 – Diagrama de Blocos do sistema

Para realizar tal controle, deve-se observar quais são as entradas do sistema em questão. Observando as Equações 2.18 e 2.20, percebe-se que o sistema possui quatro possíveis entradas: uma relacionada ao movimento linear e três, ao angular. Da Equação 2.18, temos que:

$$u_1 = \sum F_i \quad (3.2)$$

Explicitando as três entradas presentes na Equação 2.20 em um único vetor, temos:

$$\vec{u}_2 = \begin{bmatrix} L(F_2 - F_4) \\ L(F_1 - F_3) \\ M_1 + M_2 + M_3 + M_4 \end{bmatrix} \quad (3.3)$$

Logo, conclui-se que entrada u_1 é responsável pela magnitude do empuxo gerado pelos rotores, enquanto que $u_{2,1}$, $u_{2,2}$ e $u_{2,3}$ são responsáveis pela sua direção, cada qual atuando em um dos três ângulos.

3.1 CONTROLE PARA PEQUENOS ÂNGULOS

A atual seção será destinada para o desenvolvimento de equações para um controlador de pequenos ângulos [21]. Para isso, o sistema será linearizado próximo à sua estabilidade.

A primeira consideração para o desenvolvimento do presente controlador é que, durante a maior parte de seu voo, um quadricóptero opera próximo ao seu estado de flutuação (*hover state*). Logo, pode-se assumir pequenos ângulos para *roll* (ϕ) e *pitch* (θ) e, a partir disso, desenvolver as equações de um controlador para pequenos ângulos. Tais suposições implicam na linearização das funções seno e cosseno para os ângulos ϕ e θ . Assume-se também que o ângulo *yaw* (ψ) provém da trajetória planejada (Equação 3.1), e é representado por ψ_{des} . Logo, tem-se as matrizes R e T linearizadas apresentadas nas Equações 3.4 e 3.5.

$$R \approx \begin{bmatrix} c\psi_{des} - \phi\theta s\psi_{des} & -s\psi_{des} & \theta c\psi_{des} + \phi s\psi_{des} \\ s\psi_{des} + \phi\theta c\psi_{des} & c\psi_{des} & \theta s\psi_{des} - \phi c\psi_{des} \\ -\theta & \phi & 1 \end{bmatrix} \quad (3.4)$$

$$T = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\phi & 0 & c\phi c\theta \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & -\theta \\ 0 & 1 & \phi \\ \phi & 0 & 1 \end{bmatrix} \quad (3.5)$$

Além disso, assume-se que as velocidades angulares $\dot{\phi}$, $\dot{\theta}$ e $\dot{\psi}$ sejam muito pequenas, porém não desprezíveis. Substituindo na Equação 2.21 a matriz T encontrada na Equação 3.5:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & -\theta \\ 0 & 1 & \phi \\ \phi & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \theta\dot{\psi} \\ \dot{\theta} + \phi\dot{\psi} \\ \dot{\psi} + \phi\dot{\phi} \end{bmatrix} \quad (3.6)$$

Por fim, como o quadricóptero está em seu estado de flutuação, ele possui baixo movimento ao longo do eixo $\{z\}$ ¹. Para que isso ocorra, considera-se que a entrada u_1 seja aproximadamente igual à massa do quadricóptero vezes a gravidade. A partir de todas estas aproximações e considerando \ddot{x}_c , \ddot{y}_c e \ddot{z}_c acelerações de controle calculadas pelo controlador Proporcional-Derivativo (PD)

$$\begin{bmatrix} (\ddot{x}_{des} - \ddot{x}_c) \\ (\ddot{y}_{des} - \ddot{y}_c) \\ (\ddot{z}_{des} - \ddot{z}_c) \end{bmatrix} + \begin{bmatrix} K_{d,x}(\dot{x}_{des} - \dot{x}_c) \\ K_{d,y}(\dot{y}_{des} - \dot{y}_c) \\ K_{d,z}(\dot{z}_{des} - \dot{z}_c) \end{bmatrix} + \begin{bmatrix} K_{p,x}(x_{des} - x_c) \\ K_{p,y}(y_{des} - y_c) \\ K_{p,z}(z_{des} - z_c) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.7)$$

tem-se que, da Equação 2.18:

$$\phi_c = \frac{1}{g}(\ddot{x}_c \sin(\psi_{des}) - \ddot{y}_c \cos(\psi_{des})) \quad (3.8)$$

$$\theta_c = \frac{1}{g}(\ddot{x}_c \cos(\psi_{des}) + \ddot{y}_c \sin(\psi_{des})) \quad (3.9)$$

$$\psi_c = \psi_{des} \quad (3.10)$$

Possuindo os valores de \ddot{x}_c , \ddot{y}_c , \ddot{z}_c , ϕ_c , θ_c e ψ_c e realizando uma derivada discreta, em conjunto com a matriz T , para encontrar p_c , q_c e r_c , pode-se construir um controlador PD para o sistema.

$$u_1 = m(g + \ddot{z}_c) \quad (3.11)$$

$$\vec{u}_2 = \begin{bmatrix} K_{p,\phi}(\phi_c - \phi) + K_{d,\phi}(p_c - p) \\ K_{p,\theta}(\theta_c - \theta) + K_{d,\theta}(q_c - q) \\ K_{p,\psi}(\psi_c - \psi) + K_{d,\psi}(r_c - r) \end{bmatrix} \quad (3.12)$$

Combinando as Equações 3.6, 3.7, 3.8, 3.9, 3.10, 3.11 e 3.12, obtém-se o controlador para pequenos ângulos.

3.2 CONTROLE DO VETOR EMPUXO DESEJADO

A presente seção será destinada ao desenvolvimento de equações para o Controlador do Vetor Empuxo Desejado para um quadricóptero. Tal controlador foi desenvolvido com o objetivo de encontrar uma matriz R tal que

$$R \cdot \{\hat{k}\}^B = \frac{\vec{t}}{\|\vec{t}\|}. \quad (3.13)$$

Primeiramente, deve-se calcular um vetor \vec{t} , o qual representa a direção e magnitude do vetor empuxo (*thrust vector*) necessárias para que o sistema siga a trajetória desejada. Tal vetor pode ser escrito como

$$\vec{t} = m(\ddot{\vec{r}}_{des} + \vec{K}_{\dot{r}}\dot{\vec{r}} + \vec{K}_r\vec{e}_r + \{\vec{P}\}^I) \quad (3.14)$$

onde $\vec{K}_{\dot{r}}$ e \vec{K}_r representam as constantes do controlador, onde ambas possuem dimensões 3×1 , e $\{\vec{P}\}^I$ representa o vetor peso visto pelo *Frame* Inercial. A partir desse vetor, pode-se desenvolver Equação 3.13:

$$\begin{bmatrix} c\psi s\theta + c\theta s\theta s\psi \\ s\psi s\theta - c\theta s\theta c\psi \\ c\phi c\theta \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (3.15)$$

Sendo t_1 , t_2 e t_3 as componentes do vetor \vec{t} normalizado. Para que essa igualdade seja verdade, tem-se que

$$\phi_{des} = atan\left(\frac{t_1 s\psi - t_2 c\psi}{t_3}\right) \quad (3.16)$$

e

$$\theta_{des} = atan2\left(t_1 c\psi + t_2 s\psi, \frac{t_3}{c\phi}\right). \quad (3.17)$$

O ângulo ψ_{des} é calculado por meio do planejamento da trajetória. Com os três ângulos desejados, deve-se encontrar uma função de erro que será utilizada no controlador. Primeiramente, cria-se uma matriz R_{des} utilizando os três ângulos calculados e calcula-se uma matriz ΔR dada por

$$\Delta R = R_{des}^T \cdot R. \quad (3.18)$$

Depois, pode-se transformar a representação dessa rotação em torno de três eixos para a representação de uma rotação em torno de um eixo, descrito na Fórmula de Rodriguez [37]. Logo

$$\beta = \text{acos} \left(\frac{\Delta R_{(1,1)} + \Delta R_{(2,2)} + \Delta R_{(3,3)} - 1}{2} \right), \quad (3.19)$$

$$\vec{K} = \begin{bmatrix} \frac{\Delta R_{(3,2)} - \Delta R_{(2,3)}}{\sqrt{(\Delta R_{(3,2)} - \Delta R_{(2,3)})^2 + (\Delta R_{(1,3)} - \Delta R_{(3,1)})^2 + (\Delta R_{(2,1)} - \Delta R_{(1,2)})^2}} \\ \frac{\Delta R_{(1,3)} - \Delta R_{(3,1)}}{\sqrt{(\Delta R_{(3,2)} - \Delta R_{(2,3)})^2 + (\Delta R_{(1,3)} - \Delta R_{(3,1)})^2 + (\Delta R_{(2,1)} - \Delta R_{(1,2)})^2}} \\ \frac{\Delta R_{(2,1)} - \Delta R_{(1,2)}}{\sqrt{(\Delta R_{(3,2)} - \Delta R_{(2,3)})^2 + (\Delta R_{(1,3)} - \Delta R_{(3,1)})^2 + (\Delta R_{(2,1)} - \Delta R_{(1,2)})^2}} \end{bmatrix}. \quad (3.20)$$

O erro é dado por

$$\vec{e} = K \cdot \beta. \quad (3.21)$$

Então, as entradas do sistema podem ser escritas como

$$u_1 = \vec{t}^T \cdot \left(R \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \quad (3.22)$$

e

$$\vec{u}_2 = \vec{\Omega} \times I \cdot \vec{\Omega} + I \cdot (-K_R \cdot \vec{e} + K_\Omega \cdot \dot{\vec{e}}). \quad (3.23)$$

Onde I é a matriz de inércia do sistema e \vec{K}_R e \vec{K}_Ω representam constantes do controlador, ambas possuindo dimensões 3×1 .

3.3 CONTROLE POR RASTREAMENTO GEOMÉTRICO

A presente seção será destinada ao desenvolvimento de equações para o Controlador por Rastreamento Geométrico (*Geometric Tracking Control*) para um quadricóptero [22, 23]. Tal controlador possui atratividade exponencial quase global para o ponto de equilíbrio do sistema.

A base do presente controlador é encontrar uma maneira de atuar na magnitude do empuxo e nos vetores unitários do *Frame* do Corpo Rígido. Deve-se controlar a dinâmica translacional do quadricóptero por meio do cálculo do empuxo total \vec{E}^I . Para uma dada translação requerida $r_T(t)$, calcula-se o módulo empuxo total desejado $|\vec{E}^I|_{des}$ e a direção desejada para o terceiro vetor unitário do *Frame* do Corpo Rígido ($\{\hat{k}\}_{des}^B$), para que a dinâmica translacional seja estabilizada.

Uma vez que a direção desejada para o terceiro vetor unitário do *Frame* do Corpo Rígido é escolhido, existe um grau de liberdade restante a ser escolhido para a orientação desejada, o qual é correspondente à direção da frente do quadricóptero. Tal direção está presente no plano normal ao vetor $\{\hat{k}\}_{des}^B$, e é determinado pelo vetor unitário $\{\hat{i}\}_{des}^B$. Com isso, pode-se projetar uma matriz de rotação desejada

$$R_{des} = \begin{bmatrix} \{\hat{j}\}_{des}^B \times \{\hat{k}\}_{des}^B & \{\hat{j}\}_{des}^B & \{\hat{k}\}_{des}^B \end{bmatrix} \quad (3.24)$$

onde R_{des} representa a matriz de rotação R_B^I desejada e

$$\{\hat{j}\}_{des}^B = \frac{\{\hat{k}\}_{des}^B \times \{\hat{i}\}_{des}^B}{\|\{\hat{k}\}_{des}^B \times \{\hat{i}\}_{des}^B\|}.$$

Para o projeto de um controlador, é necessário definir uma função de erro para a malha de *feedback*. Define-se o erro de rastreamento para a posição e velocidade como:

$$e_{\vec{r}} = \vec{r} - \vec{r}_{des} \quad (3.25)$$

$$e_{\dot{\vec{r}}} = \dot{\vec{r}} - \dot{\vec{r}}_{des}. \quad (3.26)$$

Já os erros de rastreamento para a orientação e velocidade angular devem ser cuidadosamente escolhidos. A função de erro escolhida é apresentada como

$$\Psi(R, R_{des}) = \frac{1}{2} \text{tr}(I_{3 \times 3} - R_{des}^T R) \quad (3.27)$$

onde $\text{tr}(A)$ é o traço (soma dos elementos da diagonal principal) da matriz A . Ela é localmente positiva-definida em todos os pontos onde o ângulo de rotação entre R e R_{des} é menor que 180° [42]. Percebe-se também que $R \rightarrow R_{des}$ implica em $\Psi(R, R_{des}) \rightarrow 0$.

Considerando \hat{a} um operador que mapeia $\mathbb{R}^3 \rightarrow \text{SO}(3)$, de tal forma que

$$\hat{a} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Quando a variação da matriz de rotação pode ser expressa como $\partial R = R\hat{\eta}$ para $\eta \in \mathbb{R}^3$, a derivada da função do erro apresentada na Equação 3.27 é dada por

$$\dot{\Psi}(R, R_{des}) = \frac{1}{2} \text{tr}(R_{des}^T R \hat{\eta}) = \frac{1}{2} (R_{des}^T R - R^T R_{des})^\vee \cdot \eta \quad (3.28)$$

onde o operador vee^\vee mapeia o $so(3) \rightarrow \mathbb{R}^3$, sendo um operador inverso ao operador hat . Da Equação 3.28, pode-se definir o erro de rastreamento de orientação e_R como

$$e_R = \frac{1}{2}(R_{des}^T R - R^T R_{des})^\vee. \quad (3.29)$$

Para o cálculo do erro do rastreamento da velocidade angular e_Ω , pode-se comparar \dot{R} e \dot{R}_{des} de tal maneira que

$$\dot{R} - \dot{R}_{des}(R_{des}^T R) = R\hat{\vec{\Omega}} - R_{des}\hat{\vec{\Omega}}_{des}R_{des}^T R = R(\vec{\Omega} - R^T R_{des}\vec{\Omega}_{des})^\wedge. \quad (3.30)$$

Logo, como se deseja que a comparação apresentada na Equação 3.30 seja nula, fato que implica a igualdade entre as matrizes de rotação atual e desejada, tem-se que

$$\vec{\Omega} - R^T R_{des}\vec{\Omega}_{des} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Logo, define-se o erro de rastreamento da velocidade angular e_Ω como

$$e_\Omega = \vec{\Omega} - R^T R_{des}\vec{\Omega}_{des}. \quad (3.31)$$

Para o cálculo das entradas u_1 e \vec{u}_2 , deve-se calcular um vetor \vec{t} , o qual representa a direção e magnitude do vetor empuxo (*thrust vector*) necessárias para que o sistema siga a trajetória desejada. Tal vetor pode ser escrito como

$$\vec{t} = m(\ddot{\vec{r}}_{des} + \vec{K}_{\dot{\vec{r}}}\dot{\vec{r}} + \vec{K}_{\vec{r}}\vec{r} + \{\vec{P}\}^I) \quad (3.32)$$

onde $\vec{K}_{\dot{\vec{r}}}$ e $\vec{K}_{\vec{r}}$ representam as contantes do controlador, onde ambas possuem dimensões 3×1 , e $\{\vec{P}\}^I$ representa o vetor peso visto pelo *Frame* Inercial. Logo, a direção que o vetor unitário do *Frame* do Corpo Rígido deve apontar pode ser escrita como a normalização do vetor \vec{t} apresentado na Equação 3.32. Assim,

$$\{\hat{k}\}_{des}^B = \frac{\vec{t}}{\|\vec{t}\|}, \quad (3.33)$$

onde assume-se que

$$\|\vec{t}\| \neq 0.$$

Com o cálculo de $\{\hat{k}\}_{des}^B$ na Equação 3.33 e tendo conhecimento *a priori* de $\{\hat{i}\}_{des}^B$ devido à trajetória do ângulo *yaw* ($\psi(t)$, Equação 3.1), pode-se calcular a matriz de rotação

desejada R_{des} como foi apresentado na Equação 3.24. Utilizando as equações desenvolvidas nessa seção, pode-se concluir que as entradas do sistema para um controle do movimento de translação e rotação de um quadricóptero podem ser escritas como a seguir.

$$u_1 = \vec{t} \cdot R\{\hat{k}\}^B \quad (3.34)$$

$$\vec{u}_2 = -\vec{K}_R e_R - \vec{K}_\Omega e_\Omega + \vec{\Omega} \times I\vec{\Omega} - I(\hat{\vec{\Omega}} R^T R_{des} \vec{\Omega}_{des} - R^T R_{des} \dot{\vec{\Omega}}_{des}) \quad (3.35)$$

Onde I é a matriz de inércia do sistema e \vec{K}_R e \vec{K}_Ω representam contantes do controlador, ambas possuindo dimensões 3×1 .

O controlador desenvolvido nessa seção possui uma atratividade (*Basin of Attraction*) dada pelas equações apresentadas abaixo.

$$tr(I - (R_{des})^T R) < 2 \quad (3.36)$$

$$\|e_\omega(0)\|^2 \leq \frac{2}{\lambda_{min}(I)} k_R (1 - \frac{1}{2} tr(I - (R_{des})^T R)) \quad (3.37)$$

Onde $\lambda_{min}(I)$ se refere ao menor autovalor da matriz de inércia do sistema. Se ambas condições forem satisfeitas, o controlador apresentado irá convergir para o ponto de atração do sistema ($R = R_{des}$).

3.4 AJUSTE DOS GANHOS DOS CONTROLADORES

A presente seção discutirá os métodos utilizados para a escolha dos ganhos dos controladores descritos neste capítulo. Dois métodos distintos foram utilizados: o primeiro foi uma simples comparação entre o modelo do sistema linearizado com uma função de transferência de segunda ordem, para que se tornasse possível a escolha da frequência natural e do coeficiente de amortecimento desejados para o sistema; enquanto que o segundo foi a utilização de um algoritmo de otimização chamado *Particle Swarm Optimization* (PSO) [43].

3.4.1 Ganhos do Controlador para Pequenos Ângulos

A presente subseção apresentará o método de ajuste de ganhos *Non-Dimensional Tuning* descrito em [21]. De acordo com a referência citada, essa abordagem é comumente utilizada para veículos com uma massa que varia entre 65 e 1000 gramas.

Para que se possa ajustar os ganhos da lei de controle do Controlador para Pequenos Ângulos (Seção 3.1), confia-se no modelo dinâmico do veículo. Considere a seguinte equação de movimento para a rotação sobre o eixo x do *Frame* do Corpo Rígido, a qual pode ser

encontrada a partir das Equações 2.20 e 3.12, considerando a matriz T aproximadamente igual a identidade ($p \approx \dot{\phi}$):

$$I_{xx}\ddot{\phi} + K_{d,\phi}\dot{\phi} + K_{p,\phi}\phi = 0 \quad (3.38)$$

Como a Equação 3.38 é uma equação de um sistema de segunda ordem, pode-se comparar com a equação característica.

$$\ddot{\phi} + 2\xi\omega_n\dot{\phi} + \omega_n^2\phi = 0 \quad (3.39)$$

Na equação característica, sabe-se que ξ representa o amortecimento do sistema e o ω_n sua frequência natural. Para que se possa projetar o controlador de orientação, deve-se escolher os ganhos $K_{p,\phi}$ e $K_{d,\phi}$ de acordo com as especificações de amortecimento de frequência natural, de tal forma que

$$K_{p,\phi} = I_{xx}\omega_n^2 \quad (3.40)$$

e

$$K_{d,\phi} = 2I_{xx}\xi\omega_n. \quad (3.41)$$

O desenvolvimento descrito acima também pode ser utilizado para o ajuste dos ganhos $K_{p,\theta}$, $K_{d,\theta}$, $K_{p,\psi}$ e $K_{d,\psi}$.

Tipicamente, os controladores de orientação são projetados para serem criticamente amortecidos, ou próximos do amortecimento crítico ($\xi \approx 1$), com uma frequência natural $\omega_n \approx 9 \frac{rad}{s}$.

A resposta do controlador de orientação é limitado pelo tempo de atraso da resposta dos motores, o qual é função das hélices e do controlador do motor. Na prática, pode-se utilizar o desenvolvimento acima como um pontapé inicial para o ajuste experimental dos ganhos. Uma análise similar a essa pode ser feita para as equações relacionadas ao controlador de posição, de modo que apresentem uma resposta com amortecimento e frequência natural desejados.

3.4.2 Ganhos de quaisquer controladores

A presente subseção apresentará uma abordagem de cálculo de ganhos de controladores com base em métodos de otimização, especificamente o método *Particle Swarm Optimization*.

Foi-se desenvolvido um simulador utilizando as equações de modelo apresentadas na Seção 2.2. Sabendo a trajetória desejada $r_{des}(t)$, pode-se calcular funções erro $E(x)$, $E(y)$,

$E(z)$ e $E(\psi)$, as quais determinam o grau de acurácia do controle utilizado ao percorrer a trajetória desejada. O erro dado pelos ângulos ϕ e θ ($E(\phi)$ e $E(\theta)$) são calculados de acordo com os ângulos desejados calculados pelo controlador. A partir desses seis erros, pode-se alterar o valor dos ganhos do controlador com a finalidade de tender todas as funções erro a zero ($E(x)$, $E(y)$, $E(z)$, $E(\phi)$, $E(\theta)$, $E(\psi) \rightarrow 0$). Logo, a trajetória percorrida $r_P(t) \rightarrow r_{des}(t)$. A função erro escolhida para tal tarefa é apresentada a seguir.

$$E(x) = \frac{\sqrt{\sum_{k=1}^N (x(k) - \hat{x}(k))^2}}{N} \quad (3.42)$$

Onde $x(k)$ representa cada ponto da trajetória no eixo x e $\hat{x}(k)$ representa cada ponto percorrido no eixo x . A mesma equação é utilizada para o cálculo do erro nos outros eixos e ângulos. A partir do cálculo do erro, pode-se utilizar algum método de otimização para reduzir uma função objetivo. Para o presente trabalho, a função objetivo escolhida foi o somatório das seis funções de erro, cada uma com um peso específico, para que se pudesse comparar diferentes unidades de medida (metros e graus). Além disso, o método escolhido foi o *Particle Swarm Optimization*. Tal método encontra um mínimo de uma função objetivo por meio da movimentação de partículas num espaço dimensional apropriado. Depois de diversas iterações, as partículas têm uma tendência a convergir para um conjunto de ganhos. Caso as funções erro desejam pequenas o suficiente, os melhores ganhos encontrados podem ser utilizados pelo controlador.

É importante destacar que essa abordagem possui problemas. Os ganhos têm uma tendência de crescimento, o que pode representar incoerências quando observa-se o aspecto físico do sistema. Por exemplo, os motores, em um instante de tempo, podem desenvolver rotações contrárias às estabelecidas pelo modelo. Uma solução é penalizar a função objetivo com o crescimento desses ganhos.

Como as partículas se movimentam em um espaço n-dimensional, ele pode ser reduzido para diminuir a complexidade do problema. Como, por exemplo, o Controlador por Rastreamento Geométrico (Seção 3.3) possui 12 ganhos a serem alterados, as partículas desse método de otimização devem se movimentar por um espaço de 12 dimensões. Para simplificar o problema e diminuir o espaço de busca, algumas considerações podem ser feitas. Por exemplo, a rotação do quadricóptero em torno do eixo x e do eixo y são consideradas iguais, uma vez que $I_{xx} \approx I_{yy}$. Além disso, a movimentação do quadricóptero ao longo desses mesmos eixos também são consideradas iguais. Logo, os ganhos do controlador responsáveis por essa movimentação e rotação são considerados iguais.

Para resolver heurísticamente problemas de otimização como o apresentado, outros métodos também podem ser utilizados, como Algoritmos Genéticos [44].

4 PLANEJAMENTO DE TRAJETÓRIA

O presente capítulo será destinado ao cálculo de uma trajetória ótima dado as características do problema apresentado. Será desenvolvido um planejamento de trajetória pelo método da Trajetória com Snap Mínimo (*Minimum Snap Trajectory*), como proposto em [27].

Primeiramente, deve-se considerar as características do problema em questão. A trajetória calculada deve levar o quadricóptero de um ponto inicial \vec{r}_0 até um ponto final \vec{r}_F , podendo passar ou não por uma série de *waypoints*, de forma suave. Esse “critério de suavidade” pode ser entendido como uma minimização da velocidade com que a entrada modifica: um quadricóptero não consegue seguir trajetórias arbitrárias, as quais podem possuir mudanças abruptas. Para isso, deve-se descobrir a ordem do sistema (n) para verificar qual tipo de dinâmica a planta possui (1ª ordem - velocidade; 2ª ordem - aceleração; 3ª ordem - jerk; 4ª ordem - snap). A ordem do sistema determina a entrada do mesmo; portanto, a partir dela pode-se calcular uma trajetória factível de ser efetuado pela planta.

Para que se possa descobrir a ordem do sistema apresentado, deve-se observar as equações do modelo (2.18, 2.20 e 2.21), tal como as entradas definidas para o sistema (3.2 e 3.3):

- Na Equação 2.18 vê-se que a segunda derivada da posição ($\ddot{\vec{r}}$) depende de u_1 e da matriz de rotação R , sendo que esta última depende de ϕ , θ e ψ .
- A Equação 2.21 explicita a relação entre as velocidades angulares no *frame* do corpo rígido p , q e r com as derivadas dos ângulos ϕ , θ e ψ .
- Nas Equações 2.20 e 2.21, percebe-se que a segunda derivada da matriz de rotação R depende de \mathbf{u}_2 .
- A segunda derivada de \vec{r} é proporcional a u_1 no sistema linearizado.
- A quarta derivada de \vec{r} é proporcional a \mathbf{u}_2 no sistema linearizado.

Como a segunda derivada da matriz de rotação depende de \mathbf{u}_2 e a segunda derivada do vetor posição depende de R , percebe-se que a quarta derivada do vetor posição depende de \mathbf{u}_2 ; logo, o sistema apresentado é de quarta ordem. Isso motiva o desenvolvimento de uma Trajetória com Snap Mínimo, que tenta minimizar o quadrado da quarta derivada da posição integrada até um tempo máximo. Para o eixo x , temos:

$$x^*(t) = \operatorname{argmin} \int_0^T (\ddot{\ddot{x}})^2 dt \quad (4.1)$$

Pode-se resolver a Equação 4.1 utilizando a equação de Euler-Lagrange [45]. Tal equação é bastante aplicada em Cálculo de Variações, ou seja, quando se quer buscar máximos e mínimos de funções contínuas definidas sobre algum espaço funcional. Com a equação de Euler-Lagrange:

$$\frac{\partial L}{\partial x} - \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} + \frac{d^2}{dt^2} \frac{\partial L}{\partial \ddot{x}} - \frac{d^3}{dt^3} \frac{\partial L}{\partial \ddot{\dot{x}}} + \frac{d^4}{dt^4} \frac{\partial L}{\partial \ddot{\dot{\dot{x}}}} = 0 \quad (4.2)$$

Considerando que:

$$L = (\ddot{x})^2$$

Tem-se:

$$\begin{aligned} \frac{\partial (\ddot{x})^2}{\partial x} - \frac{d}{dt} \frac{\partial (\ddot{x})^2}{\partial \dot{x}} + \frac{d^2}{dt^2} \frac{\partial (\ddot{x})^2}{\partial \ddot{x}} - \frac{d^3}{dt^3} \frac{\partial (\ddot{x})^2}{\partial \ddot{\dot{x}}} + \frac{d^4}{dt^4} \frac{\partial (\ddot{x})^2}{\partial \ddot{\dot{\dot{x}}}} &= 0 \\ \frac{d^4}{dt^4} \frac{\partial (\ddot{x})^2}{\partial \ddot{\dot{\dot{x}}}} &= 0 \\ \stackrel{(8)}{x} &= 0 \end{aligned} \quad (4.3)$$

A Equação 4.3 indica que a oitava derivada da função $x(t)$ deva ser zero para que nossa proposição seja satisfeita. Logo:

$$x(t) = c_{7,1}t^7 + c_{6,1}t^6 + c_{5,1}t^5 + c_{4,1}t^4 + c_{3,1}t^3 + c_{2,1}t^2 + c_{1,1}t + c_{0,1} \quad (4.4)$$

A minimização apresentada na Equação 4.1 deve ocorrer nos três eixos cartesianos e no ângulo ψ . De maneira análoga, tem-se:

$$y(t) = c_{7,2}t^7 + c_{6,2}t^6 + c_{5,2}t^5 + c_{4,2}t^4 + c_{3,2}t^3 + c_{2,2}t^2 + c_{1,2}t + c_{0,2} \quad (4.5)$$

$$z(t) = c_{7,3}t^7 + c_{6,3}t^6 + c_{5,3}t^5 + c_{4,3}t^4 + c_{3,3}t^3 + c_{2,3}t^2 + c_{1,3}t + c_{0,3} \quad (4.6)$$

Porém, como visto anteriormente, a segunda derivada do o ângulo ψ está relacionado com a entrada \mathbf{u}_2 . Logo, é necessária a mudança da função custo a ser minimizada.

$$\psi^*(t) = \operatorname{argmin} \int_0^T (\ddot{\psi})^2 dt \quad (4.7)$$

Novamente, utilizando a equação de Euler-Lagrange:

$$\frac{\partial L}{\partial \psi} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\psi}} + \frac{d^2}{dt^2} \frac{\partial L}{\partial \ddot{\psi}} = 0$$

Considerando que:

$$L = (\ddot{\psi})^2$$

Tem-se:

$$\frac{\partial (\ddot{\psi})^2}{\partial \psi} - \frac{d}{dt} \frac{\partial (\ddot{\psi})^2}{\partial \dot{\psi}} + \frac{d^2}{dt^2} \frac{\partial (\ddot{\psi})^2}{\partial \ddot{\psi}} = 0$$

$$\frac{d^2}{dt^2} \frac{\partial (\ddot{\psi})^2}{\partial \ddot{\psi}} = 0$$

$$\overset{(4)}{\psi} = 0 \tag{4.8}$$

A Equação 4.8 indica que a quarta derivada da função $\psi(t)$ deva ser zero para que nossa proposição seja satisfeita. Logo:

$$\psi(t) = c_{3,4}t^3 + c_{2,4}t^2 + c_{1,4}t + c_{0,4} \tag{4.9}$$

Agora o desafio é encontrar todos os coeficientes $c_{n,m}$. Desconsiderando a presença de obstáculos ao longo do caminho desejado, pode-se obter todos os coeficientes analiticamente por meio de um simples sistema de equações. Uma vez que para a trajetória desejada no eixo x existem oito constantes desconhecidas, deve-se desenvolver oito, e apenas oito, equações linearmente independentes para que se possa calculá-las de modo que o sistema possua uma única solução. Considerando um percurso de $t = 0$ até $t = T_F$, pode-se desenvolver o seguinte sistema de equações a partir da Equação 4.4:

$$\begin{aligned} x(0) &= c_{0,1} \\ x(T_F) &= c_{7,1}T_F^7 + c_{6,1}T_F^6 + c_{5,1}T_F^5 + c_{4,1}T_F^4 + c_{3,1}T_F^3 + c_{2,1}T_F^2 + c_{1,1}T_F + c_{0,1} \\ \dot{x}(0) &= c_{1,1} \\ \ddot{x}(0) &= 2c_{2,1} \\ \dddot{x}(0) &= 6c_{3,1} \\ \dot{x}(T_F) &= 7c_{7,1}T_F^6 + 6c_{6,1}T_F^5 + 5c_{5,1}T_F^4 + 4c_{4,1}T_F^3 + 3c_{3,1}T_F^2 + 2c_{2,1}T_F + c_{1,1} \\ \ddot{x}(T_F) &= 42c_{7,1}T_F^5 + 30c_{6,1}T_F^4 + 20c_{5,1}T_F^3 + 12c_{4,1}T_F^2 + 6c_{3,1}T_F + 2c_{2,1} \\ \dddot{x}(T_F) &= 210c_{7,1}T_F^4 + 120c_{6,1}T_F^3 + 60c_{5,1}T_F^2 + 24c_{4,1}T_F + 6c_{3,1} \end{aligned}$$

Reescrevendo em forma matricial:

$$\begin{bmatrix} x(0) \\ x(T_F) \\ \dot{x}(0) \\ \ddot{x}(0) \\ \dot{x}(T_F) \\ \ddot{x}(T_F) \\ \ddot{x}(T_F) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ T_F^7 & T_F^6 & T_F^5 & T_F^4 & T_F^3 & T_F^2 & T_F & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 7T_F^6 & 6T_F^5 & 5T_F^4 & 4T_F^3 & 3T_F^2 & 2T_F & 1 & 0 \\ 42T_F^5 & 30T_F^4 & 20T_F^3 & 12T_F^2 & 6T_F & 2 & 0 & 0 \\ 210T_F^4 & 120T_F^3 & 60T_F^2 & 24T_F & 6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{7,1} \\ c_{6,1} \\ c_{5,1} \\ c_{4,1} \\ c_{3,1} \\ c_{2,1} \\ c_{1,1} \\ c_{0,1} \end{bmatrix} \quad (4.10)$$

Como todos os valores exceto $c_{n,m}$ são conhecidos e, uma vez que as equações escolhidas são linearmente independentes, a matriz

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ T_F^7 & T_F^6 & T_F^5 & T_F^4 & T_F^3 & T_F^2 & T_F & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 7T_F^6 & 6T_F^5 & 5T_F^4 & 4T_F^3 & 3T_F^2 & 2T_F & 1 & 0 \\ 42T_F^5 & 30T_F^4 & 20T_F^3 & 12T_F^2 & 6T_F & 2 & 0 & 0 \\ 210T_F^4 & 120T_F^3 & 60T_F^2 & 24T_F & 6 & 0 & 0 & 0 \end{bmatrix}$$

possui determinante diferente de zero, pode-se reescrever a Equação 4.10 de maneira a calcular todas as constantes $c_{n,m}$ da trajetória em x .

$$C = E^{-1}X \quad (4.11)$$

$$\begin{bmatrix} c_{7,1} \\ c_{6,1} \\ c_{5,1} \\ c_{4,1} \\ c_{3,1} \\ c_{2,1} \\ c_{1,1} \\ c_{0,1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ T_F^7 & T_F^6 & T_F^5 & T_F^4 & T_F^3 & T_F^2 & T_F & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 7T_F^6 & 6T_F^5 & 5T_F^4 & 4T_F^3 & 3T_F^2 & 2T_F & 1 & 0 \\ 42T_F^5 & 30T_F^4 & 20T_F^3 & 12T_F^2 & 6T_F & 2 & 0 & 0 \\ 210T_F^4 & 120T_F^3 & 60T_F^2 & 24T_F & 6 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} x(0) \\ x(T_F) \\ \dot{x}(0) \\ \ddot{x}(0) \\ \dot{x}(T_F) \\ \ddot{x}(T_F) \\ \ddot{x}(T_F) \end{bmatrix} \quad (4.12)$$

Contudo, pode-se aumentar a ordem do polinômio, caso queira-se adicionar novas restrições para o planejamento. O cálculo das constantes na Equação 4.12 leva em

Tabela 1 – Restrições para uma sequência de pontos pré-definidos

	Posição	Velocidade	Aceleração	Jerk	Snap	Crackle	Pop	Lock
$x_1(T_0)$	x_0	0	0	0	-	-	-	-
$x_1(T_1)$	x_1	v_1	a_1	j_1	s_1	c_1	p_1	l_1
$x_2(T_1)$	x_1	v_1	a_1	j_1	s_1	c_1	p_1	l_1
$x_2(T_2)$	x_2	v_2	a_2	j_2	s_2	c_2	p_2	l_2
$x_3(T_2)$	x_2	v_2	a_2	j_2	s_2	c_2	p_2	l_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$x_{n-1}(T_{n-1})$	x_{n-1}	0	0	0	-	-	-	-

Tabela 2 – Pontos arbitrária para o cálculo de uma curva

Tempo	Posição
0	0
1	1
2	2
3	0
4	1
5	2

consideração uma trajetória com apenas um ponto inicial e um ponto final. Caso se queira construir uma trajetória ótima a qual percorra uma sequência de pontos pré-definidos, deve-se construir uma matriz de tamanho $8(n-1) \times 8(n-1)$, onde n é o número de pontos pré-definidos, uma vez que serão construídas $n-1$ curvas, cada qual com suas oito constantes. Além disso, deve-se preservar a continuidade e diferenciabilidade da concatenação de todas as curvas, fazendo-se necessário que os valores das constantes no final de uma curva sejam as condições iniciais da curva seguinte. Considerando $x_n(t)$ a representação cada uma das $n-1$ curvas e T_n como o instante de tempo de cada ponto pré-definido, pode-se estabelecer uma tabela (Tabela 1) com as restrições de continuidade e diferenciabilidade para essa sequência de curvas. Para ilustrar a utilização da Tabela 1, um pseudocódigo e um exemplo simples de utilização desse algoritmo são apresentados no Apêndice A.

Utilizando tais restrições para o cálculo de uma curva que intercepta todos os pontos arbitrários apresentados na Tabela 2, obtêm-se como resultado as curvas desenhadas na Figura 5. Percebe-se a continuidade e diferenciabilidade das curvas de posição e de todas as suas derivadas.

Todo o desenvolvimento que culminou na Equação 4.12 pode ser estendida aos três eixos cartesianos, já que todos apresentam o mesmo cálculo de trajetória, como mostrado nas Equações 4.4, 4.5 e 4.6.

Entretanto, para o cálculo da trajetória do ângulo ψ existem menos restrições, já que este apresenta um polinômio de ordem menor, como mostrado na Equação 4.9. De maneira análoga à forma matricial de representação de sistemas lineares apresentada

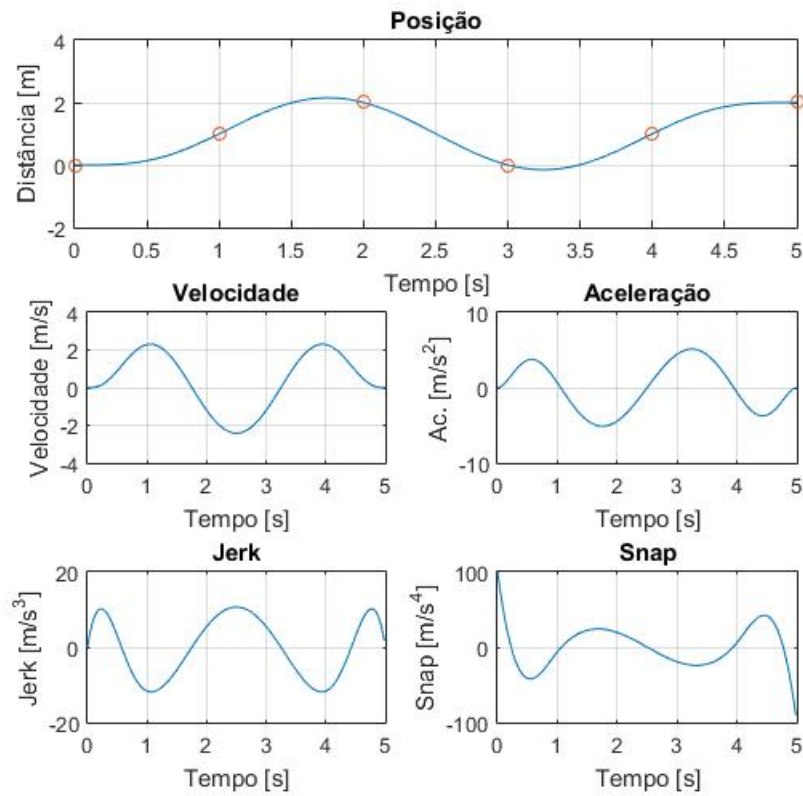


Figura 5 – Curvas obtidas seguindo restrições de continuidade e diferenciabilidade

na Equação 4.10 e considerando que são necessárias quatro, e apenas quatro, equações linearmente independentes para que se possa calcular as quatro constantes da curva de modo que o sistema possua uma única solução, chega-se à Equação 4.13.

$$\begin{bmatrix} x(0) \\ x(T_F) \\ \dot{x}(0) \\ \dot{x}(T_F) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ T_F^3 & T_F^2 & T_F & 1 \\ 0 & 0 & 1 & 0 \\ 3T_F^2 & 2T_F & 1 & 0 \end{bmatrix} \begin{bmatrix} c_{3,4} \\ c_{2,4} \\ c_{1,4} \\ c_{0,4} \end{bmatrix} \quad (4.13)$$

Sabendo de antemão que a matriz

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ T_F^3 & T_F^2 & T_F & 1 \\ 0 & 0 & 1 & 0 \\ 3T_F^2 & 2T_F & 1 & 0 \end{bmatrix}$$

Tabela 3 – Restrições para uma sequência de pontos pré-definidos para uma trajetória $\psi(t)$

	Posição	Velocidade	Aceleração	Jerk
$\psi_1(T_0)$	ψ_0	0	-	-
$\psi_1(T_1)$	ψ_1	v_1	a_1	\dot{j}_1
$\psi_2(T_1)$	ψ_1	v_1	a_1	\dot{j}_1
$\psi_2(T_2)$	ψ_2	v_2	a_2	\dot{j}_2
$\psi_3(T_2)$	ψ_2	v_2	a_2	\dot{j}_2
\vdots	\vdots	\vdots	\vdots	\vdots
$\psi_{n-1}(T_{n-1})$	ψ_{n-1}	0	-	-

é inversível, chega-se à Equação 4.14.

$$\begin{bmatrix} c_{3,4} \\ c_{2,4} \\ c_{1,4} \\ c_{0,4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ T_F^4 & T_F^3 & T_F^2 & T_F & 1 \\ 0 & 0 & 1 & 0 \\ 4T_F^3 & 3T_F^2 & 2T_F & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} x(0) \\ x(T_F) \\ \dot{x}(0) \\ \dot{x}(T_F) \end{bmatrix} \quad (4.14)$$

Contudo, pode-se aumentar a ordem do polinômio, caso queira-se adicionar novas restrições para o planejamento. Da forma análoga como fora discorrido anteriormente, o cálculo das contantes da Equação 4.14 leva em consideração uma trajetória com apenas um ponto inicial e um ponto final. Porém, o caso do cálculo da trajetória ótima para o ângulo ψ se diferencia apenas no tamanho da matriz de restrições: caso se queira construir uma trajetória ótima a qual percorra uma sequência de pontos pré-definidos, deve-se construir uma matriz de tamanho $4(n-1) \times 4(n-1)$, onde n é o número de pontos pré-definidos. Considerando $\psi_n(t)$ a representação cada uma das $n-1$ curvas e T_n como o instante de tempo de cada ponto pré-definido, pode-se estabelecer uma tabela (Tabela 3) com as restrições de continuidade e diferenciabilidade para essa sequência de curvas.

5 AMBIENTE DE SIMULAÇÕES EM MATLAB

O presente capítulo terá um enfoque no ambiente de simulações em MatLab desenvolvido para o presente trabalho.

Pode definir-se a simulação como a experimentação com um modelo que imita certos aspectos da realidade, o que permite trabalhar em condições semelhantes às reais, mas com variáveis controladas e num ambiente que se assemelha ao real, embora criado ou acondicionado artificialmente [46]. Logo, simulações são imprescindíveis no campo de engenharia, já que pode-se testar o que foi projetado num ambiente que se assemelha ao real, porém sem os custos do mesmo. Para testar todos os métodos desenvolvidos nos Capítulos 3 e 4, foi desenvolvido um ambiente de simulações a partir do GUI Environment do MatLab, utilizando as equações da modelagem desenvolvida no Capítulo 2. A Figura 6 apresenta a tela inicial do ambiente de simulação desenvolvido. É válido notar que a hélice colorida de vermelho representa o eixo x positivo no *frame* do corpo rígido.

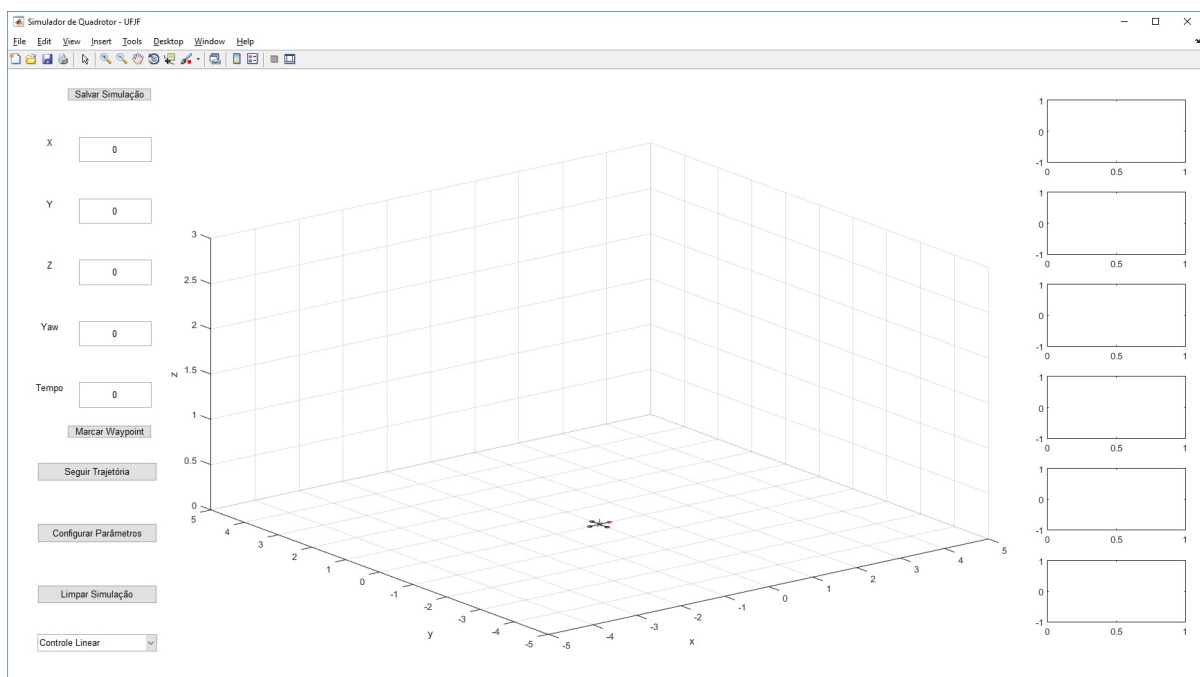


Figura 6 – Ambiente de Simulação

Neste ambiente é possível o teste de diferentes controladores em diferentes trajetórias, com todos os pontos definidos pelo usuário. Além disso, pode-se escolher os valores dos parâmetros intrínsecos relacionados à modelagem do sistema, como observado na Figura 7. Na Figura 6 percebe-se um desenho em três dimensões do ambiente simulado ao centro, seis gráficos relacionados a posição e ângulos do quadricóptero à direita, e a escolha de diferentes controladores e pontos para a trajetória à esquerda. O ambiente também permite salvar os dados de simulação em um arquivo de extensão *.mat*.

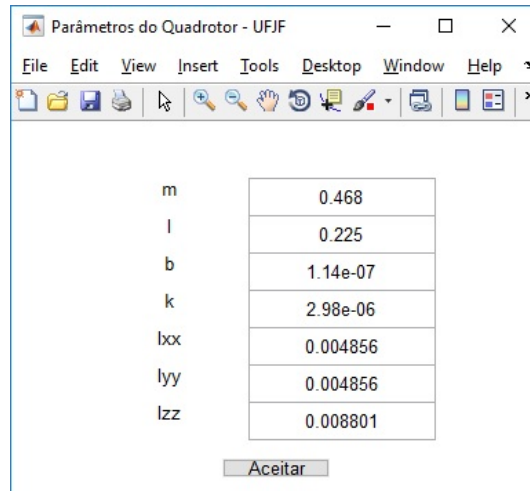


Figura 7 – Modificar parâmetros do sistema

A Figura 8 apresenta o ambiente de simulação em funcionamento, aplicando-se alguns pontos para o quadricóptero percorrer. É importante notar que a trajetória desejada é representada com uma curva pontilhada, enquanto que a trajetória percorrida, uma curva contínua. Isso se aplica ao desenho central e aos gráficos à direita.

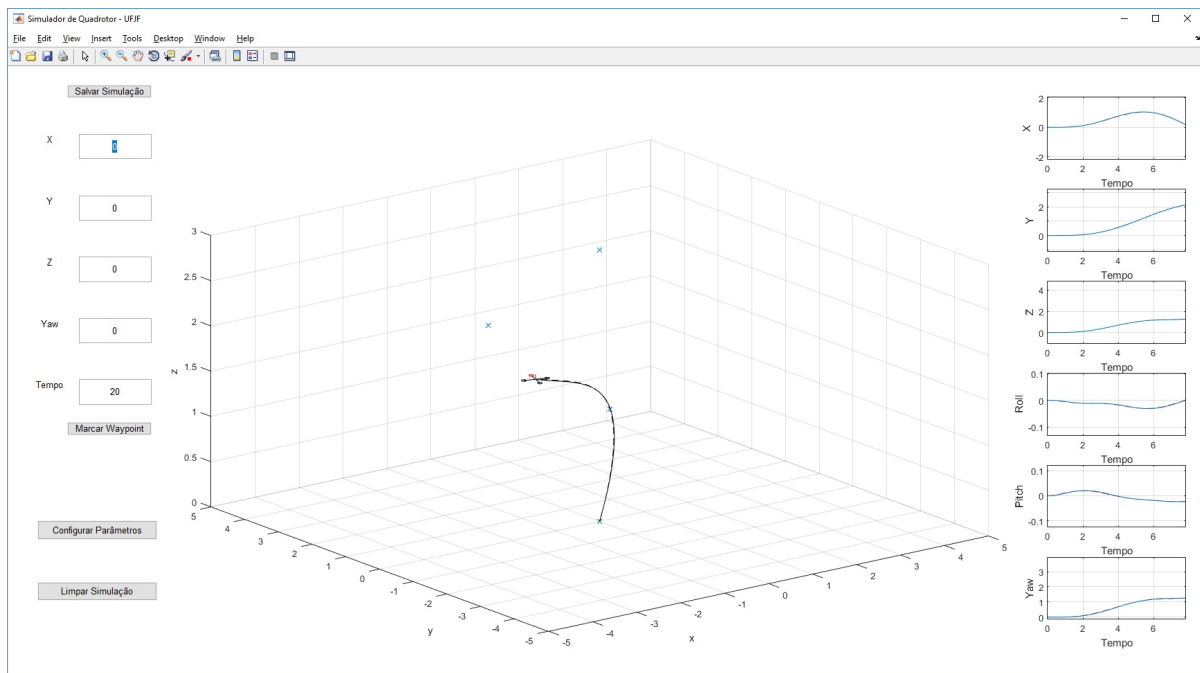


Figura 8 – Ambiente de Simulação em funcionamento

O mesmo gráfico em três dimensões pode ser desenhado com uma série de *snapshots* do quadricóptero ao longo do caminho percorrido, como apresentado na Figura 9. Para isso, deve-se apenas salvar os dados de voo da simulação e utilizar um segundo algoritmo. A visualização com *snapshots* do quadricóptero é importante para que se possa representar visualmente o comportamento do sistema ao longo do tempo.

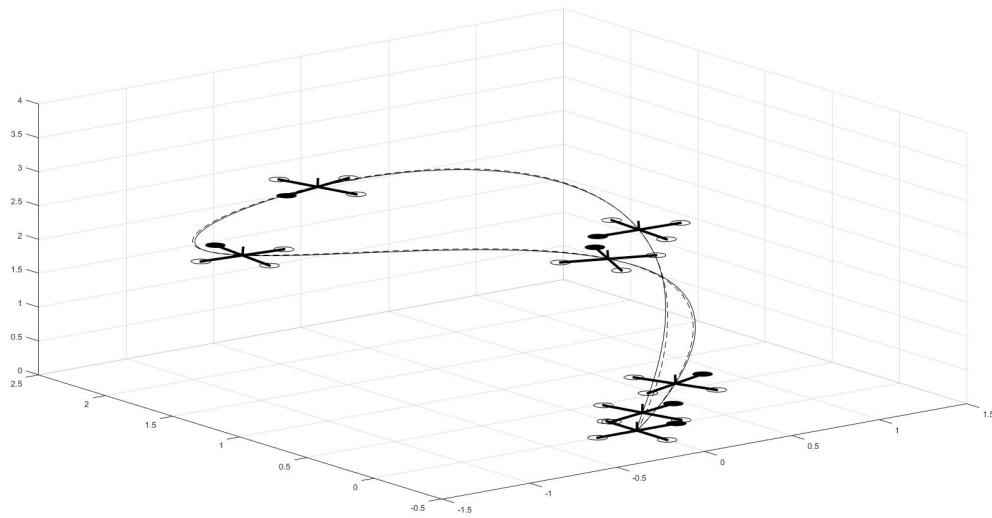


Figura 9 – Gráfico em três dimensões com *snapshots*

Outras funcionalidades estão em desenvolvimento para esse ambiente de simulação, tais como o suporte para múltiplos quadricópteros, comunicação com o controlador via *Robotics Operating System* (ROS) e implementação de obstáculos.

6 RESULTADOS

Este capítulo tem como objetivo apresentar e comparar a resposta dos controladores desenvolvidos no Capítulo 3. Para todos os controladores apresentados, serão simuladas três trajetórias:

- Trajetória 1: trajetória simples, com ponto inicial $r_T(0) = [0, 0, 0, 0]^T$ e ponto final $r_T(T_F) = [2, 2, 2, \pi]^T$. Dois diferentes T_F serão simulados: $T_F = 2s$ e $T_F = 5s$.
- Trajetória 2: trajetória helicoidal ao longo do eixo x , com ponto inicial $r_T(0) = [0, 0, 0, 0]^T$ e ponto final $r_T(T_F) = [1.8, 0, 1.6, 4\pi]^T$. A trajetória será realizada em 7,7 segundos.
- Trajetória 3: trajetória helicoidal ao longo do eixo x similar à Trajetória 2, porém com ponto inicial $r_T(0) = [-0.4, 0, 1, 0]^T$ e mesmo ponto final. Além disso, o quadricóptero apresenta uma rotação inicial em torno do eixo x de 3.1 radianos, aproximadamente 178° . A trajetória será realizada em 5,5 segundos.

Todas as trajetórias acima foram representadas na forma $[x, y, z, \psi]^T$ (sendo T o operador de transposição de vetores), com unidades metro e radiano. Para medir a performance dos controladores, foram utilizadas as medidas da Distância Euclidiana e do Erro Quadrático Médio.

6.1 CONTROLE PARA PEQUENOS ÂNGULOS

Esta seção apresentará os resultados gráficos da translação e rotação da planta utilizando o controlador apresentado na Seção 3.1. Em todos os gráficos apresentados nessa seção, a linha contínua representa o percurso tomado pelo quadricóptero, enquanto que a linha tracejada representa o percurso desejado.

6.1.1 Trajetória 1

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 1 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.1.

Para a Trajetória 1, fez-se duas diferentes simulações: a primeira, com um tempo final T_F igual a 5 segundos, enquanto que a segunda, T_F igual a 2 segundos. As Figuras 10 e 11 apresentam a ilustração do caminho executado e os gráficos da resposta do sistema para a Trajetória 1 com $T_F = 5s$, respectivamente; enquanto que as Figuras 13 e 14 apresentam essas informações para a Trajetória 1 com $T_F = 2s$.

Nas Figuras 13 e 14, pode-se perceber que o sistema não atingiu o critério requerido, o qual era percorrer a trajetória desejada em apenas dois segundos.

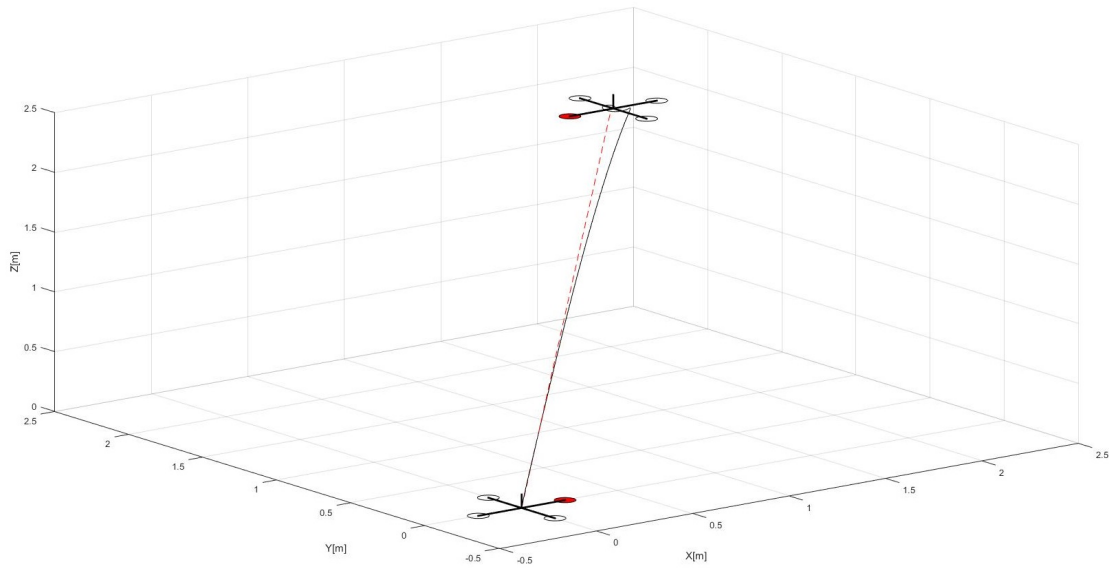


Figura 10 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 5$ segundos

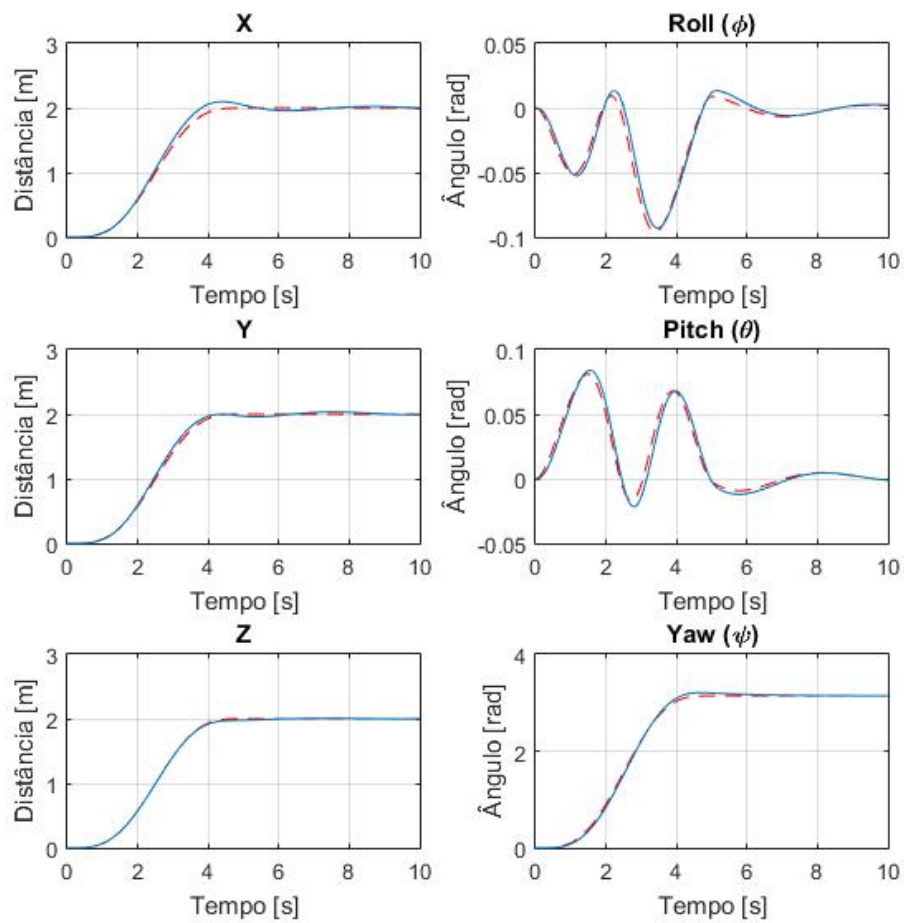


Figura 11 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 5$ segundos

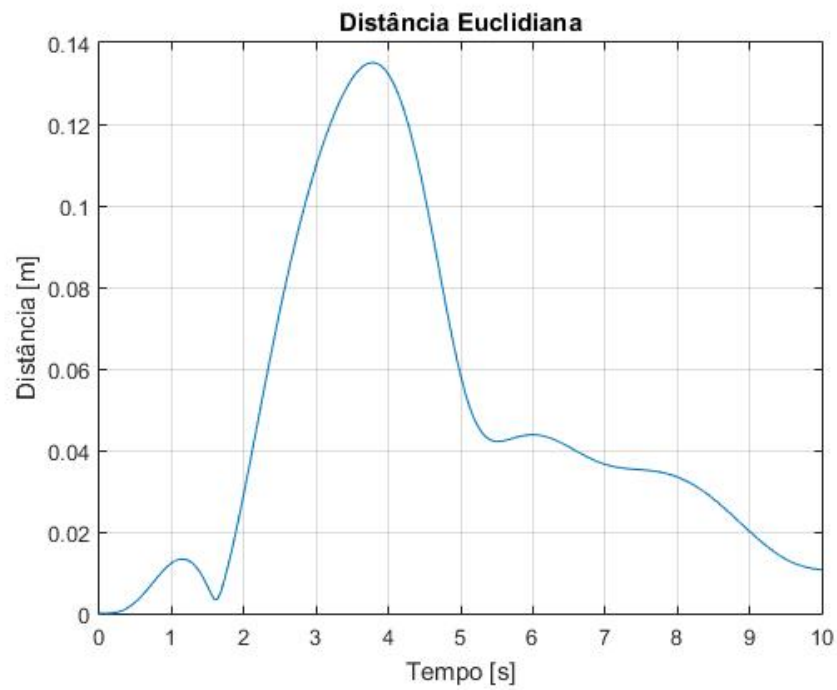


Figura 12 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 5$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.003659m^2

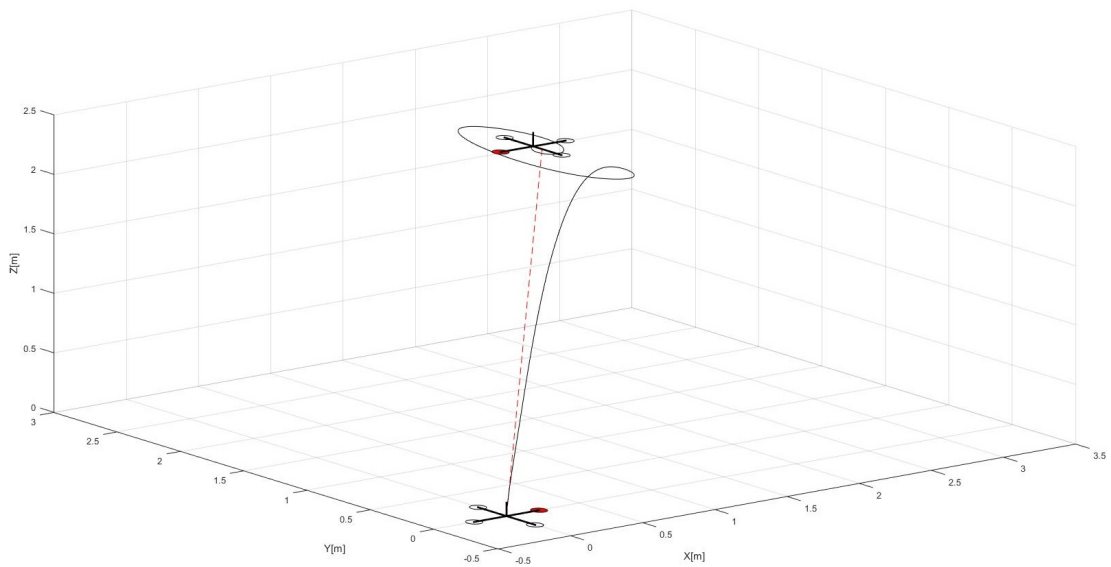


Figura 13 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 2$ segundos

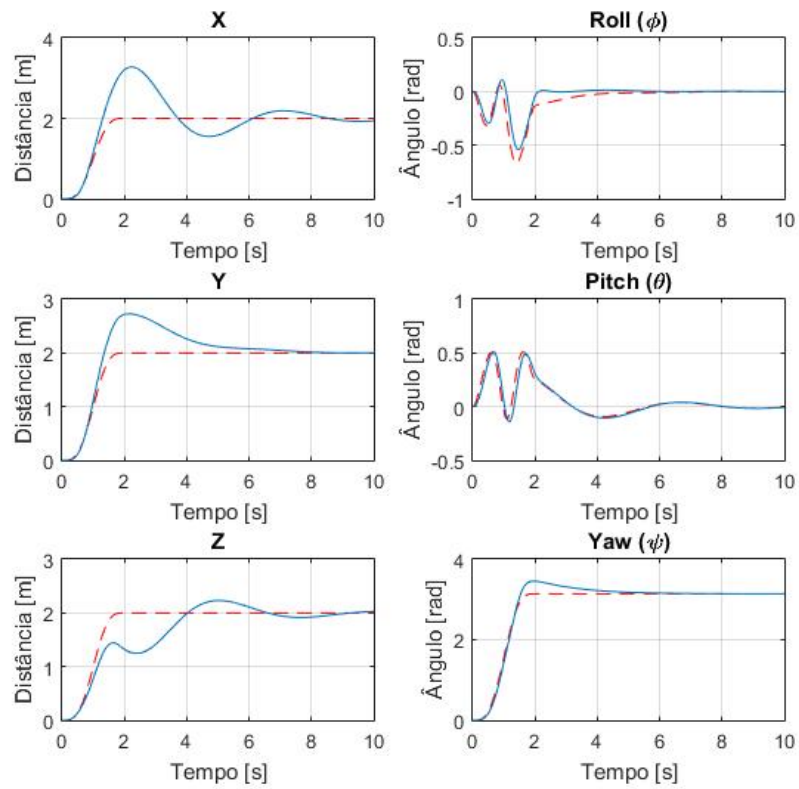


Figura 14 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 2$ segundos

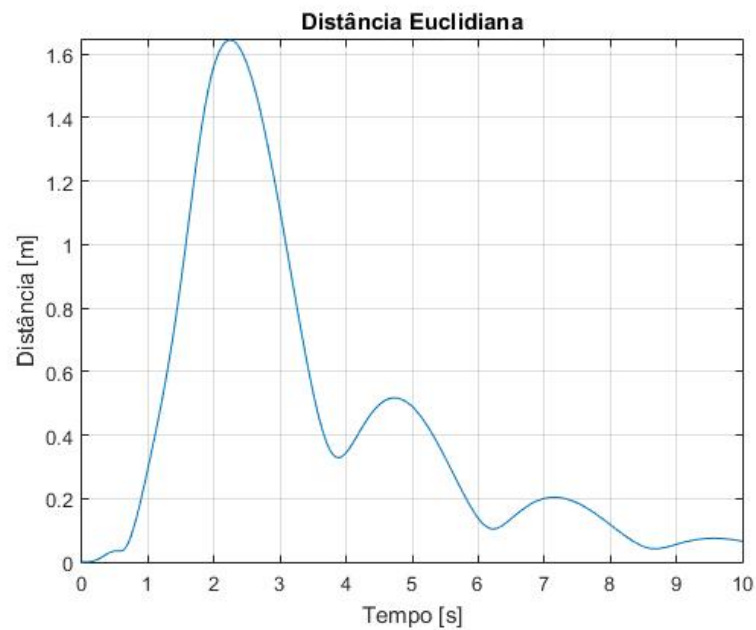


Figura 15 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 1 com $T_F = 2$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.376825m^2

6.1.2 Trajetória 2

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 2 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.1.

Para a Trajetória 2, os pontos escolhidos para formar a trajetória helicoidal são apresentados na Tabela 4.

Tabela 4 – Trajetória 2

t	x_{des}	y_{des}	z_{des}	ψ_{des}
0,00	0,00	0,00	0,00	0,00
1,40	0,00	0,00	1,00	0,00
2,10	0,20	0,00	1,60	0,00
2,80	0,40	0,40	1,00	1,57
3,50	0,60	0,00	0,40	3,14
4,20	0,80	-0,40	1,00	4,71
4,90	1,00	0,00	1,60	6,28
5,60	1,20	0,40	1,00	7,85
6,30	1,40	0,00	0,40	9,42
7,00	1,60	-0,40	1,00	11,00
7,70	1,80	0,00	1,60	12,57

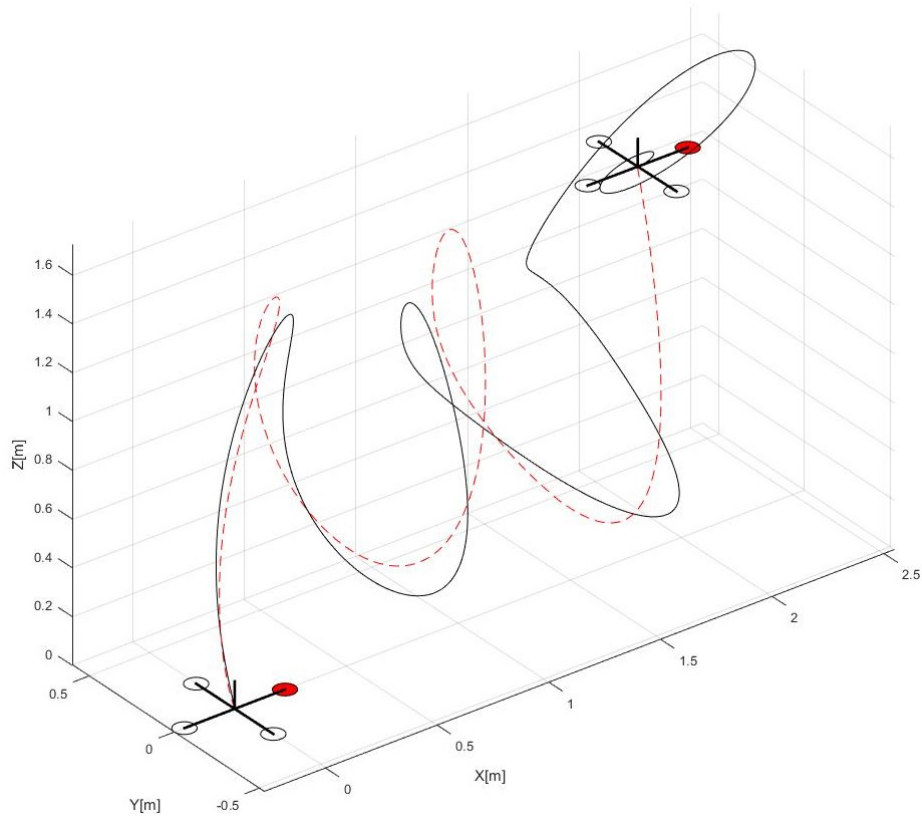


Figura 16 – Ilustração do caminho executado pelo sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 2

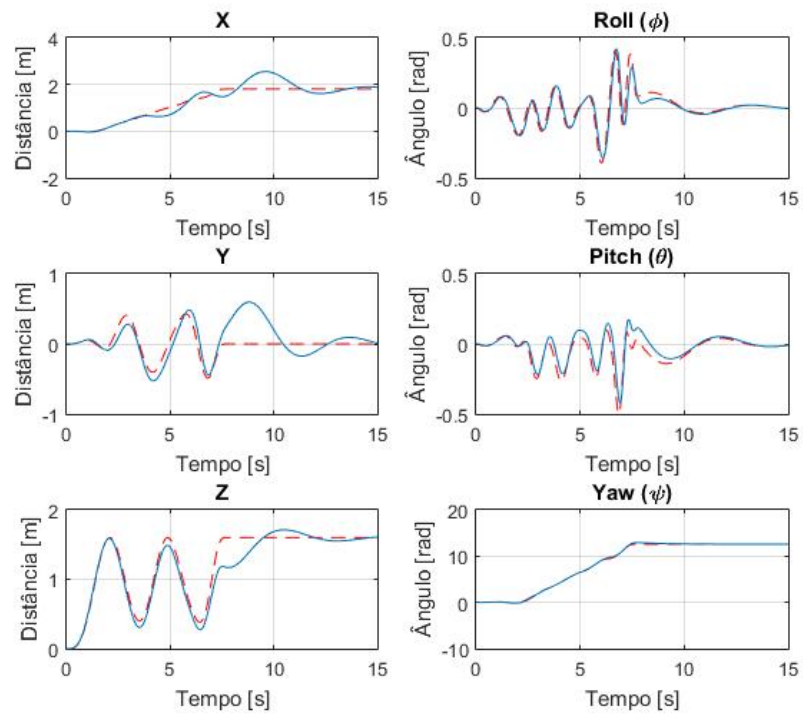


Figura 17 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 2

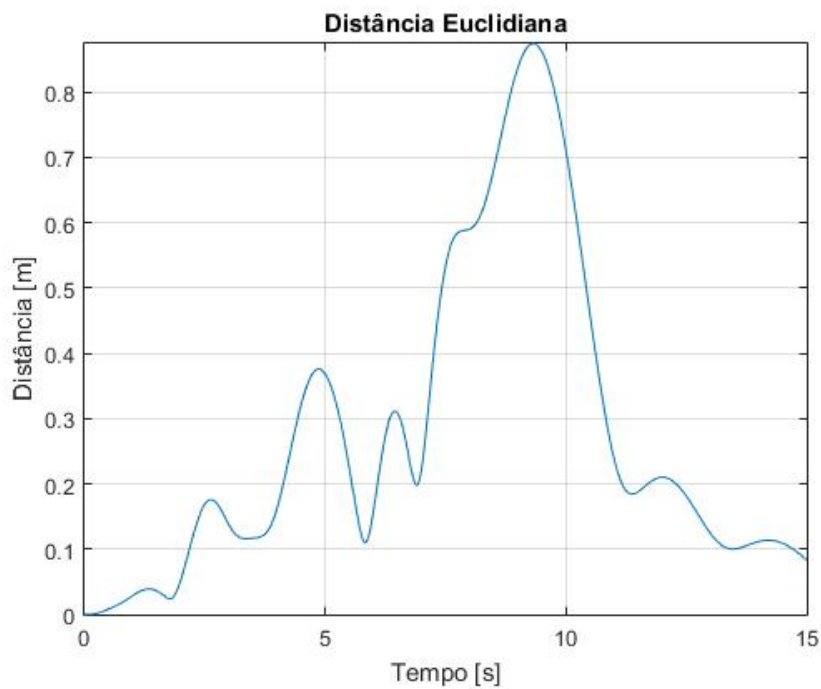


Figura 18 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 2 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.123305m^2

6.1.3 Trajetória 3

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 3 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.1.

Para a Trajetória 3, os pontos escolhidos para formar a trajetória helicoidal são apresentados na Tabela 5.

Tabela 5 – Trajetória 3

t	x_{des}	y_{des}	z_{des}	ψ_{des}
0,00	0,00	0,00	0,00	0,00
1,00	0,00	0,00	1,00	0,00
1,50	0,20	0,00	1,60	0,00
2,00	0,40	0,40	1,00	1,57
2,50	0,60	0,00	0,40	3,14
3,00	0,80	-0,40	1,00	4,71
3,50	1,00	0,00	1,60	6,28
4,00	1,20	0,40	1,00	7,85
4,50	1,40	0,00	0,40	9,42
5,00	1,60	-0,40	1,00	11,00
5,50	1,80	0,00	1,60	12,57

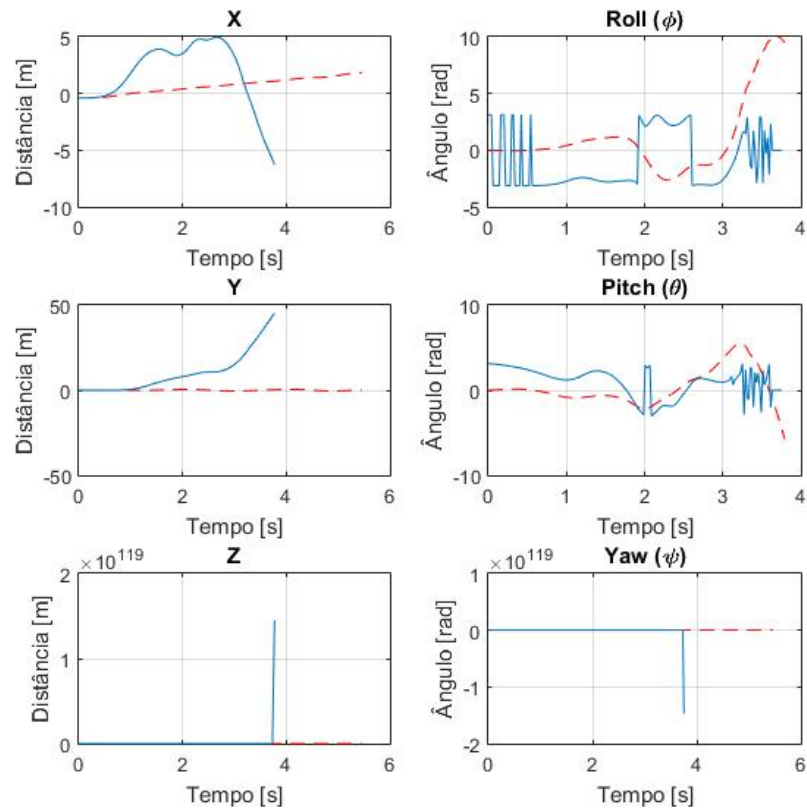


Figura 20 – Gráficos da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 3

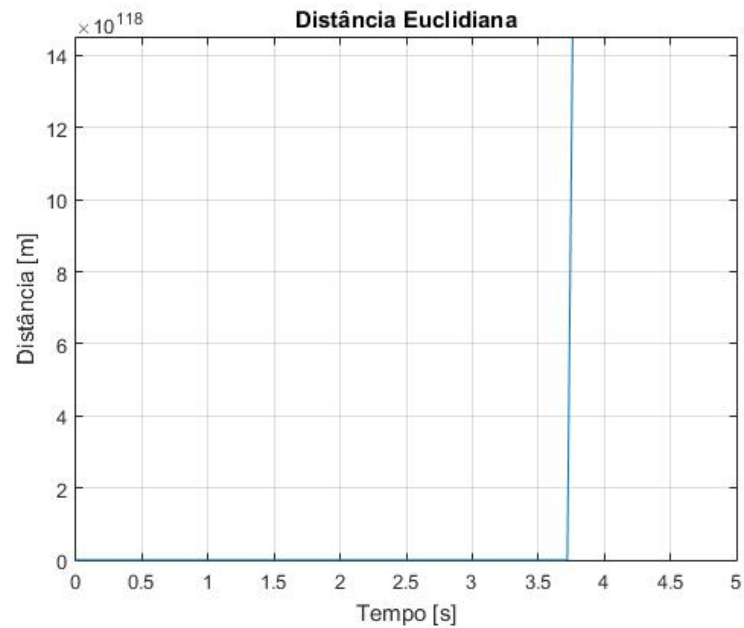


Figura 21 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle para Pequenos Ângulos percorrendo a Trajetória 3 em comparação com a trajetória desejada. O Erro Quadrático Médio tendeu ao infinito

6.2 CONTROLE DO VETOR EMPUXO DESEJADO

Esta seção apresentará os resultados gráficos da translação e rotação da planta utilizando o controlador apresentado na Seção 3.2. Em todos os gráficos apresentados nessa seção, a linha contínua representa o percurso tomado pelo quadricóptero, enquanto que a linha tracejada representa o percurso desejado.

6.2.1 Trajetória 1

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 1 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.2.

Assim como para a simulação do Controlador para Pequenos Ângulos, fez-se duas diferentes simulações para a Trajetória 1: a primeira, com um tempo final T_F igual a 5 segundos, enquanto que a segunda, T_F igual a 2 segundos. As Figuras 22 e 23 apresentam a ilustração do caminho executado e os gráficos da resposta do sistema para a Trajetória 1 com $T_F = 5s$, respectivamente; enquanto que as Figuras 25 e 26 apresentam essas informações para a Trajetória 1 com $T_F = 2s$.

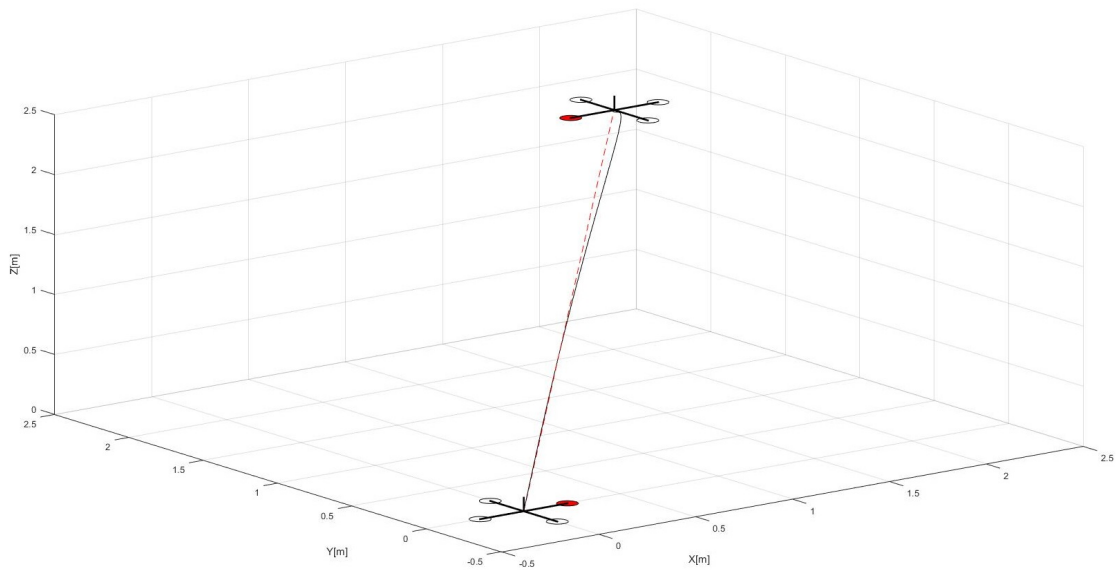


Figura 22 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 5$ segundos

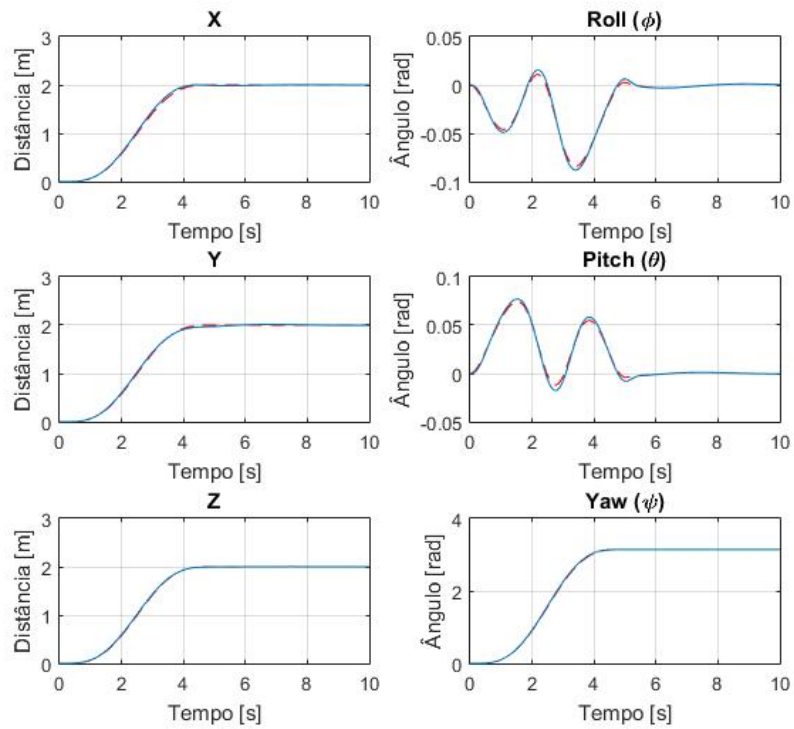


Figura 23 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 5$ segundos

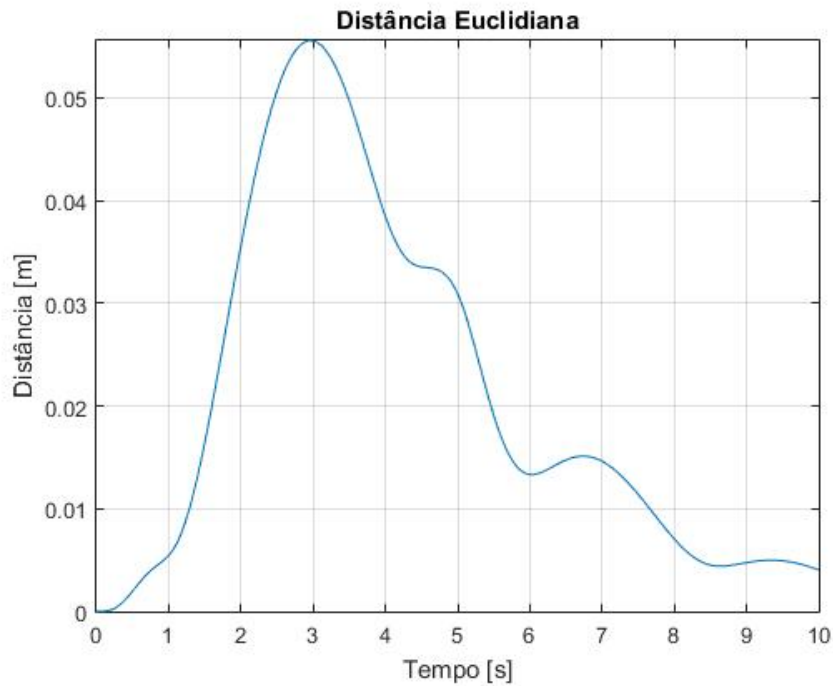


Figura 24 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 5$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.000072m^2

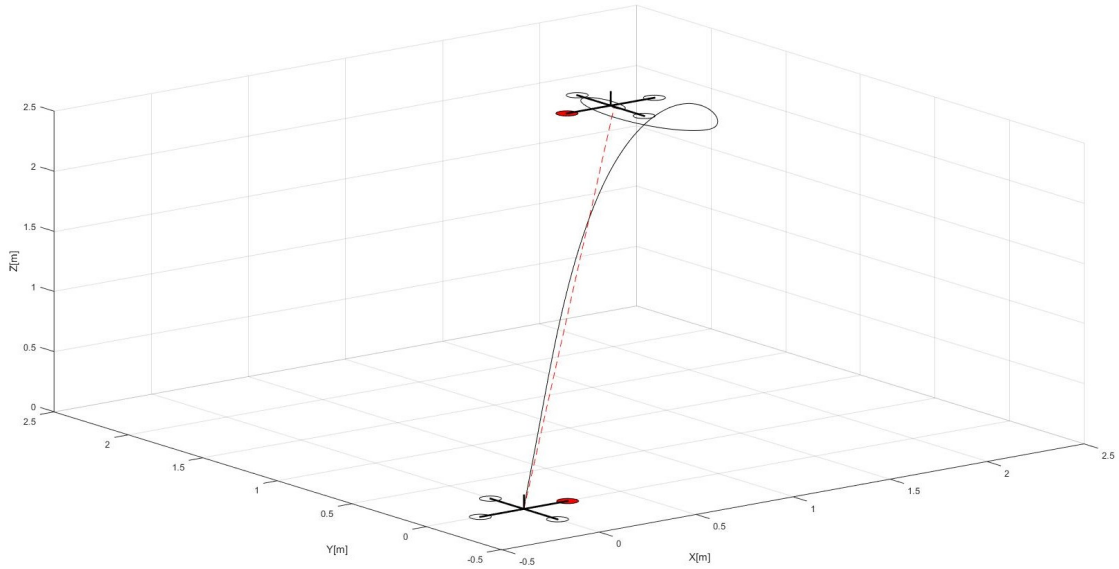


Figura 25 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 2$ segundos

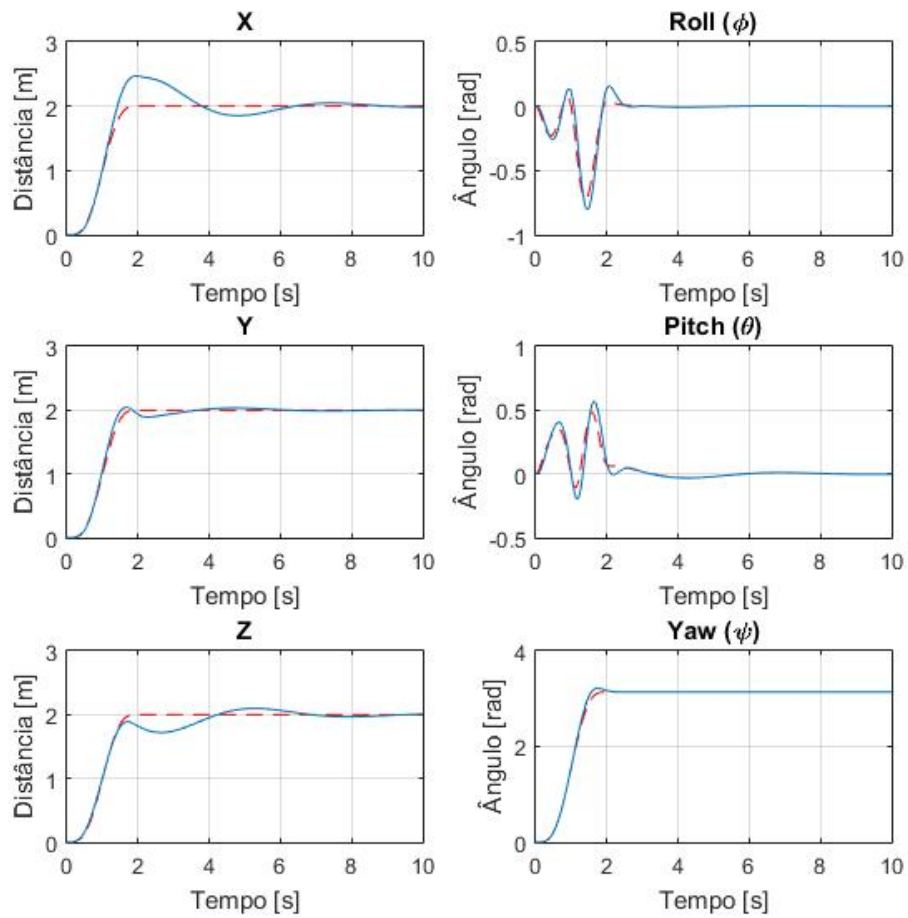


Figura 26 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 2$ segundos

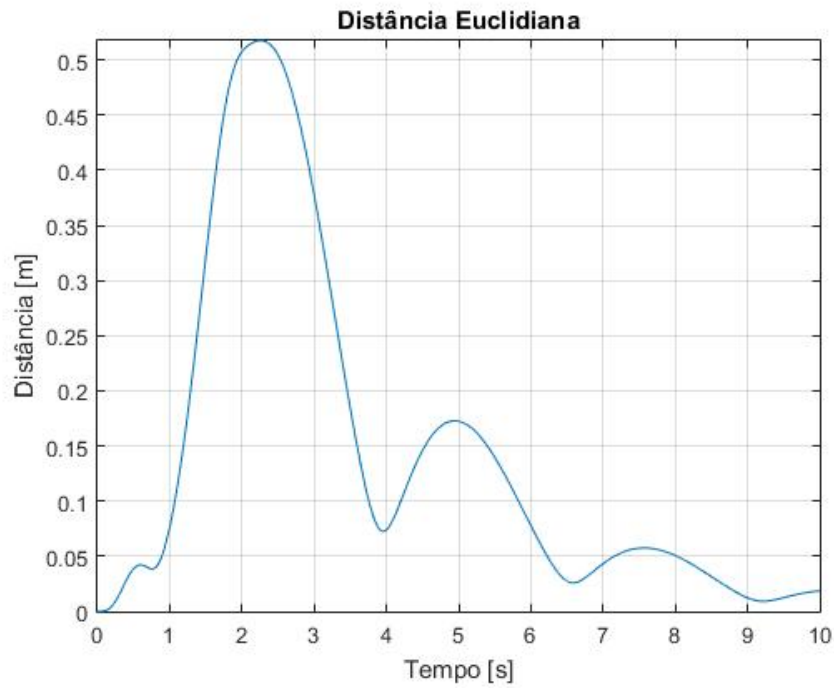


Figura 27 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 1 com $T_F = 2$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.044599m^2

6.2.2 Trajetória 2

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 2 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.2.

Para a Trajetória 2, os pontos escolhidos para formar a trajetória helicoidal são apresentados na Tabela 4.

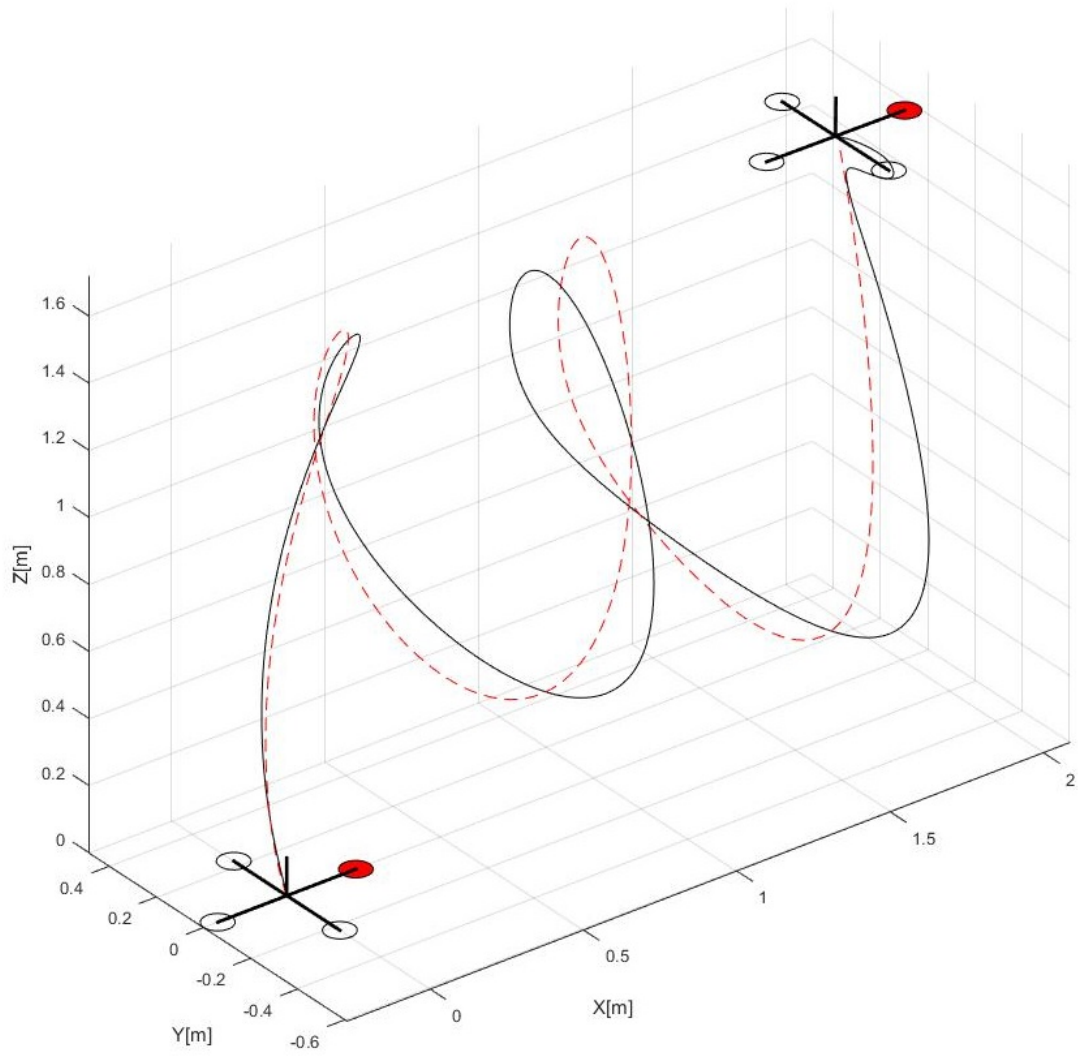


Figura 28 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 2

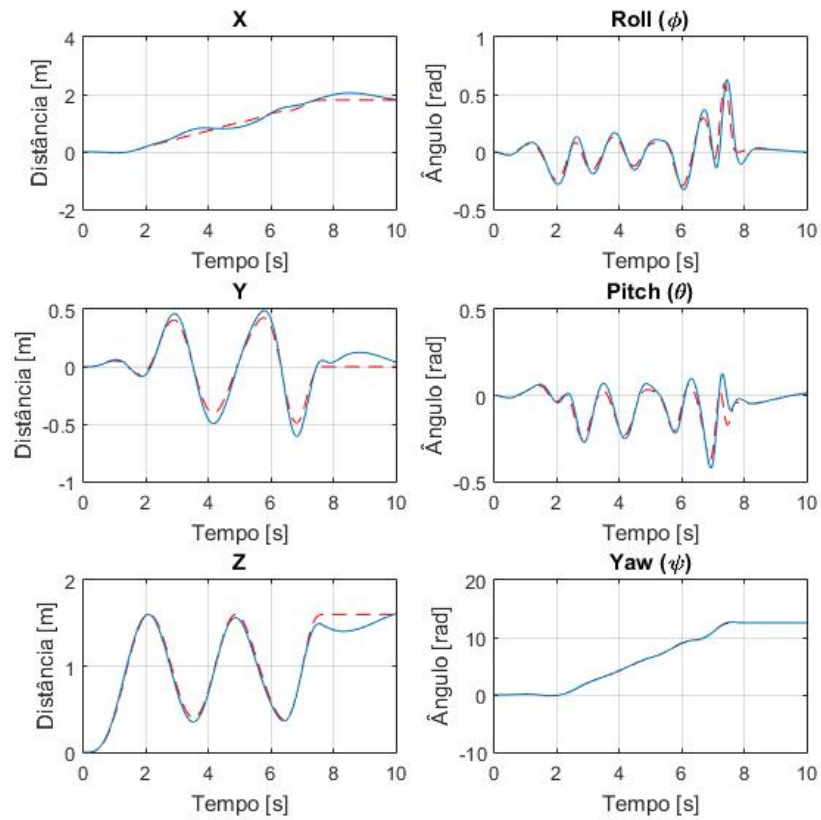


Figura 29 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 2

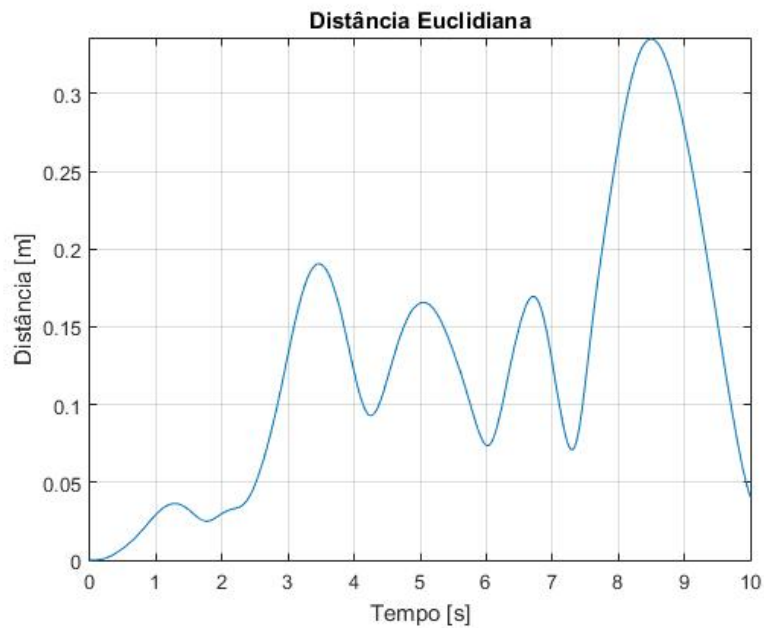


Figura 30 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 2 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.023816m^2

6.2.3 Trajetória 3

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 3 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.2.

Para a Trajetória 3, os pontos escolhidos para formar a trajetória helicoidal são apresentados na Tabela 5.

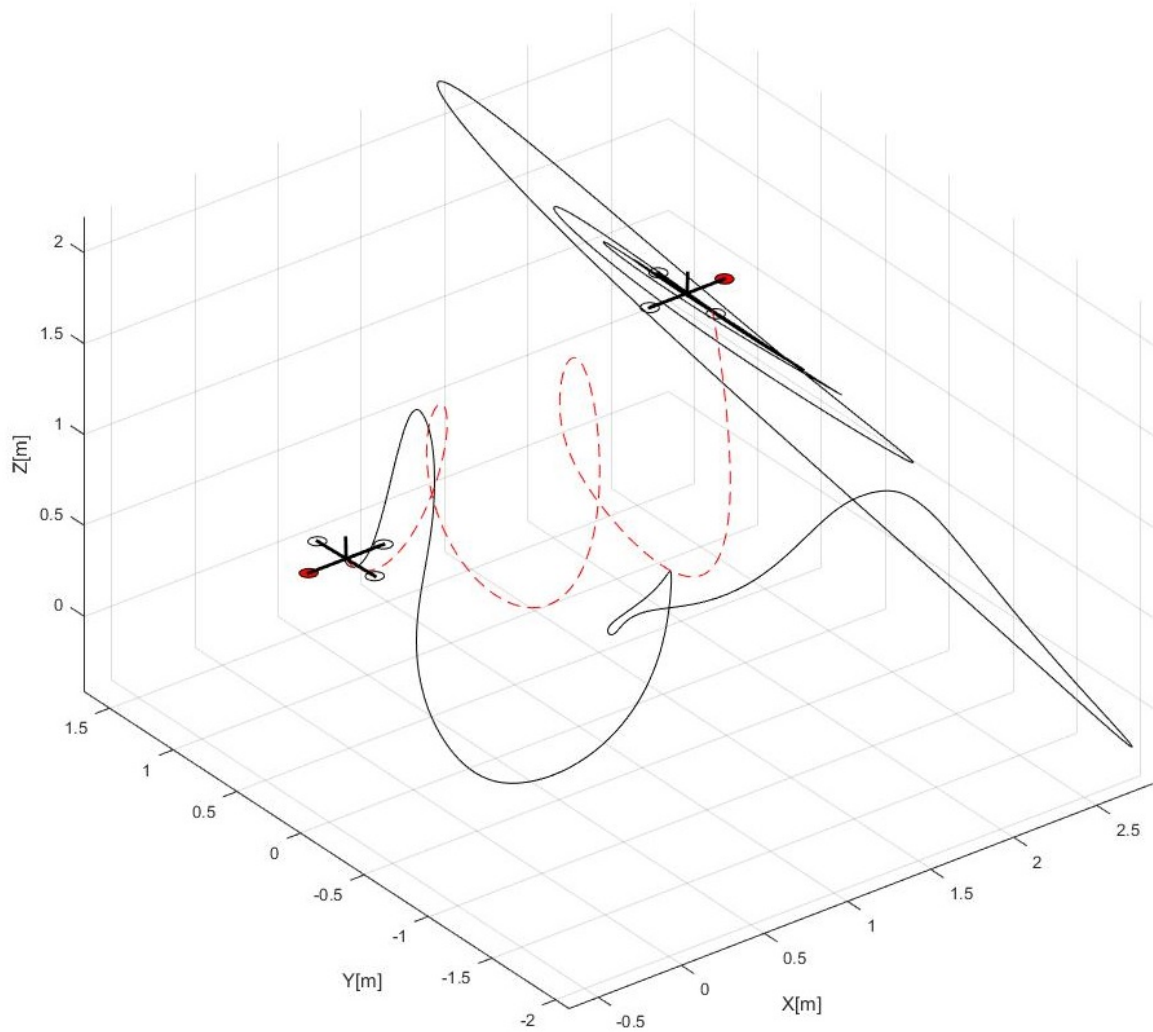


Figura 31 – Ilustração do caminho executado pelo sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 3

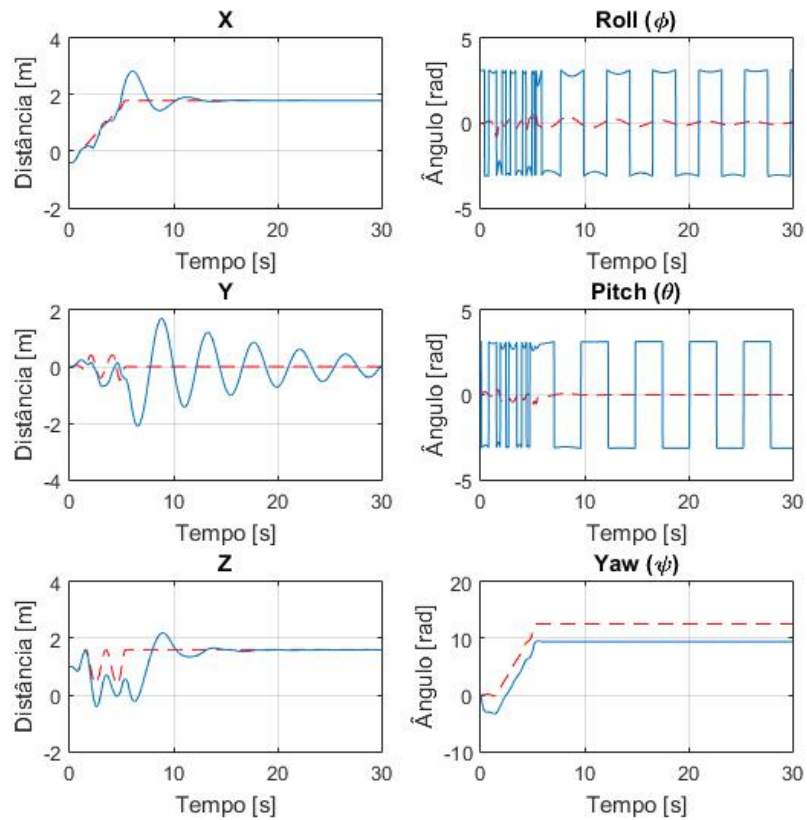


Figura 32 – Gráficos da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 3

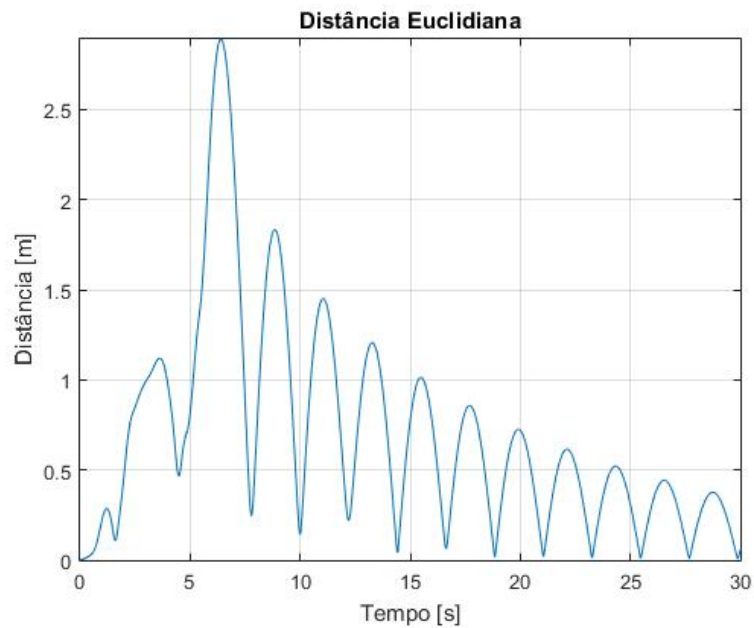


Figura 33 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle do Vetor Empuxo Desejado percorrendo a Trajetória 3 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 1.745761m^2

6.3 CONTROLE POR RASTREAMENTO GEOMÉTRICO

Esta seção apresentará os resultados gráficos da translação e rotação da planta utilizando o controlador apresentado na Seção 3.3. Em todos os gráficos apresentados nessa seção, a linha contínua representa o percurso tomado pelo quadricóptero, enquanto que a linha tracejada representa o percurso desejado.

6.3.1 Trajetória 1

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 1 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.3.

Assim como para a simulação do Controlador para Pequenos Ângulos e do Controlador do Vetor Empuxo Desejado, fez-se duas diferentes simulações para a Trajetória 1: a primeira, com um tempo final T_F igual a 5 segundos, enquanto que a segunda, T_F igual a 2 segundos. As Figuras 34 e 35 apresentam a ilustração do caminho executado e os gráficos da resposta do sistema para a Trajetória 1 com $T_F = 5s$, respectivamente; enquanto que as Figuras 37 e 38 apresentam essas informações para a Trajetória 1 com $T_F = 2s$.

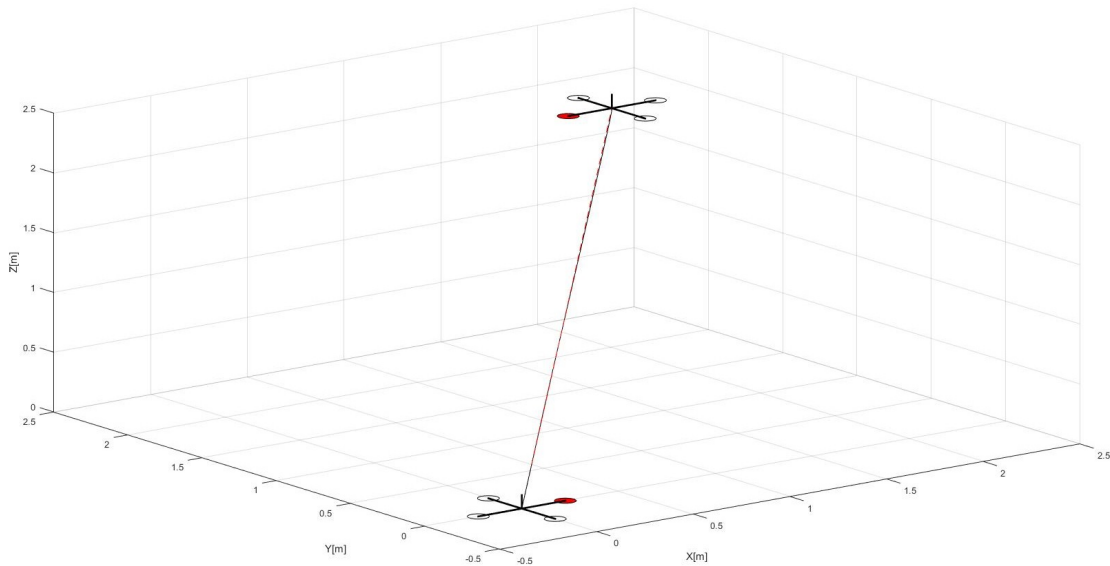


Figura 34 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 5$ segundos

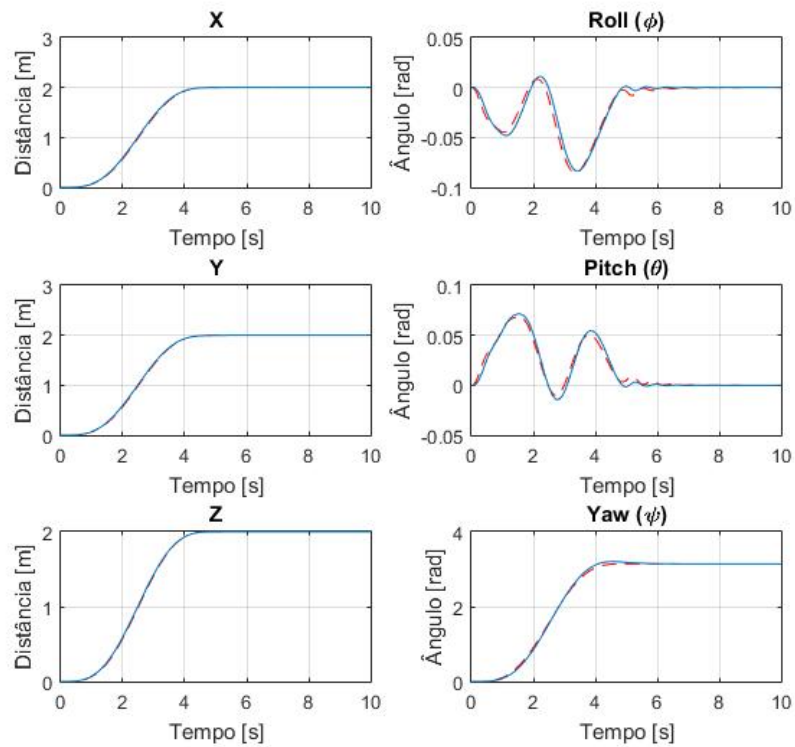


Figura 35 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 5$ segundos

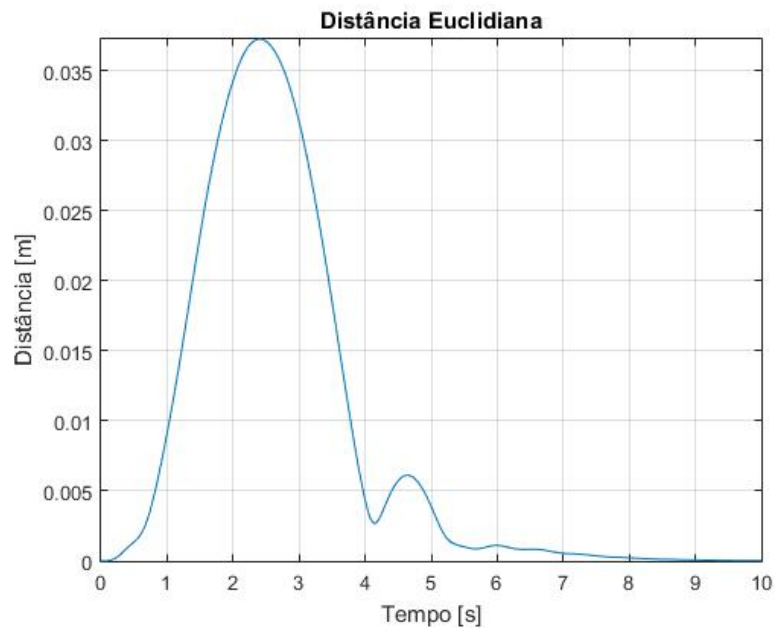


Figura 36 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 5$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.000022m^2

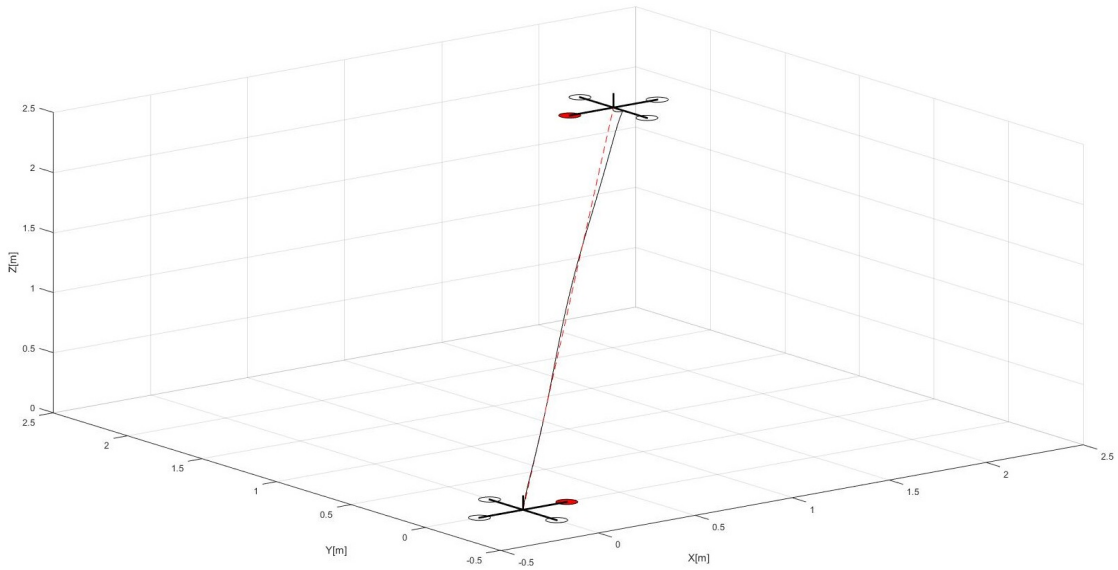


Figura 37 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 2$ segundos

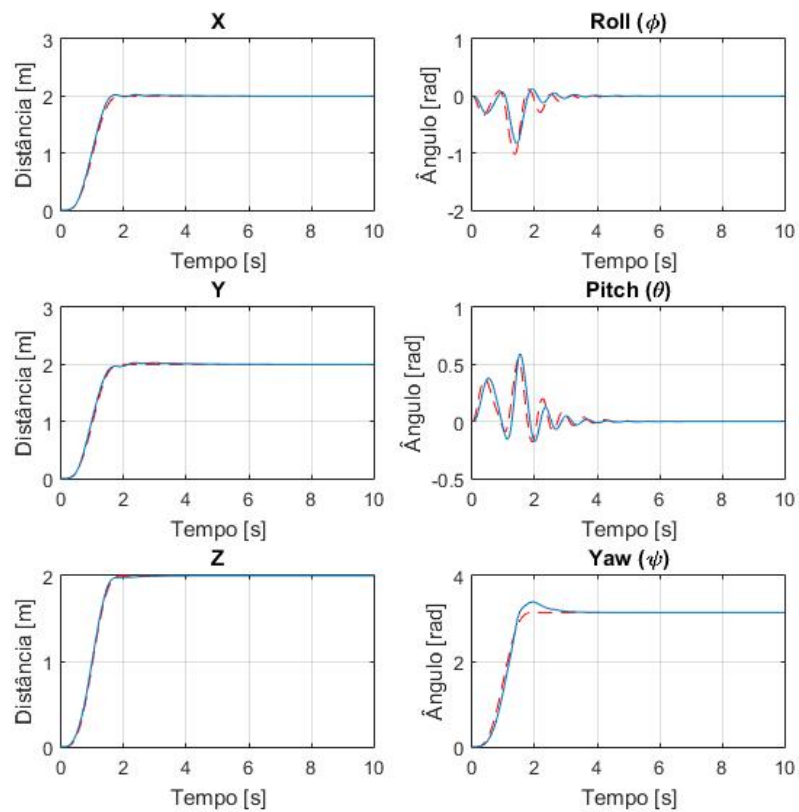


Figura 38 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 2$ segundos

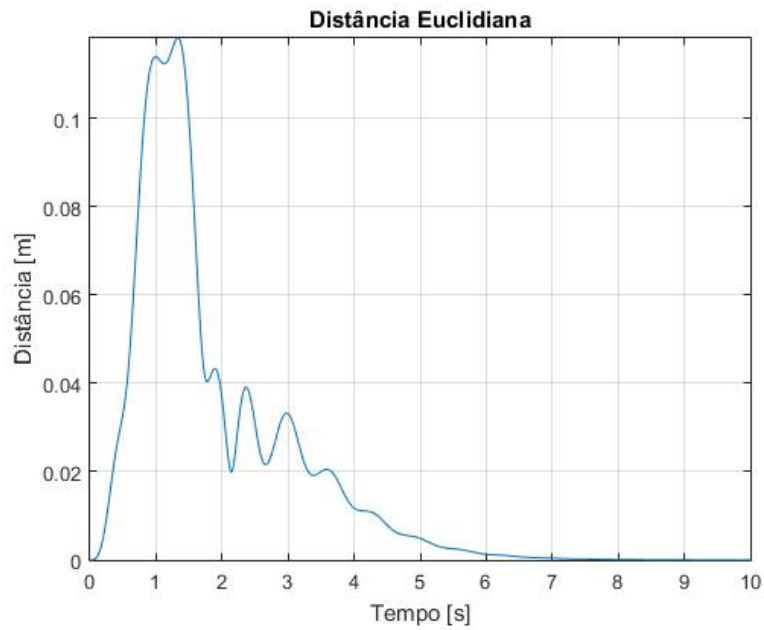


Figura 39 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 1 com $T_F = 2$ segundos em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.001320m^2

6.3.2 Trajetória 2

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 2 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.3.

Para a Trajetória 2, os pontos escolhidos para formar a trajetória helicoidal são apresentados na Tabela 4.

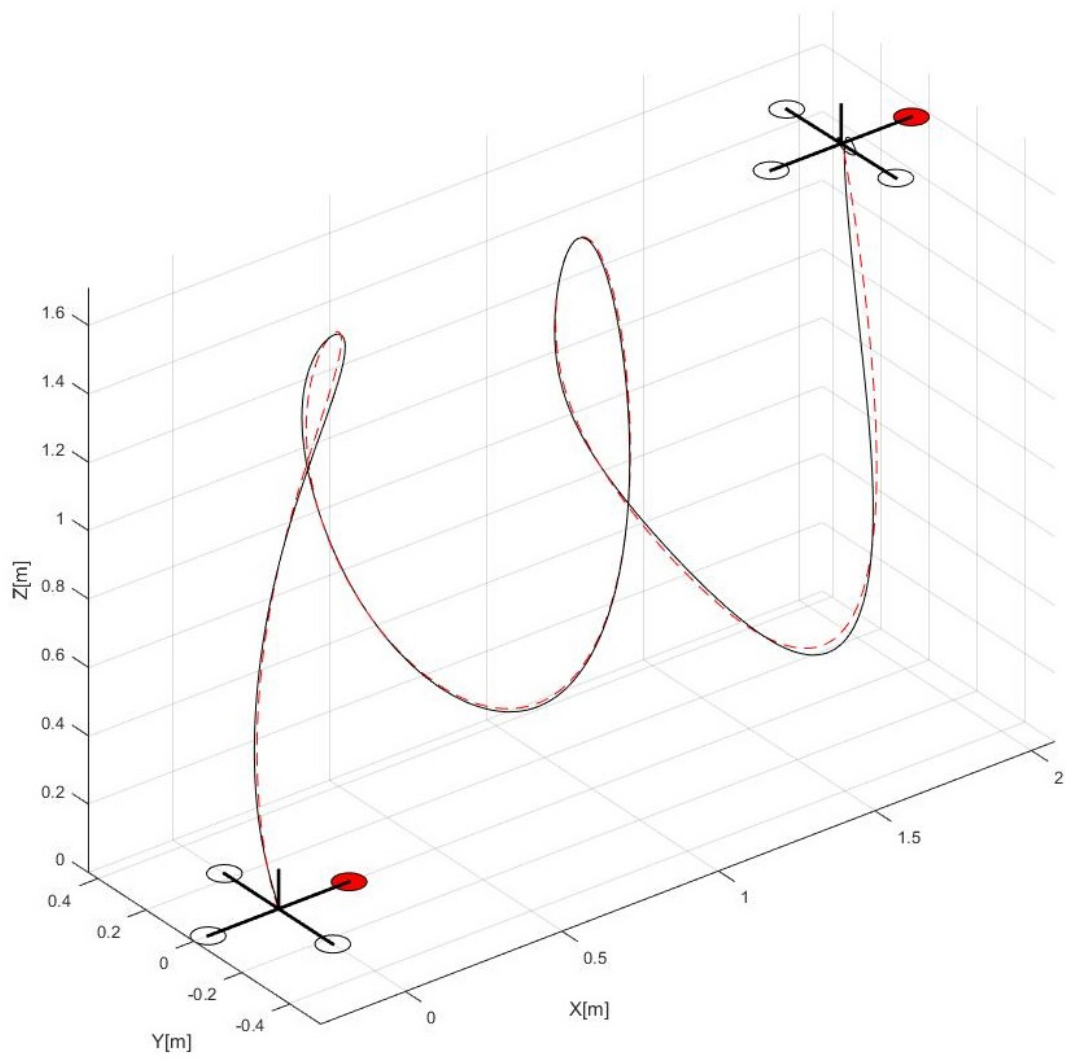


Figura 40 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 2

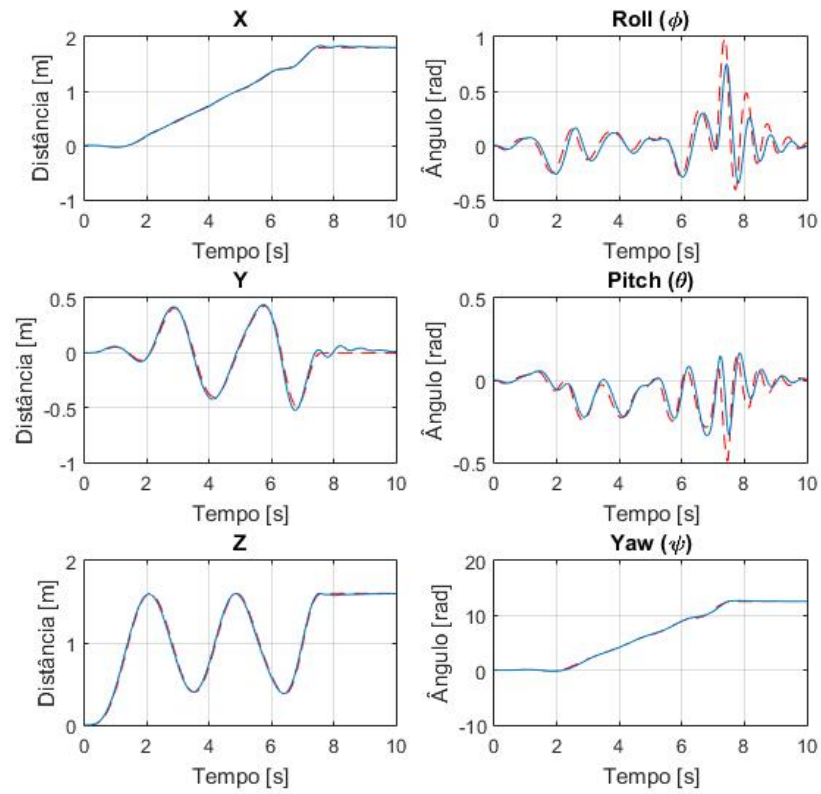


Figura 41 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 2

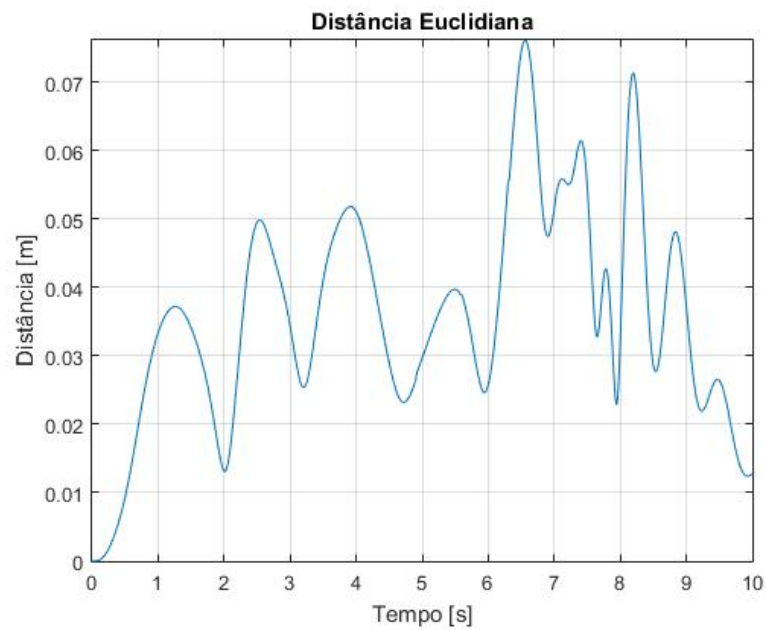


Figura 42 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 2 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.001526m^2

6.3.3 Trajetória 3

Esta subseção apresentará os resultados gráficos numa simulação onde a Trajetória 3 é a trajetória desejada, utilizando o controlador apresentado na Seção 3.3.

Para a Trajetória 3, os pontos escolhidos para formar a trajetória helicoidal são apresentados na Tabela 5.

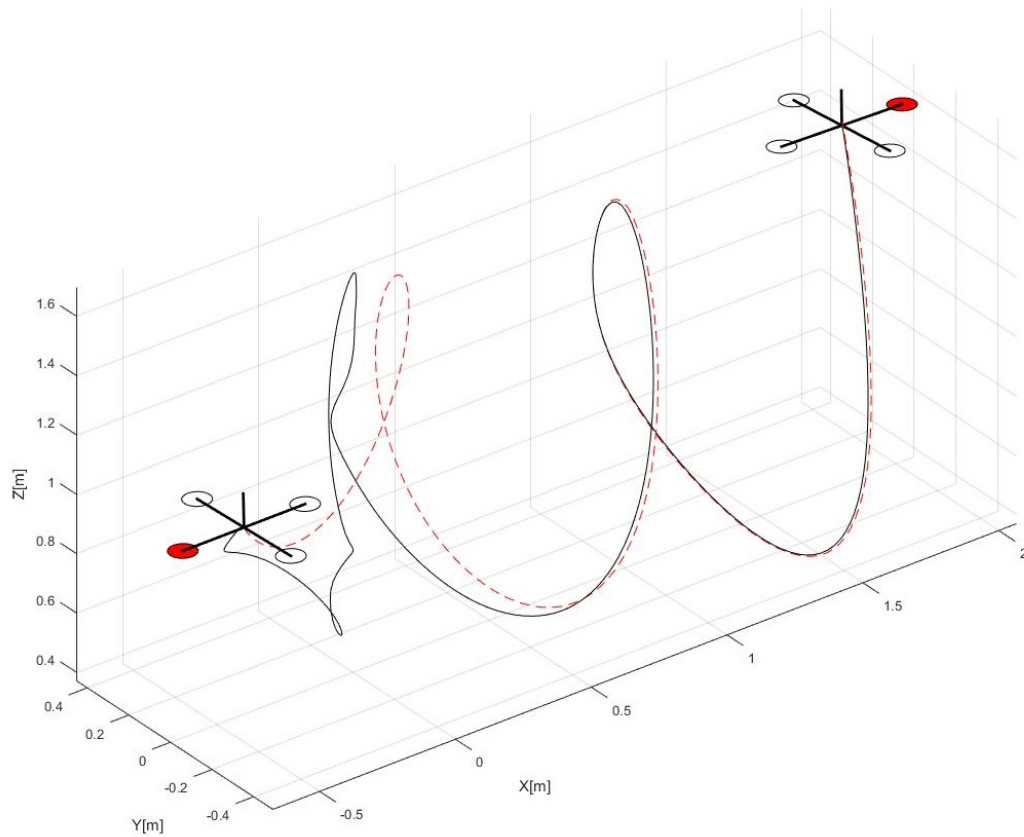


Figura 43 – Ilustração do caminho executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3

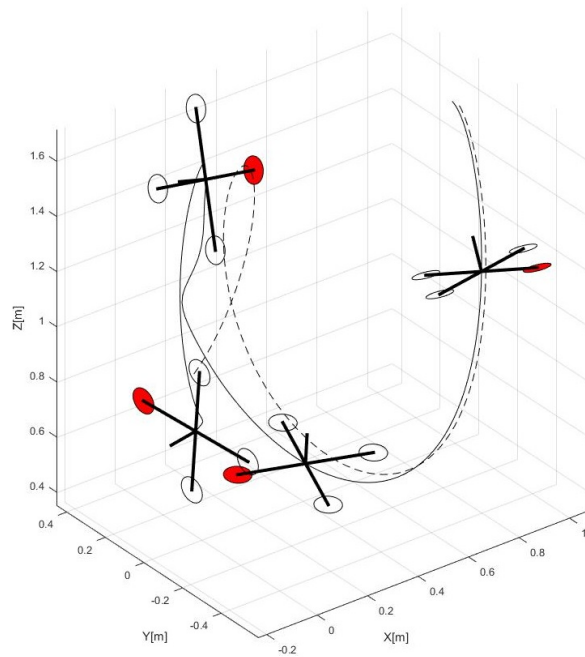


Figura 44 – Ilustração com *snapshots* da manobra de recuperação da trajetória executado pelo sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3 entre $t = 1$ segundo e $t = 3,5$ segundos

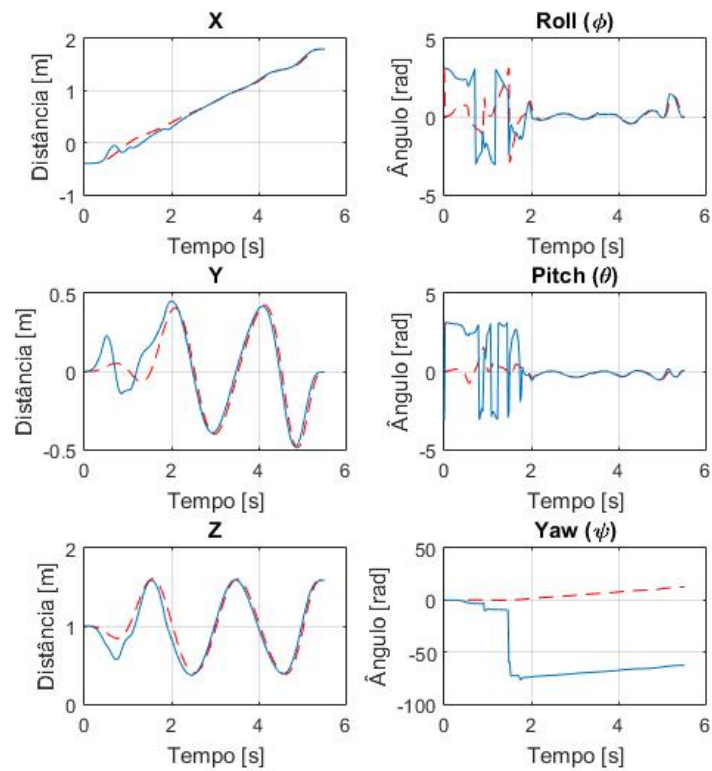


Figura 45 – Gráficos da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3

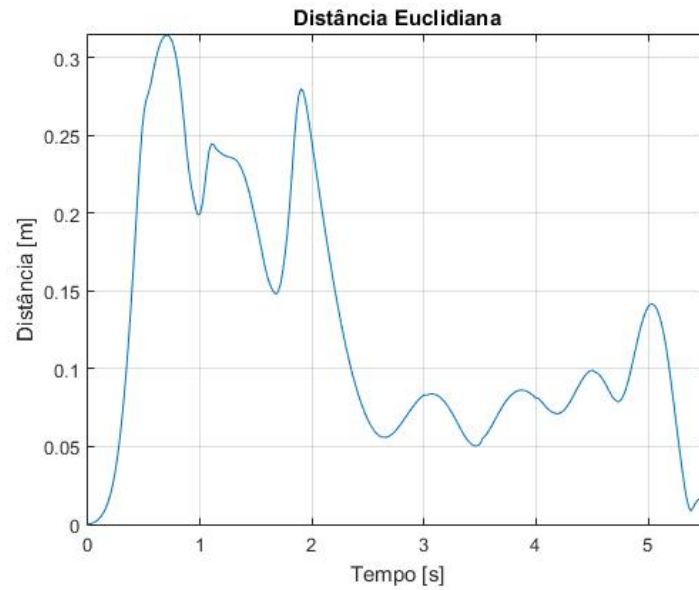


Figura 46 – Gráficos da Distância Euclidiana da resposta do sistema com o Controle por Rastreamento Geométrico percorrendo a Trajetória 3 em comparação com a trajetória desejada. O Erro Quadrático Médio foi de 0.022389m^2

6.4 COMPARAÇÃO ENTRE OS CONTROLES UTILIZADOS

A presente seção irá comparar os dados apresentados nas Seções 6.1, 6.2 e 6.3.

Comparou-se o desempenho dos três controladores citados nesse trabalho. Para uma melhor visualização, os dados foram apresentados na Tabela 6. Pode-se perceber que o Controle por Rastreamento Geométrico possui um Erro Quadrático Médio consideravelmente menor quando é necessário maiores velocidades linear e angular. Porém, em baixas velocidades, o Controlador do Vetor Empuxo Desejado apresenta uma resposta bem próxima.

Tabela 6 – Comparação do Erro Quadrático Médio entre os três controladores utilizados

Trajectoria	Controlador	Erro Quadrático Médio [m^2]
Trajetória 1 - 5s	Controlador para Pequenos Ângulos	0.003659
	Controlador do Vetor Empuxo Desejado	0.000072
	Controle Por Rastreamento Geométrico	0.000022
Trajetória 1 - 2s	Controlador para Pequenos Ângulos	0.376825
	Controlador do Vetor Empuxo Desejado	0.044599
	Controle Por Rastreamento Geométrico	0.001320
Trajetória 2	Controlador para Pequenos Ângulos	0.123305
	Controlador do Vetor Empuxo Desejado	0.023816
	Controle Por Rastreamento Geométrico	0.001526
Trajetória 3	Controlador para Pequenos Ângulos	∞
	Controlador do Vetor Empuxo Desejado	1.745761
	Controle Por Rastreamento Geométrico	0.022389

6.5 TEMPO DE PROCESSAMENTO

A presente seção apresentará o tempo de processamento de cada controlador utilizado neste trabalho.

Os controladores foram implementados na placa Arduino Mega 2560, a qual possui um ATmega2560 com uma SRAM de 8kB, memória Flash de 256kB e um cristal oscilador de 16MHz. Os tempos de processamento médio de cada iteração são apresentados na Tabela 7.

Tabela 7 – Tempo de processamento de cada iteração dos controladores na placa Arduino Mega 2560

Controlador	Tempo [ms]
Controlador para Pequenos Ângulos	1,53
Controlador do Vetor Empuxo Desejado	4,85
Controle Por Rastreamento Geométrico	6,19

7 CONCLUSÕES

O presente trabalho teve como foco a apresentação teórica de diversas partes do equacionamento de um quadricóptero. Primeiro, apresentou-se uma base teórica suficiente para o desenvolvimento de equações que representam a dinâmica de corpos rígidos. Em seguida, apresentou-se as equações de diferentes métodos de controle, além de métodos de ajuste de ganhos. Depois, descreveu-se passo-a-passo um algoritmo de planejamento de trajetória ótima. Por fim, apresentou-se diversos resultados de simulações feitas utilizando a teoria descrita neste trabalho.

Como visto no Capítulo 6, o melhor controlador, dentre os três citados, é o Controle por Rastreamento Geométrico. Ele apresentou o menor erro RMS entre as curvas desejada e realizada em todas as trajetórias simuladas. Pode-se perceber que, mesmo com uma rotação inicial muito grande (Trajetória 3: *roll angle* inicial igual a 178°), o controlador consegue convergir para a trajetória desejada. O mesmo não pode ser dito com relação aos outros dois controladores: o Controle para Pequenos Ângulos diverge, enquanto que o Controle do Vetor Empuxo Desejado converge ao ponto final, porém com uma trajetória com uma alta oscilação, diferente da trajetória desejada.

Observando as curvas realizadas pelo sistema para as Trajetórias 1 e 2 utilizando o Controle para Pequenos Ângulos, nota-se que, quando a trajetória requer grandes ângulos e altas velocidades angulares, o controlador falha em segui-la. Porém, com pequenos ângulos e baixas velocidades angulares, o controlador possui uma boa resposta, como é observado na Trajetória 1 com T_F igual a 5 segundos.

8 TRABALHOS FUTUROS

Para trabalhos futuros, pode-se desenvolver modelos matemáticos mais acurados para as dinâmicas do quadricóptero, considerando, entre outras coisas, a dinâmica dos motores e a resistência do ar. Além disso, diferentes controladores podem ser estudados e comparados, como *Backstepping* e controladores inteligentes. Na etapa de planejamento de trajetórias, pode-se estudar algoritmos que consideram obstáculos no espaço. Como dito no Capítulo 5, outras funcionalidades para o ambiente de simulação desenvolvido estão em fase de implementação, tais como o suporte para múltiplos quadricópteros, comunicação com o controlador via Robotics Operating System (ROS) e implementação de obstáculos.

Além de todos os itens citados, pode-se também terminar o desenvolvimento do controlador de voo baseado na plataforma Arduino.

REFERÊNCIAS

- [1] DECEA. Sistemas de aeronaves remotamente pilotadas e o acesso ao espaço aéreo brasileiro. <https://publicacoes.decea.gov.br/?i=publicacaooid=4510>, Março 2017.
- [2] BRANDÃO, A. S.; PIZETTA, I. H.; SARCINELLI-FILHO, M.; CARELLI, R. Modelagem e controle não linear subatuado de um quad-rotor: Parte 1. In: . c2012. p. 3102–3109.
- [3] SHKOLNIK, A.; TEDRAKE, R. High-dimensional underactuated motion planning via task space control. In: . c2008. p. 3762–3768.
- [4] DE SÃO PAULO, F. Com mais de 700 empresas, setor de drones vive aquecimento. <http://www1.folha.uol.com.br/mercado/2017/05/1887314-com-mais-de-700-empresas-setor-de-drones-vive-aquecimento.shtml>, Maio 2017.
- [5] DRONDRONES. Você sabe a diferença entre vant, drone e rpas? <https://www.dronedrones.com.br/single-post/2017/01/06/Você-sabe-a-diferença-entre-VANT-DRONE-e-RPAS>, Janeiro 2017.
- [6] ALVES, A. S. C. et al. Estudo e aplicação de técnicas de controle embarcadas para estabilização de vôo de quadricópteros. 2012.
- [7] PAW, Y. C.; BALAS, G. J. Development and application of an integrated framework for small uav flight control development. *Mechatronics*, v. 21, n. 5, p. 789–802, 2011.
- [8] COZA, C.; MACNAB, C. A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization. In: . c2006. p. 454–458.
- [9] DIKMEN, I. C.; ARISOY, A.; TEMELTAS, H. Attitude control of a quadrotor. In: . c2009. p. 722–727.
- [10] HOFFMANN, G. M.; HUANG, H.; WASLANDER, S. L.; TOMLIN, C. J. Quadrotor helicopter flight dynamics and control: Theory and experiment. In: . c2007. v. 2. p. 4.
- [11] RAWASHDEH, O. A.; YANG, H. C.; ABOUSLEIMAN, R. D.; SABABHA, B. H. Microraptor: A low-cost autonomous quadrotor system. In: . c2009.
- [12] DOMINGUES, J. M. B. Quadrotor prototype. *Universidade Tecnica de Lisboa. Dissertacio*, 2009.
- [13] MINH, L. D.; HA, C. Modeling and control of quadrotor mav using vision-based measurement. In: . c2010. p. 70–75.
- [14] BOUABDALLAH, S.; SIEGWART, R. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In: . c2005. p. 2247–2252.
- [15] ADIGBLI, P.; GR, C.; MOURET, J.-B.; DONCIEUX, S. Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques. 2007.

- [16] AL-YOUNES, Y.; JARRAH, M. Attitude stabilization of quadrotor uav using backstepping fuzzy logic & backstepping least-mean-square controllers. In: . c2008. p. 1–11.
- [17] RABHI, A.; CHADLI, M.; PEGARD, C. Robust fuzzy control for stabilization of a quadrotor. In: . c2011. p. 471–475.
- [18] NICOL, C.; MACNAB, C.; RAMIREZ-SERRANO, A. Robust neural network control of a quadrotor helicopter. In: . c2008. p. 001233–001238.
- [19] WASLANDER, S. L.; HOFFMANN, G. M.; JANG, J. S.; TOMLIN, C. J. Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In: . c2005. p. 3712–3717.
- [20] COSTA, E. B. et al. Algoritmos de controle aplicados a estabilizaçao do voo de um quadrotor. 2012.
- [21] MELLINGER, D. *Trajectory generation and control for quadrotors*. University of Pennsylvania, 2012.
- [22] LEE, T.; LEOKY, M.; MCCLAMROCH, N. H. Geometric tracking control of a quadrotor uav on se (3). In: . c2010. p. 5420–5425.
- [23] LEE, T.; LEOK, M.; MCCLAMROCH, N. H. Control of complex maneuvers for a quadrotor uav using geometric methods on se (3). *arXiv preprint arXiv:1003.2005*, 2010.
- [24] RENDÓN, M. A.; MARTINS, F. F. Path following control tuning for an autonomous unmanned quadrotor using particle swarm optimization. *IFAC-PapersOnLine*, v. 50, n. 1, p. 325–330, 2017.
- [25] NETO, A. A. Geração de trajetórias para veículos aéreos autônomos não-tripulados. 2008.
- [26] GONÇALVES, V. M.; PIMENTA, L. C.; MAIA, C. A.; PEREIRA, G. A.; DUTRA, B. C.; MICHAEL, N.; FINK, J.; KUMAR, V. Circulation of curves using vector fields: actual robot experiments in 2d and 3d workspaces. In: . c2010. p. 1136–1141.
- [27] MELLINGER, D.; KUMAR, V. Minimum snap trajectory generation and control for quadrotors. In: . c2011. p. 2520–2525.
- [28] WENDEL, A.; IRSCHARA, A.; BISCHOF, H. Natural landmark-based monocular localization for mavs. In: . c2011. p. 5792–5799.
- [29] WENDEL, A.; MAURER, M.; GRABER, G.; POCK, T.; BISCHOF, H. Dense reconstruction on-the-fly. In: . c2012. p. 1450–1457.
- [30] KAISER, G. *A friendly guide to wavelets*. Springer Science & Business Media, 2010.
- [31] CASAZZA, P. G.; KUTYNIOK, G.; PHILIPP, F. Introduction to finite frame theory. In: *Finite Frames*. Springer, 2013. p. 1–53.
- [32] BEARD, R. Quadrotor dynamics and control rev 0.1. 2008.

- [33] ÇİMEN, T. Flight dynamics, simulation and control with matlab and simulink. Workshop at International Federation of Automatic Control, 2017.
- [34] SUTTON, G. P.; BIBLARZ, O. *Rocket propulsion elements*. John Wiley & Sons, 2017.
- [35] FRESK, E.; NIKOLAKOPOULOS, G. Full quaternion based attitude control for a quadrotor. In: . c2013. p. 3864–3869.
- [36] TAYEBI, A.; MCGILVRAY, S. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on control systems technology*, v. 14, n. 3, p. 562–571, 2006.
- [37] RODRIGUEZ, O. Des lois geometriques qui regissent les déplacements d’un systeme solide dans l’espace et de la variation des coordonnees provenant de déplacements consideres independamment des causes qui peuvent les produire. *J Math Pure Appl*, v. 5, p. 380–440, 1840.
- [38] STEVENS, B. L.; LEWIS, F. L. Aircraft control and simulation, 2003.
- [39] THECURIUSASTRONOMER. Derivation of coriolis force. <https://thecuriousastronomer.wordpress.com/tag/derivation-of-coriolis-force/>, Novembro 2017.
- [40] BOUABDALLAH, S.; NOTH, A.; SIEGWART, R. Pid vs lq control techniques applied to an indoor micro quadrotor. In: . c2004. v. 3. p. 2451–2456.
- [41] FOURATI, H.; BELKHIAT, D. E. C. *Multisensor attitude estimation: Fundamental concepts and applications*. CRC Press, 2016.
- [42] BULLO, F.; LEWIS, A. D. *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Springer Science & Business Media, 2004. v. 49.
- [43] KENNEDY, J. Particle swarm optimization. In: *Encyclopedia of machine learning*. Springer, 2011. p. 760–766.
- [44] HOUCK, C. R.; JOINES, J.; KAY, M. G. A genetic algorithm for function optimization: a matlab implementation. *Ncsu-ie tr*, v. 95, n. 09, 1995.
- [45] CAETANO, E. C. Aplicações das equações de euler-lagrange. 2006.
- [46] CONCEITODE. Conceito de simulação. <https://conceito.de/simulacao>, Março 2017.

APÊNDICE A – Planejamento de Trajetória para n Pontos

Para que o algoritmo descrito no Capítulo 4 seja implementado para uma sequência de pontos pré-definidos, deve-se seguir o pseudocódigo descrito no Algoritmo 1.

Algorithm 1 Algoritmo de Planejamento de Trajetória para uma sequência de pontos pré-definidos

```

1: função PLANEJAMENTO_DE_TRAJETORIA_COM_SNAP_MINIMO
2:    $n \leftarrow$  numero de pontos pre-definidos
3:    $x_0, x_1, \dots, x_n \leftarrow$  posições pré-definidas
4:    $v_0, v_n \leftarrow$  velocidades inicial e final pré-definidas
5:    $a_0, a_n \leftarrow$  acelerações inicial e final pré-definidas
6:    $j_0, j_n \leftarrow$  jerks inicial e final pré-definidos
7:    $E \leftarrow$  matriz de zeros  $(8(n-1) \times 8(n-1))$ 
8:    $X \leftarrow$  vetor de zeros  $(8(n-1) \times 1)$ 
9:    $E(1 : 4, 1 : 8) \leftarrow$  equações da primeira curva referentes ao ponto inicial
10:   $X(1 : 4, 1) \leftarrow$  restrições da primeira curva referentes ao ponto inicial
11:   $E(8(n-1)-3 : 8(n-1), 8(n-1)-7 : 8(n-1)) \leftarrow$  equações da
    última curva referentes ao ponto final
12:   $X(8(n-1)+1-4, 8(n-1)) \leftarrow$  restrições da última curva referentes ao ponto final
13:  se  $n == 2$  então
14:    devolve  $E^{-1}X$ 
15:  fim se
16:  para  $i \leftarrow 2$  até  $n-1$  faça // Para cada uma das curvas restantes
17:     $E(8(i-1)-3 : 8(i)-4, 8(i-1)+1 : 8(i)) \leftarrow$  equações dos pontos intermediários
18:     $X(8(i-1)-3 : 8(i)-4) \leftarrow$  restrições da primeira curva referentes ao ponto inicial
19:  fim para
20:  devolve  $E^{-1}X$ 
21: fim função

```

Um exemplo de aplicação do Algoritmo 1 utilizando a Tabela 1 para uma curva com três pontos pré-definidos está destacado abaixo. Escolheu-se $T_0 = 0$, $x_0 = a$, $x_1 = b$ e $x_2 = c$.

$$\begin{bmatrix} c_{7,1,1} \\ c_{6,1,1} \\ c_{5,1,1} \\ c_{4,1,1} \\ c_{3,1,1} \\ c_{2,1,1} \\ c_{1,1,1} \\ c_{0,1,1} \\ c_{7,1,2} \\ c_{6,1,2} \\ c_{5,1,2} \\ c_{4,1,2} \\ c_{3,1,2} \\ c_{2,1,2} \\ c_{1,1,2} \\ c_{0,1,2} \end{bmatrix} = E^{-1} \begin{bmatrix} a \\ 0 \\ 0 \\ 0 \\ 0 \\ b \\ b \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ c \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

onde

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_1^7 & T_1^6 & T_1^5 & T_1^4 & T_1^3 & T_1^2 & T_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_1^7 & T_1^6 & T_1^5 & T_1^4 & T_1^3 & T_1^2 & T_1 & 1 & 1 \\ 7T_1^6 & 6T_1^5 & 5T_1^4 & 4T_1^3 & 3T_1^2 & 2T_1 & 1 & 0 & -7T_1^6 & -6T_1^5 & -5T_1^4 & -4T_1^3 & -3T_1^2 & -2T_1 & -1 & 0 & 0 \\ 42T_1^5 & 30T_1^4 & 20T_1^3 & 12T_1^2 & 6T_1 & 2 & 0 & 0 & -42T_1^5 & -30T_1^4 & -20T_1^3 & -12T_1^2 & -6T_1 & -2 & 0 & 0 & 0 \\ 210T_1^4 & 120T_1^3 & 60T_1^2 & 24T_1 & 6 & 0 & 0 & 0 & -210T_1^4 & -120T_1^3 & -60T_1^2 & -24T_1 & -6 & 0 & 0 & 0 & 0 \\ 840T_1^3 & 360T_1^2 & 120T_1 & 24 & 0 & 0 & 0 & 0 & -840T_1^3 & -360T_1^2 & -120T_1 & -24 & 0 & 0 & 0 & 0 & 0 \\ 2520T_1^2 & 720T_1 & 120 & 0 & 0 & 0 & 0 & 0 & -2520T_1^2 & -720T_1 & -120 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5040T_1 & 720 & 0 & 0 & 0 & 0 & 0 & 0 & -5040T_1 & -720 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_2^7 & T_2^6 & T_2^5 & T_2^4 & T_2^3 & T_2^2 & T_2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7T_2^6 & 6T_2^5 & 5T_2^4 & 4T_2^3 & 3T_2^2 & 2T_2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 42T_2^5 & 30T_2^4 & 20T_2^3 & 12T_2^2 & 6T_2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 210T_2^4 & 120T_2^3 & 60T_2^2 & 24T_2 & 6 & 0 & 0 & 0 & 0 \end{bmatrix}.$$