

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE ELETRÔNICA
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

FELIPE RODRIGUES DE VARGAS FERREIRA

**MODELAGEM, CONTROLE OTIMIZADO POR ALGORITMO GENÉTICO E
APLICAÇÃO DE *CELL-MAPPING* EM UM QUADRICÓPTERO**

TRABALHO DE CONCLUSÃO DE CURSO

**PONTA GROSSA
2017**

FELIPE RODRIGUES DE VARGAS FERREIRA

**MODELAGEM, CONTROLE OTIMIZADO POR ALGORITMO GENÉTICO E
APLICAÇÃO DE *CELL-MAPPING* EM UM QUADRICÓPTERO**

Trabalho de conclusão de curso apresentado ao programa de Graduação em Engenharia Eletrônica do Departamento de Eletrônica do Câmpus Ponta Grossa da UTFPR como requisito parcial para a obtenção do título de bacharel em Engenharia Eletrônica.

Orientador: Prof. Dr. Maurício dos Santos Kaster

**PONTA GROSSA
2017**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Engenharia Elétrica
Mestrado em Engenharia Elétrica



TERMO DE APROVAÇÃO

Modelagem, Controle Otimizado por Algoritmo Genético e Aplicação de *Cell-Mapping* em um
Quadricíptero

por

Felipe Rodrigues de Vargas Ferreira

Este trabalho de conclusão de curso foi julgado adequada para obtenção do Título de "Engenheiro Eletrônico" e aprovado em sua forma final pelo Departamento de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Ponta Grossa, 06/12/2017.

Prof. Dr. Jeferson Jose Gomes
Coordenador do Curso de Graduação em Engenharia

Banca Examinadora:

Prof. Dr. Maurício dos Santos Kaster
Orientador

Prof^a. Dr^a. Fernanda Cristina Correa - (UTFPR)

Prof. Dr. Frederic Conrad Janzen - (UTFPR)

"A folha de aprovação assinada encontra-se no Departamento de Registros Acadêmicos"

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, em especial minha mãe, Maria Apolônia, por sempre ter incentivado os meus estudos e por todo o apoio fornecido durante o meu desenvolvimento pessoal e profissional.

Agradeço também aos meus familiares, dentre eles minhas irmãs Juliana Vargas e Vanessa Vargas, que ajudaram em muito na minha criação e tiveram um reflexo muito positivo em minha vida.

Agradeço ao professor Doutor Maurício dos Santos Kaster pela orientação durante esses 2 anos de trabalho, obrigado por toda a paciência e dedicação ao projeto.

Agradeço aos meus colegas Rodrigo Klosowski e Fernando Constantino por todos os bons momentos que passamos juntos durante esse ano trabalhando juntos, tanto nos projetos de faculdade, quanto em outros empreendimentos.

Agradeço a todos os outros colegas com os quais eu dividi bons momentos e que sempre me apoiaram, dentre eles Guilherme Bren e Thalmó de Sá.

Agradeço também a minha namorada Talita Carvalho, por todo o apoio e paciência durante o tempo em que estive desenvolvendo esse trabalho.

Também gostaria de agradecer a UTFPR e a todos os grandes professores que tive durante o período da graduação, cada um deles me ensinou algo único, que eu irei levar para o resto da minha vida, obrigado por terem tornado o meu sonho possível.

RESUMO

Ferreira, Felipe Rodrigues de Vargas. **Modelagem , Controle Otimizado por Algoritmo Genético e Aplicação de *Cell-Mapping* em um Quadricóptero.** 2017. 62 Trabalho de Conclusão de Curso – Departamento de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

Veículos como os quadricópteros se tornam cada vez mais estudados e aplicados em diversas áreas, além das aplicações militares, os drones são utilizados para outras finalidades como topografia, fotografia, cinematografia e segurança. Esse trabalho propõe-se a realizar uma modelagem matemática de um quadricóptero, aplicando a técnica de controle PID (proporcional-integrativo-derivativo) para a estabilização dos ângulos e da posição vertical, controladores esses que foram otimizados com o uso de algoritmos genéticos, e, finalmente, a obtenção de um *cell-mapping* (CM) para a malha da altura já controlada. A modelagem matemática foi realizada utilizando o método de Newton-Euler, separando a planta em dois sub-sistemas interligados onde cada um é responsável por descrever os comportamentos angulares (sistema rotacional) e lineares (sistema translacional) do modelo. Foram propostas duas funções custo no desenvolvimento do AG para que elas pudessem ser comparadas em relação aos ganhos obtidos ao final da simulação. Por fim, é feita a aplicação de um CM para a análise da resposta do sistema controlado em um mapa de estados finitos, ainda é proposta uma aplicação preliminar de controle feita com base na resposta obtida pelo mapeamento.

Palavras-chave: Modelagem de quadricóptero. Controle linear. Algoritmo Genético. *Cell-Mapping*.

ABSTRACT

Ferreira, Felipe Rodrigues de Vargas. **Modeling, Optimized Control by Genetic Algorithm and Application of Cell-Mapping in a Quadricopter.** 2017. 62 s. Graduation Program in Eletronics Engineering, Federal University of Technology, Paraná. Ponta Grossa, 2017.

Vehicles like the quadricopter become increasingly studied and applied in several areas, besides military applications, the drones are used for other purposes like topography, photography, cinematography and security. This end-of-course monography proposes to perform a mathematical modeling of a quadricopter, applying PID (proportional–integral–derivative) control technique to stabilize the angles and the vertical position, which were optimized by using genetic algorithms, and, finally, obtained a cell-mapping (CM) for the already controlled height loop. The mathematical modeling was performed using the Newton-Euler method, separating the plant into two interconnected subsystems where each one is responsible for describing the angular (rotational system) and linear (translational system) behaviors of the model. Two cost functions were proposed in the development of the GA so that they could be compared in relation to the gains obtained at the end of the simulation. Finally, a CM is applied to the analysis of the response of the controlled system in a map of finite states, an preliminary application of control based on the response obtained by the mapping is proposed too.

Keywords: Quadricopter modeling. Linear control. Genetic Algorithm. Cell-Mapping.

LISTA DE FIGURAS

Figura 1 – <i>Aphrodite B-17F</i>	16
Figura 2 – <i>MQ-1 Predator</i>	17
Figura 3 – Empuxo Vertical	19
Figura 4 – Rolagem	19
Figura 5 – Arfagem	20
Figura 6 – Guinada	20
Figura 7 – Planos lineares do quadricóptero e da terra.	21
Figura 8 – Diagrama representativo de uma malha fechada com um PID.	31
Figura 9 – Exemplo de mapa unidimensional.	36
Figura 10 – Comparação de mapas utilizando 25 (esquerda) e 100 (direita) condições iniciais (2D).	37
Figura 11 – Comparação de mapas utilizando 25 (esquerda) e 100 (direita) condições iniciais (3D).	37
Figura 12 – Representação de uma célula no CM.	38
Figura 13 – Fluxograma proposto para o Algoritmo Genético.	41
Figura 14 – Evolução do <i>fitness</i> para diferentes valores de mutação.	44
Figura 15 – Diagrama representativo dos testes em malha aberta.	46
Figura 16 – Empuxo Vertical em malha aberta.	47
Figura 17 – Rolagem em malha aberta.	47
Figura 18 – Guinada em malha aberta.	48
Figura 19 – Arfagem em malha aberta.	48
Figura 20 – Evolução do <i>fitness</i> na melhor rodada (ISE).	49
Figura 21 – Evolução do <i>fitness</i> na melhor rodada (ITSE).	51
Figura 22 – Diagrama de blocos para o controle da estabilidade.	52
Figura 23 – Comparação entre os melhores indivíduos para as duas funções de custo (ϕ e θ).	53
Figura 24 – Comparação entre os melhores indivíduos para as duas funções de custo (ψ).	53
Figura 25 – Diagrama de blocos para o controle da altura.	54
Figura 26 – Comparação entre os melhores indivíduos para as duas funções de custo (Z).	55
Figura 27 – Variação da velocidade angular dos motores durante o período de simulação.	56
Figura 28 – CM para a malha da altura controlada.	57
Figura 29 – Variação do esforço de controle no CM.	57
Figura 30 – Comparação entre os controladores.	58

LISTA DE TABELAS

Tabela 1 – Constantes da simulação.	46
Tabela 2 – Constantes de inicialização do AG.	48
Tabela 3 – Valores obtidos para os ganhos do PID após otimização via AG (ISE).	49
Tabela 4 – Valores obtidos para os ganhos do PID após otimização via AG (ITSE).	50
Tabela 5 – Valores finais para a simulação dos controladores PD.	52
Tabela 6 – Valores para análise da performance dos controladores obtidos.	54
Tabela 7 – Valores para análise da performance dos controladores obtidos.	55
Tabela 8 – Constantes de inicialização do CM.	56

LISTA DE SIGLAS

DAELE	Departamento Acadêmico de Eletrônica
AG	Algoritmo Genético
UTFPR	Universidade Tecnológica Federal do Paraná
VANT	Veículo Aéreo não Tripulado
PID	Proporcional, Integrativo e Derivativo
PD	Proporcional e Derivativo
ESC	<i>Electronic Speed Controller</i> (em português, Controlador Eletrônico de Velocidade)
UAV	<i>Unmanned Aerial Vehicle</i> (em português, Veículo Aéreo não Tripulado)
ANAC	Agência Nacional de Aviação Civil
DECEA	Departamento de Controle do Espaço Aéreo
ANATEL	Agência Nacional de Telecomunicações
RPA	<i>Remotely Piloted Aircrafts</i> (Aeronaves Remotamente Pilotadas)
V	Volts (unidade de medida de tensão elétrica)
CM	<i>Cell Mapping</i> (Mapeamento Celular)
A	Ampère (unidade de medida de corrente elétrica)
W	Watts (unidade de medida de potência elétrica)
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i> (Semicondutor de Metal-Óxido Complementar)

LISTA DE SÍMBOLOS

Ω_r	Velocidade relativa do drone [rad/s]
Ω_n	Velocidade angular do motor n [rad/s]
ω	Vetor velocidade em relação ao plano B (ponto de referência sobre o quadricóptero) [rad/s]
J	Matriz diagonal de inércia [kg·m ²]
R	Matriz de rotação
J_r	Inércia do motor [kg·m ²]
T	Matriz de translação
$\dot{\eta}$	Aceleração angular [rad/s ²]
K_f	Constante de força aerodinâmica
K_M	Constante de momento
E	Plano inercial posicionado na Terra
B	Plano inercial posicionado no corpo do drone
U_1	Empuxo Vertical [Nm]
U_2	Rolamento [Nm]
U_3	Arfagem [Nm]
U_4	Guinada [Nm]
ϕ	Ângulo no eixo X do drone [rad]
θ	Ângulo no eixo Y do drone [rad]
ψ	Ângulo no eixo Z do drone [rad]
k_p	Ganho proporcional do controlador PID
k_i	Ganho integrativo do controlador PID
k_d	Ganho derivativo do controlador PID
SP	<i>Setpoint</i> (Ponto de estabilização)
v_x	Variável de controle

SUMÁRIO

1	INTRODUÇÃO	13
1.1	JUSTIFICATIVA	13
1.2	DELIMITAÇÃO DO TEMA	14
1.3	PROBLEMAS E PREMISSAS	14
1.3.1	Problemas	14
1.3.2	Premissas	14
1.4	OBJETIVO	14
1.4.1	Objetivos Específicos	15
1.5	VISÃO GERAL DO TRABALHO	15
2	REVISÃO DE LITERATURA	16
2.1	CONTEXTO HISTÓRICO	16
2.2	QUADRICÓPTERO	18
2.2.1	Empuxo Vertical	18
2.2.2	Rolagem	19
2.2.3	Arfagem	19
2.2.4	Guinada	20
3	MODELO MATEMÁTICO	21
3.1	MODELO CINEMÁTICO	21
3.2	MODELO DINÂMICO	24
3.2.1	Rotacional	24
3.2.2	Translacional	26
3.3	MODELO EM ESPAÇO DE ESTADOS	26
4	CONTROLE	31
4.1	PID	31
4.1.1	Saída do Controlador	32
4.1.2	Saturação	33
4.2	CELL MAPPING	33
4.2.1	Granularidade	35
4.2.2	Limites	36
4.2.3	Condições Iniciais	36
4.2.4	Armazenamento do Esforço de Controle	36
5	OTIMIZAÇÃO	39
5.1	ALGORITMO GENÉTICO	39
5.1.1	Etapas do Algoritmo Genético Clássico	40
5.1.2	<i>Fitness</i>	41
5.1.3	População Inicial	42
5.1.4	Seleção e <i>Crossover</i>	42
5.1.5	Mutação	43
5.2	FUNÇÃO CUSTO	43
6	SIMULAÇÕES	45
6.1	MALHA ABERTA	45

6.2	ALGORITMO GENÉTICO (AG)	47
6.2.1	ISE (<i>Integral Squared Error</i>)	48
6.2.2	ITSE (<i>Integral Time-Weighted Squared Error</i>)	50
6.2.3	Análise das Respostas	50
6.3	CONTROLE DA ESTABILIDADE	50
6.3.1	Análise das Respostas	52
6.4	CONTROLE DA POSIÇÃO VERTICAL	54
6.4.1	Análise das Respostas	54
6.5	ESFORÇO DE CONTROLE	55
6.6	CELL MAPPING	56
6.6.1	Controle utilizando o mapa com os esforços de controle	58
7	CONCLUSÃO	60
	REFERÊNCIAS	61

1 INTRODUÇÃO

Quando consideramos a aplicação de VANT's (Veículos Aéreos Não Tripulados) nos dias atuais, tanto em um ponto de vista acadêmico quanto comercial, é possível ver que esse tipo de tecnologia está cada vez mais sendo utilizada. Algumas de suas aplicações mais comuns incluem atividades militares, serviços de entrega, aquisições de imagens aéreas, serviços de topografia e mapeamento, entre outras (CONSTANTINO; KLOSOWSKI, 2017).

Pode-se chamar um VANT que possua 4 motores de quadrirrotor (ou quadricóptero) e alguns tipos de drones, incluindo o estudado nesse trabalho, também se encaixam nessa denominação, portanto durante o trabalho esses termos serão considerados como sinônimos.

Decorrente das demandas impostas, tecnologias envolvendo voos autônomos, controle de estabilidade e autonomia de voo tornaram-se objetivos de estudo relevantes. Assim, neste trabalho de conclusão de curso pretende-se elaborar um sistema de controle de estabilidade de voo, utilizando controladores e técnicas de otimização usando meta-heurísticas, além disso, é proposto o início de uma metodologia de *Cell Mapping* (CM) para o modelo escolhido, bem como uma análise dos dados obtidos via mapeamento.

A modelagem matemática do protótipo foi realizada utilizando o método de Newton-Euler onde o sistema é separado em dois menores, cada um responsável pela descrição de uma das partes do modelo. O primeiro subsistema seria o rotacional, que é responsável pela descrição matemática do comportamento dos ângulos, o segundo é o translacional, que diz respeito às posições lineares do modelo, com essa subdivisão fica mais fácil perceber as dependências do sistema.

No trabalho também é proposto o controle da estabilidade de voo, ou seja, dos ângulos descritos pelo sistema rotacional, bem como o controle da altura do modelo. Em todos os casos foram propostos controladores PID otimizados utilizando algoritmo genético, além disso é estudada uma metodologia para a aplicação de *cell-mapping* para a análise da malha de altura já controlada.

1.1 JUSTIFICATIVA

As justificativas do trabalho podem ser definidas como:

- Relevância do tema - Existem diversas aplicações práticas para quadricópteros, e por serem sistemas autônomos de voo são totalmente dependentes de controladores bem implementados que consigam suprir as necessidades de deslocamento do drone.
- Atualidade - Com a popularização dos quadricópteros esse é um tema muito visado atualmente, além disso técnicas como algoritmo genético e CM começaram a ser utilizadas atualmente para otimização e análise/controle de sistemas, respectivamente, tendo demonstrado bons resultados.

- Interesse do autor - Além da planta em estudo ser um tema muito interessante, a aplicação de CM para a análise de sistemas é uma técnica que possibilita diversas aplicações diferentes no futuro.

1.2 DELIMITAÇÃO DO TEMA

O tema desta pesquisa consiste em realizar a modelagem matemática do drone, desenvolver controladores PID para a estabilidade dos ângulos e o controle da altura, realizar a otimização dos controladores com Algoritmo Genético (AG) e por fim o desenvolvimento de um algoritmo para a geração de um *Cell Mapping* a partir das equações de estados da planta já controlada para a análise dos dados obtidos.

1.3 PROBLEMAS E PREMISSAS

1.3.1 Problemas

1. O drone é um sistema complexo, dotado de pelo menos 6 graus de liberdade e sujeito a muitas perturbações, pois seu movimento se dá no ar, portanto, a modelagem de tal sistema é bastante complexa.
2. Embora já tenha sido aplicado por vários pesquisadores, a compreensão e aplicação do cell-mapping e seu uso em controle ainda é uma novidade dentro dos grupos de pesquisa da Universidade Tecnológica Federal do Paraná (UTFPR).
3. Não se tinha certeza de que seria possível alcançar algum resultado satisfatório tanto no mapeamento de um controle tradicional quanto na tentativa de aplicar o mapeamento como controle.

1.3.2 Premissas

1. Não será considerado o momento giroscópico para as simulações.
2. O desenvolvimento obedecerá etapas: primeiro é feita a aplicação em apenas um eixo, no caso o eixo Z. Posteriormente, serão adicionados os demais eixos/graus de liberdade, conforme a disponibilidade de tempo do trabalho.
3. A metaheurística usada será o algoritmo genético.

1.4 OBJETIVO

Desenvolver a modelagem e o controle de um quadricóptero, bem como um algoritmo de CM genérico, que realize uma análise sobre o sistema controlado para que se possa obter as trajetórias do sistema para que uma análise em espaço de estados possa ser feita na planta em estudo.

1.4.1 Objetivos Específicos

Constituem-se objetivos deste trabalho:

- Elaborar a modelagem matemática de um quadricóptero utilizando a segunda lei de Newton e a equação de Newton-Euler;
- Desenvolver controladores PID para a posição vertical e estabilidade;
- Definir os parâmetros para a estruturação de um algoritmo de CM;
- Otimizar os controladores utilizando Algoritmo Genético;
- Simular o sistema controlado e os algoritmos utilizando MATLAB;
- Analisar a resposta obtida com o mapa das células e da transição dos estados;

1.5 VISÃO GERAL DO TRABALHO

A descrição e estrutura utilizada neste trabalho está de acordo com a divisão nos seguintes capítulos:

No capítulo 2 foi realizada uma revisão de literatura sobre o Quadricóptero, um contexto histórico é apresentado, seguido pelas normas brasileiras que regulamentam o uso de drones e por fim são apresentados os movimentos que um drone pode realizar.

No capítulo 3, a modelagem matemática da planta é realizada, separando-a em dois sistemas interligados para que o estudo possa ser feito de uma maneira simplificada, ao final do capítulo são extraídas as equações de espaço de estados para a planta.

No capítulo 4 o controlador PID é estudado bem como técnicas de digitalização e conversão de sinais, também é descrita a metodologia para a aplicação de CM na planta controlada.

No capítulo 5 a técnica de otimização (Algoritmo Genético) é apresentada, com uma descrição das etapas do processo de implementação do algoritmo.

No capítulo 6 são mostrados os resultados obtidos na simulação da planta em malha aberta, do AG, da planta controlada e do CM.

Por último, conclui-se o projeto, os resultados obtidos são discutidos e são feitas proposições para trabalhos futuros.

2 REVISÃO DE LITERATURA

O capítulo de revisão de literatura será focado em falar sobre o modelo (quadrirotor) escolhido, as outras áreas de estudo do trabalho serão referenciadas em seus próprios capítulos. Aqui será falado sobre o contexto histórico, as normas brasileiras que regulamentam e os movimentos básicos que descrevem um drone.

2.1 CONTEXTO HISTÓRICO

As aeronaves que não necessitam de um piloto para serem guiadas são chamadas de VANT's (Veículos aéreos não tripulados, em inglês *Unmanned Aerial Vehicle*), sendo controlados puramente por meios computacionais e eletrônicos.

Algumas das principais vantagens atuais dos veículos aéreos não tripulados consistem basicamente no aumento de segurança, redução de custos de produção e a possibilidade de se realizar tarefas que um piloto humano não seria capaz. Os VANT's foram desenvolvidos com o intuito de se realizar operações militares, dado ao começo da segunda guerra mundial alguns modelos de bombardeiros não tripulados começaram a ser desenvolvidos, o *Operation Aphrodite* (Figura 1) foi um dos primeiros protótipos desenvolvidos, porém o seu projeto foi prejudicado por falta de tecnologia na época. Foram feitas 14 tentativas de lançamento, sendo que nenhuma delas teve sucesso, muitas das aeronaves perderam o controle e foram neutralizadas.

Figura 1 – *Aphrodite B-17F*



Fonte – *World War 2 Today*¹.

Somente com o avanço da tecnologia, principalmente nas áreas de eletrônica, computação e controle, foi possível o desenvolvimento dessas aeronaves, em 1995 o modelo *MQ-1 Predator* (Figura

¹ Disponível em: <http://ww2today.com/wp-content/uploads/2014/08/B-24J-55-CO.jpg>. Acesso em ago. 2017.

2), fabricado pela *General Atomics* a pedido da força aérea americana entrou em operação, utilizado pela primeira vez durante a guerra do Afeganistão.

Figura 2 – MQ-1 Predator



Fonte – Army Recognition².

À medida em que foram surgindo avanços na área do desenvolvimento de VANT's as aeronaves se tornaram muito mais acessíveis num ponto de vista financeiro, fazendo com que a tecnologia, que até o momento só tinha sido utilizada pelo exército, pudesse ser utilizada para fins comerciais e de lazer podendo ser utilizada para diversas finalidades.

Uma das áreas nas quais os VANT's vem trazendo um grande impacto é a da topografia que visa estudar como certos objetos podem ser posicionados em uma certa localidade através de estudos matemáticos fornecidos pelo mapeamento da região.

Essa técnica, muito utilizada em projetos de Engenharia Civil, pode ser usada para se realizar um estudo das características do terreno, terraplanagem e efetuar um acompanhamento mais confiável de obras ou construções, também pode ser utilizada na agricultura, onde as aplicações vão de projetos para a implantação de curvas de nível até um planejamento estratégico do plantio otimizado utilizando o conhecimento sobre as dimensões do local obtidos por meio da topografia (NETO, 2014).

Além das aplicações já citadas, os veículos também são visados em outras áreas, tanto para lazer quanto para uso profissional, dentre elas:

- Entretenimento, na forma de aeromodelos para hobistas e até mesmo brinquedos;
- Logística, com sistemas de entregas em perímetros urbanos;

² Disponível em: <https://goo.gl/ARbQWa>. Acesso em ago. 2017.

- Aplicações agrícolas, utilizados para topografia, mapeamento, segurança, controle de queimadas e pragas, além de monitoramento por meio de reconhecimento de imagem;
- Monitoramento de estradas para controle de velocidade e emissão de multas;
- Segurança, com imagens de vigilância, monitoramento e investigação.

Com o aumento no número de atividades que os VANT's conseguem desempenhar foi necessária a criação de normas para o uso das aeronaves. O Regulamento Brasileiro de Aviação Civil Especial nº 94/2017 (RBAC-E nº 94/2017) da ANAC (Agência Nacional de Aviação Civil) é uma norma regulamentadora complementar às normas de operação de drones estabelecidas pelo Departamento de Controle do Espaço Aéreo (DECEA) e pela Agência Nacional de Telecomunicações (ANATEL).

Esse regulamento categoriza os VANT's em aeromodelismo, utilizados em atividades recreativas e de lazer, e RPA's (*Remotely Piloted Aircrafts*, que significa Aeronaves Pilotadas Remotamente) que são mais utilizadas para fins comerciais. Ambas categorias apresentam limitações em relação a área de operação, limitando o seu uso em áreas com no mínimo 30 metros horizontais sem nenhuma pessoa por perto.

Para operar um aeromodelo basta respeitar a distância limite de terceiros e observar as regras do DECEA e da ANATEL. Aeromodelos com peso máximo de até 250 gramas não precisam ser cadastrados, já os operados em linha de visada visual, até 400 pés acima do nível do solo, devem ser cadastrados e, nesses casos, o piloto remoto do aeromodelo deverá possuir licença e habilitação. (ANAC, 2015)

2.2 QUADRICÓPTERO

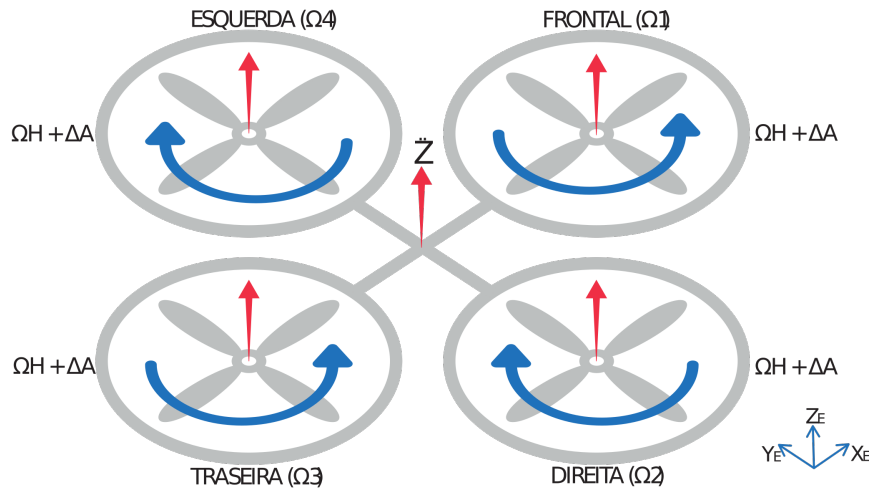
O quadricóptero, também conhecido como drone, é um VANT com quatro rotores, os quais devem ser posicionados para cima em formato de cruz, de forma equidistante do centro de massa da estrutura mecânica (LUUKKONEN, 2011).

Os propulsores traseiros e dianteiros rotacionam no sentido anti-horário e os propulsores da direita e esquerda rotacionam no sentido horário. Hélices adjacentes devem girar em sentido oposto a fim de eliminar o torque gerado pelas mesmas (BRESCIANI, 2008).

Pode-se considerar as entradas do nosso sistema como as quatro hélices. Podemos então definir quatro movimentos básicos responsáveis por fazer com que o drone se desloque pelo espaço, sendo eles empuxo vertical, rolagem, arfagem e guinada.

2.2.1 Empuxo Vertical

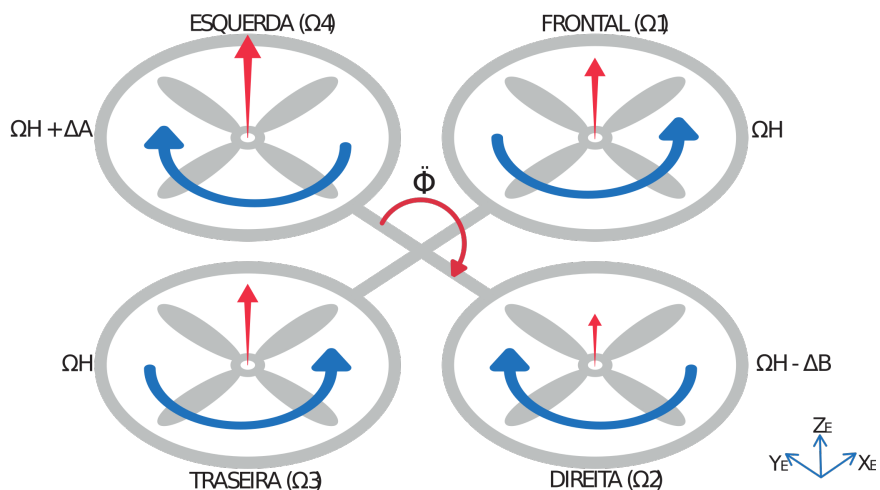
Este movimento pode ser obtido variando-se de maneira igualitária a velocidade de todos os motores. A direção da força sob o quadricóptero possui a mesma orientação do referencial do corpo B , porém em relação ao referencial inercial E , o mesmo apresenta acelerações tanto na horizontal quanto na vertical (Figura 3).

Figura 3 – Empuxo Vertical

Fonte – Modelagem e Controle de um Helicóptero Quadricóptero (CONSTANTINO; KLOSOWSKI, 2017).

2.2.2 Rolagem

Este movimento pode ser obtido aumentando-se a velocidade do rotor da esquerda e reduzindo do rotor da direita, ou vice-versa, criando um torque sob o eixo X (ângulo ϕ), responsável pelo movimento de rolagem (Figura 4).

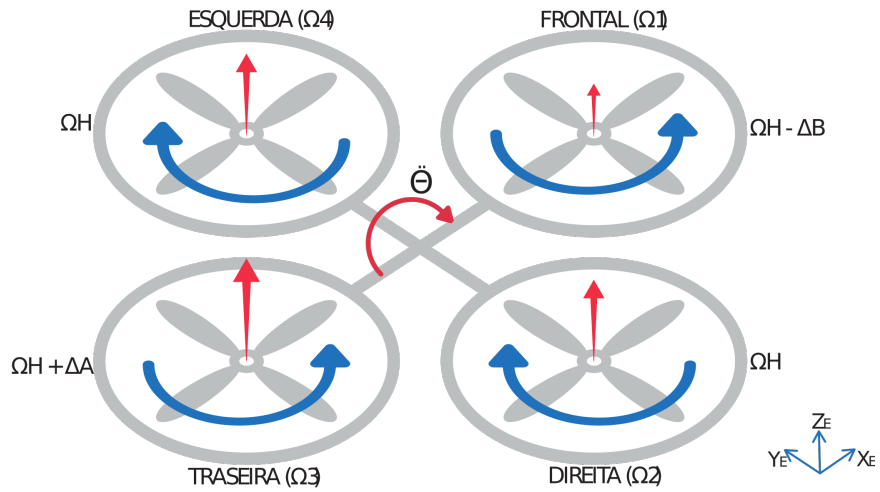
Figura 4 – Rolagem

Fonte – Modelagem e Controle de um Helicóptero Quadricóptero (CONSTANTINO; KLOSOWSKI, 2017).

2.2.3 Arfagem

Este movimento, similar à rolagem, pode ser obtido aumentando-se a velocidade do rotor frontal e reduzindo do rotor traseiro, ou vice-versa, criando um torque sob o eixo Y (ângulo θ), responsável pelo movimento de arfagem (Figura 5).

Figura 5 – Arfagem

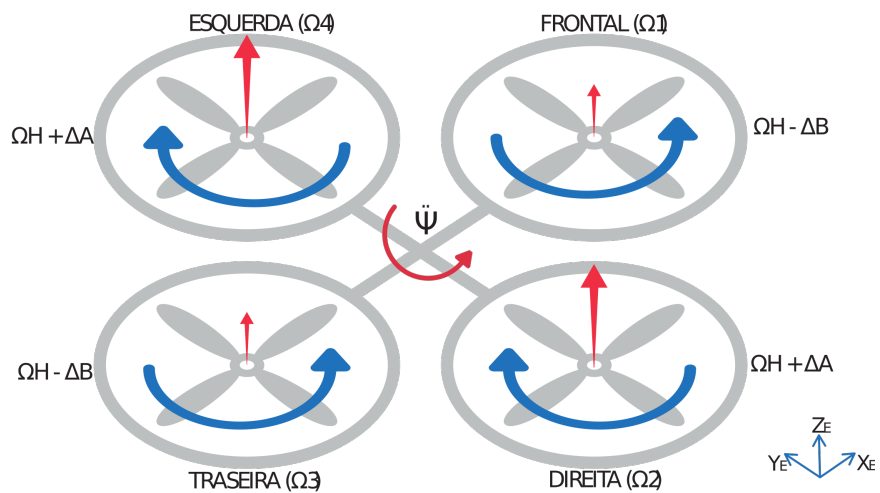


Fonte – Modelagem e Controle de um Helicóptero Quadricóptero (CONSTANTINO; KLOSOWSKI, 2017).

2.2.4 Guinada

Este movimento pode ser obtido aumentando-se simultaneamente a velocidade dos rotores da esquerda e da direita e reduzindo a velocidade dos rotores frontal e traseiro, ou vice-versa, criando um torque sob o eixo Z (ângulo ψ), responsável pelo movimento de guinada (Figura 6).

Figura 6 – Guinada



Fonte – Modelagem e Controle de um Helicóptero Quadricóptero (CONSTANTINO; KLOSOWSKI, 2017).

3 MODELO MATEMÁTICO

Aqui é feita a modelagem matemática do quadricóptero, analisando a sua cinemática e a sua dinâmica para movimentos rotacionais e translacionais. Ao final da modelagem, serão extraídas as equações de espaço de estados para o drone, que serão futuramente utilizadas na simulação (ELKHOLY, 2014).

Para a modelagem, foram feitas as seguintes considerações:

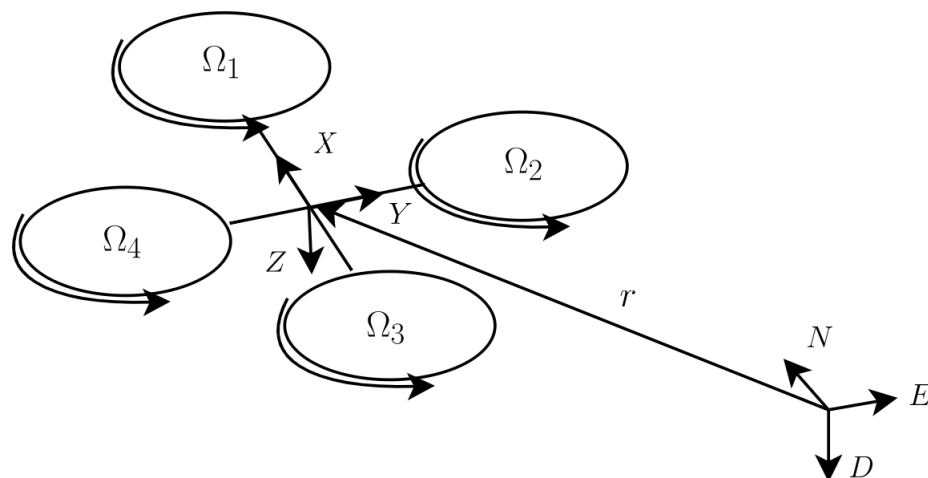
- As hélices são rígidas.
- A estrutura é rígida e simétrica.
- O centro de gravidade do drone corresponde com o centro da sua estrutura.
- Os movimentos de impulso e arrasto são proporcionais à velocidade das hélices ao quadrado.

3.1 MODELO CINEMÁTICO

O primeiro passo é a definição de dois planos de referência, o primeiro é o plano inercial posicionado na terra (E), utilizado para obter a posição linear do drone. O segundo plano de referência (B) é posicionado de forma em que o centro do eixo de referência coincida com o centro de massa do corpo do drone (figura 7) (BRESCIANI, 2008).

Assim, se for traçada uma linha r entre os dois planos, é possível obter a posição absoluta do centro de massa do drone dado que $r = [x \ y \ z]^T$. Já a orientação é descrita utilizando os ângulos de rolagem (ϕ), arfagem (θ) e guinada (ψ), que representam, respectivamente, as rotações nos eixos X , Y e Z .

Figura 7 – Planos lineares do quadricóptero e da terra.



Fonte – Autoria própria.

Logo, para que seja possível a relação entre os planos, é necessário obter matrizes que realizem a transformação das variáveis entre eles. A matriz de rotação R (equação 3.5) relaciona a transformação dos ângulos de rotação do plano B para o E , por ser ortogonal $R^{-1} = R^T$, onde R^T relaciona a transformação do referencial E para o corpo B .

Utilizando mais dois sistemas de coordenadas intermediários que sejam derivados dos dois primeiros, é possível definir a matriz R^T . O primeiro plano ($P1$) é formado a partir da rotação sob o eixo Y do plano E com uma angulação ψ , assim a matriz de transformação entre esses planos é dada por 3.1³.

$$R_E^{P1} = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

O plano inercial $P2$ é formado a partir da rotação sob o eixo Y do plano $P1$ com uma angulação θ , a matriz de transformação entre esses planos é dada por 3.2.

$$R_{P1}^{P2} = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \quad (3.2)$$

Por fim se for aplicada uma rotação em torno do eixo X de $P2$ se obtêm o plano B , a equação 3.3 descreve a matriz de transformação entre esses planos.

$$R_{P2}^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \quad (3.3)$$

Portanto a matriz R^T é definida pela equação 3.4.

$$R^T = R_E^{P1} R_{P1}^{P2} R_{P2}^B$$

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

$$R^T = \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ s\psi s\theta c\psi - c\phi s\psi & s\psi s\theta s\phi + c\psi c\phi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & s\psi s\theta c\phi - c\psi s\phi & c\theta c\phi \end{bmatrix} \quad (3.4)$$

³ Nesta equação e nas seguintes será utilizada a notação simplificada $s_K = \sin K$, $c_K = \cos K$ e $t_K = \tan K$.

Por ser ortogonal, a matriz R que relaciona os ângulos entre o plano B e o E é a transposta de R^T , dada pela equação 3.5.

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi & c\psi s\theta c\phi + s\phi s\psi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\theta & s\psi s\theta c\phi - c\psi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (3.5)$$

Essa matriz R será utilizada para a formulação do modelo dinâmico do drone, tendo em vista que é interessante que alguns dos estados sejam obtidos no plano do corpo B e outros no plano na terra E , assim para que os estados tenham relações entre si é necessária a realização dessa transformação, que é feita por meio dessa matriz.

Dado o fato de que a matriz de rotação foi obtida a partir de dois planos intermediários os ângulos só serão controláveis em uma faixa de -20 a 20 graus, isso não atrapalha em nada tendo em vista que ângulos maiores só são utilizados em drones que realizam acrobacias ou que participem de corridas assim necessitando de uma alta velocidade em curvas (LUUKKONEN, 2011).

Além disso, ainda é necessária uma outra matriz de transformação para relacionar as velocidades angulares do plano do corpo do quadricóptero B com o plano inercial na terra E , buscando uma relação entre as variações definidas por Euler (equação 3.6) com as taxas angulares do drone (equação 3.7). A relação entre as variáveis pode ser vista na equação 3.8,

$$\dot{\eta} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \quad (3.6)$$

$$\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T \quad (3.7)$$

$$\omega = R_r \dot{\eta} \quad (3.8)$$

onde R_r é a matriz de transformação para as velocidades angulares. Expandindo a equação 3.8 temos:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R(\dot{\phi}) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi)R(\dot{\theta}) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi)R(\theta)R(\dot{\psi}) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3.9)$$

$\dot{\phi}$, $\dot{\theta}$ e $\dot{\psi}$ são muito pequenos, logo é feita a aproximação $R(\dot{\phi}) = R(\dot{\theta}) = R(\dot{\psi}) = 1$, substituindo as variáveis na equação 3.9 é possível obter:

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

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.10)$$

Quando o drone está flutuando os ângulos são pequenos tendo em vista que o mesmo estaria praticamente estabilizado, fazendo com que $c\phi = c\theta = 1$ e $s\theta = s\phi = 0$, portanto extraindo a matriz R_r da equação 3.10 e aplicando esses valores, é possível observar que a matriz é aproximada a uma de identidade 3×3 , descrita pela equação 3.11.

$$R_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

3.2 MODELO DINÂMICO

Para o modelo dinâmico foi feita uma divisão em dois sistemas menores, assim temos o modelo rotacional, que define os movimentos de rolagem, arfagem e guinada, e o modelo translacional, que define a posição linear do drone (x, y e altura). Isso é feito para que a aplicação do controle seja mais fácil depois, tendo em vista que o rotacional pode ser totalmente controlável, já que suas entradas não dependem do outro modelo, em contraponto, o modelo translacional é dependente das saídas do outro sistema (ELKHOLY, 2014).

Logo, para que seja possível realizar um controle da posição linear do drone é necessário antes controlar a estabilidade de voo, então serão necessárias, no mínimo, duas malhas de controle, mas isso será explorado com mais detalhes no capítulo 4.

3.2.1 Rotacional

As equações rotacionais são obtidas a partir da aplicação do método de Newton-Euler no plano do corpo do drone, a grande vantagem de se aplicar essas equações no plano B é tornar a matriz de inércia do plano E não dependente do tempo, sendo definida pela equação 3.12,

$$J\dot{\omega} + \omega \times J\omega + M_G = M_B \quad (3.12)$$

onde:

J é a matriz diagonal de inércia do drone, definida pela equação 3.13, onde I_{xx} , I_{yy} e I_{zz} são os momentos de inércia dos respectivos eixos,

ω são as taxas angulares do corpo do drone,

M_G momento giroscópico,

M_B momento que atua no plano do corpo do quadricóptero.

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.13)$$

O momento giroscópico pode ser expandido para $M_G = \omega \times \begin{bmatrix} 0 & 0 & J_r & \Omega_r \end{bmatrix}^T$, logo substituindo esse valor na equação 3.12 tem-se:

$$J\dot{\omega} + \omega \times J\omega + \omega \times \begin{bmatrix} 0 & 0 & J_r & \Omega_r \end{bmatrix}^T = M_B \quad (3.14)$$

onde:

J_r é a inércia do motor,

Ω_r é a velocidade relativa do motor, sua definição expandida se encontra na equação 3.15.

$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (3.15)$$

Para que seja possível expandir M_B deve-se levar em conta os dois efeitos físicos que atuam em cada uma das hélices e braços do quadricóptero, sendo eles a força aerodinâmica F_n e o momento M_n , definidos pelas equações 3.16 e 3.17, respectivamente

$$F_n = K_f \Omega_n^2 \quad (3.16)$$

$$M_n = K_M \Omega_n^2 \quad (3.17)$$

onde:

K_f é a constante de força aerodinâmica,

K_M é a constante de momento,

Ω_n é a velocidade angular do motor n .

Com essas forças definidas, é possível determinar M_B , que é uma matriz que contém as equações que definem os movimentos de rolagem, arfagem e guinada (eixos X , Y e Z respectivamente). Essa matriz é dada pela equação 3.18

$$M_B = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} -F_2 l + F_4 l \\ -F_3 l + F_1 l \\ -M_2 + M_1 + M_3 - M_4 \end{bmatrix}$$

$$M_B = \begin{bmatrix} l K_f (\Omega_2^2 - \Omega_4^2) \\ l K_f (-\Omega_3^2 + \Omega_1^2) \\ K_M (-\Omega_2^2 + \Omega_1^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (3.18)$$

onde:

l é distância do centro do drone até o centro do motor.

Com a M_B é possível ter uma noção da influência da velocidade angular de cada motor nos movimentos referentes aos eixos da posição linear, assim tem-se o que será futuramente usado como as entradas do sistema rotacional.

3.2.2 Translacional

As equações para o modelo translacional são baseadas na segunda lei de Newton e derivadas no plano inercial da Terra E , a equação 3.19 mostra a matriz que representa esse movimento

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \quad (3.19)$$

onde:

$r = [X \ Y \ Z]^T$ é distância do quadricóptero até o plano E ,

m é a massa do quadricóptero,

g é a aceleração gravitacional,

F_B são as outras forças que não a gravidade que atuam no sistema (equação 3.20).

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (3.20)$$

F_B são as forças não-gravitacionais que influenciam o drone, para os eixos X e Y ela é considerada como 0, pois quando o drone está em uma orientação horizontal, sem distúrbios nos ângulos, a única força que age contra a gravidade é o empuxo vertical presente no eixo Z , definido como a soma da velocidade angular ao quadrado em cada um dos motores multiplicado por K_f .

Já que a referência é a gravidade, essa força é escrita utilizando um sinal negativo, tendo em vista que é contrária a aceleração gravitacional. F_B pode ser multiplicado pela matriz R para que assim seja possível transformar o empuxo vertical do plano do corpo do drone (B) para o plano inercial na Terra (E), isso será feito na composição do modelo de espaço de estados do quadricóptero.

3.3 MODELO EM ESPAÇO DE ESTADOS

Agora que as equações que descrevem o comportamento do drone foram obtidas, é possível buscar extrair as equações para o espaço de estados desejado para os testes em malha aberta e a aplicação de controle. Logo, as equações serão manipuladas para que possamos obter um mapeamento dos graus de liberdade do quadricóptero.

O vetor de estados X foi definido pela equação 3.21, os estados mapeados por ele estão na equação 3.22.

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \quad (3.21)$$

$$X = \left[\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y} \right]^T \quad (3.22)$$

Utilizando esses estados é possível ter uma noção completa da posição e da velocidade tanto linear quanto angular do drone.

O vetor de entrada U para o controlador foi definido utilizando as equações (3.18 e 3.20) que definem os movimentos básicos do drone, a distância entre o centro do drone até o centro do motor l é desconsiderada aqui, porém irá ser considerada nas equações para o vetor de estados X . A equação 3.23 define o vetor U

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ K_f(\Omega_2^2 - \Omega_4^2) \\ K_f(-\Omega_3^2 + \Omega_1^2) \\ K_M(-\Omega_2^2 + \Omega_1^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (3.23)$$

onde:

- U_1 representa o empuxo vertical,
- U_2 representa o movimento de rolagem,
- U_3 representa o movimento de arfagem,
- U_4 representa o movimento de guinada.

A grande vantagem de utilizar U como as entradas de controle é que isso permite o isolamento de cada parte do modelo matemático, assim, U_1 será usado para controlar o movimento no eixo Z , U_2, U_3 e U_4 os ângulos de rolagem, arfagem e guinada respectivamente.

Já que temos esse vetor para as entradas, é interessante isolar Ω_n^2 nas equações, para que assim seja possível obter uma relação entre as velocidades angulares em cada motor e as entradas de controle do sistema, a equação 3.24 mostra a relação entre as variáveis.

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} c & 0 & \frac{1}{2K_f} & \frac{1}{4K_M} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & -\frac{1}{4K_M} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & \frac{1}{4K_M} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & -\frac{1}{4K_M} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (3.24)$$

Extraindo a raiz quadrada e separando em uma equação para cada motor obtemos:

$$\Omega_1 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 + \frac{1}{4K_M}U_4}$$

$$\Omega_2 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 - \frac{1}{4K_M}U_4}$$

$$\Omega_3 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 + \frac{1}{4K_M}U_4}$$

$$\Omega_4 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 - \frac{1}{4K_M}U_4}$$

Agora que foram definidas as entradas e saídas do sistema, obtem-se a definição das equações para o espaço de estados. Primeiro, serão definidas algumas variáveis de redução, para que a notação matemática seja simplificada, as constantes de a_1 até a_4 e b_1 a b_3 .

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}} \quad a_2 = \frac{J_r}{I_{xx}} \quad a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}} \quad a_4 = \frac{J_r}{I_{yy}} \quad a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}}$$

$$b_1 = \frac{l}{I_{xx}} \quad b_2 = \frac{l}{I_{yy}} \quad b_3 = \frac{l}{I_{zz}}$$

Agora substitui-se os valores obtidos na equação 3.14, assim é obtida a descrição do sistema rotacional do drone, definida pela equação 3.25.

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} = \begin{bmatrix} IU_2 \\ IU_3 \\ U_4 \end{bmatrix}$$

Expandindo,

$$\begin{bmatrix} I_{xx} \ddot{\phi} \\ I_{yy} \ddot{\theta} \\ I_{zz} \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} \dot{\psi} I_{zz} - \dot{\theta} \dot{\psi} I_{yy} \\ \dot{\phi} \dot{\psi} I_{xx} - \dot{\phi} \dot{\psi} I_{zz} \\ \dot{\theta} \dot{\phi} I_{yy} - \dot{\theta} \dot{\phi} I_{xx} \end{bmatrix} + \begin{bmatrix} \dot{\theta} J_r \Omega_r \\ -\dot{\phi} J_r \Omega_r \\ 0 \end{bmatrix} = \begin{bmatrix} IU_2 \\ IU_3 \\ U_4 \end{bmatrix} \quad (3.25)$$

Realizando as multiplicações e substituindo os valores definidos pelas constantes a_n e b_n , obtem-se as equações que definem as velocidades angulares para os ângulos ϕ , θ e ψ (equações 3.26, 3.27 e 3.28 respectivamente).

$$\ddot{\phi} = b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6 \quad (3.26)$$

$$\ddot{\theta} = b_2 U_3 - a_4 x_2 \Omega_r + a_3 x_2 x_6 \quad (3.27)$$

$$\ddot{\psi} = b_3 U_2 + a_5 x_4 x_2 \quad (3.28)$$

Com essas equações é possível dizer que o sistema rotacional só depende das variáveis descritas dentro dele, por isso pode ser controlado de forma independente. Já o sistema translacional pode ser descrito realizando uma expansão da equação 3.19, e substituindo F_B pelo valor descrito na equação 3.20, obtemos a equação 3.29.

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi & c\psi s\theta c\phi + s\phi s\psi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\theta & s\psi s\theta c\phi - c\psi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix}$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} -U_1(s\psi s\theta c\phi - c\psi s\theta) \\ -U_1(s\psi s\theta c\phi - c\psi s\phi) \\ -U_1(c\theta c\phi) \end{bmatrix} \quad (3.29)$$

Reescrevendo e relacionando as variáveis com as escolhidas para o espaço de estados, a equação 3.29 pode ser dividida em três equações que descrevem a velocidade linear nos três eixos x , y e z (equações 3.30, 3.31 e 3.32 respectivamente).

$$\ddot{x} = \frac{-U_1}{m} (\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \quad (3.30)$$

$$\ddot{y} = \frac{-U_1}{m} (-\cos x_5 \sin x_1 + \cos x_1 \sin x_5 \sin x_4) \quad (3.31)$$

$$\ddot{z} = g - \frac{U_1}{m} (\cos x_1 \cos x_3) \quad (3.32)$$

Logo é possível ver que o sistema possui dependências de variáveis do sistema rotacional, isso nos leva a concluir que para que o controle possa ser aplicado é necessário primeiro buscar uma estabilização dos ângulos para depois se tratar sobre a posição linear do modelo.

Pode-se definir o espaço de estados para o modelo completo do drone pela equação 3.33.

$$\dot{x}_1 = \dot{\phi} = x_2$$

$$\dot{x}_2 = \ddot{\phi} = b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6$$

$$\dot{x}_3 = \dot{\theta} = x_4$$

$$\dot{x}_4 = \ddot{\theta} = b_2 U_3 - a_4 x_2 \Omega_r + a_3 x_2 x_6$$

$$\dot{x}_5 = \dot{\psi} = x_6$$

$$\dot{x}_6 = \ddot{\psi} = b_3 U_2 + a_5 x_4 x_2$$

$$\dot{x}_7 = \dot{z} = x_8$$

$$\dot{x}_8 = \ddot{z} = g - \frac{U_1}{m} (\cos x_1 \cos x_3)$$

$$\dot{x}_9 = \dot{x} = x_{10}$$

$$\dot{x}_{10} = \ddot{x} = \frac{-U_1}{m}(\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5)$$

$$\dot{x}_{11} = \dot{y} = x_{12}$$

$$\dot{x}_{12} = \ddot{y} = \frac{-U_1}{m}(-\cos x_5 \sin x_1 + \cos x_1 \sin x_5 \sin x_4)$$

$$f(X, U) = \begin{bmatrix} x_2 \\ b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6 \\ x_4 \\ b_2 U_3 - a_4 x_2 \Omega_r + a_3 x_2 x_6 \\ x_6 \\ b_3 U_2 + a_5 x_4 x_2 \\ x_8 \\ g - \frac{U_1}{m}(\cos x_1 \cos x_3) \\ x_{10} \\ \frac{-U_1}{m}(\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \\ x_{12} \\ \frac{-U_1}{m}(-\cos x_5 \sin x_1 + \cos x_1 \sin x_5 \sin x_4) \end{bmatrix} \quad (3.33)$$

Com o desenvolvimento da modelagem matemática é possível definir uma metodologia para a estruturação de um sistema de controle para o quadricóptero. O modelo levou em consideração as características físicas do sistema, o separando em dois subsistemas que possuem relações entre si fazendo com que os controladores possam ser estudados para cada uma das variáveis de estado.

No próximo capítulo os controladores que serão aplicados à planta serão desenvolvidos, levando em consideração algumas características observadas durante a modelagem.

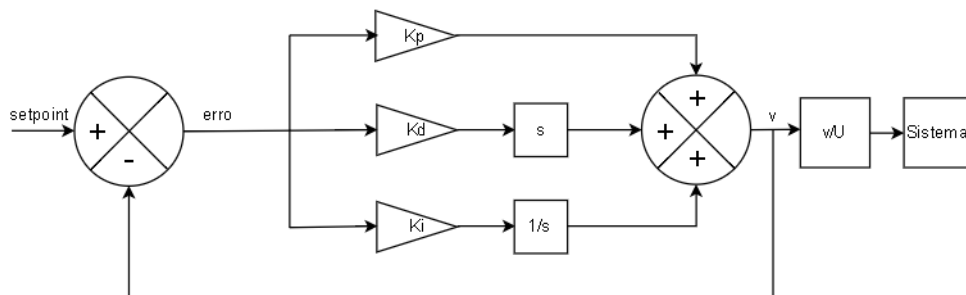
4 CONTROLE

Nesse capítulo serão tratadas as técnicas de controle aplicadas no trabalho de uma maneira mais teórica, também serão apresentadas as técnicas de otimização para os controladores que foram utilizadas. É válido lembrar que o controlador PID foi aplicado para o controle da altura e dos três ângulos do quadricóptero, ou seja, foram desenvolvidas quatro malhas de controle.

4.1 PID

O controlador Proporcional-Integral-Derivativo (PID) é a estratégia de controle mais adotada na indústria pelo seu desempenho robusto e uma alta variedade de condições em que o mesmo pode ser aplicado. A figura 8 mostra um diagrama de blocos para um PID (ANG; CHONG; LI, 2005)(WU; SU; LIU, 2014).

Figura 8 – Diagrama representativo de uma malha fechada com um PID.



Fonte – Autoria própria.

A equação 4.1 define o controle PID do ponto de vista puramente matemático

$$n_x = k_p(SP - v_x) + k_d(\dot{SP} - \dot{v}_x) + k_i \int (SP - v_x) dt \quad (4.1)$$

onde:

v_x é o valor atual da variável de controle,

SP é o valor de *setpoint* (ponto que se deseja alcançar),

k_p ganho proporcional,

k_d ganho derivativo,

k_i ganho integral.

Com isso também pode-se determinar o erro, descrito pela equação 4.2.

$$err = SP - v_x \quad (4.2)$$

Reescrevendo a equação 4.1 em função da 4.2 se obtêm

$$n_x = k_p(err) + k_d(\dot{err}) + k_i \int (err) dt \quad (4.3)$$

Porém para que o controle pudesse ser aplicado no código feito no MATLAB foram utilizadas aproximações para que a derivada e a integral se tornassem viáveis. A derivada pode ser expressa pela diferença entre o erro atual e o anterior dividida pelo passo de tempo atual, já a integral foi calculada pelo método do trapézio.

A equação 4.4 define a aproximação digital para um PID

$$n_x = k_p err + k_d \left(\frac{err - err_{ant}}{T} \right) + k_i \sum_{t_{start}}^{t_{end}} \left(\frac{err + err_{ant}}{2} \right) T \quad (4.4)$$

onde:

err_{ant} é o valor do erro para o passo de tempo anterior,

t_{start} é o tempo de início da simulação,

t_{end} é o tempo de encerramento da simulação,

T é o passo de tempo atual ($t - t_{ant}$).

4.1.1 Saída do Controlador

É válido notar que a resposta do PID sobre a planta obtida pela equação 4.4 é chamada de v_x , no caso da malha da altura por exemplo esse v_x seria dado em termos da aceleração angular necessária no eixo Z para que o drone pudesse chegar à posição desejada, logo é necessário converter esses sinais de saída do PID para sinais de esforço de controle U . As equações que realizam essas transformações são descritas.

$$U_1 = \frac{m(v_z + g)}{\cos x_1 \cos x_3}$$

$$U_2 = \frac{v_\phi - a_1 x_4 x_6}{b_1}$$

$$U_3 = \frac{v_\theta - a_3 x_2 x_6}{b_2}$$

$$U_4 = \frac{v_\psi - a_5 x_2 x_4}{b_3}$$

Essas equações são provenientes das equações do modelo de espaço de estados do quadricóptero, obtidas no final do capítulo 3. Com esses valores e após uma primeira análise das respostas obtidas pelo esforço do controle, foi possível observar a necessidade da aplicação de algumas técnicas, como a saturação, para que o PID pudesse produzir uma resposta mais estável, tendo em vista que os sinais de esforço de controle estavam apresentando ou valores muito altos ou oscilações muito bruscas.

4.1.2 Saturação

Foi aplicada uma saturação, que é a determinação de um valor máximo e mínimo para o esforço de controle. Essa limitação pode ser expressada matematicamente pela equação 4.5

$$U_x = \begin{cases} S_x & U_x \geq S_x \\ U_x & -S_x < U_x < S_x \\ -S_x & U_x \leq -S_x \end{cases} \quad (4.5)$$

onde:

S_x é o valor de saturação para o esforço de controle.

Logo os valores utilizados para a saturação para cada um dos esforços de controle foram analisados por meio das equações do modelo matemático. Utilizando a relação entre U e Ω (pode ser visto na equação 3.24), é possível obter a velocidade angular que cada motor vai ter para variações do esforço de controle. Logo em um momento em que todos estejam em um valor máximo, os valores podem ser substituídos na equação para a obtenção das velocidades angulares.

Considerando $K_m = 7,5 e^{-7}$ e $K_f = 3,13 e^{-5}$, e utilizando os seguintes valores estipulados para o esforço de controle: $S_1 = 15$ e $S_2 = S_3 = S_4 = 0,1$, onde U é considerado igual S na condição de saturação do controlador. Os valores obtidos para as velocidades angulares se encontram abaixo.

$$\Omega_1 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 + \frac{1}{4K_M}U_4} \simeq 393,37 \frac{rad}{s} \simeq 3756,40 rpm$$

$$\Omega_2 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 - \frac{1}{4K_M}U_4} \simeq 291,34 \frac{rad}{s} \simeq 2782,06 rpm$$

$$\Omega_3 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 + \frac{1}{4K_M}U_4} \simeq 389,29 \frac{rad}{s} \simeq 3717,41 rpm$$

$$\Omega_4 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 - \frac{1}{4K_M}U_4} \simeq 296,77 \frac{rad}{s} \simeq 2833,94 rpm$$

É possível concluir que os valores adotados são satisfatórios, tendo em vista que as velocidades angulares obtidas em cada motor quando todos os valores estão no máximo são possíveis de serem alcançados por um modelo real. No capítulo 6 é possível ver a resposta das velocidades angulares para a planta totalmente controlada com a aplicação da saturação.

4.2 CELL MAPPING

Desenvolver um controlador que cumpra com as necessidades de um certo sistema é um grande desafio. Além da necessidade de tornar a planta estável o controlador deve possuir um grau de robustez de modo a apresentar desempenho satisfatório mesmo sob alguma variação dos parâmetros

do sistema. No caso de motores, essa variação está particularmente relacionada à carga tracionada que costuma sofrer constantes mudanças.

Além do desenvolvimento, é de extrema importância a validação do desempenho do controlador, verificando-se os parâmetros escolhidos satisfazem os requisitos de projeto do controle. Em controladores desenvolvidos por métodos puramente matemáticos essa validação é feita de forma analítica, mas para controladores que não se baseiam nesse tipo de metodologia (Controladores Fuzzy, por exemplo) essa tarefa se torna muito mais desafiadora. Controladores lineares possuem ferramentas de projeto bem estabelecidas, porém tais controladores possuem restrições de desempenho.

Controladores não lineares podem superar essas restrições, porém seu projeto é bem mais complexo, pois não se pode fazer uso das ferramentas de projeto de controle clássicas, que só se aplicam a sistemas lineares. Várias soluções tem sido propostas pela comunidade científica, como controle adaptativo e modos deslizantes, por exemplo. Essas soluções costumam usar uma regra geral aplicada a todo o espaço de estados. Mas quando se pensa em espaço de estados particionado em células, tem-se a liberdade de definir regras particulares para cada célula, com ou sem relação com as células vizinhas.

O CM (*Cell Mapping*) consiste no particionamento do espaço de estados em células, compondo um conjunto de elementos finitos. Muitas técnicas de controle podem ser usadas para definir a saída de controle (vetor de controle) quando os estados estiverem dentro de uma célula em particular. Mas também é possível que tais vetores possam ser definidos por processos de otimização local, sem a necessidade de vinculação das células entre si.

O mapeamento de espaço de estados celular é uma poderosa técnica computacional para a análise da performance global de sistemas não-lineares. [...] Cada célula representa um número infinito de estados relacionados e é aproximada pelo seu ponto central e as trajetórias do sistema são traçadas a partir da transição entre as células (PAPA; TAI; SHENOI, 1997).

Como o mapeamento celular fornece uma maneira de avaliar o desempenho global de um controlador, é interessante utilizar essa técnica para o desenvolvimento de controladores. Isso pode ser feito utilizando uma metodologia *feedforward*, na qual o controlador é construído a partir do mapeamento, ou *feedback*, onde o controlador é feito de maneira independente porém utilizando as informações que o mapeamento celular fornece para otimizar o controle aplicado no sistema.

Isso torna a aplicação dessa técnica, principalmente em sistemas não lineares, muito interessante, pois ela permite um aumento na efetividade de um controlador já definido ou pode ser utilizada para o desenvolvimento de uma técnica de controle com uma maior robustez, tendo em vista o aspecto de análise global do espaço de estados. O *Cell Mapping* também tem se mostrado muito útil em sistemas de sintonia online, onde o aprendizado da planta e/ou controle ocorre com o sistema em pleno funcionamento.

Uma das primeiras aplicações dessa técnica foi realizada por Bose (1959), nesse trabalho é estudada a aplicação apenas algébrica de um mapeamento celular para a caracterização e otimização de sistemas não-lineares. O mesmo foi feito por Bursal (1992), onde é aplicada uma técnica de

interpolação nas células para que a discretização do sistema em tempo contínuo seja mais efetiva (BOSE, 1959)(BURSAL; TONGUE, 1992).

O *Cell Mapping* pode ser utilizado como uma técnica de auxílio na computação gráfica, conseguindo representar objetos em um plano bidimensional, essa aplicação porém apresenta um problema, tendo em vista que os vetores obtidos eram muito grandes para serem guardados na memória, logo Kuo apresenta um algoritmo otimizado de mapeamento celular e de armazenamento dessas informações (KUO, 1990).

Nos tempos atuais os estudos são focados utilizando o mapeamento celular em conjunto com outras técnicas de otimização e/ou de controle. Yousef Naranjani, desenvolveu um algoritmo híbrido que retorna um gráfico de dedicação eficiência de Pareto a partir de um problema de otimização com multiobjetivos. Primeiro é feita a busca dos melhores pontos utilizando um algoritmo genético (AG), mas o mesmo demoraria muito para encontrar os pontos ótimos, então o mapeamento celular é utilizado para realizar a busca local dentro dos pontos encontrados pelo AG (NARANJANI; SARDAHI; SUN, 2014).

Foram realizados três estudos de caso (os problemas de Fonseca, Poloni e Kursawe) e em todos foi comprovada a eficiência desse método, tanto por apresentar resultados mais certos quanto pela confiabilidade apresentada em resolver problemas de diversos graus de complexidade. O ponto negativo é o aumento modesto do processamento computacional necessário devido ao fato do algoritmo utilizar duas técnicas computacionais em conjunto (COELLO; LAMONT; VELDHUIZEN, 2007).

Outros exemplos são a aplicação de CM em um CMOL, que é um CMOS híbrido que utiliza uma arquitetura de circuito molecular, o mapeamento é utilizado como uma tentativa de otimização do circuito (CHU et al., 2010) (XIA et al., 2011).

Nesse trabalho o CM foi aplicado como uma técnica para a análise da planta controlada, em específico a malha responsável pelo controle da altura do quadricóptero, além disso foi proposto um controlador baseado nos valores obtidos no mapa. A seguir será explicada a metodologia para o desenvolvimento do CM.

4.2.1 Granularidade

A granularidade representa o número de células por eixo, definindo a faixa de valores contida em uma célula; quanto maior esse valor, maior a similaridade entre o mapeamento celular e o espaço de estados real da planta. Porém esse valor deve ser escolhido com cuidado, pois afeta o esforço computacional de forma linear. Nesse caso, o ideal seria buscar a máxima redução do erro sem prejudicar as simulações. A equação 4.6 define o efeito da granularidade na matriz do CM,

$$CM_{(i,j)} = \begin{pmatrix} i_1 & j_1 \\ i_2 & j_2 \\ \vdots & \vdots \\ i_{G_{CM}} & j_{G_{CM}} \end{pmatrix} \quad (4.6)$$

onde:

$CM_{(i,j)}$ é o mapa $[i] \times [j]$,

G_{CM} é a granularidade (quantidade de células), onde $G_{CM} \in I$.

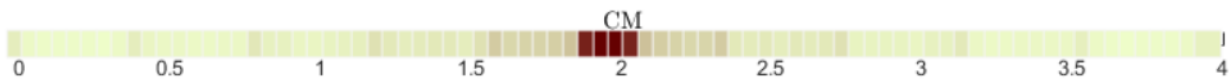
4.2.2 Limites

Os limites devem ser escolhidos de acordo com as variáveis analisadas, eles determinam o intervalo discreto do CM. Para um mapa unidimensional $CM_{(i)}$ da variável X_1 de tamanho G_{CM} , com limites definidos entre $X_{1i} < X_{1i} < X_{1f}$ consegue-se definir a variação de valores para cada célula (X_{1div}) pela equação 4.7.

$$X_{1div} = \frac{X_{1f} - X_{1i}}{G_{CM}} \quad (4.7)$$

As figuras citadas no texto a seguir referem-se à aplicação de um controle PI aplicado ao drone previamente modelado. Esse controle foi mapeado no espaço de estados para um conjunto de condições iniciais. A figura 9 mostra um exemplo de um mapa unidimensional para: $X_{1i} = 0$, $X_{1f} = 4$ e $G_{CM} = 80$.

Figura 9 – Exemplo de mapa unidimensional.



Fonte – Autoria própria.

4.2.3 Condições Iniciais

A quantidade de condições iniciais simuladas no CM é de extrema importância, esse é o valor que determina o mapeamento do sistema. A figura 10 mostra a comparação entre mapas com duas dimensões para diferentes valores de condições iniciais mapeadas, já a figura 11 mostra a comparação entre tridimensionais, utilizando o esforço de controle (saída) como uma terceira dimensão.

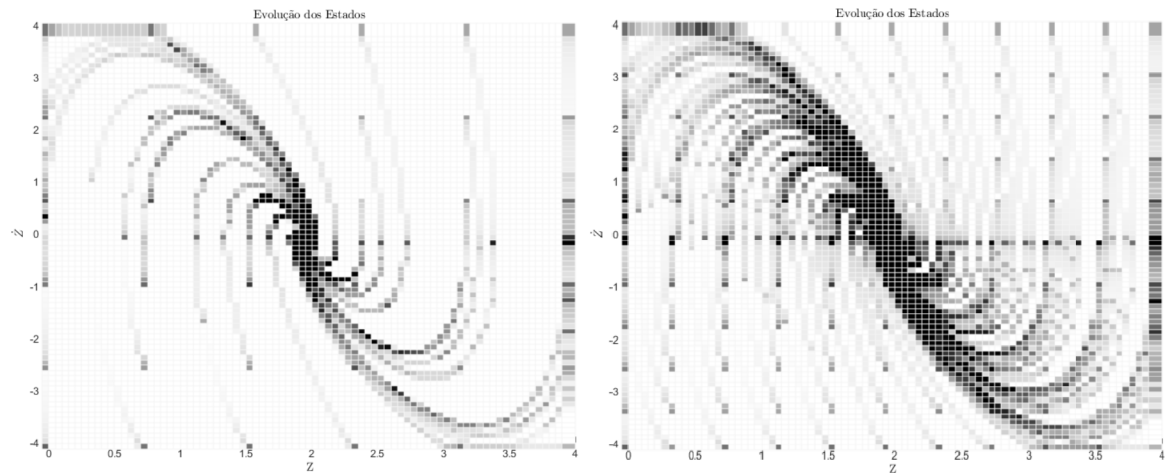
Esses exemplos foram retirados das próprias simulações do trabalho. No capítulo 6 serão mostradas os resultados obtidos para as simulações finais do CM.

É possível perceber nas imagens a relevância desse parâmetro, ele é quem preenche o mapa com os valores obtidos, no caso das imagens esse valor é o esforço de controle fornecido pelo PD para cada uma das posições e velocidades de Z.

4.2.4 Armazenamento do Esforço de Controle

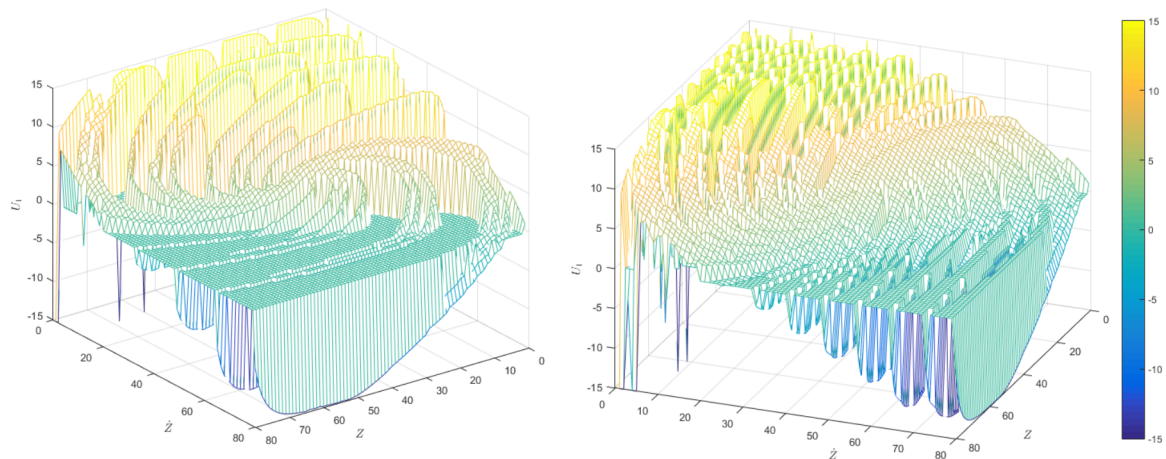
Como as simulações são feitas variando as condições iniciais do sistema a mesma célula pode receber valores diferentes de esforço de controle, em um primeiro momento era feita apenas a média simples dos valores em caso de duplicados porém isso estava comprometendo a qualidade do mapa, portanto uma nova metodologia foi proposta. A figura 12 mostra um exemplo de uma célula com um

Figura 10 – Comparação de mapas utilizando 25 (esquerda) e 100 (direita) condições iniciais (2D).



Fonte – Autoria própria.

Figura 11 – Comparação de mapas utilizando 25 (esquerda) e 100 (direita) condições iniciais (3D).



Fonte – Autoria própria.

ponto central C, o ponto P é o valor da leitura atual da simulação, logo é feita uma transformação entre a posição dela na célula para o centro.

Com isso quanto mais próximo do centro mais o valor irá influenciar no valor atual de esforço de controle armazenado na célula. a equação 4.8 mostra esse cálculo,

$$\alpha_{cm} = \left(\frac{X_{1p} - X_{1c}}{X_{1c}} + \frac{X_{2p} - X_{2c}}{X_{2c}} \right) / 2$$

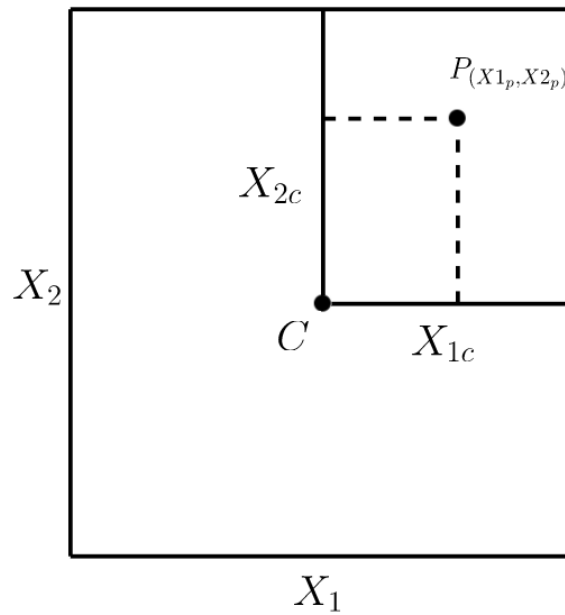
$$U = \alpha_{cm} U_p + (1 - \alpha_{cm}) U_c \quad (4.8)$$

onde:

α_{cm} é a porcentagem da influência do valor novo no valor da célula,

U_p é o esforço de controle do ponto atual,

Figura 12 – Representação de uma célula no CM.



Fonte – Autoria própria.

U_c é o esforço de controle armazenado na célula.

Nesse capítulo as estratégias de controle e de análise que serão aplicadas na planta em estudo foram explicados, os controladores PID serão otimizados via Algoritmo Genético e após isso será extraído o CM, utilizando a metodologia explicada nesse capítulo, para uma análise da malha da altura controlada. O próximo capítulo irá tratar sobre o AG, explicando o passo a passo para a implementação do algoritmo bem como um breve referencial teórico que explica a origem da técnica.

5 OTIMIZAÇÃO

Encontrar máximos e mínimos em funções sempre foi um problema muito explorado na matemática, com o desenvolvimento do cálculo diferencial e integral (desenvolvido simultaneamente em trabalhos independentes por Gottfried Wilhelm Leibniz (1646-1716) e por Isaac Newton (1643-1727)) esse problema se tornou solúvel por uma forma analítica de encontrar os zeros do gradiente de uma função. A falta de um método universal para a otimização de problemas mais complexos que ocorrem na realidade, levou ao desenvolvimento de diferentes estratégias de otimização impulsionadas pela observação da natureza, graças aos avanços da matemática e principalmente pelo aumento da capacidade computacional.

Todos os controladores aplicados nesse trabalho foram otimizados utilizando-se Algoritmo Genético (AG), uma técnica heurística que realiza a otimização e a busca de valores máximos globais dentro de um sistema. A inspiração para o algoritmo vem da teoria moderna da evolução das espécies, na qual é proposta a seleção natural. A técnica é baseada na ideia de que a solução de um problema é codificada em um indivíduo e o seu genótipo é relacionado diretamente aos parâmetros livres do modelo.

Uma função de *fitness* deve ser definida para medir o grau de adaptabilidade do indivíduo ao ambiente, o que é a expressão do custo da função a ser otimizada. O processo iterativo entre a função de *fitness* e os operadores genéticos levam aos indivíduos a mudarem o seu genótipo. Esse processo permite a exploração do espaço de busca, assim, buscando as soluções aos problemas propostos. A descrição de uma estrutura geral para o algoritmo vai de acordo com alguns princípios, o processo de busca é iterativo, onde uma nova população (o conjunto dos indivíduos), é criada de acordo com a evolução do anterior (HOLLAND, 1992).

5.1 ALGORITMO GENÉTICO

O AG é o principal representante dos algoritmos evolutivos sendo altamente aplicável em diversos problemas de otimização. Como característica marcante eles são adaptativos e inspirados na evolução natural descrita pela primeira vez por Darwin em seus livros “*On the Origin of Species*” e “*The preservation of favored races in the struggle for life*”, aonde foram descritos, de forma pioneira, os conceitos de seleção natural, recombinação sexual e mutação. (DARWIN, 1859)

Estes conceitos base foram os inspiradores para elaboração dos algoritmos genéticos (AG's) introduzidos por Holland e mais tarde aperfeiçoados por Goldberg e Michalewicz. AG's são métodos de busca que visam a procura do ótimo global, sem impor restrições sobre a função a ser otimizada, como continuidade ou dimensionalidade. O AG é aplicado com sucesso em diferentes campos que variam desde a otimização de processos de usinagem, previsão de séries temporais, otimização de carga em contêineres, teoria de jogos, projeto de circuito eletrônico, identificação criminal, a economia, a previsão do mercado de ações e preços de câmbio (GOLDBERG, 1989) (MICHALEWICZ, 1996).

A expressão da evolução natural ocorre nos cromossomos e, por consequência, no DNA de cada ser vivo, com essa premissa uma possível solução de um problema é representada por uma versão codificada dos parâmetros, aqui chamados de genes, em um algoritmo genético. A sequência linear de genes formam um indivíduo, a prescrição que mapeia o genótipo (espaço dos dados) ao fenótipo (espaço da qualidade das soluções) correspondente depende inteiramente da estrutura do problema a ser adaptado (função custo).

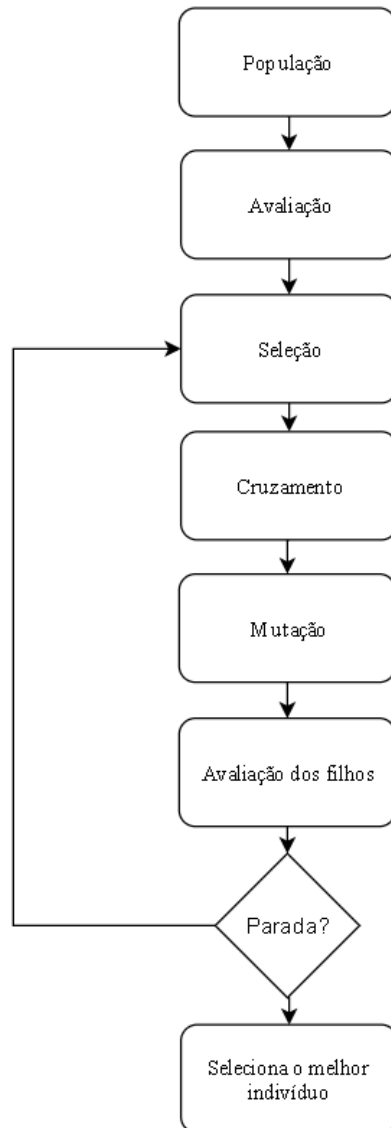
Após a inicialização da primeira população, um número é atribuído a cada indivíduo para determinar o grau de adaptação ao ambiente. Dois indivíduos são escolhidos com probabilidade proporcional à posição relativa da população atual e acasalados para produzir novos indivíduos (prole). Este passo é repetido até que os indivíduos criados sejam suficientes para formar a próxima população. A genética destes indivíduos pode ser novamente modificada por mutação que ocorre com uma probabilidade fixa, mas pequena. Desta forma, geração após geração a população é avaliada preservando sempre aqueles com o maior grau de adaptabilidade (*fitness*) (SIQUEIRA, 2009) .

5.1.1 Etapas do Algoritmo Genético Clássico

As etapas básicas de um clássico algoritmo genético seguindo a proposta de Holland podem ser descritas pelo seguinte pseudocódigo e também a figura 13 apresenta um fluxograma que representa as iterações do genético:

1. Gerar uma população inicial com N_{pop} .
2. Determinar o número máximo de iterações.
3. Calcular a aptidão dos indivíduos da população inicial (*fitness*).
4. Em cada iteração, fazer:
 - a) Selecionar 2 pais da população para *Crossover*.
 - b) Realizar o *Crossover*.
 - c) Realizar Mutação no filho gerado pelo *crossover*.
 - d) Calcular a aptidão do filho.
5. A busca prossegue até atingir o número máximo de iterações.
6. Seleciona os melhores indivíduos para a próxima geração.
7. Repete os passos para todas as gerações.

Figura 13 – Fluxograma proposto para o Algoritmo Genético.



Fonte – Autoria própria.

5.1.2 *Fitness*

Para cada indivíduo de uma população é atribuída uma nota, graduando assim a influência do ambiente sobre eles. Em outras palavras, indivíduos mais bem adaptados – ou cromossomos que resultem em melhores resultados – possuem maior *fitness*; quanto maior, mais adaptado ao meio e maior sua probabilidade de sobrevivência. Esta conversão depende da função custo desejada e normalmente busca-se reduzir a função custo para aumentar o grau de adaptação do indivíduo (BOCCATO et al., 2009).

A função *fitness* é definida pela equação

$$fitness = \frac{1}{Custo} K_{fit} \quad (5.1)$$

onde:

K_{fit} é a constante do *fitness*.

A constante do *fitness* foi utilizada somente para se obter um valor melhor para a plotagem dos gráficos, tendo em vista que o quadricóptero é um sistema relativamente lento o que causava um valor muito elevado na função custo.

5.1.3 População Inicial

A população é inicializada inserindo as características desejadas em cada indivíduo de uma forma aleatória e uniforme, visando maximizar a distribuição dos mesmos no espaço amostral. Estas características são norteadas pelo objetivo da busca: por exemplo, se deseja-se otimizar um controlador PID as características de cada indivíduo (cromossomos) são as variáveis k_i, k_p, k_d . Como não se sabia muito do comportamento do sistema controlado, os valores iniciais para os genes dos indivíduos da população inicial foram determinados com um número de variação randômica entre 0 e 1.

5.1.4 Seleção e Crossover

Foi utilizada uma seleção em duas etapas para o algoritmo, logo no começo do processamento da geração os indivíduos são colocados em ordem pelo seu *fitness*, assim selecionando apenas os que se encontram entre os 50% melhores. Essa mesma seleção é feita novamente após o *crossover*, portanto apenas os melhores indivíduos permanecem na próxima geração. Esse método de seleção onde o indivíduo com o maior *fitness* é escolhido se chama torneio, aqui ainda é aplicado o elitismo, o que significa que apenas os melhores indivíduos continuam nas próximas gerações, fazendo com que a variação do *fitness* seja sempre positiva.

O *crossover* é uma técnica utilizada para se realizar a combinação entre os genes dos indivíduos, produzindo filhos que irão obter outros valores de *fitness*, garantindo a exploração do espaço de busca do sistema. Para codificação de números reais é possível utilizar o método desenvolvido por Michalewicz, aonde cada indivíduo é combinado com outro selecionado de maneira randômica gerando três filhos, as equações 5.2, 5.3 e 5.4 definem a lógica utilizada na combinação dos genes dos pais,

$$Gene_{F1} = (1 - \alpha)Gene_{P1} + \alpha Gene_{P2} \quad (5.2)$$

$$Gene_{F2} = (1 - \alpha)Gene_{P2} + \alpha Gene_{P1} \quad (5.3)$$

$$Gene_{F3} = \frac{Gene_{P1} + Gene_{P2}}{2} \quad (5.4)$$

onde:

$Gene_{Pn}$ é o gene do pai n,

$Gene_{Fn}$ é o gene do filho n,

α é a constante randômica de combinação.

Portanto, realizando a combinação para todos os genes dos indivíduos pais é possível produzir os filhos. Os dois primeiros são feitos pela combinação dos dois genes dado uma constante que é gerada de forma randômica para cada combinação, o terceiro é a média entre os valores dos dois genes.

5.1.5 Mutação

Após a geração dos valores dos filhos por meio do *crossover* todos os genes tem uma chance (15%) de sofrerem mutação (GOLDBERG, 1989).

A equação 5.5 define o cálculo utilizado para a mutação

$$Gene'_{Fx} = Gene_{Fx} + \beta Gene_{Fx} \quad (5.5)$$

onde:

β é uma variável randômica que segue a distribuição discreta $\beta \sim R([-0.2, 0.2])$.

Assim, a mutação pode causar grandes aumentos ou reduções nos valores do ganho, fazendo com ocorra uma redução na chance de estagnação nos valores dos ganhos. A utilização de valores muito grandes na mutação pode fazer com que o algoritmo seja guiado a valores maiores de *fitness* de uma maneira mais rápida, porém faz com que o AG perca um pouco a sua característica de busca local, o que não é desejado.

A figura 14 mostra o que uma diferença no valor de mutação pode causar no sistema, para a linha em vermelho foi utilizada uma mutação entre -100% à 100% do valor, na linha em azul de -20% a 20%. É possível observar que para a mutação mais agressiva o algoritmo converge muito mais rápido, porém para o outro valor foi encontrado um *fitness* final maior, isso ocorreu pois o nível de mutação influencia diretamente na capacidade de busca local do algoritmo.

5.2 FUNÇÃO CUSTO

Foram propostas duas funções custo, para a aplicação em todas as malhas, as equações 5.6 e 5.7 definem essas funções,

$$ISE = \left(\sum_{i=t_{start}}^{t_{end}} (x(i) - SP)^2 \right) + k_{OS}(V_{pico} - setpoint)^2 \quad (5.6)$$

$$ITSE = \left(\sum_{i=t_{start}}^{t_{end}} i(x(i) - SP)^2 \right) + k_{OS}(V_{pico} - setpoint)^2 \quad (5.7)$$

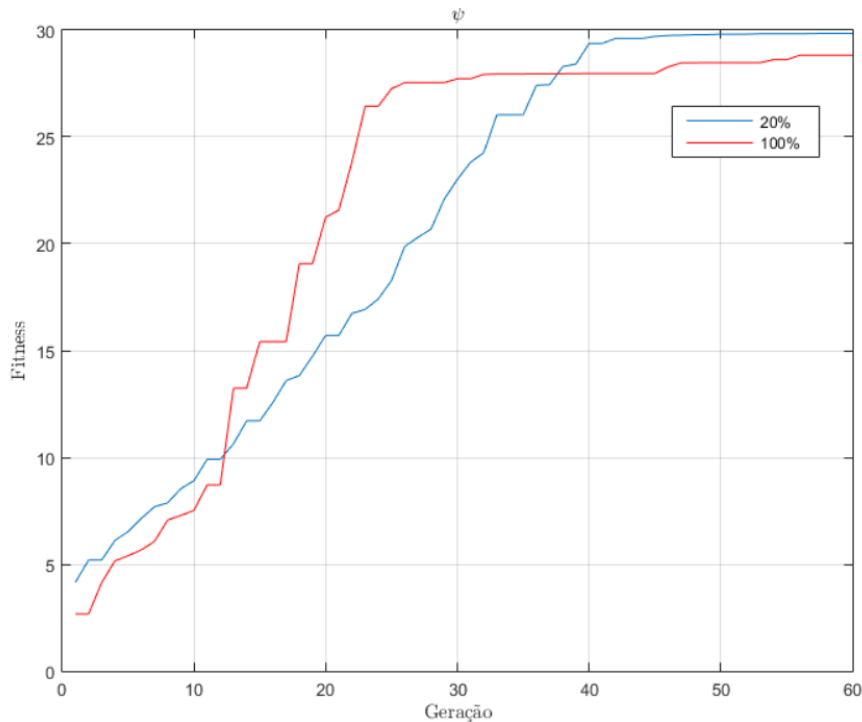
onde:

$x(i)$ é o valor da variável de controle dado o tempo i ,

k_{OS} constante ajustável para a influência do *overshoot* no custo,

V_{pico} é o valor de pico.

Figura 14 – Evolução do *fitness* para diferentes valores de mutação.



Fonte – Autoria própria.

Ou seja, a função custo é a somatória do erro quadrático lido pela malha a cada instante de tempo somado com um termo que representa o *overshoot* obtido, essa função de custo é conhecida na literatura como ISE (*Integral Squared Error*, que significa Integral do Erro Quadrático). Esse termo que representa o *overshoot* é dado como a subtração entre o ponto mais alto obtido após o tempo de estabilização e o ponto que se deseja alcançar ao quadrado, multiplicados por uma constante para tornar o valor significativo em relação a somatória do erro.

A outra função proposta apenas adiciona o tempo como fator multiplicativo no erro quadrático, conhecida como ITSE (*Integral Time-weighted Squared Error*, que significa Integral com peso Temporal do Erro Quadrático), essa função tem como intuito fazer com que erros no regime permanente tenham uma maior influência no valor de custo dado o peso que o tempo insere no valor.

Nesse capítulo foi determinada a metodologia para a aplicação de um AG para a otimização dos ganhos dos controladores PID, assim sendo possível começar as simulações para a validação em malha fechada do sistema assim que os ganhos do PID forem obtidos através da busca do algoritmo. No próximo capítulo serão mostrados todos os resultados obtidos na simulação do modelo, do AG e dos controladores aplicados, assim como uma análise dos dados obtidos.

6 SIMULAÇÕES

Nesse capítulo os resultados obtidos em simulação serão mostrados, em conjunto com diagramas explicando a montagem dos blocos. A planta foi simulada utilizando o *software* MATLAB, foi feito um *script* utilizando o algoritmo ode23 para a avaliação do sistema, e *Simulink*, aonde foram criados blocos para a representação da planta.

A escolha do ode23 ao invés do ode45 (que é o algoritmo mais comum para simulação) se dá pelo fato de que o segundo apresenta muitos passos retroativos no tempo, ele utiliza essa ferramenta para obter uma maior precisão, porém isso acabava afetando muito a saída do esforço de controle, tendo em vista que o controlador PD leva em consideração o passo atual de tempo. De acordo com a documentação do MATLAB nesses casos é recomendado utilizar o ode23. O mesmo foi simulado utilizando um maior refinamento nos seus parâmetros para que a qualidade da simulação não fosse tão afetada.

A primeira simulação realizada foi a verificação do modelo matemático por uma simulação em malha aberta, após isso os ganhos para os controladores PID foram obtidos por meio de AG. Com os ganhos dos controladores foi possível simular o controle de estabilidade dos ângulos e de altura, após isso foi analisada a aplicação de CM.

Foi utilizado o modelo em espaço de estados definido pela equação 3.33 com uma simplificação, o momento giroscópico foi considerado como nulo portanto os termos da aceleração angular em ϕ e θ na simulação são descritos pelas equações 6.1 e 6.2, respectivamente.

$$\dot{x}_2 = \ddot{\phi} = b_1 U_2 + a_1 x_4 x_6 \quad (6.1)$$

$$\dot{x}_4 = \ddot{\theta} = b_2 U_3 + a_3 x_2 x_6 \quad (6.2)$$

As outras equações de espaço de estados não foram simplificadas, também é válido notar que o eixo Z foi considerado invertido para que os gráficos tivessem uma variação positiva, apenas para uma melhor visualização das informações. A tabela 1 mostra os parâmetros utilizados para a simulação.

6.1 MALHA ABERTA

Os testes em malha aberta foram realizados utilizando as equações obtidas no capítulo 3, após a separação em dois subsistemas (Translacional e Rotacional) representados pela figura 15, foram medidos os valores das posições lineares e angulares para variações de E, avaliando se as respostas condizem com o esperado na teoria.

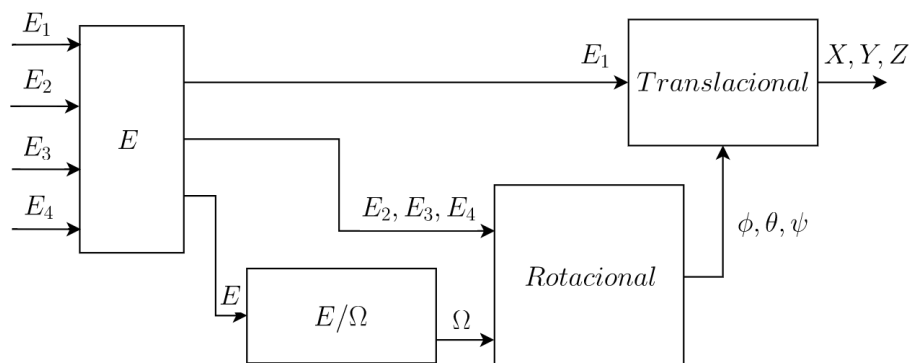
Todas as condições iniciais $(x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$ foram consideradas como zero, assim, o drone partiria sem nenhuma variação nos seus ângulos e nenhuma aceleração linear ou angular.

Tabela 1 – Constantes da simulação.

Variável	Valor	Descrição
t_{start}	0	Tempo de início da simulação
t_{end}	10	Tempo de encerramento da simulação
T	0,01	Passo da simulação
g	9,81	Constante gravitacional
m	0,5	Massa do quadricóptero
L	0,1	Distância do centro do quadricóptero até o rotor
K_m	$1 E^{-7}$	Constante relacionada à aerodinâmica (<i>thrust</i>)
K_f	$3 E^{-6}$	Constante de arrasto (<i>drag</i>)
J_r	$6 E^{-5}$	Inércia do rotor
I_{xx}	$5 E^{-3}$	Coefficiente de inércia sobre o eixo x
I_{yy}	$5 E^{-3}$	Coefficiente de inércia sobre o eixo y
I_{zz}	$10 E^{-2}$	Coefficiente de inércia sobre o eixo z

Fonte – Adaptada de Modelagem e Controle de um Helicóptero Quadricóptero (CONSTANTINO; KLOSOWSKI, 2017).

Figura 15 – Diagrama representativo dos testes em malha aberta.



Fonte – Autoria própria.

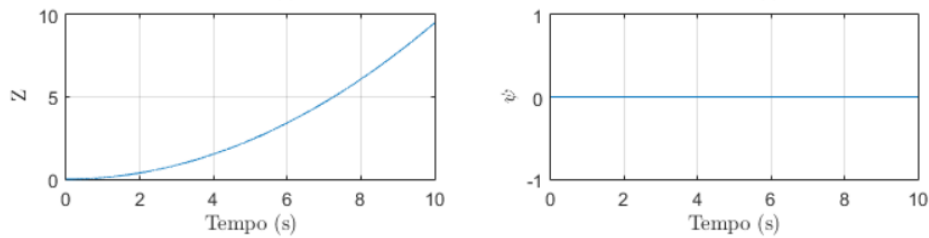
O primeiro teste feito foi a aplicação de um degrau no sinal de entrada E_1 (representa o empuxo vertical), teoricamente o drone deveria apresentar uma variação em z apenas, os ângulos aqui não deveriam variar dadas as condições iniciais. A figura 16 mostra a variação da posição linear e angular para esse teste.

Após isso foi feita uma simulação para cada movimento básico do drone (rolagem, arfagem e guinada), as figuras 17, 18 e 19 mostram as respostas obtidas. Vale ressaltar que quando o eixo z apresenta valores negativos isso significa que o drone está em queda livre.

Para a rolagem (figura 17) o esperado era uma grande variação no ângulo ϕ e na posição y , isso foi obtido e ainda ocorreu uma pequena variação tanto no ângulo θ quanto na posição x mas isso ocorre dada a influência que um tem no outro, processo descrito na modelagem matemática.

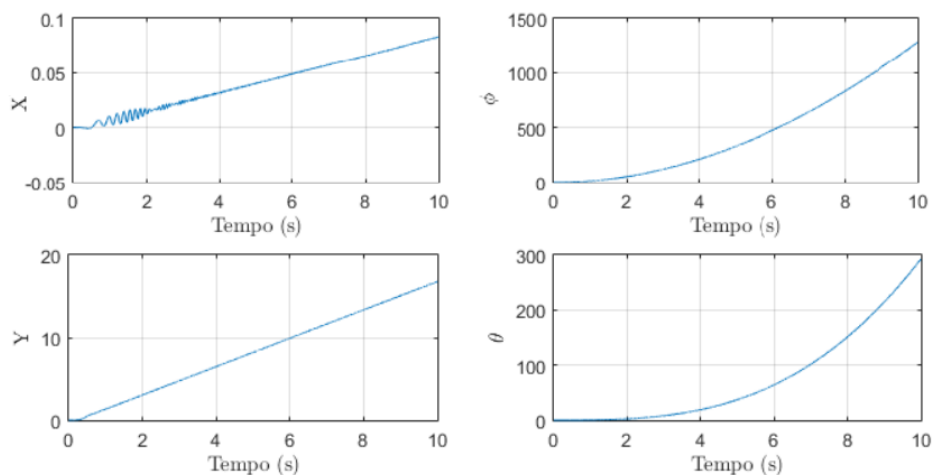
Para a guinada (figura 18) o esperado era uma grande variação no ângulo θ e na posição x ,

Figura 16 – Empuxo Vertical em malha aberta.



Fonte – Autoria própria.

Figura 17 – Rolagem em malha aberta.



Fonte – Autoria própria.

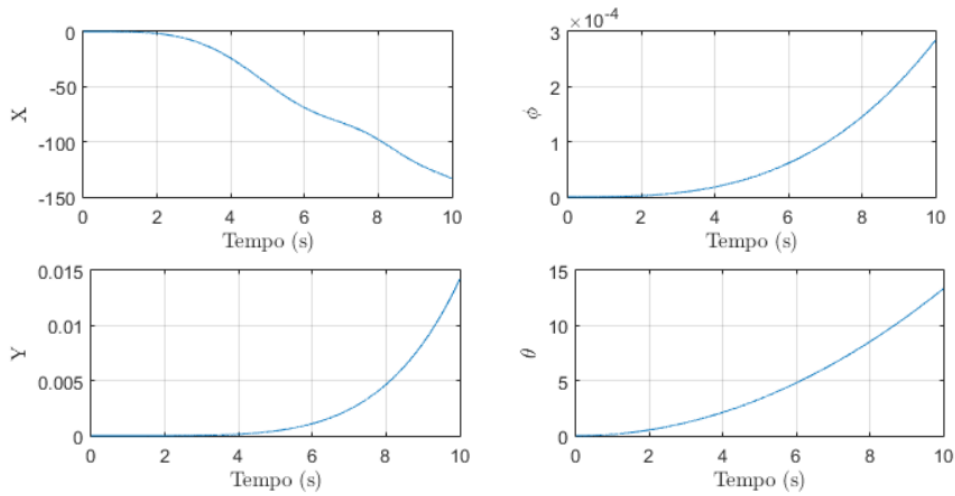
isso foi obtido com o mesmo comportamento do movimento anterior, os outros sofreram pequenas variações pela interdependência das variáveis. Já a arfagem é apenas a variação do ângulo ψ no eixo z , observado na figura 19.

6.2 ALGORITMO GENÉTICO (AG)

O AG foi simulado com 60 gerações e 14 indivíduos, cada um deles com três genes (cada um dos ganhos do PID), na primeira geração de cada rodada os ganhos eram gerados de forma randômica. Os parâmetros utilizados na simulação do AG estão na tabela 2.

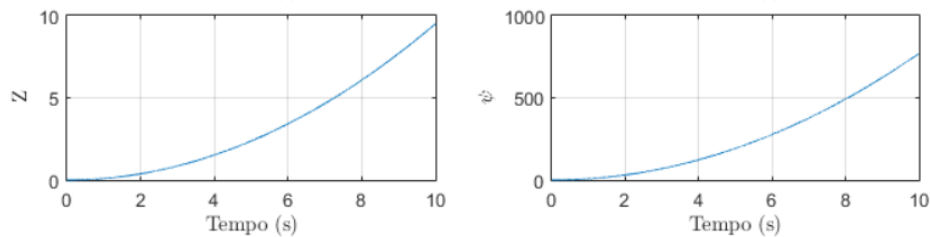
Para que os dados pudessem ser avaliados de uma melhor maneira foram simuladas 5 rodadas do genético para cada um dos controladores, porém somente o melhor indivíduo de cada uma das malhas (selecionados pelo maior número de *fitness*) será analisado. Como foram aplicadas duas funções distintas para a avaliação dos indivíduos, ambas serão discutidas e após isso será feita uma análise dos resultados para a escolha dos melhores ganhos, o valor de *fitness* obtido pelos indivíduos nas duas funções tem diferenças pela maneira em que elas avaliam o desempenho do sistema logo os ganhos obtidos serão comparados utilizando o percentual de sobressinal e o tempo de estabilização

Figura 18 – Guinada em malha aberta.



Fonte – Autoria própria.

Figura 19 – Arfagem em malha aberta.



Fonte – Autoria própria.

Tabela 2 – Constantes de inicialização do AG.

Variável	Valor	Descrição
k_{OS}	100	Constante da influência do <i>overshoot</i> no <i>fitness</i>
k_{fit}	300	Constante do <i>fitness</i>
Ger	60	Número de gerações
N_{pop}	14	Número de indivíduos

Fonte – Autoria própria.

obtidos durante a simulação dos controladores.

6.2.1 ISE (*Integral Squared Error*)

A tabela 3 contém os valores dos melhores indivíduos após 5 rodadas de simulação para cada uma das malhas controladas pelo PID.

Para todos os casos os algoritmos convergiram para valores muito pequenos de k_i , ou seja, em nenhum deles a parte integrativa do PID causou uma influência positiva no *fitness*, portanto para as simulações o k_i foi considerado como 0 aplicando somente as partes proporcional e derivativa do

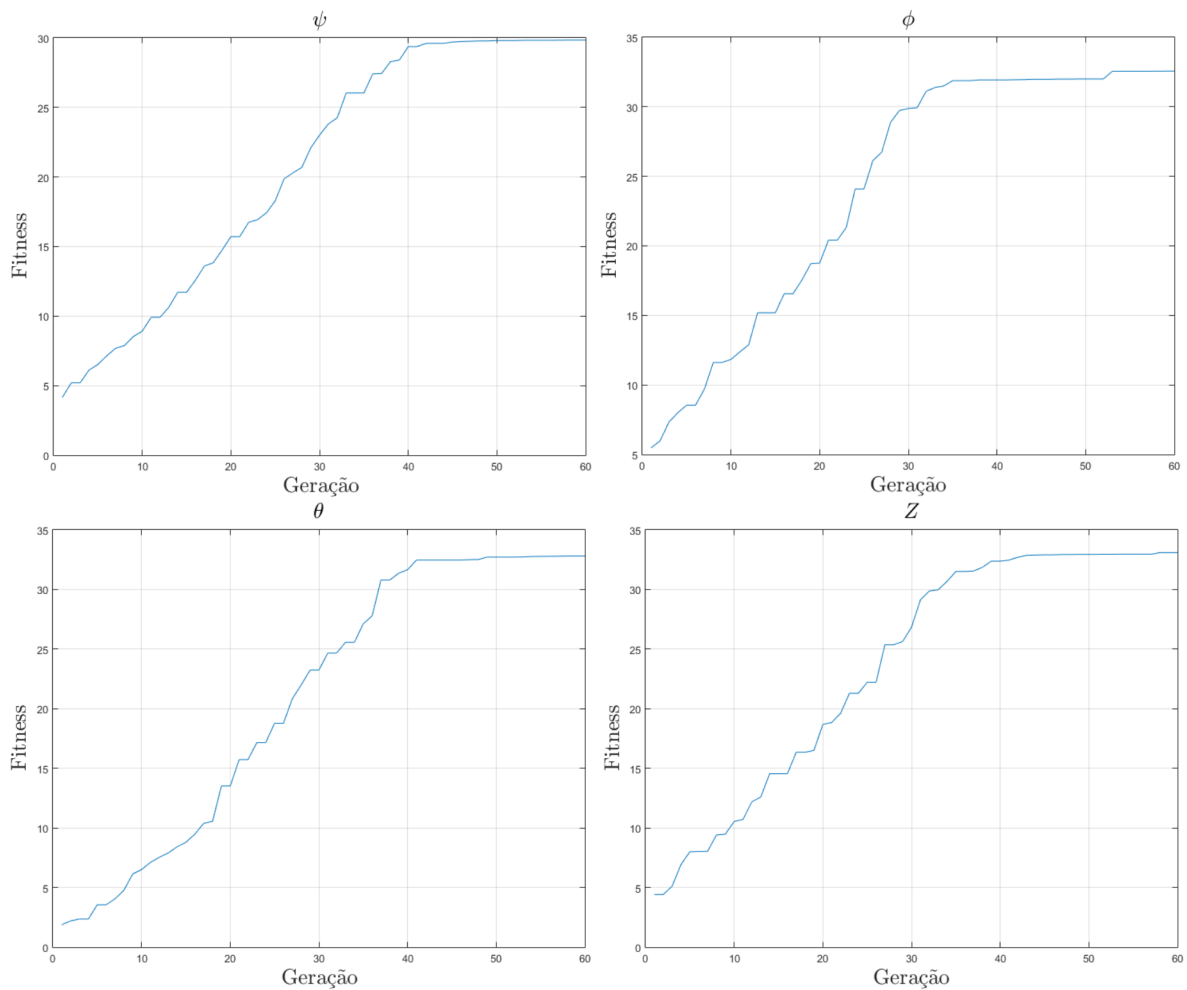
controle, podendo ser resumido a um controlador PD. A figura 20 mostra a evolução do *fitness* para as 4 variáveis de controle.

Tabela 3 – Valores obtidos para os ganhos do PID após otimização via AG (ISE).

Variável	k_p	k_i	k_d	<i>Fitness</i>
z	14,929	$3,540 E^{-3}$	5,570	33,0923
ϕ	14,847	$1,0278 E^{-3}$	5,1553	32,5691
θ	14,4373	$4,790 E^{-2}$	5,5191	32,8086
ψ	14,1932	$5,176 E^{-3}$	5,7258	29,8470

Fonte – Autoria própria.

Figura 20 – Evolução do *fitness* na melhor rodada (ISE).



Fonte – Autoria própria.

6.2.2 ITSE (*Integral Time-Weighted Squared Error*)

Os melhores indivíduos para cada uma das malhas utilizando a função custo ITSE se encontram na tabela 4. Aqui foi possível observar que o valor de *fitness* foi maior do que para a outra função, isso ocorre pelo fator multiplicativo do tempo que acaba gerando valores menores de custo para erros que se encontram no transitório do sistema. Os ganhos k_i aqui encontrados foram maiores que os da ISE porém ainda sim foram muito pequenos para serem levados em consideração, além disso quando simulados sem a parte integral os indivíduos apresentam valores ainda maiores de *fitness*.

Tabela 4 – Valores obtidos para os ganhos do PID após otimização via AG (ITSE).

Variável	k_p	k_i	k_d	<i>Fitness</i>
z	14,9195	$1,690 E^{-2}$	4,6765	184,3635
ϕ	14,8879	$1,185 E^{-1}$	4,8673	189,9838
θ	14,9091	$1,560 E^{-2}$	4,7479	192,6591
ψ	14,9529	$3,180 E^{-2}$	4,7258	162,1021

Fonte – Autoria própria.

6.2.3 Análise das Respostas

Durante as simulações para as duas funções custo foi possível observar que o ganho k_i se aproximou muito de zero, até mesmo quando o tempo foi considerado como um ganho no erro. Portanto a integral acaba mais prejudicando do que melhorando o sinal, o que é um comportamento até de certa forma esperado, devido ao fato de que um quadricóptero muito dificilmente obterá uma estabilidade completa em seus três ângulos, além disso os valores de erro estacionário para a estabilidade são muito pequenos se analisados em suas devidas grandezas físicas.

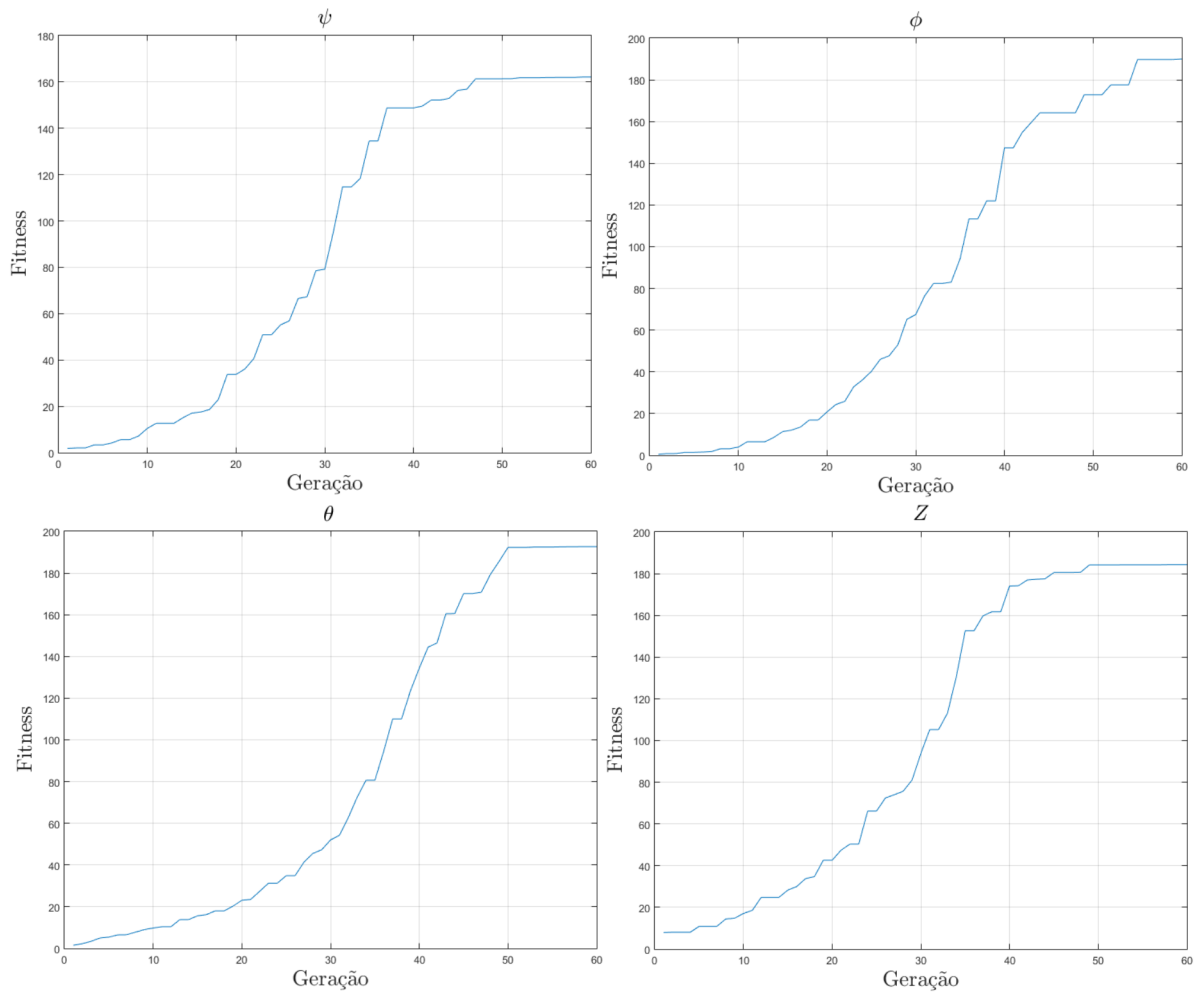
Tendo isso em mente, para uma comparação mais honesta entre os controladores obtidos o termo integrativo foi desconsiderado, ou seja, foi aplicado nas malhas apenas um controlador PD. Isso não descarta o estudo feito no capítulo 4, tendo em vista que o funcionamento do controlador irá ocorrer praticamente da mesma forma, somente retirando a parte integral do erro, o impacto da remoção da integral será melhor analisado na seção de simulações do controlador.

A tabela 5 mostra os valores finais para os ganhos dos controladores PD para as quatro malhas.

6.3 CONTROLE DA ESTABILIDADE

Agora iremos falar sobre o controle para a estabilidade dos ângulos do quadricóptero. Como dito na seção anterior todos os ganhos integrativos apresentaram valores muito baixos logo o controlador

Figura 21 – Evolução do *fitness* na melhor rodada (ITSE).



Fonte – Autoria própria.

foi resumido a apenas um PD. A teoria apresentada no capítulo 4 ainda é válida, apenas é necessário desconsiderar a parte da integral do controlador.

Para o controle da estabilidade é necessário fechar as malhas para os três ângulos (ϕ , θ e ψ) assim os mesmos irão representar a variável de controle. Como explicado no capítulo 4 todos os controladores aplicados utilizaram a mesma arquitetura para o seu projeto, produzindo assim a resposta como uma derivada de segunda ordem da variável de controle, necessitando assim de uma transformação para o esforço de controle E . A figura 22 mostra um diagrama de blocos que exemplifica o controle da estabilidade.

Como explicado no capítulo 3 o controle da estabilidade pode ser feito como uma malha a parte, tendo em vista que todas as variáveis controladas fazem parte do sistema rotacional. Os ganhos utilizados pelos controladores são os mesmos obtidos na melhor rodada do genético para cada uma das malhas, a tabela 5 contém as informações com os ganhos.

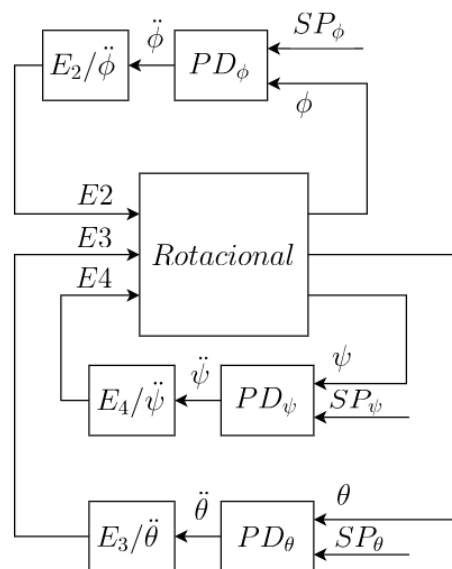
Para as simulações todos os *setpoints* para o controle da estabilidade foram determinados como 0 por ser o valor desejado para que a planta fique estável em relação a sua posição angular. O qua-

Tabela 5 – Valores finais para a simulação dos controladores PD.

	ISE		ITSE	
	k_p	k_d	k_p	k_d
z	14,929	5,570	14,9195	4,6765
ϕ	14,847	5,3553	14,8879	4,8673
θ	14,4373	5,5191	14,9091	4,7479
ψ	14,1932	5,7258	14,9529	4,9291

Fonte – Autoria própria.

Figura 22 – Diagrama de blocos para o controle da estabilidade.



Fonte – Autoria própria.

drirrotor foi iniciado com um distúrbio randômico que varia de -12° a 12° em todos os seus ângulos fazendo com que o controle tente realizar a estabilização.

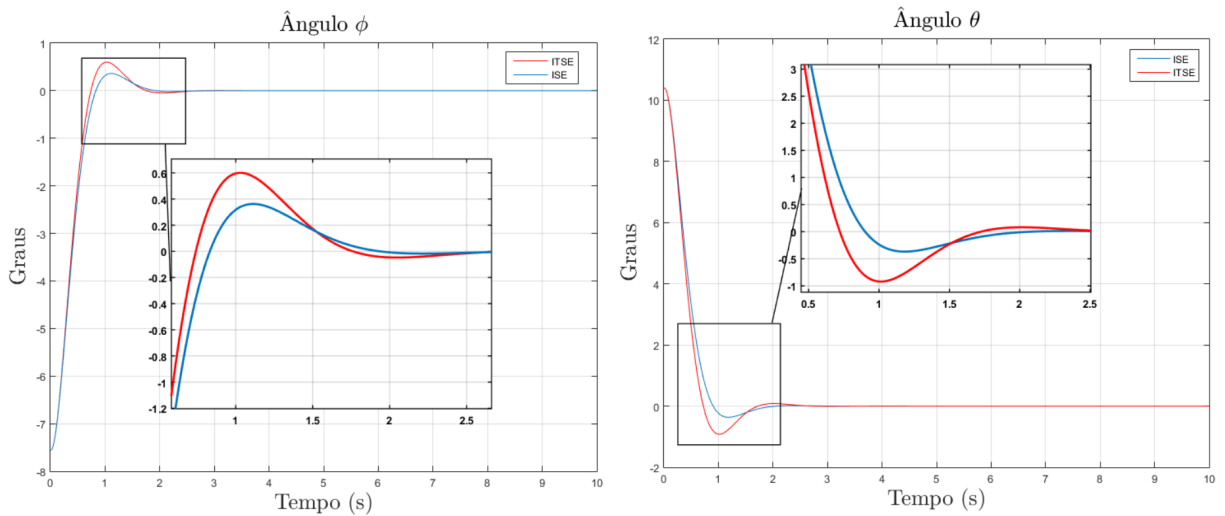
Uma comparação entre as duas funções custo será feita para cada um dos controladores, os parâmetros para comparação são o tempo de estabilização em 2% (TE), o percentual de *overshoot* (sobressinal) e o erro no regime permanente em graus. Tendo em vista que no controle da estabilização dos ângulos o *setpoint* é zero os valores iniciais de cada simulação foram utilizados como a base para o cálculo.

As figuras 23 e 24 mostram as respostas do melhor indivíduo de cada uma das funções custo de uma forma sobreposta para os ângulos ϕ , θ e ψ .

6.3.1 Análise das Respostas

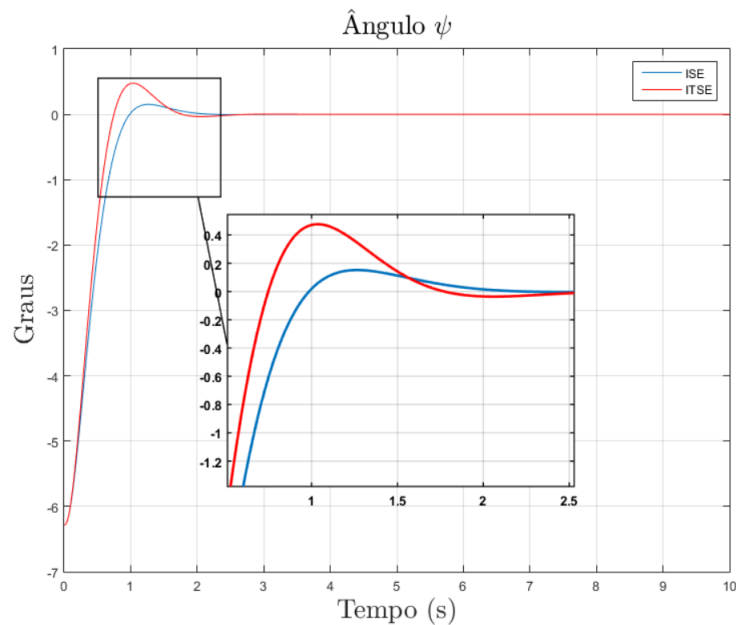
A tabela 6 mostra os valores obtidos para cada uma das funções.

Figura 23 – Comparação entre os melhores indivíduos para as duas funções de custo (ϕ e θ).



Fonte – Autoria própria.

Figura 24 – Comparação entre os melhores indivíduos para as duas funções de custo (ψ).



Fonte – Autoria própria.

A primeira coisa a se notar é que o valor de erro em regime permanente para todas as simulações foi muito baixo, ainda mais quando consideramos a sua grandeza física. Esse comportamento é esperado considerando a planta tendo em vista que os ângulos são interdependentes. Para os valores obtidos por meio da função custo ITSE, que leva em consideração o tempo, o tempo de estabilização para todos os controladores foi menor, porém o *overshoot* foi consideravelmente maior. Isso ocorreu pela baixa penalização que a função aplica nos erros durante o transitório, já para a ISE o tempo foi um pouco maior, porém o sobressinal foi bem menor ficando sempre abaixo de 5%.

Tabela 6 – Valores para análise da performance dos controladores obtidos.

	<i>ISE</i>			<i>ITSE</i>		
	TE (s)	Overshoot (%)	Erro (°)	TE (s)	Overshoot (%)	Erro (°)
ϕ	0,89	4,7582	$6,723 E^{-2}$	0,78	7,9075	$1,4 E^{-3}$
θ	0,85	3,5444	$-4,350 E^{-2}$	0,70	8,8667	$-2,1 E^{-4}$
ψ	1,14	2,4123	$-1.5536 E^{-2}$	0,79	7,5746	$-1,1 E^{-3}$

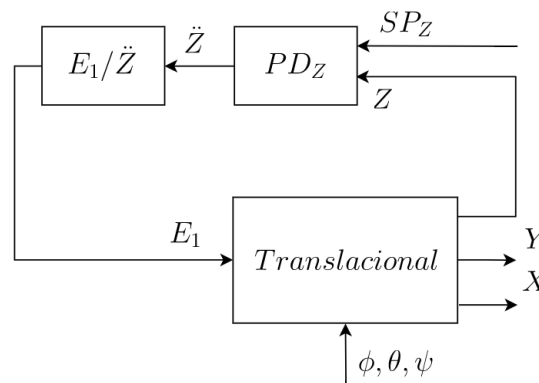
Fonte – Autoria própria.

Portanto para as três malhas do controle da estabilidade do drone os valores finais escolhidos para os controladores são os obtidos pela função custo ISE.

6.4 CONTROLE DA POSIÇÃO VERTICAL

O controle da posição vertical, quando partimos do pressuposto de que os ângulos já estão estáveis, depende somente da variável de controle Z , a figura 25 mostra um diagrama de blocos exemplificado para o controlador PD.

Figura 25 – Diagrama de blocos para o controle da altura.



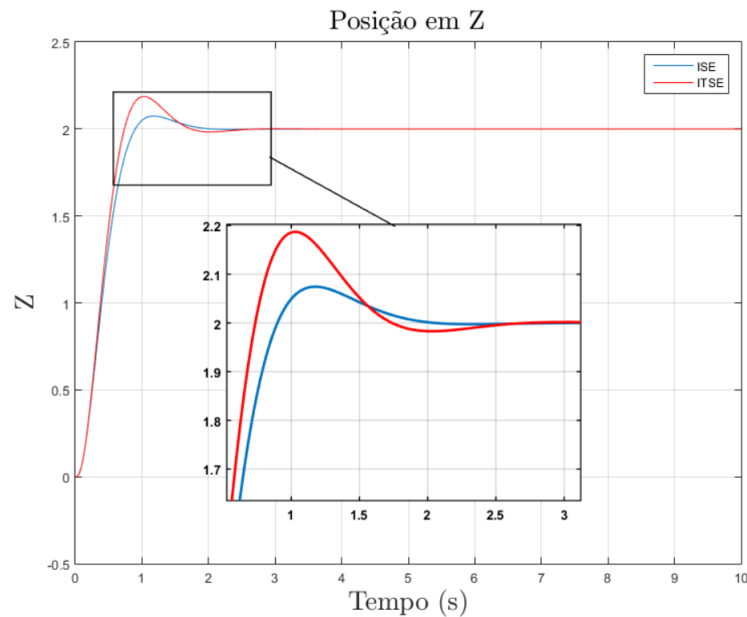
Fonte – Autoria própria.

Diferente dos controladores da estabilidade, a ausência do termo integrativo aqui não tem o efeito de não anular o erro em regime permanente, tendo em vista que a própria planta apresenta essa característica. A figura 26 mostra a comparação entre os melhores indivíduos obtidos nas duas funções de custo aplicadas durante o AG.

6.4.1 Análise das Respostas

Os valores obtidos para as duas funções de custo estão na tabela 7, aqui o mesmo comportamento observado para o controle da estabilidade dos ângulos foi observado, com a função ISE tendo um melhor percentual de *overshoot* e a ITSE um melhor tempo de estabilização. Os ganhos pro-

Figura 26 – Comparação entre os melhores indivíduos para as duas funções de custo (Z).



Fonte – Autoria própria.

venientes da função ISE foram novamente escolhidos pela grande redução obtida no percentual de sobressinal.

Tabela 7 – Valores para análise da performance dos controladores obtidos.

Função	TE (s)	Overshoot (%)	Erro (m)
<i>ISE</i>	0,85	3,7156	0
<i>ITSE</i>	0,72	9,3501	0

Fonte – Autoria própria.

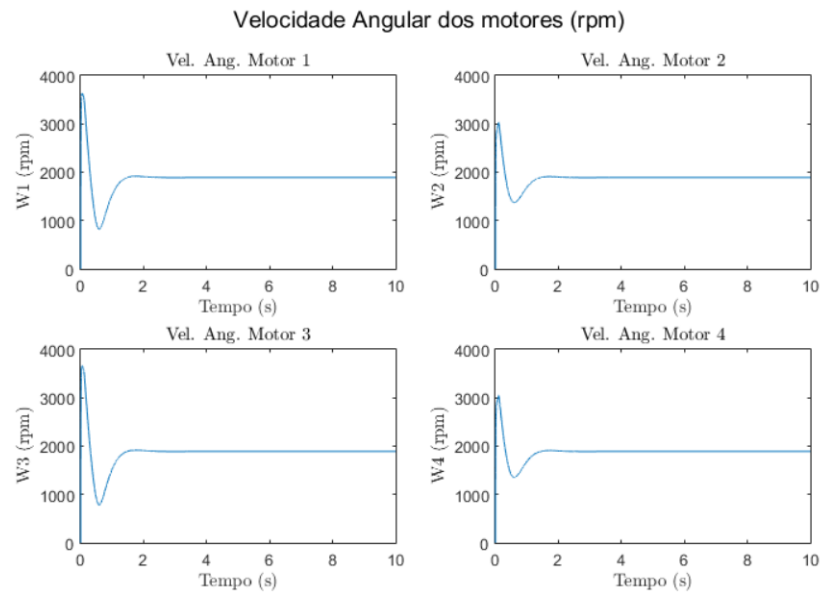
6.5 ESFORÇO DE CONTROLE

Agora que os controladores foram definidos, podemos fazer uma análise da saída da variável de controle, como o esforço de controle é descrito pelo empuxo vertical e o torque em cada um dos eixos é complicado realizar uma análise com valores que seriam observados na planta real.

Para que isso pudesse ser feito, todos os esforços de controle foram transformados em velocidade angular, utilizando a relação entre U e Ω (pode ser visto na equação 3.24).

A figura 27 mostra essa saída expressa em RPM para cada um dos motores, na simulação todos começam com um valor máximo, para o modelo real seria necessário um tempo de aceleração para que os motores chegassem a essas velocidades, porém o transitório de estabilização seria praticamente o mesmo. Todos os motores estabilizam em valores abaixo de 2000 rpm, o que é

Figura 27 – Variação da velocidade angular dos motores durante o período de simulação.



Fonte – Autoria própria.

um valor aceitável para um funcionamento estacionário do modelo tendo em vista que os motores utilizados em drones podem chegar em até 9000 rpm.

6.6 CELL MAPPING

O CM foi aplicado na malha que controla a altura do drone, como dito anteriormente essa malha tem dependência de apenas uma variável de controle, assim o mapa do espaço de estados seria bidimensional, contendo a sua variável e a sua derivada ($X_1 = Z$ e $X_2 = \dot{Z}$). Os parâmetros utilizados para a obtenção do mapa estão na tabela 8.

Tabela 8 – Constantes de inicialização do CM.

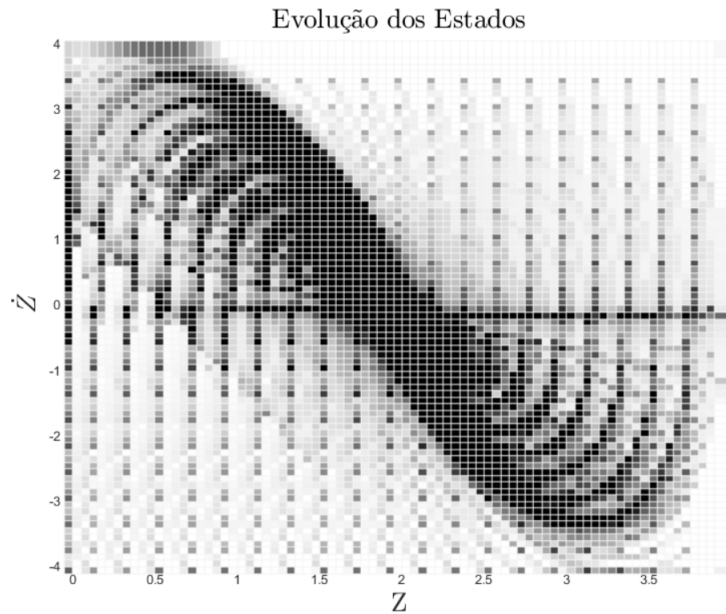
Variável	Valor	Descrição
G_{CM}	100	Granularidade
X_{1i}	0 m	Valor inicial para X_1
X_{1f}	4 m	Valor final para X_1
X_{2i}	$-4 m/s$	Valor inicial para X_2
X_{2f}	$4 m/s$	Valor final para X_2

Fonte – Autoria própria.

Foram feitas 3 rodadas de simulação, a diferença entre elas era o número de condições iniciais utilizadas para o mapeamento, foram utilizadas 400 condições iniciais diferentes. Essas condições iniciais tiveram um intervalo igualmente distribuído por todo o espaço do mapa, a quantidade de condições iniciais influencia diretamente na qualidade do mapa.

A figura 28 mostra o mapeamento das células, quanto mais escura a cor mais a célula foi visitada, significando que um maior número de leituras foi feito em cima dela.

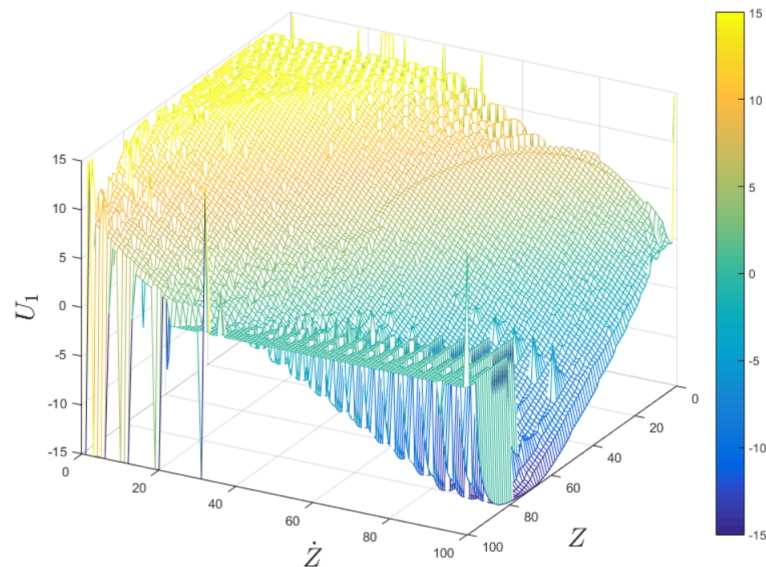
Figura 28 – CM para a malha da altura controlada.



Fonte – Autoria própria.

Também é possível ver os valores de esforço de controle U_1 para cada célula, a figura 29 mostra essa relação entre o esforço de controle e o CM.

Figura 29 – Variação do esforço de controle no CM.



Fonte – Autoria própria.

É possível observar que ocorreram alguns pontos onde o esforço de controle sofreu saturação, isso ocorre sempre no extremo das velocidades, significando talvez uma não-linearidade do modelo

para esses valores. Além disso é possível ver que o mapa ainda tem diversos pontos onde ocorre uma oscilação muito grande entre os valores de esforço de controle, isso ocorre pois não é feita uma normalização dos valores.

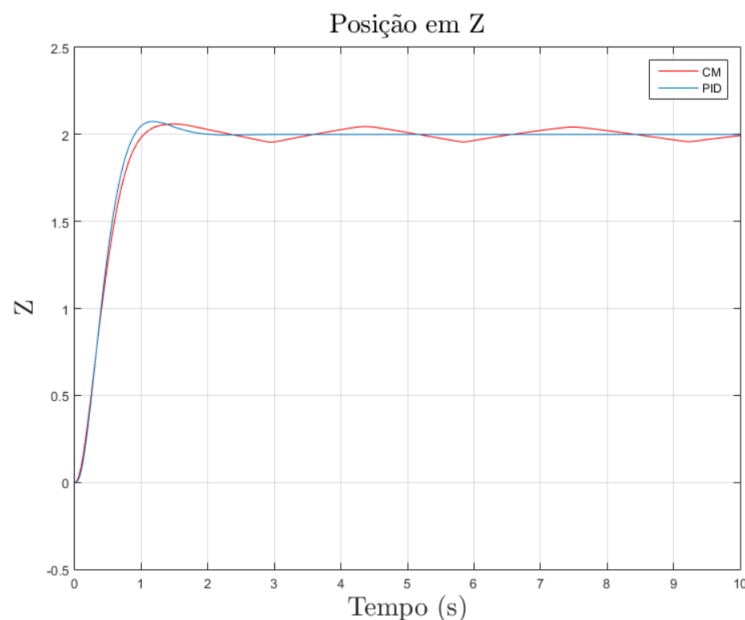
6.6.1 Controle utilizando o mapa com os esforços de controle

Foi feita também uma aplicação preliminar de controle utilizando o mapa obtido no CM, substituindo o controlador PID pelo mapa com os esforços de controle dada a velocidade e a posição em Z atual do sistema. É válido destacar aqui que o controlador não teve uma resposta muito exata tendo em vista que a varredura não conseguiu preencher todas as células. Além disso existem certos pontos aonde células vizinhas tem valores muito discrepantes de esforço de controle, o que causa instabilidades.

Essa seção está no trabalho pois pode ser vista como um primeiro passo do estudo da aplicação de controle por essa técnica, existem muitos estudos que podem ser feitos utilizando o CM, com aplicações de técnicas tanto *on-line* (*Reinforced Learning*) quanto *off-line* (A aplicação que foi tentada nesse trabalho) de controle.

O resultado obtido pelo controlador utilizando a tabela de esforços de controle comparada com o PID que a originou se encontra na figura 30.

Figura 30 – Comparação entre os controladores.



Fonte – Autoria própria.

O resultado obtido é proveniente do fato de que algumas células não foram "visitadas" pela trajetória dos estados e, portanto, ficaram sem valor mapeado. Isso acarretou um mapeamento com buracos justificando o resultado do controle não ter ficado bom. Seria necessário aplicar técnicas adicionais para estimar ou interpolar valores nas células vazias e ainda suavizar a superfície mapeada

final afim de retirar eventuais picos e vales. A aplicação dessas técnicas ficou fora do escopo do trabalho.

É valido lembrar que essa foi uma aplicação preliminar da técnica, não podendo ser considerado um resultado final, porém esse estudo é interessante devido a alguns pontos:

- Alguns centros de pesquisa tem estudado a técnica de controle *control-adjointing cell-mapping* (CACM) usando em conjunto uma técnica de otimização online *Reinforced Learning*. Trata-se de uma técnica que ainda tem muito a ser explorada (GÓMEZ; ARRIBAS; SÁNCHEZ, 2012).
- Esse estudo é novidade dentro da UTFPR, está sendo estudado pela primeira vez.
- O desenvolvimento desses estudos está seguindo etapas; primeiro seria o estudo e compreensão do *cell-mapping*, depois a tentativa de replicar resultados de artigos científicos onde CACM é aplicado e conseguir aplicar em outros sistemas e, por último, encontrar formas de otimização do controle CACM, primeiramente *off-line* e depois *on-line*.
- O que pôde ser desenvolvido no escopo deste trabalho foi a primeira etapa: estudo e compreensão do *cell-mapping* e a realização do mapeamento de um controle clássico PI já otimizado. Os resultados obtiveram apenas êxito parcial na replicação do controle mapeado, porém permitiram elucidar alguns dos mecanismos e dificuldades que essa técnica possui.

Com os resultados obtidos na simulação é possível ter uma ideia do funcionamento do modelo em malha aberta, a evolução do algoritmo genético utilizando duas funções distintas para o custo, os resultados obtidos para a planta controlada, o mapa de espaço de estados finitos da planta já controlada para a malha da altura e uma aplicação preliminar de controle via CM.

7 CONCLUSÃO

Neste trabalho foi apresentada uma modelagem matemática para o quadricóptero, desenvolvimento de um controlador PID para o controle de quatro malhas diferentes (os três ângulos e a altura) esse que foi otimizado utilizando um Algoritmo Genético e por fim a aplicação de *cell-mapping* e o começo de um controle utilizando o CM obtido.

A modelagem matemática foi baseada nas equações de Newton-Euler, separando a planta em dois subsistemas interligados, para que assim pudesse ser mais fácil a visualização do funcionamento do sistema. Os controladores PID foram aplicados de maneira digital, realizando uma conversão de suas saídas para o esforço de controle desejado.

Dentre as principais contribuições do trabalho, pode-se destacar a modelagem matemática, feita baseada nas equações de Newton-Euler, a aplicação de controladores PID que foram otimizados via algoritmo genético em quatro malhas de controle (os três ângulos e a altura), a extração do CM da malha controlada da altura e uma aplicação preliminar de um controlador baseado na resposta do CM.

Com as respostas obtidas é possível chegar a algumas conclusões: as respostas obtidas pela função custo ISE foram superiores as da ITSE, isso ocorreu dado ao fato que a segunda tende a punir mais erros que ocorrem após o transitório (dada a influência do tempo na equação) e acabava por prejudicar muito o transitório. Além disso a resposta obtida via CM é muito interessante de ser observada, é possível ver a superfície da atuação do esforço de controle no espaço de estados contido no mapa.

Após a tentativa de controlar-se a planta utilizando o mapa com os esforços de controle foi possível observar a necessidade de técnicas para a melhoria do mapa, a superfície não está totalmente suavizada, ainda acontecendo muitas discrepâncias entre os valores para o esforço em células vizinhas, aqui é possível, em um trabalho futuro, de se realizar um estudo sobre a influência de métodos de estimação, interpolação e filtragem dos valores mapeados.

Ainda existem outras sugestões para implementações futuras no trabalho, sendo: efetuar o controle da posição linear (X e Y) aplicando outras malhas de controle posicionadas entre os dois subsistemas, a utilização de outras técnicas de otimização para os ganhos do PID, efetuar o mapeamento de outras técnicas de controle, como realimentação de estados, controle *Fuzzy* ou controle adaptativos, buscar um controle por otimização do próprio CM, explorando técnicas como o *Reinforced Learning* e otimização *on-line*.

REFERÊNCIAS

- ANAC. **Regulamento Brasileiro da Aviação Civil Especial**. 2015. Acessado em 10 de março de 2017. Disponível em: <<http://www.anac.gov.br/assuntos/paginas-tematicas/drones>>.
- ANG, K. H.; CHONG, G.; LI, Y. Pid control system analysis, design, and technology. **IEEE Transactions on Control Systems Technology**, v. 13, n. 4, p. 559–576, July 2005. ISSN 1063-6536.
- BOCCATO, L. et al. Um estudo da aplicação de algoritmos bio-inspirados ao problema de estimação de direção de chegada. **Revista Controle Automação**, v. 20, p. p. 609–624, 2009.
- BOSE, A. Nonlinear system characterization and optimization. **IRE Transactions on Information Theory**, v. 5, n. 5, p. 30–40, May 1959. ISSN 0096-1000.
- BRESCIANI, T. **Modelling, Identification and Control of a Quadrotor Helicopter**. Tese (Doutorado) — Lund University, 2008.
- BURSAL, F. H.; TONGUE, B. H. A new method of nonlinear system identification using interpolated cell mapping. In: **1992 American Control Conference**. [S.l.: s.n.], 1992. p. 3160–3164.
- CHU, Z. et al. A memetic approach for nanoscale hybrid circuit cell mapping. In: **2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools**. [S.l.: s.n.], 2010. p. 681–688.
- COELLO, C. C.; LAMONT, G. B.; VELDHUIZEN, D. A. van. **Evolutionary Algorithms for Solving Multi-Objective Problems**. Springer US, 2007. ISBN 9780387367972. Disponível em: <http://www.ebook.de/de/product/19293629/carlos_coello_coello_gary_b_lamont_david_a_van_veldhuizen_evolutionary_algorithms_for_solving_multi_objective_problems.html>.
- CONSTANTINO, F. A. S.; KLOSOWSKI, R. F. **Modelagem e Controle de um Helicóptero Quadrirotor**. 2017.
- DARWIN, C. **The Origin of Species by Means of Natural Selection**. [S.l.]: John Murray, Albemarle Street, 1859.
- ELKHOLY, H. talla M. N. **Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches**. Dissertação (Mestrado) — The American University in Cairo (AUC) - School of Sciences and Engineering, 2014.
- GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. [S.l.]: ADDISON-WESLEY PUBLISHING COMPANY, INC., 1989.
- GÓMEZ, M.; ARRIBAS, T.; SÁNCHEZ, S. Optimal control based on cacm-rl in a two-wheeled inverted pendulum. **International Journal of Advanced Robotic Systems**, 2012.
- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. Cambridge, MA, USA: MIT Press, 1992. ISBN 0262082136.
- KUO, Y. S. Representing large cell maps. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 9, n. 11, p. 1238–1241, Nov 1990. ISSN 0278-0070.

LUUKKONEN, T. **Modelling and control of quadcopter**. [S.l.]: Aalto University, 2011.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. [S.l.]: Springer, 1996.

NARANJANI, Y.; SARDAHI, Y.; SUN, J. Q. A genetic algorithm and cell mapping hybrid method for multi-objective optimization problems. In: **2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)**. [S.l.: s.n.], 2014. p. 1–5.

NETO, M. S. **Topografia com drones, a evolucao tecnologica**. 2014. Acessado em 10 de fevereiro de 2017. Disponível em: <<http://blog.droneng.com.br/topografia-com-drones/>>.

PAPA, M.; TAI, H.-M.; SHENOI, S. Cell mapping for controller design and evaluation. **IEEE Control Systems**, v. 17, n. 2, p. 52–65, April 1997. ISSN 1066-033X.

SIQUEIRA, H. **Previsão de Séries de Vazões com Redes Neurais Artificiais e Modelos Lineares Ajustados por Algoritmos Bio-Inspirados**. Dissertação (Mestrado) — Universidade Estadual de Campinas (UNICAMP-SP) Faculdade de Engenharia Elétrica e Computação Departamento de Engenharia de Sistemas, 2009.

WU, H.; SU, W.; LIU, Z. Pid controllers: Design and tuning methods. In: **2014 9th IEEE Conference on Industrial Electronics and Applications**. [S.l.: s.n.], 2014. p. 808–813. ISSN 2156-2318.

XIA, Y. et al. An integrated optimization approach for nanohybrid circuit cell mapping. **IEEE Transactions on Nanotechnology**, v. 10, n. 6, p. 1275–1284, Nov 2011. ISSN 1536-125X.