



**UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE
E AUTOMAÇÃO - CEAU**



THAYS CRISTINA DA SILVA ALVES

**SIMULAÇÃO DE UM SISTEMA DE CONTROLE DE REENTRADA
NA ATMOSFERA PARA VEÍCULOS AEROSPACIAIS
REUTILIZÁVEIS**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO**

Ouro Preto, 2022

THAYS CRISTINA DA SILVA ALVES

**SIMULAÇÃO DE UM SISTEMA DE CONTROLE DE REENTRADA
NA ATMOSFERA PARA VEÍCULOS AEROSPACIAIS
REUTILIZÁVEIS**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenharia de Controle e Automação.

Orientador: Prof. Dr. Fernando Cortez Sica

Coorientador: Prof. Dr^a. Adrielle de Carvalho Santana

**Ouro Preto
Escola de Minas – UFOP
2022**



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Thays Cristina da Silva Alves

Simulação de um Sistema de Controle de Reentrada na Atmosfera para Veículos Aeroespaciais Reutilizáveis

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia de Controle e Automação

Aprovada em 04 de Novembro de 2022

Membros da banca

Prof. Dr. Fernando Cortez Sica - Orientador (DECOM - Universidade Federal de Ouro Preto)
Profa. Dra. Adrielle de Carvalho Santana - Coorientadora (DECAT - Universidade Federal de Ouro Preto)
Prof. M.Sc. João Carlos Vilela de Castro - Convidado (DECAT - Universidade Federal de Ouro Preto)
Prof. M.Sc. Vinícius Antônio de Oliveira Martins - Convidado (DECOM - Universidade Federal de Ouro Preto)

Fernando Cortez Sica, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 07/11/2022



Documento assinado eletronicamente por **Fernando Cortez Sica, PROFESSOR DE MAGISTERIO SUPERIOR**, em 07/11/2022, às 16:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0421621** e o código CRC **A93C106F**.

*Este trabalho é dedicado às mulheres que
me fizeram acreditar que é possível alcançar as estrelas.*

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a minha mãe por sempre me dar todo o apoio necessário para que eu chegasse e concluísse essa etapa. À toda a minha família que sonhou junto comigo. Aos orientadores Dr. Fernando Cortez e Dra. Adrielle de Carvalho pelo aprendizado e compreensão para a realização deste trabalho. E a Universidade Federal de Ouro Preto pelo conhecimento adquirido durante a minha graduação e pelas oportunidades como o Movimento Empresa Júnior.

*“Um pequeno passo para o homem,
mas um grande salto para a humanidade.” (Neil Armstrong)*

RESUMO

Diversos veículos aeroespaciais tem suas peças separadas e descartadas durante as etapas do voo, as quais podem cair no mar ou ficar na órbita da Terra. Para evitar o lixo espacial e para reutilizar as peças, nos últimos anos estão sendo desenvolvidos veículos de lançamento reutilizáveis. Neste sentido, este trabalho propõe um sistema de controle de atitude utilizado para controlar a etapa de reentrada na atmosfera. Foram utilizados dois Algoritmos Genéticos, sendo um para gerar os ângulos necessários para as raquetes corrigirem o ângulo do foguete e, o outro, capaz de gerar os melhores parâmetros K_p e K_d do controlador PD responsável pelo controle do Empuxo. Para o desenvolvimento deste sistema de controle, foram elaborados os cálculos necessários para traçar a trajetória do foguete. Como conclusão, o sistema de controle proposto neste projeto conseguiu corrigir o ângulo de ataque do veículo para $1,5^\circ$ em torno de 45s.

Palavras-chaves: Controlador PD. Foguetes. Controle de Atitude. Algoritmo Genético.

ABSTRACT

An aerospace vehicle has its parts separate and disposable during the flight stages, and these parts can fall into the sea or stay in Earth orbit. To avoid space junk and to reuse the parts, in recent years reusable launch vehicles are being developed. Therefore, this work proposes an attitude control system used to control the re-entry stage in the atmosphere. Two Genetic Algorithms were used, one to generate the necessary angles for the rackets to correct the rocket's angle of attack and the other, capable of generating the best coefficients K_p and K_d of the PD controller responsible for controlling the Thrust. For the development of this control system, the necessary calculations were made to trace the trajectory of the rocket, and *Matlab* and *Visual Studio Code* (C language) were used to develop the code and analyze the results obtained. In conclusion, the control system proposed in this project was able to correct the vehicle's angle of attack to 1.5° around 45s.

Key-words: PD controller. Rockets. Attitude Control. Genetic Algorithm.

LISTA DE ILUSTRAÇÕES

Figura 1 – Relação entre o eixo inercial, XYZ, e o eixo móvel. Fonte: (MENEGALDO, 2020)	16
Figura 2 – Ângulos Pitch, Yaw e Roll. Fonte: Autoria Própria (2021)	17
Figura 3 – Referencial de Coordenadas de Reentrada. Fonte: Autoria Própria (2021)	17
Figura 4 – Modelo do Space Shuttler. Fonte: Autoria Própria (2021)	19
Figura 5 – Representação das etapas do pouso da Falcon 9. Fonte: (SPACEEX, 2006)	20
Figura 6 – Forças sobre um foguete. Fonte: (ALVES et al., 2020) - Adaptado pela autora.	21
Figura 7 – Forças aerodinâmica no foguete. Fonte: (LEAL, 2017) - Adaptado pela autora.	23
Figura 8 – Fluxograma Algoritmo Genético. Fonte: (TAVARES; NEDJAH; MOURELLE, 2015) - Adaptado pela autora.	26
Figura 9 – Forças atuando no foguete em relação ao um eixo: (YAMAMOTO, 2022)	30
Figura 10 – Diagrama de blocos de uma malha fechada. Fonte: (OGATA, 2010) - Adaptado pela autora.	31
Figura 11 – Malha fechada do sistema de controle. Fonte: Autoria própria	32
Figura 12 – Fluxograma Algoritmo Genético. Fonte: Autoria própria	33
Figura 13 – Trajetória do foguete sem o controlador. Fonte: Autoria Própria.	34
Figura 14 – Valores da Força Empuxo em relação à altura sem o controle. Fonte: Autoria Própria.	35
Figura 15 – Ângulo de ataque em função da distância sem o controlador. Fonte: Autoria Própria.	36
Figura 16 – Angulação do foguete, em relação ao tempo, utilizando o Controlador PD e algoritmo genético para a angulação das raquetes. Fonte: Autoria Própria.	37
Figura 17 – Dispersão do Empuxo gerado pelo AG durante os testes. Fonte: Autoria Própria.	38
Figura 18 – Dispersão do Empuxo igual a 20000N gerado pelo AG durante os testes. Fonte: Autoria Própria.	38
Figura 19 – Valores da altura em relação ao tempo após a correção do PD. Fonte: Autoria Própria.	39
Figura 20 – Valores de velocidade em relação ao tempo após a correção do PD. Fonte: Autoria Própria.	40

LISTA DE TABELAS

Tabela 1 – Valores iniciais.	29
Tabela 2 – Valores da dispersão do Empuxo gerado pela AG e o Empuxo teste.	39

LISTA DE ABREVIATURAS E SIGLAS

VLR	Veículo de Lançamento Reutilizável
P	Controladores Proporcional-Derivativos
NASA	National Aeronautics and Space Administration
FE	Força Empuxo
P	Força Peso
FA	Força Aerodinâmicas
Ve	Gases de Escapes
FG	Força Gravitacional
AG	Algoritmo Genético
CM	Centro de Gravidade
Kp	Ganho proporcional
Ki	Ganho integral
Kd	Ganho derivativo

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	14
1.2	Estruturação da Monografia	14
2	REVISÃO DA LITERATURA	16
2.1	Atitude de um Veículo Aeroespacial	16
2.2	Reentrada de um Veículo Aeroespacial	17
2.3	Modelos de Veículos Reutilizáveis	18
2.4	Forças que agem sobre um foguete	21
2.4.1	<i>Força de Empuxo (FE)</i>	21
2.4.2	<i>Força gravitacional (FG)</i>	22
2.4.3	<i>Força Aerodinâmica (FA)</i>	23
2.5	Propelentes	24
2.6	Controle de Atitude	24
2.7	Algoritmos Genéticos	26
2.8	Estado da Arte	27
3	DESENVOLVIMENTO	29
3.1	Condições iniciais considerados para os cálculos.	29
3.2	Cálculo da trajetória	29
3.3	Sistema de controle	31
3.4	Algoritmo Genético	32
4	RESULTADOS	34
4.1	Simulação da trajetória de referência	34
4.2	Controle dos ângulos das raquetes	36
4.3	Controle do Empuxo após o Algoritmo Genético	37
5	CONCLUSÃO	41
	REFERÊNCIAS	42
	APÊNDICE A – CÓDIGO TRAJETÓRIA	44
	APÊNDICE B – CÓDIGO DO ALGORITMO GENÉTICO - ÂNGULO DAS RAQUETES	47

APÊNDICE C – CÓDIGO DO ALGORITMO GENÉTICO - COEFICIENTES KP E KD	51
---	-----------

1 INTRODUÇÃO

Um foguete é um veículo aeroespacial utilizado para levar satélites, sondas ou tripulantes para o espaço. Durante as etapas/estágios do voo, algumas partes do veículo aeroespacial são separadas do veículo e descartadas para diminuir a sua carga e manter a aceleração, mesmo com a redução do empuxo em altas altitudes. Como algumas partes do foguete são descartáveis, só uma pequena fração da massa levada para o espaço é considerada carga ativa. De acordo com [Carvalho, Lima e Gonçalves \(2021\)](#), no período entre 1958 e 2020, houve mais de 5.980 lançamentos que geraram em torno de 25.297 objetos rastreados na órbita da Terra, ou seja, existe mais lixo espacial do que satélites operacionais.

Um veículo de lançamento reutilizável (VLR) é um veículo aeroespacial que pode ter suas peças reutilizadas após o seu lançamento, evitando assim, o lixo espacial. Esse tipo de veículo está sendo desenvolvido por empresas com o objetivo de gerar menos lixo espacial e reutilizar peças. A reutilização de peças, proporciona, também, a redução de custos das viagens e missões espaciais. As etapas de reentrada na atmosfera são gerenciadas por sistemas de controle de atitude.

O objetivo deste trabalho consiste na modelagem e implementação de um controlador de veículos aeroespaciais reutilizáveis no que tange à etapa de reentrada na atmosfera terrestre. Para tanto, o controlador receberá seus parâmetros de uma camada de *software* baseada em algoritmos genéticos (evolutivos). Os algoritmos evolutivos toam por base, de forma antecipada, as variações do meio externo no qual o veículo se encontra (densidade do ar, temperatura, velocidade, etc). Para validação do sistema, são realizadas simulações a partir de dados reais de voo.

1.1 Justificativa

Este estudo se justifica por ser uma área de estudo recente, com escassez de literatura sobre o tema. Além disso, é uma área promissora, que pode auxiliar na redução de custos a partir da reutilização de peças, e com isso contribuir no avanço da exploração espacial. A sua importância é que, com a corrida espacial para a próxima missão que levará humanos à Lua, algumas empresas de veículos aeroespaciais, como a SpaceX e a Blue Origin, estão desenvolvendo foguetes parcialmente ou totalmente reutilizáveis através de sistemas de controle de atitude, tornando a área promissora e com oportunidades para desenvolvimento de outros tipos de sistema de controle.

1.2 Estruturação da Monografia

No capítulo 2 são apresentados os conceitos, através de uma revisão de literatura, para a facilitar a compreensão deste trabalho, enquanto o capítulo 3 contém o desenvolvimento deste

trabalho apresentando os cálculos realizados para determinar as equações de forças e a malha fechada do sistema utilizando o controlador do tipo PD. Nos capítulos 4 e 5, é apresentado o resultado e a conclusão deste trabalho.

2 REVISÃO DA LITERATURA

Para um melhor entendimento do desenvolvimento deste projeto, alguns conceitos deverão, inicialmente, ser abordados de modo que se tenha uma visão geral dos pontos a serem tratados. Como conceitos básicos pode-se mencionar: atitude de um foguete, etapa de reentrada na atmosfera, modelos de veículos reutilizáveis, controle de atitude, forças que agem sobre o foguete, propelentes, controlador PID e Algoritmos Genéticos descritos a seguir.

2.1 Atitude de um Veículo Aeroespacial

De acordo com (MENEGALDO, 2020), a atitude é a orientação de um corpo rígido em relação a um referencial inercial e pode ser representada como: matriz de cossenos diretores, ângulos de Euler, eixos de Euler, quatérnios, dentre outros.

Os elementos do sistema de coordenadas móvel (x, y, z) podem ser expressos no sistema de coordenadas inercial (X, Y, Z) em função dos seus componentes (1).

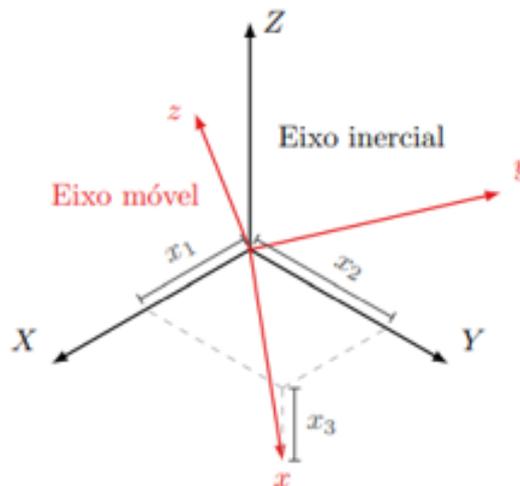


Figura 1 – Relação entre o eixo inercial, XYZ, e o eixo móvel. Fonte: (MENEGALDO, 2020)

Analisando essa relação entre os sistemas de coordenadas, é possível obter uma matriz do sistema móvel em relação ao sistema inercial, sendo representada pela Equação (1.1).

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ z1 & z2 & z3 \end{bmatrix} \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \quad (2.1)$$

A atitude de um foguete é dada pela orientação de rotação em torno do centro de massa do veículo por meio dos ângulos *Pitch* (arfagem) – considerado o mais importante para calcular

a trajetória de voo pela sua movimentação em torno do eixo horizontal –, *Yaw* (guinada) e o *Roll* (rolagem). A Figura 2 mostra as incidências dos ângulos *Pitch*, *Yaw* e *Roll* em relação à fuselagem de uma aeronave.

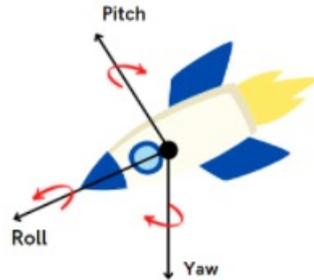


Figura 2 – Ângulos Pitch, Yaw e Roll. Fonte: Autoria Própria (2021)

Pode-se observar, pela figura 2, que os ângulos *Pitch* e *Yaw* podem ser idênticos por causa da simetria geométrica. Desta forma, pode-se utilizar o mesmo controle para ambos, porém, é importante salientar que o controle para o ângulo *Roll* precisa ser notadamente diferente (PORTO, 2017).

2.2 Reentrada de um Veículo Aeroespacial

A reentrada de um veículo na atmosfera é considerada crítica pela sua complexidade, isto porque os fenômenos físicos envolvidos nessa etapa são diferentes do que se usa na superfície terrestre (AFEITOS, 2014). Todo veículo aeroespacial tem um referencial de coordenadas de reentrada, onde os vetores \hat{X} , \hat{Z} , \hat{V} representam, respectivamente, o eixo horizontal, vertical e o de velocidade. A Figura 3 ilustra o referencial de coordenadas de reentrada. O ângulo formado na reentrada, conhecido como ângulo de ataque, é dado por γ que denota o valor angular entre o plano horizontal e o vetor de velocidade. O controle do valor do ângulo γ tem a sua importância para que a aeronave não se desintegre assim que atinja a atmosfera e, também, para anular as forças aerodinâmicas que atuam no foguete. Desta forma, o ângulo γ representa um dos fatores mais importantes no processo da reentrada.

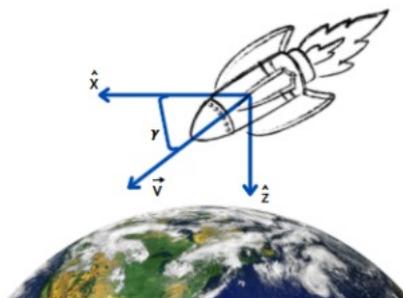


Figura 3 – Referencial de Coordenadas de Reentrada. Fonte: Autoria Própria (2021)

Logo, é preciso ter conhecimento sobre os princípios fundamentais e as etapas da reentrada para conseguir fazer um controle de atitude. Os princípios fundamentais se baseiam na desaceleração, no aquecimento e impacto que ocorrem na parte externa do veículo. A etapa de desaceleração é a mais importante porque se a desaceleração for muito rápida, o veículo não irá atingir a superfície terrestre e terá, como consequência, a volta para o espaço. Porém, se a velocidade na etapa da reentrada for muito baixa, o veículo pode se chocar com a superfície terrestre. O aquecimento do veículo ocorre devido à fricção de sua superfície com a atmosfera. Por mais que a parte externa do veículo aeroespacial seja desenvolvida para aguentar temperaturas extremas, é preciso ter um controle dessa temperatura para não afetar negativamente os possíveis tripulantes e os circuitos eletrônicos da aeronave. Por fim, o impacto que ocorre na superfície terrestre, também conhecido como aterragem, é o momento de pouso do veículo em um determinado lugar (AFEITOS, 2014).

Os tipos de reentradas podem ser classificados como (PREADO; ALMEIDA, 2007):

- **Reentrada balística:** o veículo irá cair quando entrar na atmosfera. É preciso ter um controle da posição do eixo central do veículo (eixo da velocidade), pois se o veículo tende muito para um dos lados, terá consequências devido a resistência do ar;
- **Reentrada de voo planado (*glide*):** é uma reentrada em que o veículo entra na atmosfera com um ângulo de 30° ou mais garantindo, assim, uma estabilidade entre as forças de sustentação e a gravidade (AFEITOS, 2014). É preciso controle para se ter um equilíbrio entre as forças de sustentação e gravitacional, a força de arrasto tem como objetivo diminuir a velocidade no período que a atmosfera não é densa até que a força gravitacional seja maior que a força de sustentação;
- **Reentrada de voo salto (*skip*):** o foguete dá “saltos”, ou seja, irá subir e depois descer durante a trajetória na intenção de alterar a intensidade em altitudes mais altas provocando assim uma desaceleração do veículo de forma menos brusca, resultando em um tipo de reentrada mais adequada para missões com tripulantes.

O tipo de controle de atitude na etapa de reentrada de um foguete é importante para que a trajetória do veículo tenda a ser a mais estável e segura possível. Em foguetes, este controle é utilizado na etapa de reentrada para fazer o pouso do foguete controlando a queima do combustível, posição e o ângulo do veículo.

2.3 Modelos de Veículos Reutilizáveis

O primeiro tipo de sistema reutilizável foi desenvolvido pela *National Aeronautics and Space Administration* (NASA), e é conhecido como *Space Shuttle* (Ônibus Espacial). Seu lançamento era executado como o de foguete, as manobras na órbita eram executadas como

uma espaçonave e o seu pouso era feito como um avião (??). O Ônibus Espacial é representado Figura 4, e tinha suas peças, exceto o tanque externo, jogadas no mar e depois recicladas.

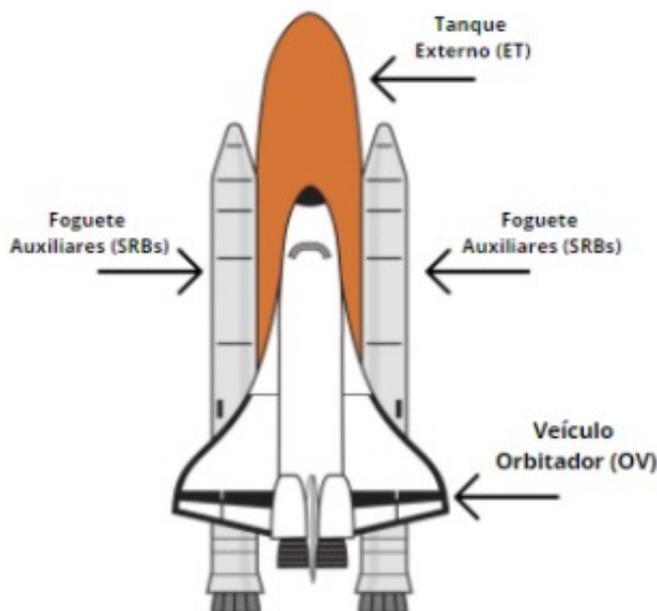


Figura 4 – Modelo do Space Shuttler. Fonte: Autoria Própria (2021)

No total, a NASA desenvolveu 6 VLRs, sendo eles: *Enterprise, Columbia, Challenger, Discovery, Atlantis e o Endeavor*. O sistema de controle de atitude na fase de reentrada desses veículos foi feito por meio do piloto automático, que executou diversas voltas de rotação para diminuir a velocidade do veículo (MARS, 2021). O controle de voo desses veículos aeroespaciais era feito por quatro computadores IBM AP-101 de 32 bits, que tinham um desempenho parecido com os computadores pessoais da época (TOMAYKO, 1988). Em 12 de abril de 1981 o modelo *Columbia* se tornou o primeiro foguete reutilizável a ser lançado, ficando dois dias em órbita. Desde 2011 a NASA não desenvolve mais os ônibus espaciais, mas a tecnologia usada neles está sendo utilizada no foguete *Space Launch System*, que será utilizado para levar astronautas à Lua em 2024 pelo programa Artemis.

Em 2015 a empresa de foguetes *Space Exploration Technologies Corp*, também conhecida como SpaceX, lançou a Falcon 9 e conseguiu recuperar o propulsor do primeiro estágio do veículo por meio do controle de atitude dos seus propulsores, se tornando o primeiro foguete reutilizável de classe orbital do mundo. A Falcon 9 é um foguete com dois estágios, que recebe esse nome por ter 9 propulsores Merlins que queimam oxigênio líquido e querosene refinado. Na etapa de lançamento, os 9 motores Merlin levam o foguete até uma altitude de 100 Km, e após alcançar essa altitude começa a etapa *Main Engine Cutoff*, em que motores são desligados, liberando o primeiro estágio e se preparando para voltar para a Terra. O pouso é efetuado por meio do controle do propulsor *Booster* (SPACEX, 2006).

O lugar do pouso da Falcon 9 é determinado de acordo com a sua carga: se for leve é feita na Terra, e se for pesada o pouso é feito em uma balsa autônoma. As aletas são abertas a partir do *Main Engine Cutoff* seguindo a sua rota em direção à Terra, tendo pequenos ajustes com jatos dos motores de atitude. Depois é a fase de *Return Burn*, em que os três motores são ligados para diminuir a velocidade e evitar que ele seja queimado na atmosfera. Logo após, os motores são desligados e o *Booster*, que é controlado pelas aletas, continua caindo até chegar a altitude correta para que apenas um motor faça a queima na hora do pouso. Durante a reentrada o veículo fica posicionado para o mar e só tem sua posição centralizada na balsa quando o motor é ligado. Todo esse processo é representado pela Figura 5.

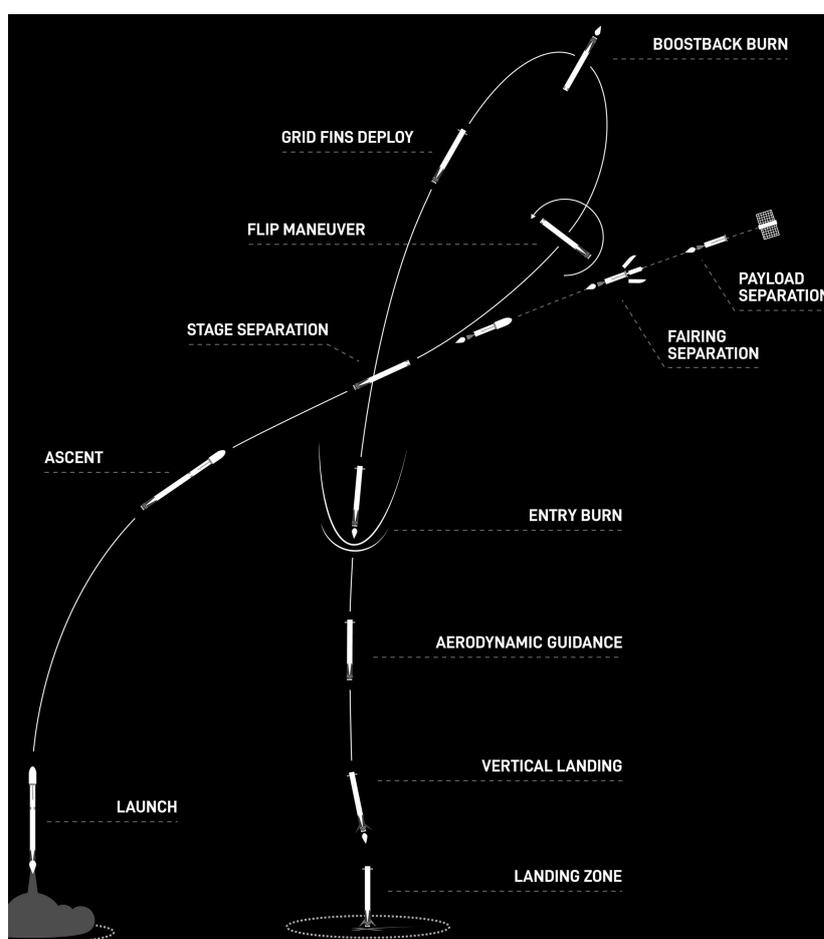


Figura 5 – Representação das etapas do pouso da Falcon 9. Fonte: (SPACE X, 2006)

A *New Shepard* é um modelo de foguete totalmente reutilizável criado pela Blue Origin, empresa aeroespacial. O tipo de pouso desse foguete é feito através da tecnologia de pouso e decolagem vertical (VTVL), o mesmo utilizado com a Falcon 9. O controle de atitude na etapa de reentrada se encontra nos motores-foguetes principais que vão desacelerar e realizar um pouso vertical propulsivo de forma autônoma, para garantir que o veículo de lançamento reutilizável consiga receber as instruções de orientação para um pouso seguro. (AN et al., 2019).

2.4 Forças que agem sobre um foguete

As forças que agem em um foguete durante seu movimento na atmosfera são as forças de empuxo (FE), força peso (P) e aerodinâmicas (FA) (ALVES et al., 2020). Na Figura 6 é possível analisar as forças agindo no foguete.

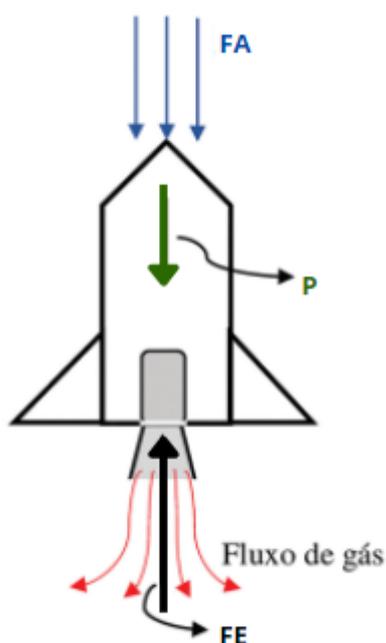


Figura 6 – Forças sobre um foguete. Fonte: (ALVES et al., 2020) - Adaptado pela autora.

2.4.1 Força de Empuxo (FE)

É a força gerada pela pluma do foguete, ou seja, ocorre através da velocidade do fluxo dos gases de escapes (V_e), que é a velocidade mínima necessária para que um corpo escape da atração gravitacional de um corpo celeste, quando o combustível está sendo queimado em variação do tempo ($\frac{m}{dt}$). Como o combustível consumido pelo foguete é considerado negativo por causa da sua queima, a força de tração tem o sentido contrário a velocidade de escape, tornando a força de empuxo como:

$$FE = -V_e \left[\frac{m}{dt} \right] \quad (2.2)$$

Mas, para o controle de atitude de empuxo do motor-foguete, é necessário considerar outros aspectos como representado melhor na fórmula abaixo:

$$FE = -V_e \left[\frac{m}{dt} \right] + (p_0 - P_{atm}) A_b \quad (2.3)$$

Onde,

V_e = velocidade de escape dos gases;

p_0 = Pressão em determinada altura;

P_{atm} = pressão atmosférica;

A_b = área do bocal do motor.

A velocidade de escape dos gases pode ser representada pela seguinte fórmula:

$$V_e = \sqrt{\frac{2GM}{R}} \quad (2.4)$$

Usando os valores da Terra, é possível concluir que:

$$V_e = \sqrt{\frac{2 * (6,67 \cdot 10^{11})(5,58 \cdot 10^{24})}{6,38 \cdot 10^6}} \quad (2.5)$$

$$V_e = 11,2 \text{ Km/s} \quad (2.6)$$

2.4.2 Força gravitacional (FG)

Quando o foguete está se movimentando, a força de todos os corpos celestes que são exercidas no corpo do mesmo, é denominada força gravitacional (FG). Por meio da Lei da Gravitação Universal, proposta por Newton, é possível relacionar a força gravitacional entre dois corpos.

$$FG = \frac{G(m_1 \cdot m_2)}{r^2} \quad (2.7)$$

Onde,

G = gravidade universal;

m_1 = massa do primeiro corpo;

m_2 = massa do segundo corpo;

r = distância entre os corpos 1 e 2.

Já na segunda Lei de Newton, em que Força de um objeto é igual a sua massa vezes a aceleração do mesmo, a aceleração de um veículo espacial que está orbitando a Terra pode ser representada por:

$$a = \frac{F}{m} \quad (2.8)$$

Fazendo uma associação entre as Equações (2.8) e (2.9), considerando que $m1$ é a massa da Terra e $m2$ é a massa do veículo, temos:

$$a = \frac{\frac{G(m1,m2)}{r^2}}{m2} = \frac{G.m1}{r^2} \quad (2.9)$$

Entretanto, essa fórmula só é possível quando não são apresentados distúrbios externos e quando o objeto está em uma circunferência perfeita, ou seja, ambas restrições não são aceitas na Terra. Com isso, se vê necessário o estudo das possíveis perturbações que podem interferir na aceleração do foguete durante sua trajetória.

2.4.3 Força Aerodinâmica (FA)

Trata-se das forças, em sentido oposto, que o ar faz sobre o foguete quando ele está em movimento. De acordo com Zanardi (1990), essa força pode ser separada em uma Força de Arrasto (F_{ar}) e em uma Força de Sustentação, ou Força normal (F_n).

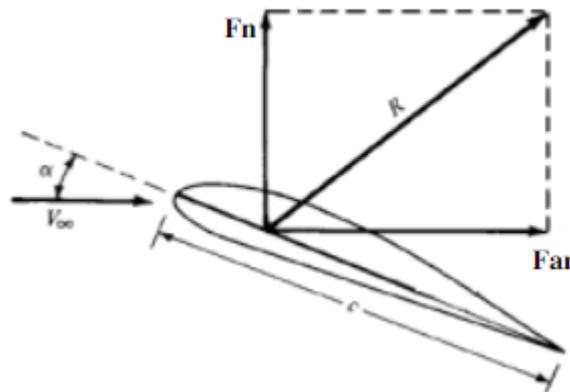


Figura 7 – Forças aerodinâmicas no foguete. Fonte: (LEAL, 2017) - Adaptado pela autora.

A força de arrasto é a força projetada na mesma direção do escoamento. E a força de sustentação é equivalente a força normal na direção do escoamento (LEAL, 2017). Representadas pelas fórmulas abaixo:

$$F_{ar} = \frac{\rho.Cd.A.V_o^2}{2} \quad (2.10)$$

$$F_n = \frac{\rho.Cl.A.V_o^2}{2} \quad (2.11)$$

Onde,

$\rho =$ densidade do ar;

C_d e C_l = coeficientes de arrasto e sustentação;

A = área do veículo;

V_o = velocidade relativa

Os valores de C_l e C_d são grandezas adimensionais, podendo ser aproximadas como o número de Mach e do ângulo de ataque γ . Como explicado na seção 2.1, o ângulo γ é formado entre o eixo horizontal e a direção de velocidade do foguete. Quando a FA está no sentido oposto ao centro de gravidade (CM), ou seja, no eixo no eixo horizontal do foguete, o ângulo de ataque será nulo, tornando a força aerodinâmica nula. Para o cálculo é considerado que a atmosfera tem a mesma velocidade de rotação da Terra.

Na etapa de reentrada de um veículo, o controle do pouso é feito pelos propulsores, logo a pluma dos gases de exaustão tem uma influência na aerodinâmica do foguete. Em [Zanardi \(1990\)](#) é citado que, os valores do coeficiente de arrasto sofrem uma redução drástica quando o motor é ligado, porque a pressão na parte inferior do foguete diminui por causa do fluxo de jato contrário ao escoamento.

2.5 Propelentes

Segundo a segunda Lei de Newton, “Toda ação gera uma reação”. Nesse caso o que irá gerar o movimento do foguete é a queima dos propelentes pelos seus motores. Os propelentes podem ser do tipo sólido, líquido e híbrido. De acordo com [Zanatta \(2019\)](#), o tipo de propelente sólido foi utilizado nos modelos antigos de foguetes. Uma vez iniciada, não há como interromper ou controlar a queima. No entanto, é possível ter um controle da vazão de massa por meio da evolução da área superficial da cavidade do grão, que por sua vez resulta em diferentes perfis da curva de empuxo ao longo do tempo. Já o tipo de propelente líquido, tem um complexo sistema de distribuição, que inclui turbinas, compressores e válvulas, sendo responsável pela circulação dos propelentes nas linhas de distribuição e pela regulagem da pressão nas mesmas.

Os propelentes do tipo híbrido têm a junção do tipo líquido e sólido, um será o tipo de oxidante e o outro o combustível. Este tipo de propelente tem como vantagens a segurança, pois é possível controlar a sua queima, baixo custo, reinicialização de empuxo e um impulso maior em comparação aos motores que utilizam apenas o propelente sólido ([OLIVEIRA, 2016](#)).

2.6 Controle de Atitude

O controle de atitude em veículos aeroespaciais tem como objetivo principal minimizar os impactos das perturbações externas ao veículo para que ele consiga manter a sua estabilidade ([SAMPAIO, 2006](#)). Logo, é preciso estudar e analisar qual tipo de controlador melhor se adequa para cada tipo de projeto. É possível utilizar mais de um tipo de controlador sendo eles já tradicionais ou, ainda, os baseados em sistemas conexionistas ou evolucionários. Como

apresentado nos tópicos anteriores, durante a trajetória, o veículo irá encontrar perturbações, logo se vê necessário um sistema de controle robusto que seja capaz de mantê-lo na trajetória e atitude desejada. Ou seja, é preciso que o sistema de controle gere sinais ou esforços – o controlador terá que controlar os ângulos de arfagem e guinada. Para tal, se vê a necessidade de um estudo sobre controladores para determinar qual seria o melhor de acordo com as necessidades do projeto.

“A escolha do tipo de controlador a ser utilizado deve ser decidida com base na natureza da planta e nas condições de operação, incluindo certas considerações como segurança, custo, disponibilidade, confiabilidade, precisão, peso e tamanho.” (OGATA, 2010, p. 30)

Existem 6 tipos principais de controladores (OGATA, 2010):

- Controladores de duas posições ou on-off;
- Controladores proporcionais (P);
- Controladores integrais (I);
- Controladores proporcional-integrais (PI);
- Controladores proporcional-derivativos (PD);
- Controladores proporcional-integral-derivativos (PID);

O controlador PID é um dos mais utilizados e conhecidos pelo fato de que, quando ajustados corretamente, eles podem conduzir o sistema de controle linear e invariante no tempo, a um desempenho satisfatório (BEZERRA et al., 2007). Esse controlador recebe esse nome porque contém três coeficientes que são variados para obter a resposta desejada (OGATA, 2010):

- **Ganho proporcional (K_p):** o valor da saída vai ser proporcional com o valor do erro em função do tempo. Esse controlador vai atuar na resposta transitória e no regime permanente, ampliando a velocidade de resposta e a oscilação, causando uma redução do erro de regime estacionário. Em resumo, quanto maior for o valor do erro, maior vai ser o controle desse sistema. Sua função de transferência é representada pela Equação 2.12:

$$\frac{U(s)}{E(s)} = K_p \quad (2.12)$$

- **Ganho integral (K_i):** seu desempenho é mais focado no regime permanente. Eliminando completamente o erro estacionário e, em contrapartida, interferindo na resposta transitória

que é responsável pela estabilidade, aumentando assim a oscilação do sistema. A função de transferência desse controlador é descrita pela Equação 2.13:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.13)$$

- **Ganho derivativo (Kd):** esse controle é proporcional à variação do sinal do erro em função do tempo. Age na resposta transitória, reduzindo a oscilação sem afetar o regime permanente. Sua função de transferência é descrita pela Equação 2.14:

$$\frac{U(s)}{E(s)} = K_d s \quad (2.14)$$

Tendo o conhecimento que o controlador PID é a junção do tipo proporcional, integral e derivativo, é possível concluir que a sua função de transferência é dada pela Equação 2.15:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2.15)$$

2.7 Algoritmos Genéticos

Os Algoritmos Genéticos (AG) são inspirados na teoria de evolução proposta por Darwin, em que a seleção privilegia os indivíduos mais aptos com maior probabilidade de reprodução. Sendo assim, os AG buscam uma melhor solução para um específico problema, através de populações de soluções codificadas através de cromossomos artificiais e após diversos ciclos de evolução a população irá conter indivíduos mais aptos (PINHO et al., 2013).

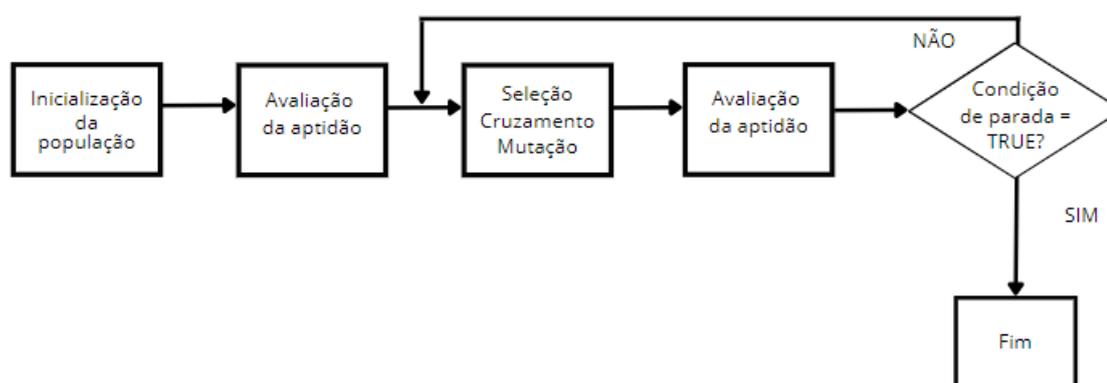


Figura 8 – Fluxograma Algoritmo Genético. Fonte: (TAVARES; NEDJAH; MOURELLE, 2015) - Adaptado pela autora.

Na Figura 8 é possível ver um fluxograma que explica como funcionam as etapas para a execução de um algoritmo genético. De acordo com Tavares, Nedjah e Mourelle (2015), na

primeira etapa é gerada, aleatoriamente, a população inicial de possíveis soluções em que cadeias de cromossomos binários formam indivíduos para o problema. Estes indivíduos podem ser representados por números reais, binários, permutação de símbolos ou símbolos repetidos. A segunda etapa é considerada o elo entre o AG e o mundo externo (PACHECO et al., 1999), pois é nela que será avaliado se os indivíduos estão aptos para a solução do problema através da avaliação de aptidão (*Fitness Function*). Como é mostrado na Figura 8, caso a melhor escolha para a solução seja encontrada, é possível finalizar o processo e exibir a solução ótima encontrada. Caso não seja encontrada uma solução ótima, os indivíduos passam pela terceira etapa, que é o processo de Seleção, Cruzamento e Mutação. Nesta etapa é feita uma seleção dos indivíduos mais aptos para se reproduzirem e gerarem outra população de indivíduos aptos para a solução do problema, a seleção dos indivíduos pode ser feita por ranking, torneio, normalização linear, normalização exponencial ou por truncamento. O Cruzamento (*Crossover*) tem como objetivo criar novos indivíduos a partir da troca do material genético em que os descendentes são diferentes dos seus pais, mas obtêm características de ambos os pais. Geralmente é utilizado o cruzamento simples ou de um ponto, em que os cromossomos de cada um dos genitores são divididos em pontos aleatórios e depois são alterados de forma aleatória. Para evitar que a população tenha indivíduos semelhantes, é utilizada a Mutação que é responsável pela diversidade da população. Após todo esse processo, o critério de parada se dá normalmente pela quantidade de interações que são propostas ou após múltiplos processos de reprodução.

2.8 Estado da Arte

No trabalho realizado por Zou et al. (2015), foi desenvolvido um controlador de atitude para que o VLR consiga rastrear o comando de orientação sob restrição de entrada, incerteza do modelo e perturbações externas. O modelo orientador foi um sistema de *feedback* estrito com incertezas, e após foi projetado um controle *backstepping* adaptativo robusto. O método *backstepping* tem como objetivo manter o ponto de equilíbrio estável e realizar a trajetória de referência. Apresenta a elaboração de uma função de Lyapunov (usada para analisar a estabilidade transitória de um sistema) para os sistemas não lineares. Como resultado, a restrição de entrada é tratada e o controle *backstepping* adaptativo robusto obteve sucesso no controle de rastreamento dos ângulos de atitude.

O projeto desenvolvido por Mao et al. (2018) consistiu em elaborar um sistema de controle de atitude para o modelo aero servovelástico que consiga garantir o rastreamento de comando das orientações para a etapa de reentrada. Para que isso seja possível, foi utilizado o método de *backstepping* para desenvolver um controlador de subsistemas de dois circuitos. Para garantir o rastreamento dos comandos de orientação foi utilizado um controlador *backstepping* e, para melhorar o desempenho do rastreamento dos ângulos de atitude na etapa de reentrada, foi utilizada uma lógica *fuzzy* adaptável tipo 2. Tais técnicas podem ser usadas em situações onde muitas incertezas estão presentes para proporcionar um desempenho superior em uma série

de aplicações onde é necessário, por exemplo, estimar as matrizes de função não linear. Estas estimativas garantem os ajustes dos parâmetros do controlador.

Em [Gao e Yao \(2018\)](#), foi elaborado um sistema de controle capaz de tolerar falhas do atuador, atraso do controle, saturação de entrada, distúrbios externos na etapa da reentrada e incertezas de parâmetros do tempo. Isso foi feito através da rede *neural radial-basis function* (função de ativação de base radial) junto com o método de controle tolerante a falhas adaptativas. Este método consiste em um sistema de controle desenvolvido estrategicamente para garantir a estabilidade do sistema e uma performance aceitável mesmo com a presença de falhas no sistema. A rede *neural radial-basis function*, que recebe esse nome por causa da ativação dos neurônios, denota a distância em relação a um ponto. A saída deste tipo de rede neural assemelha-se à uma parábola. O que difere esse controlador dos outros é o melhoramento da confiabilidade do sistema de atitude, isso porque ele foi projetado considerando as falhas do atuador, o atraso de controle, as incertezas do modelo dinâmico variante no tempo e os distúrbios externos.

Por sua vez, em [Fialho et al. \(2020\)](#) é proposto um sistema para o controle de atitude durante o pouso vertical utilizando o controlador PID, que é utilizado para compensar a dinâmica de diversos tipos de sistemas. Esse controlador tem esse nome por conta de seus três ganhos: proporcional (K_p), integral (K_i) e derivativos (K_d), que são variados para obter a resposta desejada. Logo, o objetivo do projeto foi desenvolver três controladores PID (um para cada grau de liberdade), utilizando o software *Simulink*, que faz o controle de atitude na fase de reentrada de um foguete auxiliar. Foi utilizado o foguete *Electron* da empresa *Rocket Lab*. Para o desenvolvimento do sistema, foram criados dois tipos de modelos: o modelo completo e o modelo simplificado. A diferença entre esses modelos é a força aerodinâmica que não foi considerada no modelo simplificado. Como resultado, o sistema proposto conseguiu fazer o seu pouso utilizando 81 kg de combustível a partir da queda livre vertical.

3 DESENVOLVIMENTO

Como citado anteriormente, este trabalho tem como principal objetivo fazer uma simulação de pouso vertical de um foguete. Para que isto seja possível será elaborado uma simulação da etapa de reentrada de um foguete em que o controlador será responsável por calcular o empuxo necessário para se conseguir o controle de velocidade do pouso e mantenha o ângulo de ataque inferior à 1.5° .

O controlador utilizado neste projeto será o PD pelo seu desempenho satisfatório em sistemas de controle linear e invariante no tempo. Para gerar os valores dos coeficientes do controlador PD, será desenvolvido um algoritmo genético responsável por determinar os valores de K_p e K_d do controlador, e outro algoritmo genético atuando nas raquetes do foguete para o controle do ângulo de ataque.

Para desenvolver as simulações desta pesquisa, os códigos serão gerados em C pelo software *Visual Studio Code* e o *Matlab 2014a* é utilizado para gerar os gráficos.

3.1 Condições iniciais considerados para os cálculos.

Na Tabela 1 são apresentados os valores utilizados para a simulação da etapa de reentrada, em que o valor da área do bocal do motor é o mesmo do motor Merlin e o valor de queima do combustível é referente ao propelente utilizado na Falcon 9, sendo um propelente do tipo híbrido, pela sua facilidade de controle da queima, com oxigênio líquido (LOX) e querosene refinado (RP-1).

Tabela 1 – Valores iniciais.

Massa do combustível	4.000 Kg
Área do bocal	10,75 m ²
Gravidade	9,8 m/s ²
Velocidade de escape dos gases	1,2 m/s
Pressão atmosférica	101.325 kPa
Tempo de queima do combustível	200 s
Raio terrestre	6.370 Km
Ângulo inicial	60°

3.2 Cálculo da trajetória

Pela segunda Lei de Newton é determinado que a força resultante que age sobre um corpo em um determinado tempo deve ser igual ao produto da massa pela aceleração, representada pela Equação 3.1:

$$F(t) = m * a(t) \quad (3.1)$$

Logo, é possível determinar os valores de velocidade e aceleração do foguete por meio das forças atuantes que são a força de empuxo, arrasto e peso. Na Figura 9 estão apresentadas as forças e suas posições em relação a um eixo inercial.

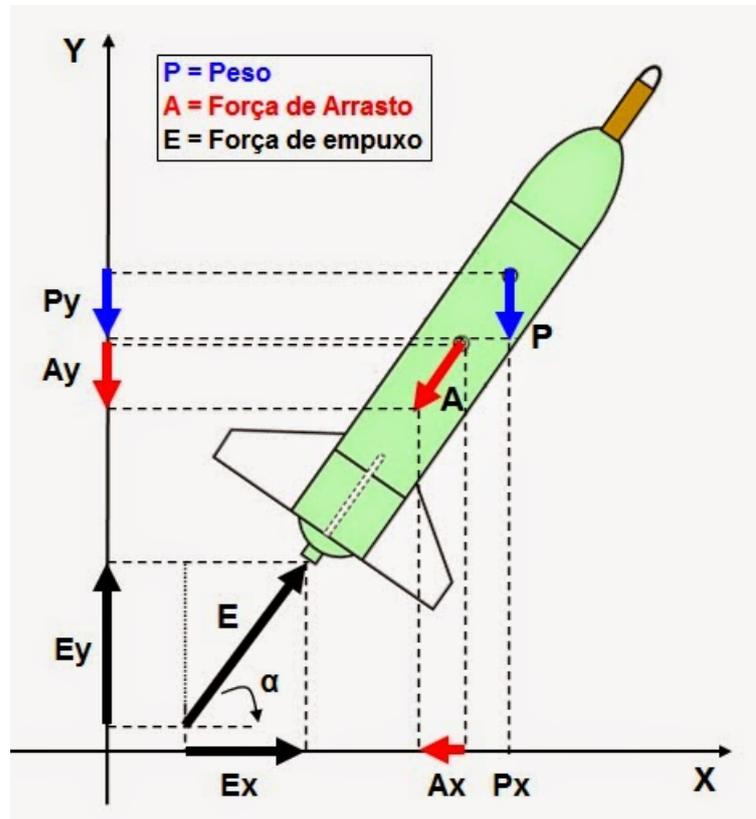


Figura 9 – Forças atuando no foguete em relação a um eixo: (YAMAMOTO, 2022)

As Equações 3.2 e 3.3 apresentam, respectivamente, os somatórios das forças que agem sobre o foguete no eixo X e Y.

$$F_x = F_{emp} * \cos(\alpha) - F_{nor} * \sin(\alpha) + F_{arr} * \cos(\alpha) \quad (3.2)$$

$$F_y = F_{emp} * \sin(\alpha) - P - F_{arr} * \sin(\alpha) \quad (3.3)$$

Onde o ângulo α é o ângulo de ataque do foguete, F_{emp} é o empuxo e F_{arr} a Força de arrasto. Igualando a Equação 3.1 com o somatório das forças, é possível obter a aceleração do foguete, sendo representada pela equação abaixo:

$$A_x(t) = \frac{F_x(t)}{m} \quad (3.4)$$

$$A_y(t) = \frac{F_y(t)}{m} \quad (3.5)$$

Logo, é possível concluir que a velocidade instantânea do foguete será:

$$Vx(t) = V_0 + Ax(t).t \quad (3.6)$$

$$Vy(t) = V_0 + Ay(t).t \quad (3.7)$$

Utilizando os valores da velocidade determinada nas Equações 3.4 e 3.5, é possível determinar o movimento em x e y pela fórmula da função da posição no movimento uniforme, representada pelas equações abaixo:

$$x(t) = X_o + Vx(t).t \quad (3.8)$$

$$y(t) = Y_o + Vy(t).t \quad (3.9)$$

3.3 Sistema de controle

O sistema de controle é feito em uma malha fechada onde um controlador recebe o valor de entrada, gera os coeficientes de Kp e Kd. A diferença de um malha aberta para uma malha fechada é que a na malha fechada é possível obter um valor de erro que será subtraído na realimentação do sistema (OGATA, 2010). Esse esquema pode ser representado pela Figura 10:

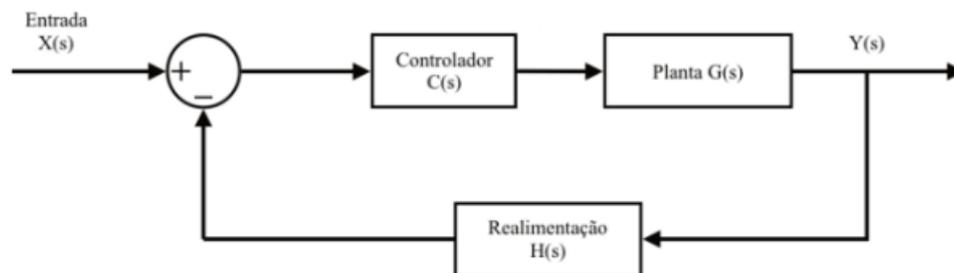


Figura 10 – Diagrama de blocos de uma malha fechada. Fonte: (OGATA, 2010) - Adaptado pela autora.

A Equação 3.10 representa a equação da malha fechada:

$$\frac{Y(s)}{X(s)} = \frac{C(s)G(s)}{1 + H(s)C(s)G(s)} \quad (3.10)$$

Com o objetivo de que o sistema gere correções entre o valor de entrada e o sinal de saída, foi desenvolvido uma planta em malha fechada que irá receber como entrada o um valor de empuxo como referência. A fim de que o sistema seja capaz de fazer as alterações para que o empuxo permaneça estável (Y(s)), será implementado um controlador PD no qual o

ganho proporcional (K_p) corrigirá os desvios de atitude em cada instante de tempo e o ganho derivativo (K_d) seja proporcional à variação do erro em função do tempo (CAMPOS, 2005). A representação da Malha Fechada do sistema é mostrada na Figura 11.

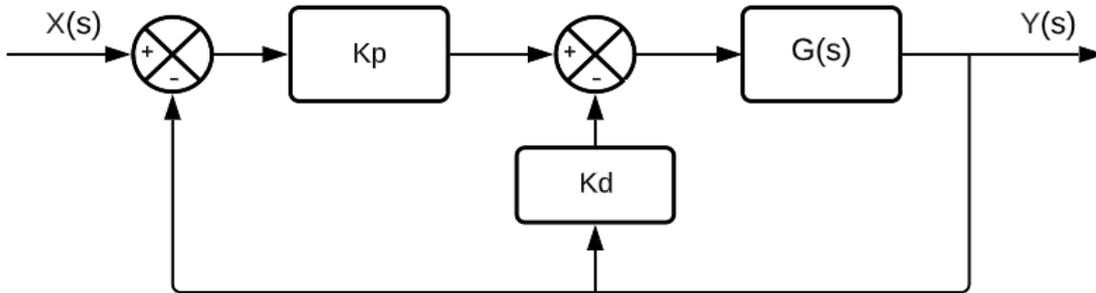


Figura 11 – Malha fechada do sistema de controle. Fonte: Autoria própria

Onde a função de transferência $G(s)$ representa a transformada de Laplace do empuxo em função do tempo e as variáveis K_p e K_d representa, respectivamente, os ganhos proporcional e derivativo do controlador PD. O ganho derivativo foi aplicado na realimentação com o objetivo de obter um sistema de segunda ordem (FIALHO et al., 2020). A função de transferência proposta neste trabalho é representada pela Equação 3.11:

$$\frac{Y(s)}{X(s)} = \frac{(Femp * K_p)}{s^2 + (Femp * K_d)s + Femp * K_p} \quad (3.11)$$

Por ser um sistema variante no tempo, ou seja, a entrada do sistema altera de acordo com o tempo, foi utilizado um algoritmo genético para determinar os melhores valores dos coeficientes do controlador.

3.4 Algoritmo Genético

Como supracitado, o algoritmo genético será responsável por determinar os coeficientes do controlador PD e determinar os ângulos das raquetes do foguete para que o pouso seja considerado estável. O código desenvolvido para o algoritmo genético foi desenvolvido na linguagem C e está no apêndice deste trabalho. Na Figura 12 é possível analisar o fluxograma do algoritmo genético desenvolvido nesta pesquisa:

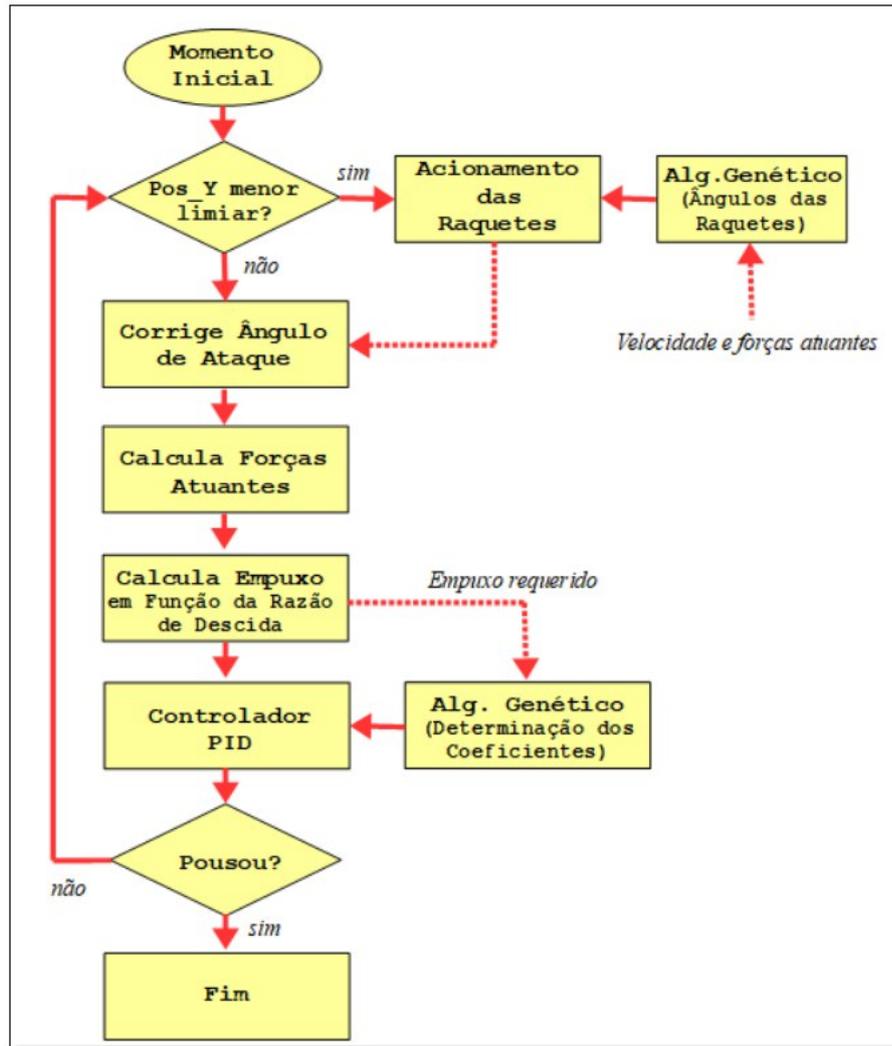


Figura 12 – Fluxograma Algoritmo Genético. Fonte: Autoria própria

O fluxograma é iniciado com os valores apresentados na Tabela 1. Todo o código irá rodar enquanto o foguete não estiver pousado, ou seja, quando o valor da posição de Y for maior que 0, irá dar início ao *looping* do código. Ao ser iniciado, as raquetes serão acionadas para começar a fazer o controle do ângulo de ataque do foguete. Com o novo valor do ângulo, gerado pelo AG, será refeito os cálculos das forças atuantes no foguete. O novo valor do Empuxo é enviado para o AG que será responsável para gerar os ganhos Kp e Kd necessário gerando um novo valor de empuxo.

4 RESULTADOS

4.1 Simulação da trajetória de referência

Para a simulação dos valores de entrada do sistema, foi utilizado o *software* MATLAB para analisar a trajetória do foguete. Logo, foi utilizado um laço para que, enquanto altura do foguete em relação ao solo fosse maior que zero, fosse feito o cálculo das forças que agem no foguete, sendo elas: aceleração, velocidade e sua posição atual. No gráfico representado na Figura 13 é possível analisar a trajetória do foguete com as considerações iniciais apresentadas na Seção 3.1. Para estes calculos iniciais, foi considerado que a queima do combustível é linear, resultando em uma trajetória quase linear.

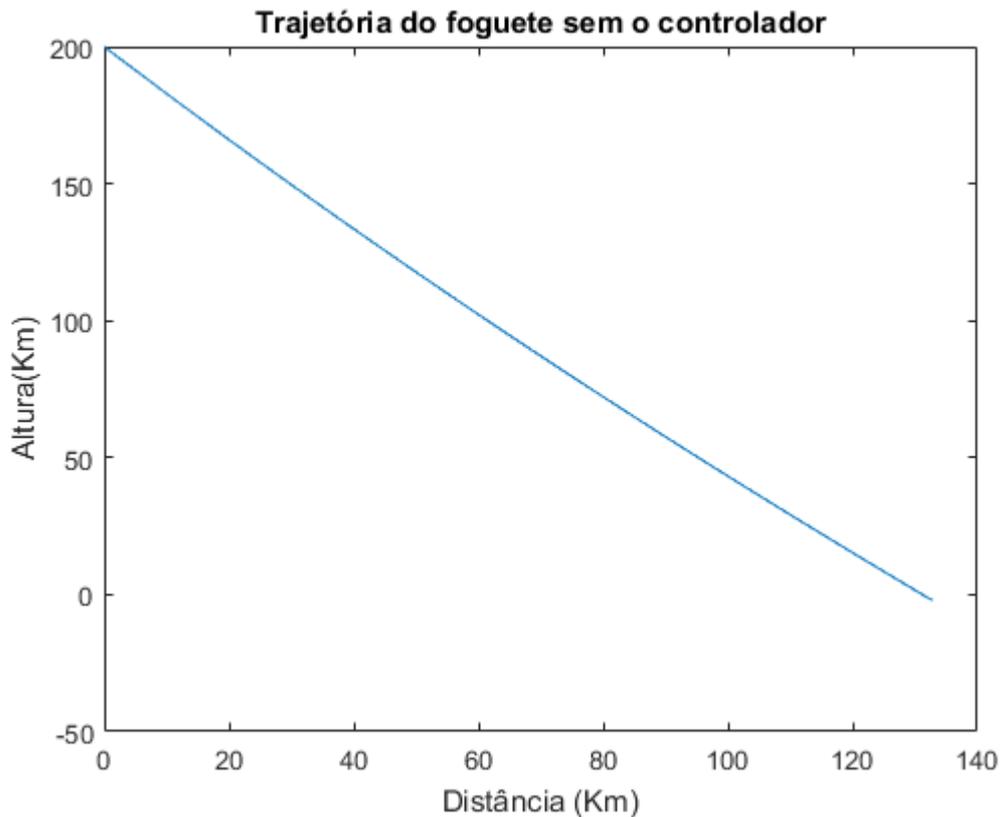


Figura 13 – Trajetória do foguete sem o controlador. Fonte: Autoria Própria.

Na Figura 14 é possível ver a relação da Força de Empuxo de acordo com a distância. Por não ter um controle de pouso e a queima do combustível ser linear, o valor do empuxo gerado durante a trajetória é linear também.

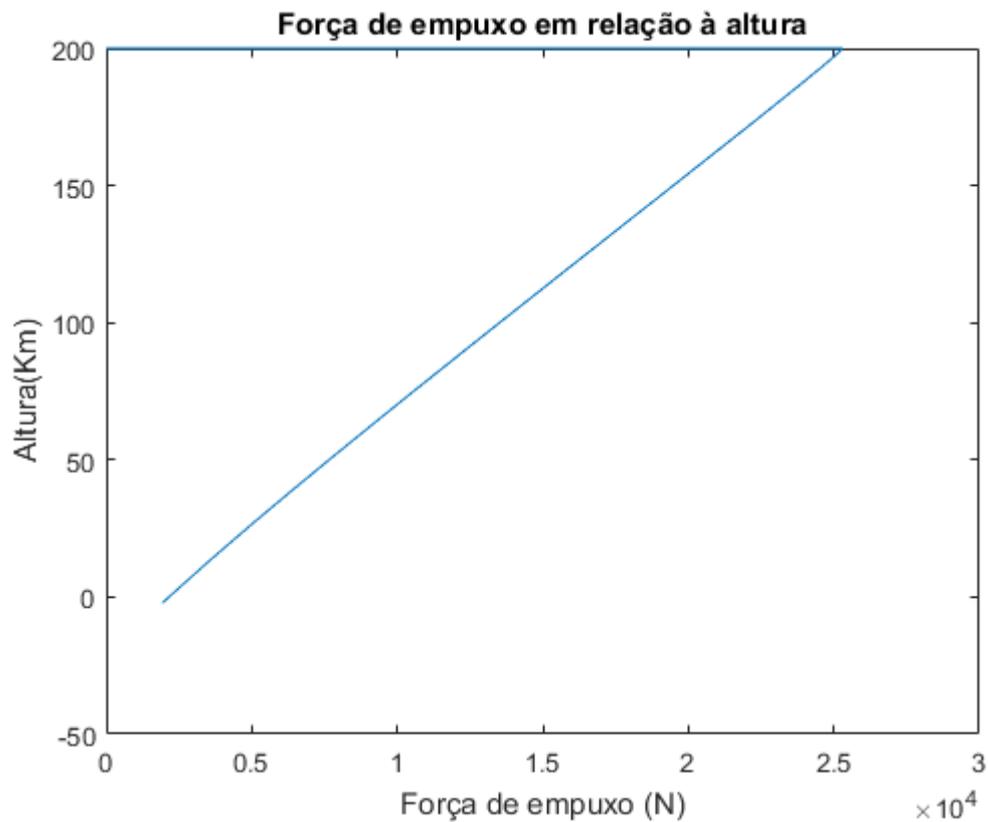


Figura 14 – Valores da Força Empuxo em relação à altura sem o controle. Fonte: Autoria Própria.

Os valores do ângulo do foguete durante a trajetória são representados pelo gráfico na Figura 15. Analisando este gráfico, é possível determinar que, sem o controlador, o foguete iria pousar com um ângulo entre 54° e 53°, não sendo considerado um valor para pouso vertical estável. Logo, se vê a necessidade de elaborar um sistema que seja capaz de fazer o controle do valor de Empuxo para que o pouso seja estável.

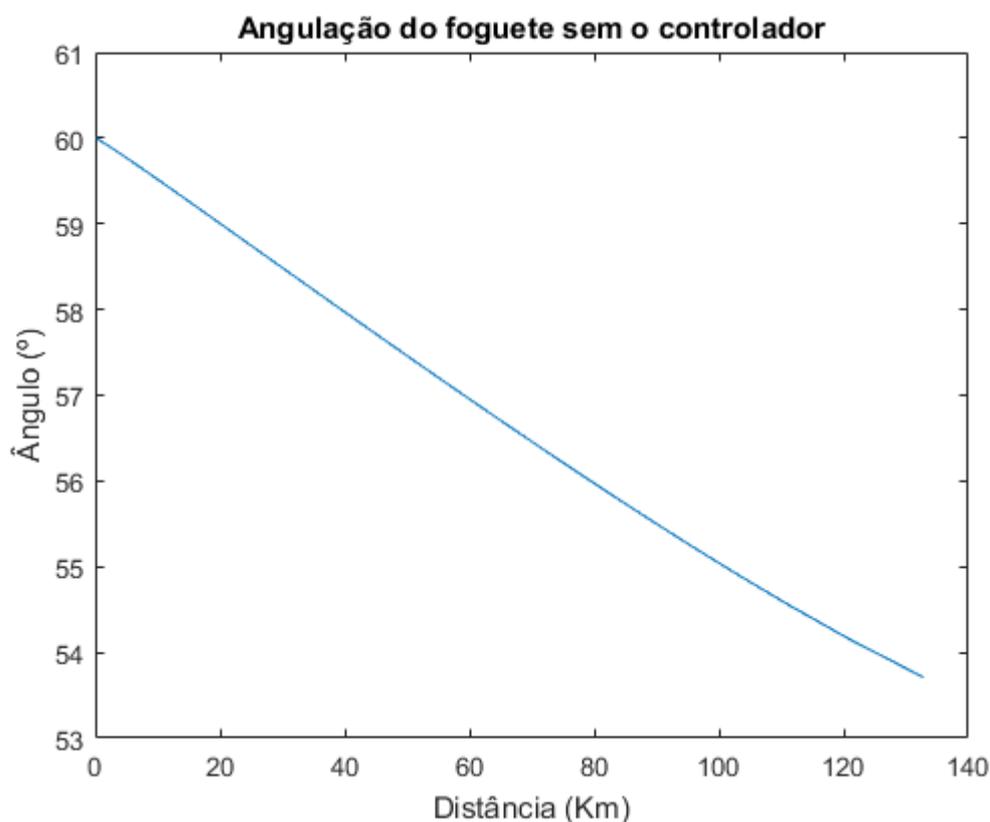


Figura 15 – Ângulo de ataque em função da distância sem o controlador. Fonte: Autoria Própria.

Na Figura 15, convém mencionar que a distância corresponde ao deslocamento horizontal sofrido pelo foguete até o seu momento do pouso conforme ilustrado na Figura 13. Nota-se que, sem a atuação do controlador, a angulação permaneceu demasiadamente alta para um pouso eficaz.

4.2 Controle dos ângulos das raquetes

Como representado na Seção 3.4, o Algoritmo Genético será responsável por determinar os ângulos necessários para que as raquetes conseguissem fazer a correção do ângulo de ataque mantendo em um valor inferior à $1,5^\circ$. Para os cálculos das raquetes, foi estipulando um tamanho de $0,5\text{m} \times 0,5\text{m}$ para as raquetes e que apenas teria a Força de Arrasto atuando contra elas.

Como pode ser notado na Figura 16, o foguete corrigiu a sua angulação, iniciada em 60° , no momento 45s após o início de sua reentrada na atmosfera. A partir deste momento, o foguete manteve a sua angulação próxima a 0° em relação ao eixo vertical.

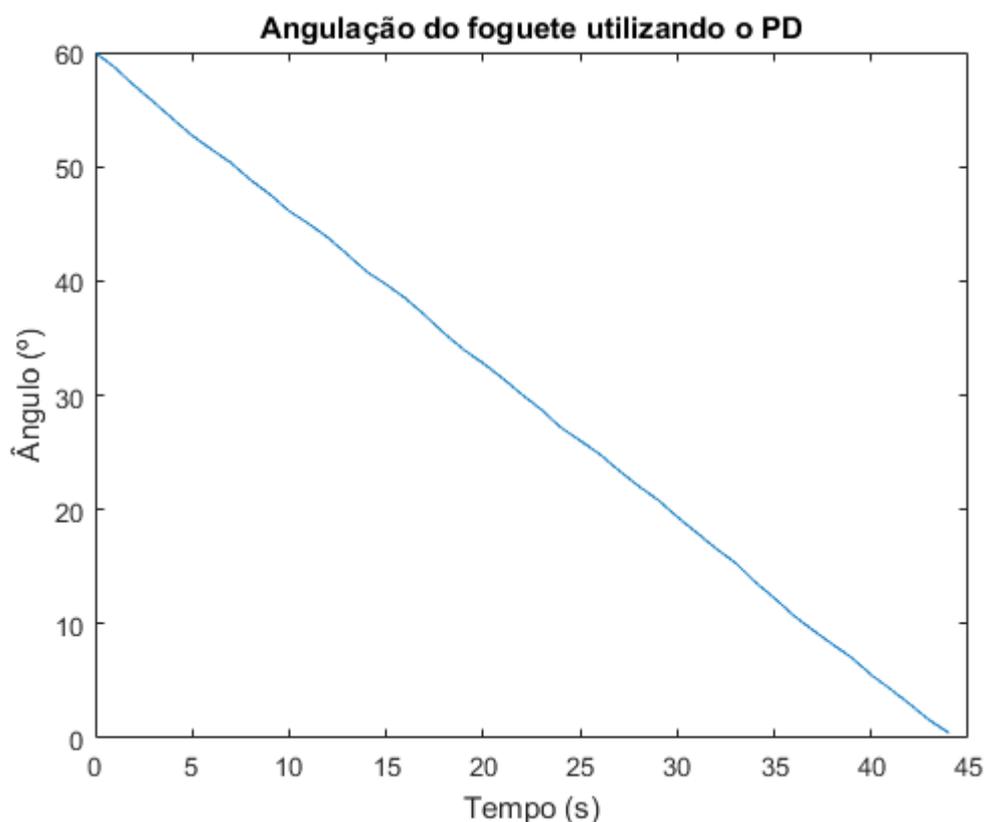


Figura 16 – Angulação do foguete, em relação ao tempo, utilizando o Controlador PD e algoritmo genético para a angulação das raquetes. Fonte: Autoria Própria.

4.3 Controle do Empuxo após o Algoritmo Genético

Para que o Algoritmo Genético desenvolvesse os melhores valores de K_p e K_d é necessário que ele receba os valores do Empuxo para gerar os coeficientes que, ao ser aplicado no controlador PD, fosse responsável por gerar uma saída mais próxima do valor de entrada referencial. É preciso salientar que o sistema de controle do pouso vertical proposto neste trabalho é dinâmico visto que os valores são diferentes no decorrer do tempo. Logo, foram elaborados 50 testes com 5 valores fixos de Empuxo para analisar como seria o desvio padrão dos valores gerados pelo AG. Para estes testes, foram utilizados Empuxos iguais a 100N, 500N, 1000N, 2000N e 20000N. Os resultados gerados pelos testes é apresentado nas Figuras 17 e 18.

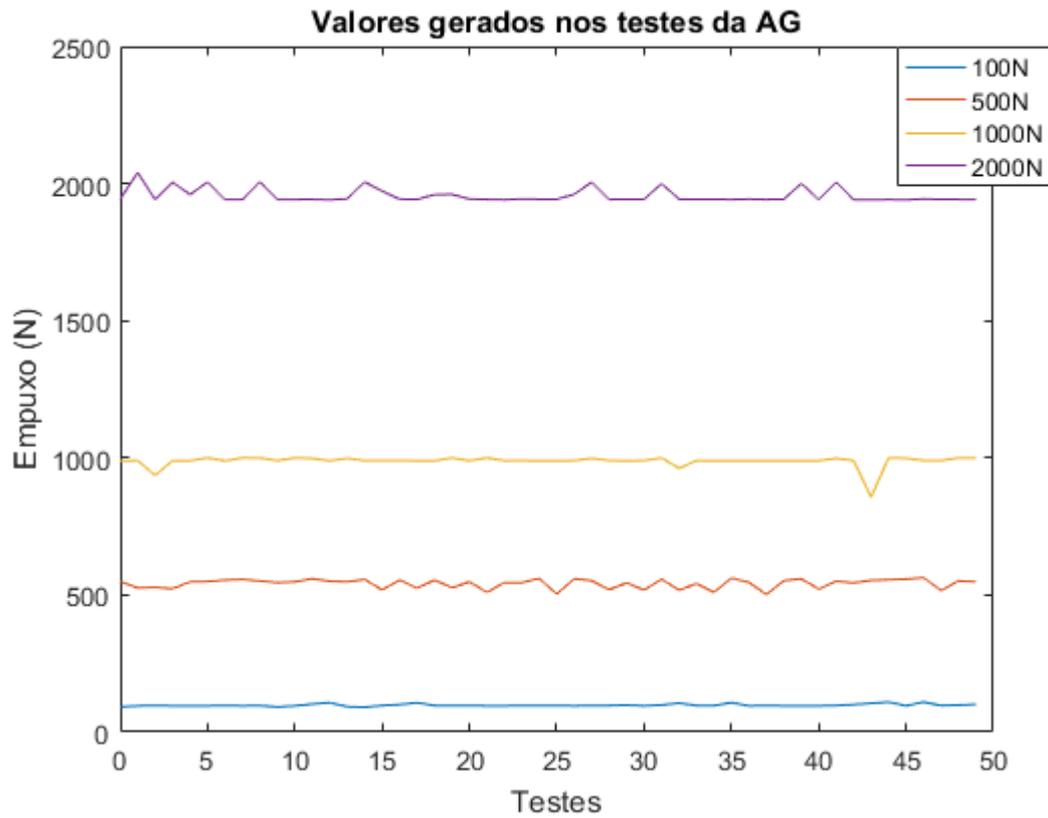


Figura 17 – Dispersão do Empuxo gerado pelo AG durante os testes. Fonte: Autoria Própria.

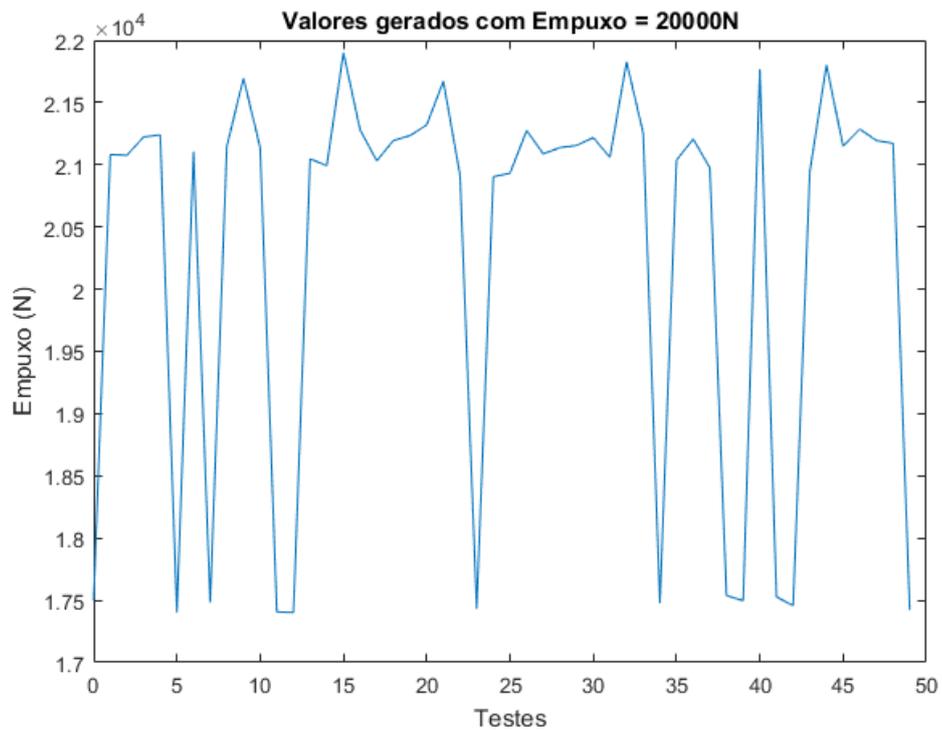


Figura 18 – Dispersão do Empuxo igual a 20000N gerado pelo AG durante os testes. Fonte: Autoria Própria.

O desvio padrão dos valores gerados pelo AG é representado na Tabela 2. Como o valor de empuxo vai diminuindo de acordo com o tempo, é possível concluir que o AG se mostra eficiente para a geração dos coeficientes do controlador PD.

Tabela 2 – Valores da dispersão do Empuxo gerado pela AG e o Empuxo teste.

Empuxo (N)	Desvio padrão
100	4,36
500	17,22
1000	21,51
2000	25,36
20000	$1,6414 \cdot 10^3$

A variação da altura do foguete após a aplicação do sistema de controle é demonstrando na Figura 19. É possível analisar que o ângulo é corrigido em um período de 45s, resultando em uma trajetória perpendicular ao eixo x.

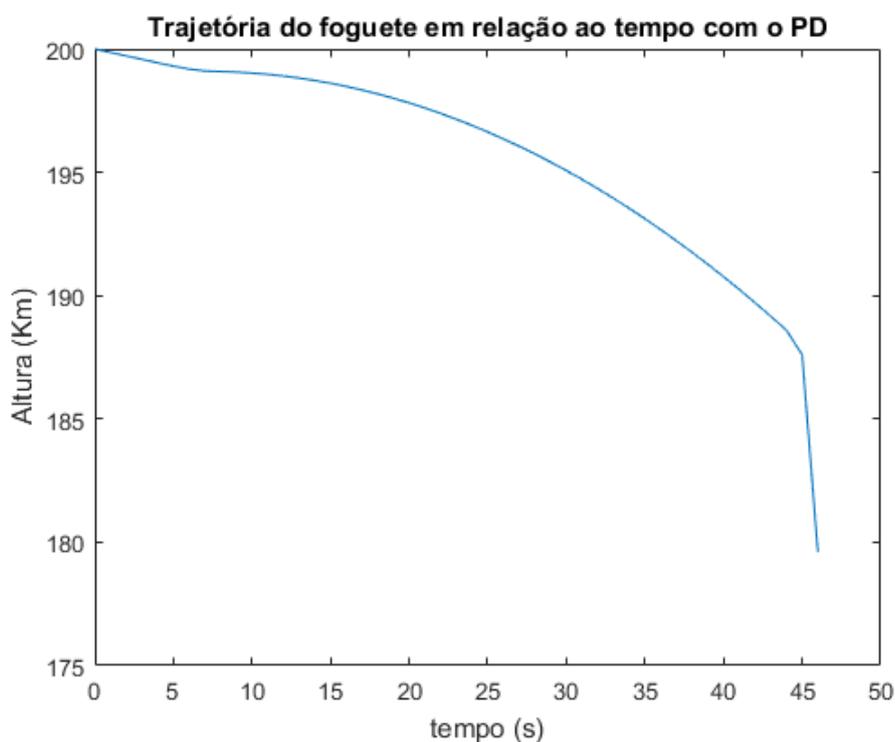


Figura 19 – Valores da altura em relação ao tempo após a correção do PD. Fonte: Autoria Própria.

Na Figura 20 é possível analisar que a velocidade tem um aumento significativo em função da ação gravitacional e, depois, a diminuição da mesma devido ao controle do Empuxo do motor.

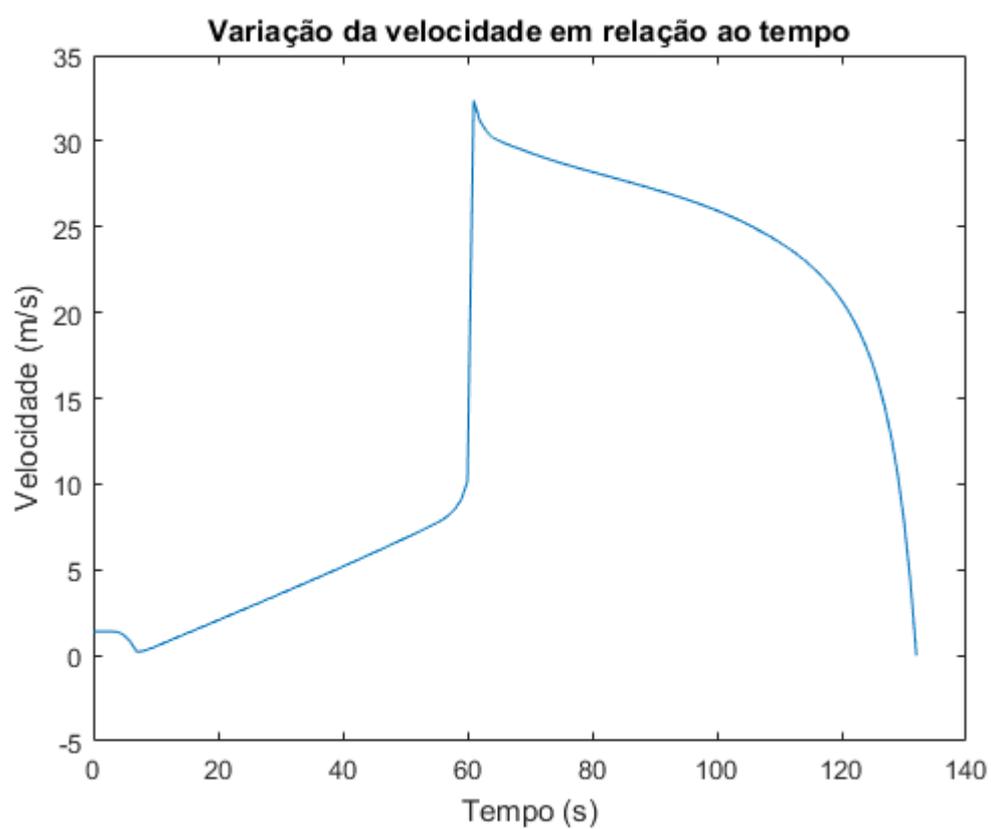


Figura 20 – Valores de velocidade em relação ao tempo após a correção do PD. Fonte: Autoria Própria.

5 CONCLUSÃO

O objetivo deste estudo foi desenvolver uma simulação de pouso vertical de um foguete em que o sistema de controle é formado por um controlador PD e o algoritmo genético que determina os valores dos coeficientes do PD e os ângulos das raquetes para que o foguete pouse com um ângulo inferior a 1.5° .

O sistema proposto neste estudo foi capaz de fazer o controle de atitude de um foguete na etapa de reentrada utilizando Algoritmos Genéticos para determinar os valores de K_p e K_d do controlador PD e a angulação necessária das raquetas para corrigir a angulação de ataque do foguete. Como resultado, o ângulo de ataque do foguete teve seu valor de estabilidade respeitada em até 45s após o início da etapa de reentrada do veículo na atmosfera.

Como trabalhos futuros sugere-se a realização de estudos com outros tipos de controladores a configuração uma trajetória limite. Para a realização dos resultados, não foi considerado os valores referente ao Momento do foguete para simplificação e isso também pode ser considerado em trabalhos futuros. A queima do combustível também pode ser levada em conta nos cálculos.

REFERÊNCIAS

- AFEITOS, L. D. d. *Controlo ótimo de trajetórias explícitas de reentrada atmosférica*. Tese (Doutorado) — Universidade da Beira Interior, 2014. Citado 2 vezes nas páginas 17 e 18.
- ALVES, A. L. et al. Minifoguete a propelente sólido: aspectos teóricos e propostas experimentais para o ensino de física. *Revista Brasileira de Ensino de Física*, 2020. SciELO Brasil, v. 42, 2020. Citado 2 vezes nas páginas 9 e 21.
- AN, B. et al. Adaptive terminal sliding mode control for reentry vehicle based on nonlinear disturbance observer. *IEEE Access*, 2019. IEEE, v. 7, p. 154502–154514, 2019. Citado na página 20.
- BEZERRA, J. H. et al. Estudo de sistema de controle usando o controlador pid em processos industriais. 2007. [sn], 2007. Citado na página 25.
- CAMPOS, D. C. de. Estudo de um método para cálculo de ganhos da malha de controle de atitude de um lançador de satélites. *CAMPOS*, 2005. v. 629, p. 062–2, 2005. Citado na página 32.
- CARVALHO, J. P. dos S.; LIMA, J. dos S.; GONÇALVES, C. M. Poluição do ambiente espacial: O problema do lixo no espaço. *Scientia: Revista Científica Multidisciplinar*, 2021. v. 6, n. 2, p. 61–80, 2021. Citado na página 14.
- FIALHO, Y. L. et al. Sistema de controle para pouso vertical autônomo de foguetes auxiliares. 2020. Joinville, SC, 2020. Citado 2 vezes nas páginas 28 e 32.
- GAO, M.; YAO, J. Finite-time H_∞ adaptive attitude fault-tolerant control for reentry vehicle involving control delay. *Aerospace Science and Technology*, 2018. Elsevier, v. 79, p. 246–254, 2018. Citado na página 28.
- LEAL, F. H. Estudo numérico de cargas aerodinâmicas em foguete. 2017. 2017. Citado 2 vezes nas páginas 9 e 23.
- MAO, Q. et al. Reentry attitude control for a reusable launch vehicle with aeroservoelastic model using type-2 adaptive fuzzy sliding mode control. *International Journal of Robust and Nonlinear Control*, 2018. Wiley Online Library, v. 28, n. 18, p. 5858–5875, 2018. Citado na página 27.
- MARS, K. *Space Shuttle System*. 2021. Publicado em 14/02/2021. Acessado em 06/10/2021. Disponível em: <<https://www.nasa.gov/feature/40-years-ago-space-shuttle-columbia-returns-home>>. Citado na página 19.
- MENEGALDO, C. G. *Simulador de controle de atitude e propagação de órbita aplicado a nanossatélites em órbita baixa terrestre: desenvolvimento e validação com dados de voo do nanossatélite PicSat*. Tese (Doutorado) — Universidade de São Paulo, 2020. Citado 2 vezes nas páginas 9 e 16.
- OGATA, K. *Engenharia de Controle Moderno*. [S.l.]: Pearson, 2010. Citado 3 vezes nas páginas 9, 25 e 31.

OLIVEIRA, L. H. dos S. Desenvolvimento de um motor propelente híbrido para testes experimentais. 2016. 2016. Citado na página 24.

PACHECO, M. A. C. et al. Algoritmos genéticos: princípios e aplicações. *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, 1999. v. 28, 1999. Citado na página 27.

PINHO, A. F. de et al. Algoritmos genéticos: Fundamentos e aplicações. *Meta-Heurísticas em Pesquisa Operacional, edited by HS Lopes, LC de Abreu Rodrigues and MTA Steiner*, 2013. p. 21–32, 2013. Citado na página 26.

PORTO, M. C. F. Sistema de controle de atitude para foguete de pequeno porte. 2017. 2017. Citado na página 17.

PREADO; ALMEIDA, A. F. B. de. *A Conquista do Espaço do Sputnik a Missão Centenário*. [S.l.]: Editora Livraria da Física, 2007. Citado na página 18.

SAMPAIO, R. P. Sistema de controle de atitude embarcado para voo autônomo de aviões em escala. 2006. Escola Politécnica/Instituto de Matemática, 2006. Citado na página 24.

SPACEEX. *Landing*. 2006. Disponível em: <<https://www.spacex.com/mission/>>. Citado 3 vezes nas páginas 9, 19 e 20.

TAVARES, Y.; NEDJAH, N.; MOURELLE, L. de M. Utilização de otimização por enxame de partículas e algoritmos genéticos em rastreamento de padrões. In: *12 Congresso Brasileiro de Inteligência Computacional*. [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 9 e 26.

TOMAYKO, J. E. Computers in spaceflight: The nasa experience. 1988. 1988. Citado na página 19.

YAMAMOTO, T. K. *Foguete de Garrafa Pet*. 2022. Disponível em: <<http://maquetesdicas.blogspot.com/2014/07/foguete-de-garrafa-pet.html>>. Citado 2 vezes nas páginas 9 e 30.

ZANARDI, M. Fundamentos da astronáutica. *Apostila do ITA*, 1990. 1990. Citado 2 vezes nas páginas 23 e 24.

ZANATTA, R. Modelagem, simulação e controle de um vetor de lançamento orbital. 2019. 2019. Citado na página 24.

ZOU, Q. et al. Robust adaptive constrained backstepping flight controller design for re-entry reusable launch vehicle under input constraint. *Advances in Mechanical Engineering*, 2015. SAGE Publications Sage UK: London, England, v. 7, n. 9, p. 1687814015606308, 2015. Citado na página 27.

APÊNDICE A – CÓDIGO TRAJETÓRIA

% ————— DADOS INICIAIS

```

Delta = 0.1;                % step tempo
t= zeros(1, interacoes); %tempo
Femp = zeros(1, interacoes);
Fart = zeros(1, interacoes);
Fnor= zeros(1, interacoes);
Fnor(1) = 0;
mt = zeros(1, interacoes);
mt(1) = 4500; %massa tot foguete kg
mcom = 4500; %massa do combustivel kg
Ab = 10.75; %area do ve culo
g = 9.8; %gravidade
Ve = 1.2; %velocidade de escape do g s
patm = 101.325; %kPa
bt = 200; %tempo de queima
rt = 63.7814; %valor m dio para o raio terrestre
pesotot = zeros(1, interacoes);
pesotot (1) = 44100;
ang = zeros(1, interacoes); %angulo inicial
Fx = zeros(1, interacoes);
Ax = zeros(1, interacoes);
Vx = zeros(1, interacoes);
x = zeros(1, interacoes);
Fy = zeros(1, interacoes);
Ay = zeros(1, interacoes);
Vy = zeros(1, interacoes);
y = zeros(1, interacoes);
p = zeros(1, interacoes);
p(1) = 0;
xm = zeros(1, interacoes);
p2 = zeros(1, interacoes);
Fx(1) = 0;
Ax(1) = 0;
Vx(1) = 0;
x(1) = 0;

```

```

Fy(1) = 0;
Ay(1) = 0;
Vy(1) = 0;
ang(1) = 60;           %angulo inicial
y(1) = 200;           %posicao inicial
t(1) = 0;
C = 0.4;               % coeficiente Drag
Rho = 1.2;             % densidade do ar (kg/m^3)
tempfinal= zeros(1, interacoes);
kp = zeros(1, interacoes);
ki = zeros(1, interacoes);
i=1;
PD = zeros(1, interacoes);
% ----- POSICOES
n=1;
num = zeros(1, interacoes);
den = zeros(1, interacoes);
sys = zeros(1, interacoes);
tempfinal = [];
while y(n) > 0 %Laco enquanto o foguete nao atinge o ch o .
    n = n+1;
    t(n)= (n-1)*Delta;
    tempfinal = t(n-1)+t(n);
    %ARRASTO:
    den = 1.2;
    Fart(n) =0.5*C*Rho*Ab*(Vx(n-1)^2+Vy(n-1)^2);
    %EMPUXO:
    p(n) = patm + den.*g.*y(n-1);
    p2(n) = (p(n)-patm)*Ab;
    xm(n) = ((t(n)*mcom)/bt);
    Femp(n) = ((xm(n).*Ve)+p2(n));
    mt(n) = mcom-xm(n);
    %NORMAL:
    g2(n) = g*((rt/(rt+y(n-1)))^2); %Gravidade de acordo com a altura
    pesocomb(n) = mt(n).*g2(n);
    pesotot(n) = pesotot(n-1)-pesocomb(n);
    Fnor(n) = pesotot(n).*g2(n).*cosd(ang(n-1));

    %FORCAS SOBRE O FOGUETE:

```

```

    % Forca em x
    Fx(n) = (Femp(n)*cosd(ang(n-1))) - (Fnor(n)*sind(ang(n-1))) + (Fart
    Ax(n) = Fx(n)/mt(n);           % Aceleracao em x
    Vx(n) = Vx(n-1)+Ax(n)*Delta;   % Velocidade em x
    x(n) = x(n-1)+(Vx(n)*Delta);    % Posicao x
    % Forca em y
    Fy(n) = (Femp(n)*sind(ang(n-1)))-(pesotot(n).*g2(n))+(Fnor(n)*cosd(
    Ay(n) = Fy(n)/mt(n);           % Aceleracao em y
    Vy(n) = Vy(n-1)+Ay(n)*Delta;   % Velocidade em y
    y(n)= y(n-1)-(Vy(n)*Delta);    % Posicao y
    ang(n) = atand(Vy(n)/Vx(n));

```

```
end
```

APÊNDICE B – CÓDIGO DO ALGORITMO GENÉTICO - ÂNGULO DAS RAQUETES

```

#include <time.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

typedef struct
{
    float c[2], //c[1]=esq c[0]=dir
        cor_ang;
} TRaq;

#define MUTA 50
#define COMBINA 50
#define NFILHOSMAX 100
#define SELEC 20
#define PRECISAOANG 0.01
#define MAXINTERACOES 30

TRaq fraq[NFILHOSMAX];
float Vy[200], ang[200], pesotot[200];
int n=0;

void CombinarFilhosRaq()
{
    float tmp;
    int pos1, pos2, campo;
    for (int i=0; i<COMBINA; i++)
    {
        do{
            pos1 = rand()%SELEC;

```

```

        pos2 = rand()%SELEC;
    } while (pos1 == pos2);
    campo = rand()%2;
    tmp = fraq[pos1].c[campo];
    fraq[pos1].c[campo] = fraq[pos2].c[campo];
    fraq[pos2].c[campo] = tmp;
} // final for ...
for (int i=COMBINA; i<NFILHOSMAX; i++)
{
    do{
        pos1 = rand()%SELEC;
        pos2 = rand()%SELEC;
    } while (pos1 == pos2);
    fraq[i].c[0] = fraq[pos1].c[0];
    fraq[i].c[1] = fraq[pos2].c[1];
} // final for ...
} // final CombinarFilhosRaq

void MutacoesRaq()
{
    int pos1, campo, sinal, valor;
    for (int i=0; i<MUTA; i++)
        fraq[rand()%NFILHOSMAX].c[rand()%2]+=((float)(rand()%10)*0.01234)*
} // final MutacoesRaq

float CalculaAnguloVertical(float angE, float angD)
{
    //supondo raquete tamanho 0.5mx0.5m
    float arrasto1, arrasto2, aceler1, aceler2, V1, V2;
    arrasto1 = (1.05*0.5* sin(angE)*0.5*Vy[n]*Vy[n])/2;
    arrasto2 = (1.05*0.5* sin(angD)*0.5*Vy[n]*Vy[n])/2;
    aceler1 = arrasto1 / pesotot[n];
    aceler2 = arrasto2 / pesotot[n];
    V1 = Vy[n] - fabs(aceler1);
    V2 = Vy[n] - fabs(aceler2);
    arrasto1=(asin((3.1415*(V1 - V2)/180)));
    return arrasto1;
} // final CalculaAnguloVertical

```

```

//retorno o indice do individuo mais especializado para ser a configuracao
//o pai da proxima geracao
float FuncaoFitnessRaq()
{
    int i=0,j;
    TRaq tmp;
    for(; i<NFILHOSMAX-1; i++)
    {
        fraq[i].cor_ang = CalculaAnguloVertical(fraq[i].c[1], fraq[i].c[0]);
        // printf("cor_ang: %f\n", fraq[i].cor_ang);
    }
    for(i=0; i<NFILHOSMAX-1; i++)
        for(j=i+1; j<NFILHOSMAX; j++)
            if((ang[n]+frac[i].cor_ang) > (ang[n]+frac[j].cor_ang))
                {
                    memcpy(&tmp, &frac[i], sizeof(TRaq));
                    memcpy(&frac[i], &frac[j], sizeof(TRaq));
                    memcpy(&frac[j], &tmp, sizeof(TRaq));
                } // final if
    // printf("retornarah %f\n", fraq[0].cor_ang);
    return fraq[0].cor_ang;
} // final FuncaoFitnessRaq

//Gerar os individuos de cada geracao a partir do individuo atual
void PopulacaoInicialRaq()
{
    for(int nfilhos = 0; nfilhos < NFILHOSMAX; nfilhos++)
    {
        fraq[nfilhos].c[0] = ((float)(rand()%10)*0.0873474)*pow(-1, rand()%2);
        fraq[nfilhos].c[1] = ((float)(rand()%10)*0.0526351)*pow(-1, rand()%2);
    } // final for
} // final PopulacaoInicialRaq

float GeneticoRaq()
{
    float ret;

```

```

int ninteracoes = 0;
PopulacaoInicialRaq ();
for (; ninteracoes < MAXINTERACOES; ninteracoes++)
{
    ret = FuncaoFitnessRaq (); // retorna o melhor individuo para aquele
    if (ang[n] < PRECISAOANG) return ret;
    CombinarFilhosRaq (); // na combinacao, escolher alguns individuos
    MutacoesRaq (); // para a mutacao, escolher alguns individuos
}
return ret; // mudar para angulo resultante
} // final GeneticoRaq

```

```

int main()
{
    float ret;
    srand(time(0));
    ang[0] = 1.047;
    printf("1.047,0,0\n");
    Vy[0] = 4;
    for (int nteste = 0; nteste < 132 ; nteste++)
    {
        pesotot[n] = 1;
        ret = GeneticoRaq ();
        ang[n] += ret;
        if (ang[n] < 0) break;
        printf("%f,%f,%f\n", ang[n], fraq[0].c[1], fraq[0].c[0]);
        // Vy[0]--;
    }
    return 0;
}

```

APÊNDICE C – CÓDIGO DO ALGORITMO GENÉTICO - COEFICIENTES KP E KD

```
#include <time.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

typedef struct
{
    float c[2],    //c[0]=kd; c[1]=kp;
    pid;
} TPid;

#define MUTA 50
#define COMBINA 50
#define NFILHOSMAX 100
#define SELEC 20
#define PRECISAOPID 10
#define MAXINTERACOES 3000

TPid fpid[NFILHOSMAX];

void CombinarFilhosPID()
{
    float tmp;
    int pos1, pos2, pos3, campo;
    for (int i=0; i<COMBINA; i++)
    {
        do{
            pos1 = rand()%SELEC;
            pos2 = rand()%SELEC;
        } while(pos1 == pos2);
    }
}
```

```

        campo = rand()%2;
        tmp = fpid[pos1].c[campo];
        fpid[pos1].c[campo] = fpid[pos2].c[campo];
        fpid[pos2].c[campo] = tmp;
    } // final for ...
for (int i=COMBINA; i<NFILHOSMAX; i++)
    {
        do{
            pos1 = rand()%SELEC;
            pos2 = rand()%SELEC;
        }while(pos1 == pos2);
        fpid[i].c[0] = fpid[pos1].c[0];
        fpid[1].c[1] = fpid[pos2].c[1];
    } // final for ...
} // final CombinarFilhosPID

void MutacoesPID ()
{
    int pos1 ,campo , sinal , valor ;
    for (int i=0; i<MUTA; i++)
        {
            campo = rand()%2;
            pos1 = rand()%NFILHOSMAX;
            if(campo) fpid[pos1].c[1]+=((float)(rand()%10)*1.14535)*pow(-1,rand()%2);
            else
                {
                    do
                        {
                            fpid[pos1].c[0]+(((float)(rand()%10)*0.095235)*pow(-1,rand()%2));
                        }while(fpid[pos1].c[0]==0);
                }
        }
} // final MutacoesPID

//retorno o indice do individuo mais especializado para ser a configura
//o pai da proxima gera o

```

```

float FuncaoFitnessPID(float empuxo,int t)
{
    int i,j;
    double x1,x2;
    TPid tmp;          //c[0] = kd;c[1] = kp
    for(i=0; i<NFILHOSMAX-1; i++)
        fpid[i].pid=(empuxo*fpid[i].c[1])/(t*t+fpid[i].c[0]*empuxo*t+fpid[i].c[0]*empuxo)
    for(i=0;i<NFILHOSMAX-1;i++)
        for(j=i+1;j<NFILHOSMAX;j++)
            if((fabs(fpid[i].pid - empuxo)) > (fabs(fpid[j].pid-empuxo)))
                {
                    memcpy(&tmp,&fpid[i], sizeof(TPid));
                    memcpy(&fpid[i],&fpid[j], sizeof(TPid));
                    memcpy(&fpid[j],&tmp, sizeof(TPid));
                } // final if
    return fpid[0].pid;
} // final FuncaoFitness

```

```

//Gerar os indiv duos de cada gera o a partir do indiv duo atual
void PopulacaoInicialPID()
{
    for(int nfilhos = 0; nfilhos < NFILHOSMAX; nfilhos++)
        {
            fpid[nfilhos].c[0] = -((float)(rand()%10)*20.23342);
            fpid[nfilhos].c[1] = ((float)(rand()%10)*82.1234);
        }
} // final PopulacaoInicialPID

```

```

//Gerar popula o de descendentes a partir de um indiv duo selecionado
void PreencheFilhos()
{
    int i=1;
    for(;i<NFILHOSMAX;i++) memcpy(&fpid[i],&fpid[0], sizeof(TPid));
} // final PreencheFilhos

```

```

// Sequencia execucao algoritmo genético
int GeneticoPID(float empuxo,int t)
{
    float ret;
    int ninteracoes = 0;
    PopulacaoInicialPID();
    for(; ninteracoes < MAXINTERACOES; ninteracoes++)
        {
            ret = FuncaoFitnessPID(empuxo,t); //retorna o melhor individuo p
            if((fabs(ret - empuxo) < PRECISAOPID) || (ninteracoes == MAXINTERACOES
            PreencheFilhos();
            MutacoesPID();
            CombinarFilhosPID();
        }
    return ret; //mudar para angulo resultante
} //final GeneticoPID

```

```

// funcao que representa o controlador
void ControladorPID(float empuxo,int t)
{
    int ret;
    ret = GeneticoPID(empuxo,t);
} //final ControladorPID

```

```

int main()
{
    srand(time(0));
    FILE *arq_agpid;
    float emp=10; //empuxo para teste do algoritmo
    int tempo=20; //tempo para teste do algoritmo
    if((arq_agpid = fopen("agpid.csv","w"))==NULL)
        {
            printf("Erro abertura arquivo.\n");
            return 0;
        }
    for(int nteste = 0; nteste < 50 ; nteste++)
        {

```

```
    if(nteste) fprintf(arq_agpid, "\n");
    ControladorPID(emp, tempo);
    printf("%f\t%f\t%f\t%f\n", emp, fpid[0].pid, fpid[0].c[1], fpid[0].c[2]);
    fprintf(arq_agpid, "%f,%f,%d,%f,%f,%f", emp, fpid[0].pid, tempo, emp-f, fpid[0].c[1], fpid[0].c[2]);
}
fclose(arq_agpid);
return 0;
}
```