

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

GUILHERME ANGELO PIAIA

**DESENVOLVIMENTO DE UM CONTROLADOR PID
COM AUTO SINTONIA USANDO REDES NEURAIAS
ARTIFICIAIS E REGRESSÃO NÃO-LINEAR ROBUSTA
PARA O CONTROLE DE ATITUDE DE UM SIMULADOR
DE SATÉLITES COM RODAS DE REAÇÃO**

São Leopoldo, RS

2018

GUILHERME ANGELO PIAIA

**DESENVOLVIMENTO DE UM CONTROLADOR PID
COM AUTO SINTONIA USANDO REDES NEURAIAS
ARTIFICIAIS E REGRESSÃO NÃO-LINEAR ROBUSTA
PARA O CONTROLE DE ATITUDE DE UM SIMULADOR
DE SATÉLITES COM RODAS DE REAÇÃO**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação pelo curso de Engenharia de Controle e Automação da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador:

Prof. Dr. Rodrigo Marques de Figueiredo

São Leopoldo, RS

2018

Este trabalho é dedicado aos meus pais que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

“Study hard what interests you the most in the most undisciplined, irreverent and original manner possible.”
(Richard Feynman)

RESUMO

As técnicas de controle e sintonia de controladores foram se desenvolvendo e incorporando ferramentas desenvolvidas com o tempo. No presente panorama, o uso de algoritmos que simulam o comportamento de organismo e métodos de otimização estão ganhando espaço, disputando com métodos clássicos em aplicações complexas e que exigem eficiência, elevada controlabilidade, como é o caso do controle de atitude de satélites. O presente trabalho apresenta um estudo e uma proposta de técnica de sintonia de controladores PID, com o objetivo de controlar a dinâmica de um satélite. Portanto, esse trabalho propõe o desenvolvimento de um protótipo de satélite com rodas de reação e de um método automático de sintonia de controladores PID usando redes neurais artificiais e regressão não-linear robusta, afim de melhorar a dinâmica do simulador de satélites. Por fim, são apresentados os resultados e análises do protótipo e do método automático de sintonia.

Palavras-chaves: Controladores PID, Redes Neurais Artificiais. Satélite. Controle de Atitude. Regressão Não-Linear Robusta. Rodas de Reação.

LISTA DE FIGURAS

Figura 1 – Movimento de translação de um satélite artificial.	17
Figura 2 – Representação de \vec{r}_{com}	19
Figura 3 – Distribuição de massa.	20
Figura 4 – Representação de um corpo rígido com rodas de reação.	24
Figura 5 – Parâmetros de uma resposta ao degrau de um sistema de segunda ordem ou superior.	26
Figura 6 – Representação do modelo paralelo de um controlador PID com distúrbios.	27
Figura 7 – Modelo de um controlador PID com histerese.	27
Figura 8 – Conversão de um sinal contínuo em discreto.	29
Figura 9 – Diagrama de blocos do filtro de Kalman.	30
Figura 10 – Parâmetros para a sintonia de controladores PID a partir da resposta ao degrau.	31
Figura 11 – Fluxograma para a escolha do método de sintonia.	33
Figura 12 – Modelo do método de auto sintonia via relé.	33
Figura 13 – Célula de um neurônio natural e modelo matemático respectivo.	35
Figura 14 – Representação de uma rede neural.	35
Figura 15 – Controlador PID com rede neural.	36
Figura 16 – Comparação entre os métodos de regressão não-lineares.	37
Figura 17 – Desenho mecânico completo do simulador de satélite.	38
Figura 18 – Corpo do simulador de satélites.	39
Figura 19 – Desenho mecânico do mancal a ar.	40
Figura 20 – Geometria básica para o projeto de rodas de reação.	41
Figura 21 – Desenho mecânico do conjunto motor-roda.	42
Figura 22 – Base do simulador de satélites prototipado.	43
Figura 23 – Corpo do simulador de satélites prototipado.	44
Figura 24 – Simulador de satélites prototipado.	45
Figura 25 – Modelo do motor DC.	46
Figura 26 – Diagrama de blocos em malha fechada.	48
Figura 27 – Função de transferência dos motores.	48
Figura 28 – Modelo em malha fechada com um controlador PID.	49
Figura 29 – Diagrama de blocos do controlador PID com Anti-Windup.	49
Figura 30 – Representação geral do sistema.	50
Figura 31 – Fluxograma do <i>software</i> de controle.	51
Figura 32 – Características dos sinais amostrados pelo giroscópio.	52
Figura 33 – Diagrama de blocos do sintonizador e otimizador.	53
Figura 34 – Resultados do método do relé.	55
Figura 35 – Saída da RNA treinada e o sinal real.	56

Figura 36 – Respostas ao degrau com sintonia via método do relé.	57
Figura 37 – Velocidade angular instantânea e sinal de controle.	58

LISTA DE CÓDIGOS

Código 1 – Minimização da função do controlador PID	54
---	----

LISTA DE TABELAS

Tabela 1 – Parâmetros PID pelo método de Ziegler-Nichols - resposta ao degrau	32
Tabela 2 – Parâmetros PID pelo método de Ziegler-Nichols - resposta em frequência .	32
Tabela 3 – Dimensões das rodas de reação.	42
Tabela 4 – Correção do <i>bias</i> do giroscópio.	51
Tabela 5 – Parâmetros da RNA.	54
Tabela 6 – Parâmetros dos Controladores	58

LISTA DE ABREVIATURAS E SIGLAS

AD	<i>Analógico-Digital</i>
DC	<i>Direct Current</i> (Corrente contínua)
ESC	<i>Electronic-Speed-Control</i> (Controlador eletrônico de velocidade)
GMV	<i>General Minimum Variance</i> (Variância Mínima Geral)
GPIO	<i>General Purpose Input/Output</i> (Entradas e Saídas de uso Geral)
I2C	<i>Inter-Integrated Circuit</i> (Comunicação Entre circuitos Integrados)
MMQ	<i>Método dos Mínimos Quadrados</i>
PI	<i>Proportionl and Integral</i> (Proporcional e Integral)
PID	<i>Proportionl, Integral and Derivative</i> (Proporcional, Integral e Derivativo)
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulso)
RPM	<i>Rotações por Minuto</i>
RMSE	<i>Root Mean Square Error</i> (Erro quadrático médio)
RNA	<i>Rede Neural Artificial</i>
Rpi	<i>Raspberry Pi Zero W</i>
SSH	<i>Secure Shell</i> (Shell Seguro)
UNISINOS	<i>Universidade do Vale do Rio dos Sinos</i>
URSS	<i>União das Repúblicas Socialistas Soviéticas</i>
ZOH	<i>Zero-Order-Hold</i> (Amostrador de Ordem Zero)

LISTA DE SÍMBOLOS

F	Força
m	Massa
t	Tempo
dt	Diferencial de tempo
\vec{p}	Vetor momento linear
\vec{v}	Vetor velocidade
\vec{a}	Vetor aceleração
\vec{F}_i	Força total sobre uma partícula
\vec{f}_{ie}	Força externa aplicada em uma partícula
\vec{f}_{ij}	Força interna aplicada por cada partícula j
m_i	Massa de uma partícula i
\vec{a}_i	Aceleração de uma partícula i
\vec{F}_e	Somatório das forças sobre um corpo
\vec{F}_{ie}	Somatório de todas as forças sobre as partículas
\vec{r}_{com}	Vetor posição
M_T	Massa total de um corpo rígido
x, y, z	Coordenadas cartesianas
$I_{xx}, IA, I_{yy}, IB, I_{zz}, IC$	Momentos de inércia de um corpo rígido simétricos
$\omega, \vec{\omega}$	Velocidade angular e vetor velocidade angular
$\dot{\vec{\omega}}$	Primeira derivada do vetor velocidade angular
ω_0	Velocidade angular inicial
α	Aceleração angular
$\vec{\tau}$	Vetor torque
\vec{L}	Vetor momento angular

I	Inércia
r_{sat}	Distância dos motores em relação ao centro de massa do satélite
r_e	Raio total da roda de reação
r_d	Raio do disco da roda de reação
h_r	Espessura da borda da roda de reação
h_d	Espessura do disco da roda de reação
I_{sat}	Inércia do satélite
ψ, θ, ϕ	Posição angular em relação aos eixos cartesianos
θ_e	Valores desconhecidos de uma função não-linear
\vec{L}_s	Momento angular do satélite
\vec{L}_{total}	Soma do momento angular do satélite e das rodas de reação
\vec{L}_{roda}	Momento angular de uma roda de reação
$\vec{\omega}_{roda}$	Vetor velocidade angular de uma roda de reação
\vec{L}_ω	Momento angular da roda de reação
J_ω, J_{ω_2}	Momento de inércia da roda de reação
$\vec{\psi}_\omega$	Variação instantânea da posição angular da roda de reação
τ_ω	Torque do conjunto motor/roda de reação
τ_1, τ_2, τ_3	Torque do conjunto motor/roda de reação, um em cada eixo
J_{ω_1}	Momento de inércia do rotor
J	Momento de inércia do rotor somado com a da roda de reação
$\omega_1, \omega_2, \omega_3$	Velocidade angular em relação aos eixos x, y e z, respectivamente
β	Posição
β_0	Posição inicial
t_0	Tempo inicial
M_p	Sobressinal
t_0	Tempo de atraso de transporte

t_d	Tempo de atraso de transporte
t_r	Tempo até sinal de referência
t_p	Tempo até o sobressinal
t_s	Tempo até o regime estacionário
e, err	Erro
T_i	Tempo integral
T_d	Tempo derivativo
K, K_p	Ganho Proporcional
P	Variável de processo
K_i	Ganho Integral
K_d	Ganho Derivativo
τ_m	Torque do motor
β_{com}	Posição de referência
G_e	Função de transferência do controlador PID
D	Distúrbio
G_p	Função de transferência da planta
ω_{sp}	Velocidade angular de referência
y	Saída do sistema
u	Sinal computado; Sinal do controlador
T	Período
z	Variável do plano z
s	Variável do plano s
\hat{y}	Sinal na saída de um ZOH; Saída estimada; Saída predita
y^*	Trem de pulsos
\hat{y}	Saída predita
K_u	Ganho que leva o sistema à marginalidade

d	Amplitude de oscilação do relé
ε	Histerese do relé
T_u	É o período de oscilação
H	Ganho do filtro de Kalman
M_A	Matriz de transição de estados
M_B	Matriz de entrada
M_C	Matriz de saída
λ_n	Ganho do neurônio
f_{rn}	Função excitação do neurônio
rn	Saída de um neurônio
p_n	Enésimo dendrito
a	Sinal de saída do neurônio; Amplitude de oscilação do sinal de saída
R_a	Resistência de armadura
L_a	Indutância de armadura
i_a	Corrente de armadura
e_a	Tensão de armadura
e_b	Tensão induzida
K_w	Constante de velocidade contra-eletromotriz
K_t	Constante de torque
B	Atrito viscoso
V_a	Tensão da fonte

SUMÁRIO

1	INTRODUÇÃO	16
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Satélites Artificiais	17
2.1.1	Dinâmica de um Satélite	17
2.1.1.1	<i>Rodas de Reação</i>	24
2.2	Controladores PID	25
2.2.1	Resposta ao Degrau do um Sistema	25
2.2.2	Controlador PID	26
2.2.3	Controle Digital	28
2.2.3.1	<i>Transformada Z</i>	28
2.2.3.2	<i>Amostragem</i>	28
2.2.3.3	<i>Controlador PID Digital</i>	29
2.2.3.4	<i>Filtro de Kalman</i>	30
2.3	Sintonia de Controladores	31
2.3.1	Método de sintonia de Ziegler-Nichols	31
2.3.2	Sintonia Automática de Controladores	32
2.3.2.1	<i>Método do Relé</i>	33
2.4	Estado da Arte - Controladores Inteligentes	34
2.4.1	Controle com Redes Neurais Artificiais	34
2.4.1.1	<i>Regressão Não-Linear Robusta</i>	36
3	METODOLOGIA	38
3.1	Hardware	38
3.1.1	Corpo do Simulador de Satélite	39
3.1.2	Mancal a Ar	40
3.1.3	Motor DC e Rodas de Reação	40
3.1.3.1	<i>Projeto das Rodas de Reação</i>	40
3.1.3.2	<i>Escolha dos Motores DC</i>	42
3.1.4	Elementos Eletrônicos	43
3.2	Sistemas de Controle	45
3.2.1	Modelo Motor DC	45
3.2.2	Modelo Completo do Satélite	47
3.3	Software	50
3.3.1	Implementação do Controlador PID e Métodos de Sintonia	50
3.3.2	Análise dos Sinais e Filtragem	51

3.3.3	Métodos de Sintonia	52
3.3.3.1	<i>Método do Relé</i>	53
3.3.3.2	<i>Método via RNA e Regressão Não-Linear</i>	53
4	ANÁLISE DOS RESULTADOS	55
4.0.1	Métodos Clássicos de Sintonia	55
4.0.2	Método de Sintonia Automático usando RNA e Regrão Não-Linear Robusta	56
4.0.3	Respostas ao degrau	57
5	CONCLUSÃO	60
	REFERÊNCIAS BIBLIOGRÁFICAS	61

1 INTRODUÇÃO

O controle de atitude de satélites é estudado desde o final dos anos 50, quando os primeiros satélites artificiais foram colocados em órbita. O controle acurado da posição é necessário para manter o alinhamento com o planeta ou alvo. Isso se dá através de dispositivos que pela variação do momento angular, realizam movimentos de rotação no corpo do satélite. Para solucionar esse problema, diversas técnicas de controle e sintonia já foram empregadas no decorrer das últimas décadas, dentre elas, o controlador PID (Proporcional-Integral-Derivativo) que é usado para controlar a velocidade angular dos motores que são utilizados para o movimento de rotação do corpo do satélite.

Os controladores PID precisam de sintonia, ou seja, dependendo da dinâmica do sistema em que eles estão inseridos, ou ainda, alguma mudança que ocorra no sistema, o controlador precisa ser reajustado para que atenda as especificações previamente estabelecidas pelo projetista. Como os satélites estão inacessíveis e em um ambiente hostil, técnicas automáticas e adaptativas são irrecusáveis para garantir o perfeito funcionamento e a vida útil do satélite. Algumas técnicas de sintonia automática já foram desenvolvidas, algumas usam princípios clássicos de controle e tabelas para especificações engessadas, outras, modernas, fazem o uso de inteligência artificial para buscar os parâmetros ótimos.

Este trabalho tem por principal objetivo a implementação em um sistema embarcado de um controlador com sintonia automática utilizando conceitos de controle inteligente. Essa implementação será embarcada em um dispositivo com sistema operacional Linux e fará o controle de atitude de um simulador de satélites, que é a planta do sistema de controle. Com a utilização de diferentes técnicas de controle, uma comparação entre as diferentes técnicas será feita, com o objetivo de avaliar de forma quantitativa e qualitativa a implementação do controle inteligente.

O trabalho está organizado em capítulos, onde o primeiro é a presente introdução. O segundo é o desenvolvimento teórico, que aborda os principais conceitos dos elementos básicos para o desenvolvimento do trabalho. Já o terceiro, é a metodologia, onde os materiais, desenhos mecânicos e todos os passos para o desenvolvimento estão descritos. No quarto, temos a análise dos resultados, onde os principais resultados são descritos. E por último, a conclusão, onde estão reunidas as conclusões e trabalhos futuros.

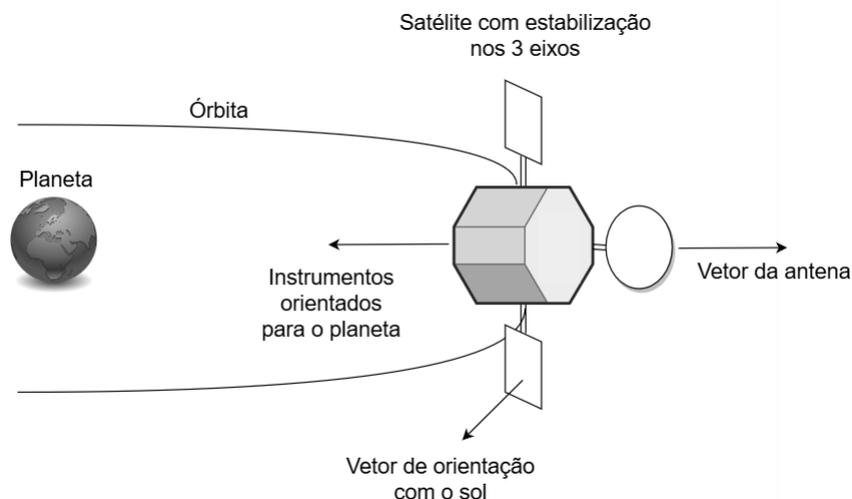
2 REVISÃO BIBLIOGRÁFICA

No presente capítulo, são apresentados conceitos básicos para a compreensão do trabalho. Nessa etapa, busca-se através da revisão de trabalhos já elucidados a contextualização do projeto proposto.

2.1 Satélites Artificiais

Desde o lançamento do satélite Sputnik I, primeiro satélite artificial, que foi desenvolvido e lançado pela extinta URSS (União das Repúblicas Socialistas Soviéticas) no dia 4 de Outubro de 1957, novas formas de controle foram desenvolvidas. Como o principal fim de um satélite é a comunicação e a navegação, que exigem orientação acurada, juntamente com todas as forças que atuam sobre o satélite, se torna imprescindível o uso de técnicas de controle para se garantir a translação e a rotação do satélite (BROWN, 2002). Na figura 1, podemos ver os movimentos de rotação e translação de um satélite ao redor de um planeta.

Figura 1 – Movimento de translação de um satélite artificial.



Fonte – Adaptado de Brown (2002).

A dinâmica de rotação é descrita no próximo item, juntamente com todas as ferramentas matemáticas para sua modelagem.

2.1.1 Dinâmica de um Satélite

A dinâmica é dividida em duas partes, onde uma representa os movimentos de translação e a outra os de rotação. A *dinâmica de translação* de um corpo em relação ao outro é um caso

de *Dinâmica de Corpo Rígido*. A fundação da Dinâmica de corpos rígidos foi feita pelo Físico Inglês Isaac Newton em forma de três leis, das quais, a 2ª e a 3ª são primordiais para descrever a *dinâmica de rotação* (SNIDER, 2010). A 2ª lei de Newton pode ser escrita das seguintes forma:

$$\vec{F} = \frac{d}{dt}\vec{p} \quad \vec{p} = m\vec{v} \quad (1)$$

$$\vec{F} = m\vec{a} \quad (2)$$

Onde \vec{F} é a força, \vec{p} é o momento linear, \vec{v} é a velocidade e \vec{a} é aceleração. Ainda, podemos descrever a força total sobre uma partícula \vec{F}_i da seguinte forma:

$$\vec{F}_i = \vec{f}_{ie} + \sum_{i \neq j}^N \vec{f}_{ij} = m_i \vec{a}_i \quad (3)$$

Onde, \vec{f}_{ie} é a força externa e \vec{f}_{ij} é a força interna de cada parte j que compõe o corpo i e $m_i \vec{a}_i$ é a massa e a força resultante no corpo rígido. Ao unirmos N partículas para formar um corpo rígido, estamos somando todas as forças externas e as forças internas de todos os corpos da seguinte forma:

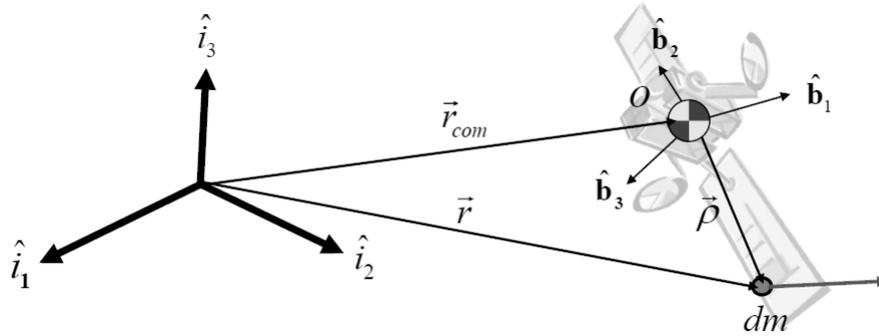
$$\sum_{i=1}^N \vec{F}_i = \sum_{i=1}^N \vec{f}_{ie} + \sum_{i=1}^N \sum_{i \neq j}^N \vec{f}_{ij} = \sum_{i=1}^N m_i \vec{a}_i \quad (4)$$

Segundo a 3ª Lei de Newton, como o corpo j exerce uma força \vec{f}_{ji} em módulo igual a força \vec{f}_{ij} que o corpo i exerce sobre o corpo j, só que oposta. Logo, as forças internas se anulam. Quando temos um conjunto de partículas unidas, podemos descrever a força total sobre o corpo (\vec{F}_e) como o somatório de todas as forças sobre as partículas (\vec{F}_{ie}). Isso pode ser escrito da seguinte forma:

$$\vec{F}_e = \sum_{i=1}^N \vec{f}_{ie} = \sum_{i=1}^N m_i \vec{a}_i \quad (5)$$

A imagem 2, representa o vetor posição \vec{r}_{com} de um corpo rígido em relação a origem. A descrição matemática desse vetor pode ser visto na equação 6.

Figura 2 – Representação de \vec{r}_{com}



Fonte – Adaptado de Snider (2010).

$$\vec{r}_{com} = \frac{1}{M_T} \sum_{i=1}^N m_i \vec{r}_i \quad (6)$$

Onde, M_T é a massa total do corpo rígido.

Como a aceleração é a derivada segunda da posição, derivamos duas vezes a posição \vec{r}_{com} e multiplicamos pela massa total em ambos os lado da equação 6. Assim temos:

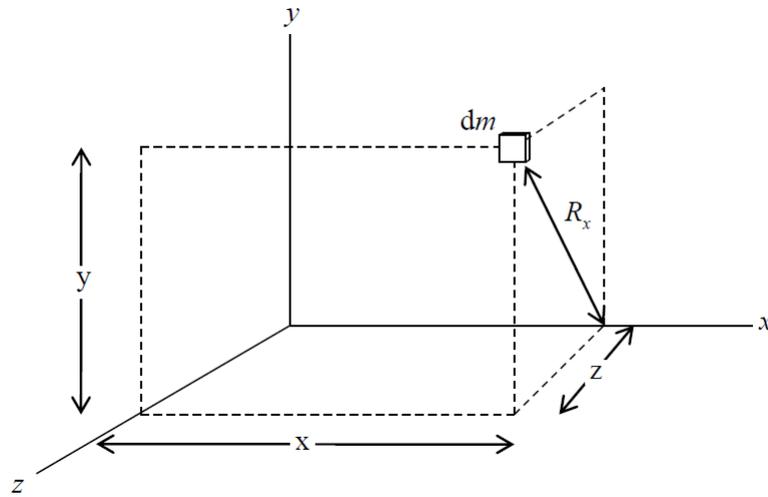
$$M_T \frac{d^2}{dt^2} \vec{r}_{com} = \sum_{i=1}^N m_i \vec{a}_i \quad (7)$$

Igualando as equações 5 e 7, temos a força total sobre um corpo rígido que desempenha movimento de translação. Isso pode ser visto na sequência.

$$\vec{F}_e = M_T \frac{d^2}{dt^2} \vec{r}_{com} \quad (8)$$

A *dinâmica de rotação* descreve o movimento de um corpo rígido em relação a um ponto. Para descrevermos os movimentos de rotação, devemos conhecer a distribuição de massa do corpo rígido (SNIDER, 2010). A figura 3 ilustra a distribuição de massa infinitesimal de um corpo tridimensional. Por definição, o momento de inércia infinitesimal disposto sobre um eixo (I_{xx} , I_{yy} e I_{zz}) é o produto da massa pela distância do ponto de referência. Se fizermos isso para N partículas, temos:

Figura 3 – Distribuição de massa.



Fonte – Adaptado de Snider (2010).

$$I_{xx} = \sum_{i=1}^N m_i (y_i^2 + z_i^2) \quad (9)$$

$$I_{yy} = \sum_{i=1}^N m_i (x_i^2 + z_i^2) \quad (10)$$

$$I_{zz} = \sum_{i=1}^N m_i (x_i^2 + y_i^2) \quad (11)$$

Já quando as partículas estão dispostas sobre os planos xy, xz e yz, temos:

$$I_{xy} = - \sum_{i=1}^N m_i x_i y_i = I_{yx} \quad (12)$$

$$I_{xz} = - \sum_{i=1}^N m_i x_i z_i = I_{zx} \quad (13)$$

$$I_{yz} = - \sum_{i=1}^N m_i y_i z_i = I_{zy} \quad (14)$$

Com isso, podemos criar uma matriz que representa o momento de inércia total do corpo rígido:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (15)$$

Caso o satélite seja simétrico, os momentos de inércia nos planos xy , xz e yz são de mesmo módulo só que com sinal oposto aos momentos de inércia dos planos xy , zx e zy , anulando todos esses termos. Isso pode ser visto na seguinte matriz diagonal:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} IA & 0 & 0 \\ 0 & IB & 0 \\ 0 & 0 & IC \end{bmatrix} \quad (16)$$

Para facilitar a escrita das equações, os momentos de inércia I_{xx} , I_{yy} e I_{zz} serão descritos como IA , IB e IC respectivamente. Após a descrição matemática do momento de inércia, podemos começar a analisar o movimento de rotação do corpo rígido. Vamos começar pela velocidade angular instantânea ω , que pode ser descrita como a derivada primeira da posição β angular, segundo o teorema fundamental do cálculo:

$$\omega \equiv \lim_{\Delta t \rightarrow 0} \frac{\Delta \beta}{\Delta t} = \frac{d\beta}{dt} \quad (17)$$

A aceleração angular α , nada mais é do que a derivada primeira da velocidade angular:

$$\alpha \equiv \lim_{\Delta t \rightarrow 0} \frac{\Delta \omega}{\Delta t} = \frac{d\omega}{dt} \quad (18)$$

Outra equação importante para descrever a maioria dos tipos de movimento, é aquela que relaciona a posição β com a posição inicial β_0 , velocidade angular inicial ω_0 , o tempo inicial do movimento t_0 , o tempo atual t e a aceleração angular α . A expressão pode ser vista na sequência.

$$\beta = \beta_0 + \omega_0(t - t_0) + \frac{1}{2}\alpha(t - t_0)^2 \quad (19)$$

Não menos importante, é a expressão que relaciona a velocidade angular ω com a velocidade angular inicial ω_0 , aceleração angular e o deslocamento ($\beta - \beta_0$). Isso é descrito na equação da sequência.

$$\omega^2 = \omega_0^2 + 2\alpha(\beta - \beta_0) \quad (20)$$

Agora, precisamos relacionar os momentos e a aceleração angular α . Para isso vamos começar relacionando o torque $\vec{\tau}$ com a derivada do momento angular \vec{L} :

$$\vec{\tau} = \dot{\vec{L}} \quad (21)$$

Mas como o momento angular é o produto do momento de inércia do corpo I pela velocidade angular, assim:

$$\vec{L} = I\vec{\omega} \quad (22)$$

Com isso, já conhecendo as relações entre velocidade e aceleração, temos:

$$\vec{\tau} = I\dot{\vec{\omega}} = I\vec{\alpha} \quad (23)$$

Se usarmos o teorema do transporte cinemático, o momento de inércia pode ser escrito da seguinte forma:

$$\dot{\vec{L}} = I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} \quad (24)$$

Por consequência, descrevemos o torque da seguinte maneira:

$$\vec{\tau} = I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} \quad (25)$$

Como temos o momento de inércia do corpo rígido representado na forma matricial, precisamos mudar a representação a matricial. Começaremos pela velocidade angular $\vec{\omega}$, que em módulo, pode ser descrita da seguinte forma:

$$|\vec{\omega}| = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_2 \end{bmatrix} \quad (26)$$

Se fizermos o mesmo com a equação 25, temos:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} IA\dot{\omega}_1 \\ IB\dot{\omega}_2 \\ IC\dot{\omega}_3 \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \times \begin{bmatrix} IA\omega_1 \\ IB\omega_2 \\ IC\omega_3 \end{bmatrix} \quad (27)$$

Como podemos ver, já conseguimos relacionar os momentos de inércia do corpo rígido simétrico com as demais variáveis de interesse. Além disso, podemos sair da forma matricial e apresentarmos os torques da forma algébrica, isso pode ser visto nas 3 equações a seguir:

$$\tau_1 = IA\dot{\omega}_1 - (IB - IC)\omega_2\omega_3 \quad (28)$$

$$\tau_2 = IB\dot{\omega}_2 - (IC - IA)\omega_1\omega_3 \quad (29)$$

$$\tau_3 = IC\dot{\omega}_3 - (IA - IB)\omega_1\omega_2 \quad (30)$$

Se utilizarmos um corpo não simétrico, podemos recorrer a matriz da equação 15, e obteremos as 3 seguintes equações para descrever o torque:

$$\tau_1 = I_{xx}\dot{\omega}_1 - I_{xy}(\dot{\omega}_2 - \omega_1\omega_3) - I_{xz}(\dot{\omega}_3 + \omega_1\omega_2) - (I_{yy} - I_{zz})\omega_2\omega_3 - I_{yz}(\omega_2^2 - \omega_3^2) \quad (31)$$

$$\tau_2 = I_{zz}\dot{\omega}_3 - I_{yz}(\dot{\omega}_3 - \omega_1\omega_2) - I_{xy}(\dot{\omega}_1 + \omega_2\omega_3) - (I_{zz} - I_{xx})\omega_1\omega_3 - I_{xz}(\omega_3^2 - \omega_1^2) \quad (32)$$

$$\tau_3 = I_{zz}\dot{\omega}_1 - I_{xz}(\dot{\omega}_1 - \omega_2\omega_3) - I_{yz}(\dot{\omega}_2 + \omega_1\omega_3) - (I_{xx} - I_{yy})\omega_1\omega_2 - I_{xy}(\omega_1^2 - \omega_2^2) \quad (33)$$

Se isolarmos a derivada da velocidade angular na nas equações 31, 32 e 33, temos:

$$\dot{\omega}_1 = \frac{\tau_1}{IA} + \left(\frac{IB - IC}{IA} \right) \omega_2\omega_3 \quad (34)$$

$$\dot{\omega}_2 = \frac{\tau_2}{IB} + \left(\frac{IC - IA}{IB} \right) \omega_1\omega_3 \quad (35)$$

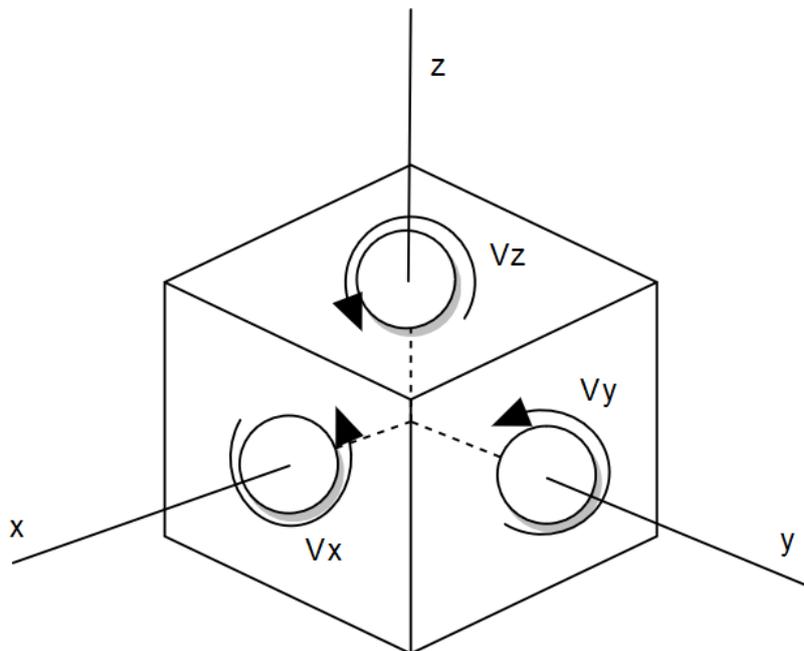
$$\dot{\omega}_3 = \frac{\tau_3}{IC} + \left(\frac{IA - IB}{IC} \right) \omega_1\omega_2 \quad (36)$$

Assim, conseguimos a relação da velocidade com o torque aplicado. Essa é uma expressão que representa a reação em forma de variação de velocidade que um torque provoca.

2.1.1.1 Rodas de Reação

Uma roda de reação consiste em um dispositivo que é fixado em um eixo de motor, com o objetivo de interferir na posição de um corpo através da variação de velocidade angular das rodas. Quando a roda sofre uma variação da sua velocidade angular, ela exerce um torque em uma determinada direção (Bong Wie, 2001). Esse efeito ocorre devido à conservação do momento angular do corpo, logo, se a roda está sendo desacelerada, o corpo tende se opor a esse movimento. Na figura 4 podemos ver 3 rodas de reação dispostas nos 3 eixos de um corpo rígido.

Figura 4 – Representação de um corpo rígido com rodas de reação.



Fonte – Adaptado de Levine (1996).

Como já foram descritos nas outras seções, precisamos encontrar uma relação que expresse o torque criado pelas rodas de reação, para que consigamos unir ao modelo do satélite. Para isso, começamos pelo momento angular total \vec{L}_{total} , que pode ser visto na sequência:

$$\vec{L}_{total} = \vec{L}_s + \vec{L}_\omega = constante \quad (37)$$

Onde L_s representa o momento angular do satélite e \vec{L}_ω o momento angular da roda de reação. Como já vimos na equação 22, o momento angular pode ser descrito da seguinte forma:

$$\vec{L}_s = I\vec{\omega} \quad (38)$$

E o momento angular nas rodas de reação pode ser escrito da seguinte forma:

$$\vec{L}_\omega = J_\omega \vec{\dot{\psi}}_\omega \quad (39)$$

Onde J_ω é o momento de inércia da roda de reação e $\vec{\dot{\psi}}_\omega$ é variação instantânea da posição angular da roda. Com isso, conseguimos encontrar o torque provido pela roda de reação τ_ω :

$$\vec{\tau}_\omega = \dot{\vec{L}}_\omega = J_\omega \ddot{\vec{\psi}}_\omega \quad (40)$$

E por fim, podemos relacionar as velocidades angulares do satélite com a inércia e as velocidades das rodas, isso se dá da seguinte forma:

$$\dot{\omega}_1 = \frac{J_{\omega 1} \ddot{\psi}_{\omega 1}}{IA} + \left(\frac{IB - IC}{IA} \right) \omega_2 \omega_3 \quad (41)$$

$$\dot{\omega}_2 = \frac{J_{\omega 2} \ddot{\psi}_{\omega 2}}{IB} + \left(\frac{IC - IA}{IB} \right) \omega_1 \omega_3 \quad (42)$$

$$\dot{\omega}_3 = \frac{J_{\omega 3} \ddot{\psi}_{\omega 3}}{IC} + \left(\frac{IA - IB}{IC} \right) \omega_1 \omega_2 \quad (43)$$

Para isso é usada a equação 37 e a relação entre torque e momento angular da equação 40. Assim, temos a última equação que faltava para descrever toda a dinâmica de um satélite artificial simétrico.

2.2 Controladores PID

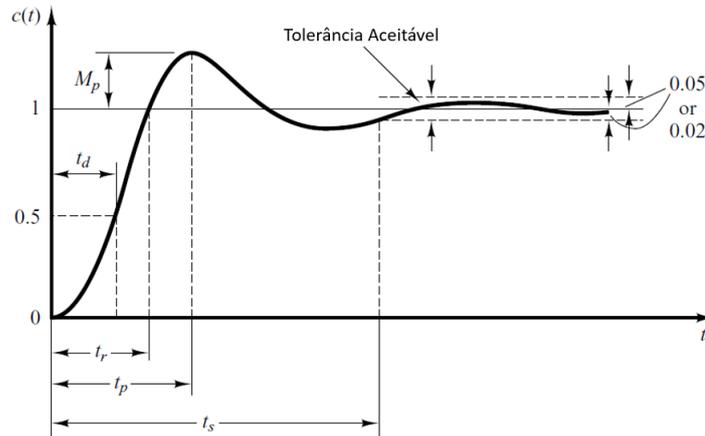
Como já temos as equações diferenciais que descrevem o modelo satélite, precisamos desenvolver os conceitos básicos de controle para controlarmos as variáveis de interesse. Esses conceitos estão descritos nas próximas seções

2.2.1 Resposta ao Degrau do um Sistema

Um problema fundamental em engenharia, é prever e modelar sistemas naturais ou artificiais para tirarmos o melhor proveito de suas características. Para isso, muitas técnicas foram desenvolvidas para se conseguir controlar essas variáveis de interesse (LEVINE, 1996). Uma forma clássica de representar uma resposta de uma variável de interesse, é através da resposta ao degrau. Essa pode ser vista na figura 5, onde temos as principais características da resposta ao degrau de um sistema de segunda ordem ou superior. Onde M_p (*maximum overshoot*) é o sobressinal da variável de interesse em percentual, t_d (*Delay Time*) é o tempo atraso de transporte,

t_r (rise time) é o tempo necessário para atingir o valor do sinal de referência, t_p é o tempo de pico (peak time) e t_s (settling time) é o tempo para o sistema entrar em regime, observando um critério de erro em regime (OGATA; HALL, 2010).

Figura 5 – Parâmetros de uma resposta ao degrau de um sistema de segunda ordem ou superior.



Fonte – Adaptado de Ogata e Hall (2010).

A resposta ao degrau nos diz muito sobre os sistemas de interesse, pois nela podemos ver claramente a atuação dos *polos* e *zeros* dominantes e o ganho de baixa frequência da função de transferência do sistema. Os métodos clássicos para se calcular os parâmetros de controladores são baseados na resposta ao degrau (OGATA; HALL, 2010).

2.2.2 Controlador PID

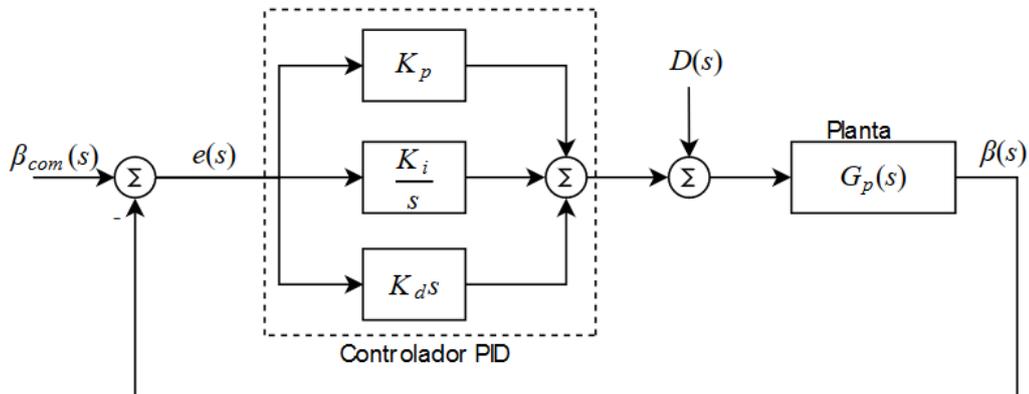
O controlador proporcional-integral-derivativo é um dos controladores mais utilizados nas aplicações industriais. Essa topologia de controlador consegue, em diferentes configurações atender entre 90 e 95% de todos os sistemas que necessitam de controladores (LEVINE, 1996). A forma descritiva matemática mais comum de se encontrar um controlador PID no domínio do tempo é a seguinte:

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (44)$$

Onde $e(t)$ é o erro, T_i é o tempo integral, T_d é o tempo derivativo e K o ganho proporcional. Uma outra forma de se representar os tempos integral e derivativo, é através dos ganhos K_i que é o ganho integral e o K_d , que é o ganho derivativo (ÅSTRÖM; HÄGGLUND, 1995).

Na figura 6, podemos ver a configuração mais utilizada do controlador PID, onde $\beta_{com}(S)$ é o valor do sinal de referência no domínio da frequência, $e(S)$ o erro, K_p é o ganho proporcional, K_i é o ganho integral, K_d é o ganho derivativo, $D(S)$ é um distúrbio, $G_p(S)$ é a função de transferência da planta e por fim, $\beta(S)$ que é a variável de interesse (SNIDER, 2010).

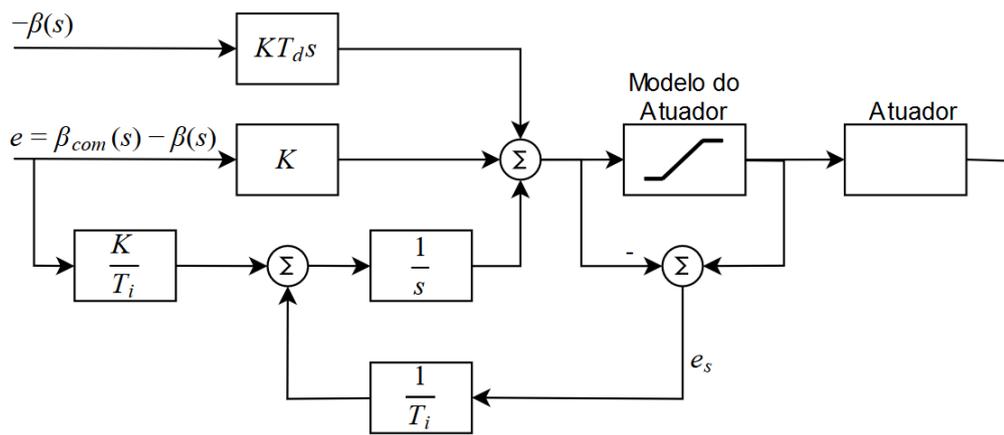
Figura 6 – Representação do modelo paralelo de um controlador PID com distúrbios.



Fonte – Adaptado de Snider (2010).

Como alguns sistemas podem admitir grandes e rápidas variações de sinais de referência, exigindo uma grande força de controle e energia no atuador, algumas topologias foram desenvolvidas para limitar a atuação de alguns fatores dos controladores. Um bom exemplo é o *anti-windup*, onde essa topologia limita a saturação do atuador quando o sistema atinge o regime, causado pela ação integral. Essa topologia pode ser vista no modelo da figura 7 (ÅSTRÖM; HÄGGLUND, 1995).

Figura 7 – Modelo de um controlador PID com histerese.



Fonte – Adaptado de Åström e Hägglund (1995).

Existem muitas combinações de controladores PID, cada uma com suas peculiaridades e vantagens de uso. Uma combinação muito usada é a PI, onde a ação integral de zera o erro em regime e uma boa resposta transitória já satisfazem as especificações (ÅSTRÖM; HÄGGLUND, 1995).

2.2.3 Controle Digital

Quando utilizamos um dispositivo digital como controlador, temos que representar os sinais de forma discretizada, ou seja, em níveis que possam ser transformados em um conjunto de bits. Só assim o microcontrolador ou microprocessador poderá realizar as operações matemáticas desejadas. Para essa transformação, usamos conversores AD (analógico-digital) que convertem um sinal analógico (contínuo) em um sinal binário (Bong Wie, 2001). Um sinal analógico é representado digitalmente com um período T , da seguinte forma:

$$y(0), y(T), y(2T), \dots \quad (45)$$

Se expressarmos usando notação $y[kT]$, podemos representar uma equação diferencial da seguinte forma:

$$u(k) = a_0y[k] + a_1y[k-1] + \dots + a_ny[k-n] - b_1y[k-1] - b_2y[k-2] - \dots - b_ny[k-n] \quad (46)$$

Onde $u(k)$ denota o sinal computado e $y(k)$ os valores o sinal analógico nos instante k -ésimos.

2.2.3.1 Transformada Z

A transformada z é usada para aquisição e processamento de dados em sistemas discretos. Ela possui relação matemática com o domínio s da seguinte forma:

$$Z = e^{-Ts} \quad (47)$$

Por definição, a transformada z é um somatório do produto de z^{-k} com o valor do sinal analógico amostrado no momento k , $y(k)$. Isso pode ser conferido na equação as seguir.

$$y(z) = \sum_{k=0}^{\infty} y[k]z^{-k} = y[0] + y[1]z^{-1} + y[2]z^{-2} \dots \quad (48)$$

Se usarmos os Teoremas da Tradução, Valor Inicial e Valor Final, conseguimos expressar uma função da seguinte forma (Bong Wie, 2001):

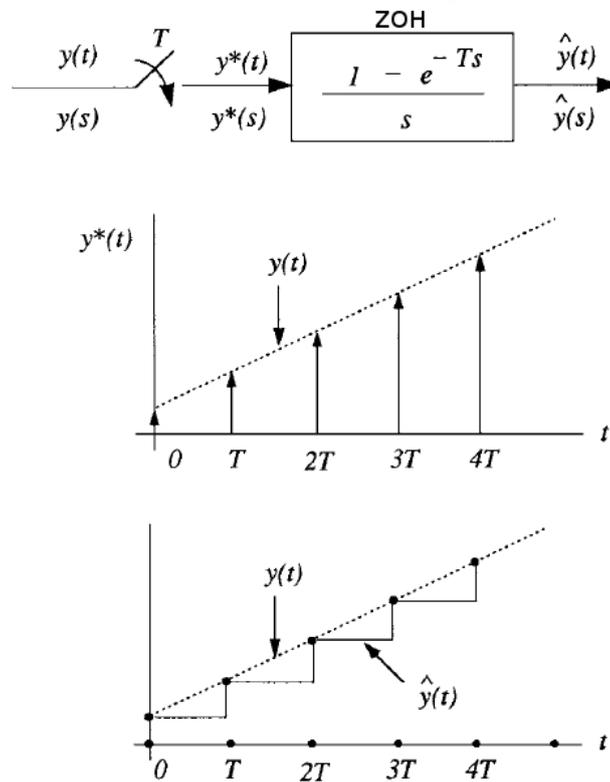
$$\frac{u(z)}{y(z)} = \frac{a_0 + a_1z^{-1} + \dots + a_nz^{-n}}{1 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n}} \quad (49)$$

2.2.3.2 Amostragem

Em um sistema discreto, uma forma de amostrar um sinal analógico é usar um trem de impulsos com amplitude do sinal $y(t)$ nos tempos T e reter esses dados com um ZOH (Zero-

Order-Hold - Amostrador de Ordem Zero) (Bong Wie, 2001). A figura 8 representa muito bem esse processo.

Figura 8 – Conversão de um sinal contínuo em discreto.



Fonte – Adaptado de Bong Wie (2001).

Onde $y(t)$ é o sinal que está sendo amostrado, $y^*(t)$ é o trem de pulsos amostrados e $\hat{y}(t)$ é o sinal resultante. Um ZOH ideal pode ser representado da seguinte forma:

$$\hat{y}(s) \approx \frac{1 - e^{-Ts}}{Ts} y(s) \tag{50}$$

Essa expressão é de suma importância para a aquisição das variáveis que queremos controlar usando técnicas digitais.

2.2.3.3 Controlador PID Digital

Agora, dese-se descrever a representação do controlador PID da equação 44 de forma discreta, para isso devemos usar a aproximação de Euler:

$$s = \frac{1 - z^{-1}}{T} \tag{51}$$

Conseguimos expressar o controlador PID no domínio z da seguinte forma:

$$u(t) = -K \left[1 + \frac{1}{T_i} \frac{T}{1 - z^{-1}} + T_d \frac{1 - z^{-1}}{T} \right] e(t) \quad (52)$$

Com isso, podemos representar de forma discreta o controlador PID, que pode ser visto na equação da sequência (Bong Wie, 2001):

$$u(t) = -K_p \left[y[k] - \frac{1}{T_i} \hat{u}[k] - T_d \frac{y[k] - y[k-1]}{T} \right] \quad (53)$$

Onde

$$\hat{u}[k] = \hat{u}[k-1] + T y[k] \quad (54)$$

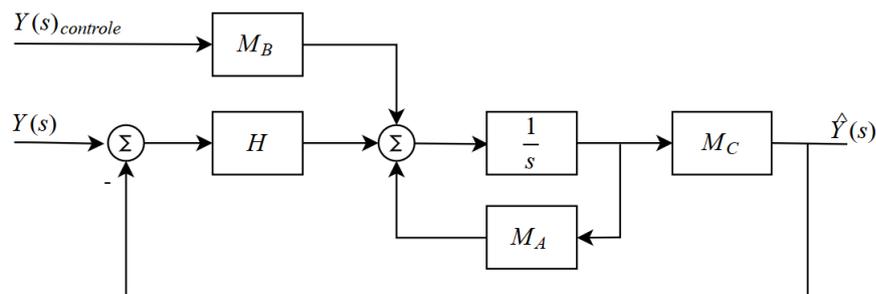
Desse modo, concluímos a modelagem de um controlador PID, com um conjunto de equações que podem ser, de forma simples, implementadas em um sistema digital.

2.2.3.4 Filtro de Kalman

Quando se trabalha na aquisição de grandezas físicas, encontramos ruídos nos sinais amostrados. Esses ruídos podem inviabilizar o controle de uma planta, pois, não informam o verdadeiro estado do sinal amostrado, fazendo com que o atuador atue de forma equivocada, levando à instabilidade do sistema. Uma forma de minimizar esse efeito, é utilizando o filtro de Kalman.

Filtro de Kalman é um algoritmo usado para estimar estados de variáveis de um sistema, sujeito a ruídos estocásticos. Esse algoritmo relaciona as informações sobre a dinâmica da planta e informações sobre o comportamento estocástico dos sinais com o objetivo de estimar variáveis de estados da planta (LEVINE, 1996). Na figura 9, temos a representação em diagrama de blocos do filtro de Kalman.

Figura 9 – Diagrama de blocos do filtro de Kalman.



Onde $U(S)$ é o sinal do controlador, $Y(S)$ é o sinal amostrado da saída da planta, $\hat{Y}(S)$ é a saída estimada, H é o ganho do filtro e M_A , M_B e M_C são as matrizes em espaço de estados da dinâmica da planta.

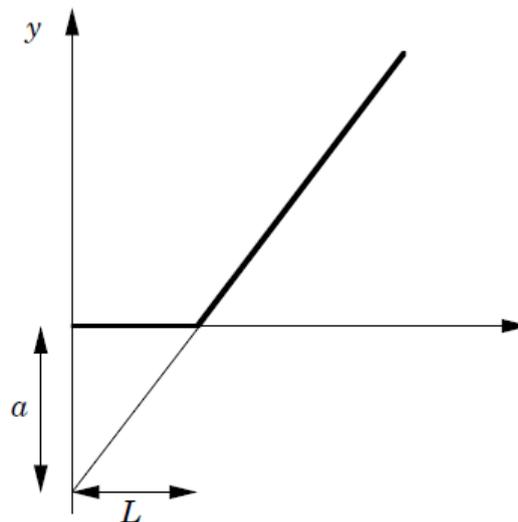
2.3 Sintonia de Controladores

Como podemos ver na equação 44, os valores de ajuste K_p , T_i e T_d podem assumir infinitos valores, sendo necessário a escolha do melhor conjunto desses valores para que a planta desempenhe o comportamento esperado (OGATA; HALL, 2010).

2.3.1 Método de sintonia de Ziegler-Nichols

Dois métodos clássicos de sintonia foram desenvolvidos em 1942 por Ziegler e Nichols. Esses dois métodos são baseados em características da resposta ao degrau e com a resposta em frequência. Na figura 10 podemos ver a resposta ao degrau e os parâmetros de atraso (L) e velocidade da resposta (a), que são usados para o cálculos dos parâmetros do controlador (ÅSTRÖM; HÄGGLUND, 1995).

Figura 10 – Parâmetros para a sintonia de controladores PID a partir da resposta ao degrau.



Fonte – Åström e Hägglund (1995).

Na tabela 1, podemos ver as relações dos parâmetros K_p , T_i , T_d e T_p com os vistos na figura 10.

Tabela 1 – Parâmetros PID pelo método de Ziegler-Nichols - resposta ao degrau

Controlador	K	T_i	T_d	T_p
P	1/a			4L
PI	0.9/a	3L		5.7L
PID	1.2/a	2L	L/2	3.4L

Fonte – Adaptado de Åström e Hägglund (1995)

O outro método de se estimar os valores do controlador, é através de duas características da resposta em frequência, que usa a seguinte tabela para encontrar os parâmetros do controlador.

Tabela 2 – Parâmetros PID pelo método de Ziegler-Nichols - resposta em frequência

Controlador	K	T_i	T_d	T_p
P	$0.5K_u$			T_u
PI	$0.4K_u$	$0.8T_u$		$1.4T_u$
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$	$0.85T_u$

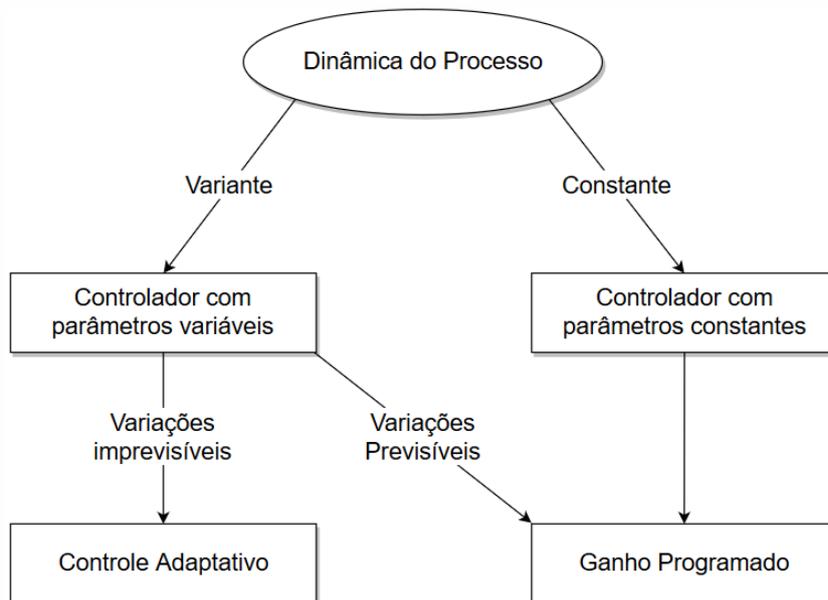
Fonte – Adaptado de Åström e Hägglund (1995)

Para usarmos a tabela 2, precisamos encontrar dois parâmetros, um é o ganho que deixa o sistema marginalmente estável (K_u), e a outro, é o período do sinal de referência que deixa o sistema marginalmente estável (T_u). Como podemos ver na tabela 2 as relações entre K , T_i , T_d , T_p com (K_u) e (T_u). Após a apresentação dos métodos clássicos, precisamos compreender os métodos automáticos.

2.3.2 Sintonia Automática de Controladores

Como muitos sistemas sofrem variações temporais, distúrbios e o interesse do usuário em modificar constantemente a resposta da planta, cria-se a necessidade de automatizar o processo de sintonia, a um simples comando do usuário. Além da simplicidade do conceito de implementação, os controladores PID são muito utilizados pela possibilidade de auto sintonia e adaptatividade dos controladores, pois a partir do comportamento da resposta ao degrau, podemos calcular os parâmetros do controlador e aplicarmos sem a intervenção humana (ÅSTRÖM; HÄGGLUND, 1995). Na figura 11, podemos a partir do comportamento da planta, escolher a, ou as técnicas mais adequadas que devemos implementar para controlar de forma satisfatória a planta.

Figura 11 – Fluxograma para a escolha do método de sintonia.



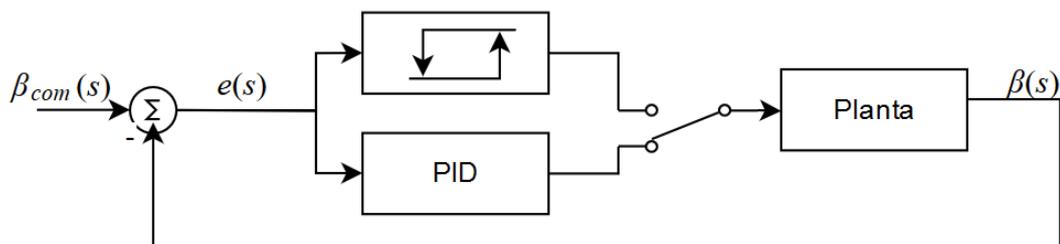
Fonte – Adaptado de Åström e Hägglund (1995).

Uma alternativa possível para sintonia de controladores, é o método do relé, ele, no modo clássico, se encontra entre os controladores com parâmetros constantes.

2.3.2.1 Método do Relé

Características da resposta em frequência podem ser descobertas se somarmos ao sinal de referência, uma onda retangular ao mesmo tempo em que o controlador PID está desconectado. Esse é o princípio do método de sintonia usando relé, onde o relé desempenha o papel do chaveamento e cria um sinal retangular sobreposto ao sinal de referência (LEVINE, 1996). Na imagem 12 podemos ver o conceito básico do método do relé. O modelo matemático usado para descrever o comportamento de um relé, pode ser visto na equação 55 que está na sequência.

Figura 12 – Modelo do método de auto sintonia via relé.



Fonte – Adaptado de Åström e Hägglund (1995).

$$N(a) = \frac{4d}{\pi a} \left(\sqrt{1 - \left(\frac{\varepsilon}{a}\right)^2} - i\frac{\varepsilon}{a} \right) \quad (55)$$

Onde d é a amplitude de oscilação do relé (normalmente até 10% do sinal de referência), ε é a histerese do relé e a é a amplitude de oscilação da saída (LEVINE, 1996). Com isso, podemos calcular os parâmetros intermediários para a sintonia do controlador da seguinte forma:

$$K_u \approx \frac{4d}{\pi a} \quad (56)$$

Onde T_u é o período de oscilação do sinal de interesse. Munido desses valores, podemos recorrer a tabela do método Ziegler-Nichols em resposta em frequência (Tabela 2).

2.4 Estado da Arte - Controladores Inteligentes

O estado da arte dentro de controle, está na criação de controladores inteligentes. Controle inteligente implementa conceitos de inteligência de organismos biológicos em forma de algoritmos. Esses algoritmos são usados para automatizar os processos de sintonia, promover a adaptação do controle perante as mudanças que a planta possa apresentar ou otimizar os parâmetros atuais. Uma das principais formas de controle inteligente utilizadas na atualidade, é a por *redes neurais artificiais*, será explicada na sequência.

2.4.1 Controle com Redes Neurais Artificiais

Uma *Rede Neural* (RNA) é um modelo matemático generalizado da cognição humana ou de um organismo biológico. O elemento básico de uma RNA é o neurônio, que associado em camada, forma uma rede de neurônios. Um neurônio pode ter n entradas e p saídas, e dentro de uma mesma camada, todos os neurônios devem ser idênticos. O processo de cognição é caracterizado como um comportamento não-linear, a função (f_{rn}) que descreve a excitação um neurônio binário unipolar, pode ser vista na sequência (ÜNAL et al., 2013).

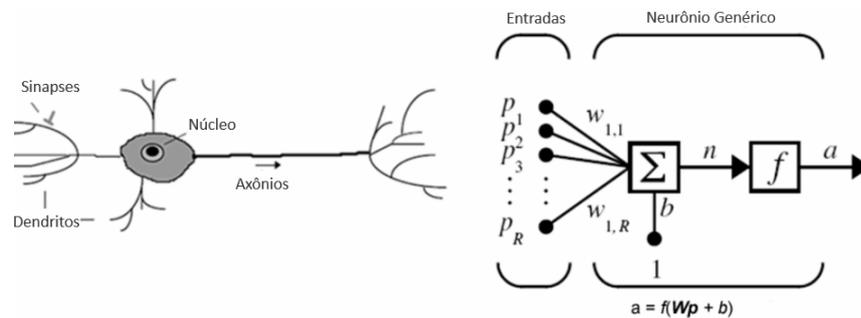
$$f(rn) \equiv \frac{1}{1 + e^{-\lambda_n}} \quad (57)$$

também

$$f(rn) \equiv \text{sgn}(rn) = \begin{cases} 1, & rn > 0 \\ 1, & rn < 0 \end{cases} \quad (58)$$

Onde λ_n é o ganho do neurônio. Uma representação gráfica de um neurônio natural comparado com um artificial pode ser visto na figura 13, onde os dendritos são representados pelas entradas p_1, p_2, \dots, p_R , uma soma dessas entradas que posteriormente é aplicada a função da equação 57, onde obtemos a saída a .

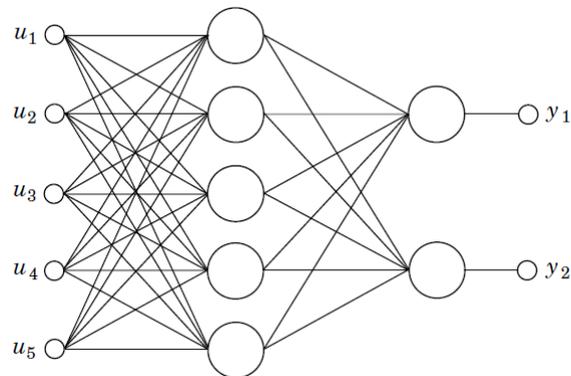
Figura 13 – Célula de um neurônio natural e modelo matemático respectivo.



Fonte – Adaptado de Únal et al. (2013).

Ao paralelizar um grupo de neurônios, criamos uma camada de neurônios, cada elemento dessa camada é conectada em todas as entradas, ou em todos os neurônios da camada anterior, ou posterior, ou ainda, nas saídas. Uma representação de uma rede neural pode ser vista na figura 14, onde temos 5 entradas (u_1, \dots, u_5) e duas saídas (y_1, y_2).

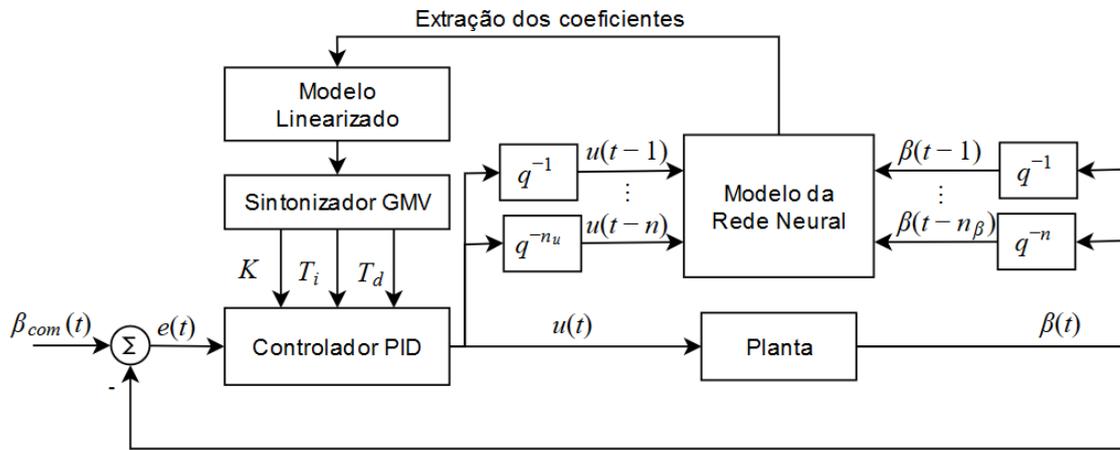
Figura 14 – Representação de uma rede neural.



Fonte – Adaptado de Åström e Hägglund (1995).

Em controle, as RNAs são usadas principalmente em sistemas não lineares, onde existem regiões que a planta não é controlável pelas técnicas de controle clássicas Chen e Huang (2004). Nesse caso, a RNA recebe os sinais de controle do controlador e da saídas da planta, realiza as operações e encontra os novos parâmetros para o controlador PID. Um bom exemplo dessa topologia pode ser visto na figura 15, onde os parâmetros obtidos pela rede neural ainda são linearizados para que sejam utilizados pelo controlador.

Figura 15 – Controlador PID com rede neural.



Fonte – Adaptado de Chen e Huang (2004).

Na imagem anterior, existe o GMV (*General Minimum Variance* - Variância Mínima Geral) que é usado para extrair os parâmetros K , T_i e T_d do modelo da rede neural linearizada Chen e Huang (2004).

2.4.1.1 Regressão Não-Linear Robusta

Regressão é um método matemático para o ajuste de curvas através da otimização de parâmetros que compõem a equação característica do sinal. O caso de interesse, é a regressão para sinais não-lineares, os quais são provenientes da RNA, que precisarão ser otimizados sobre a equação característica do controlador PID (equação 53). De forma genérica, a equação que descreve um modelo não-linear pode ser descrita da seguinte forma (Riazoshams, H., Midi, H., & Ghilagaber, 2018):

$$y_i = f(x_i; \theta_e) + \varepsilon_i, \quad i = 1, \dots, n \quad (59)$$

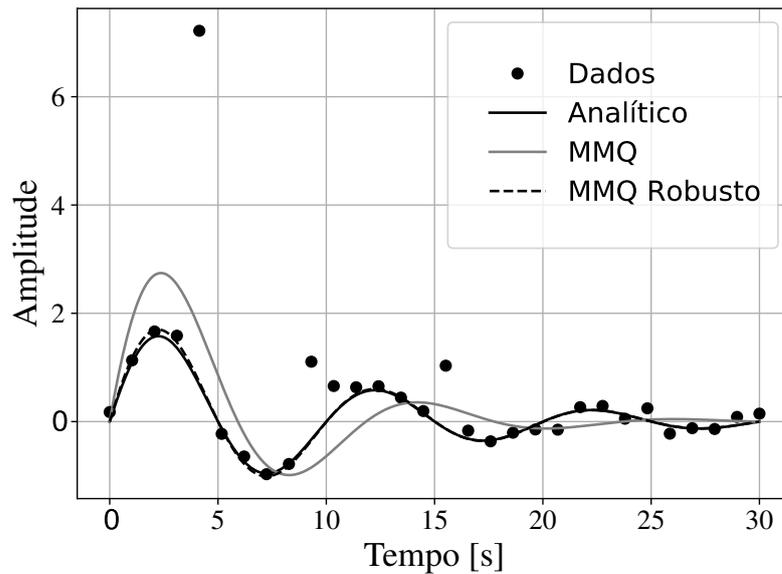
Onde y_i é a resposta observável, x_i é o vetor k-dimensional com os valores conhecidos e θ_e é p-dimensional com valores desconhecidos que influenciam a resposta observável e ε_i é o erro que satisfaz alguma distribuição estocástica.

Um dos métodos mais usados para regressão, independente da linearidade do sistema, é o *Método dos mínimos quadrados* (MMQ), que pode ser representado da seguinte forma:

$$S(\theta_e) = \sum_{i=1}^n [y_i - f(x_i; \theta_e)]^2 \quad (60)$$

Esse método por definição, é usado onde as amostras obedecem uma distribuição normal. Quando isso não é verdadeiro, usa-se o método MMQ robusto, onde o erro pode assumir outras distribuições, garantindo uma boa resposta da regressão. Na figura 16, podemos ver a comparação entre os métodos MMQ e MMQ Robusto, ficando clara a melhor resposta do MMQ Robusto(MAYOROV, 2015).

Figura 16 – Comparação entre os métodos de regressão não-lineares.



Fonte – Adaptado de Mayorov (2015).

Os últimos itens abordados, apontam algumas das modernas técnicas aplicadas em controle, as quais, foram aplicadas no presente trabalho de forma conjunta, propondo uma técnica afim para a sintonia de controladores. Após elucidados os conceitos básicos para a realização do trabalho, todos os passos para a implementação da mecânica, do software e o sistema de controle serão descritos no próximo capítulo.

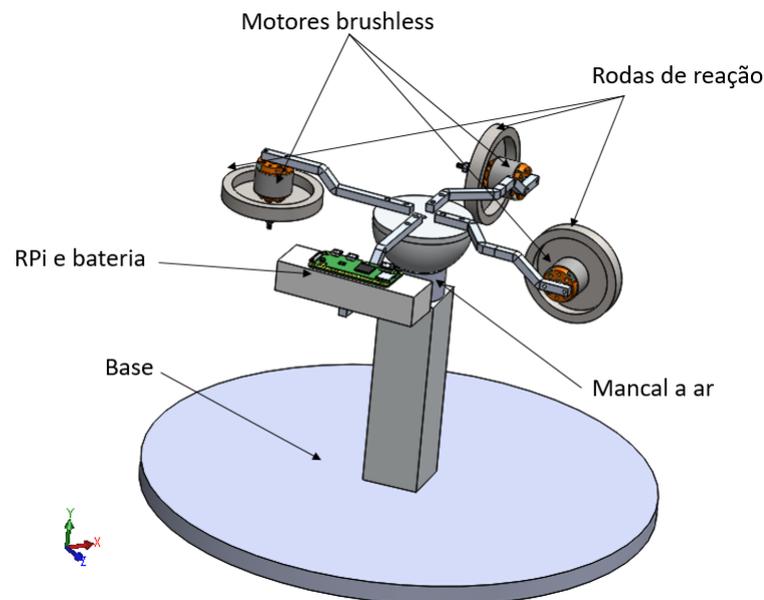
3 METODOLOGIA

A metodologia do presente trabalho é dividida em três principais partes. A primeira é a etapa de projeto dos componentes mecânicos que fazem parte do satélite, entre eles: as rodas de reação, motor DC (*Direct Current* - corrente contínua), mancal a ar e hastes. Também contempla as especificação dos componentes eletrônicos que constituem o satélite. É na segunda etapa que ocorre a modelagem completa do sistema e especificações das limitações de movimentação do satélite, juntamente com a elaboração dos diagramas de blocos do controle. Nessa etapa também é feita a modelagem do motor DC, para que possa fornecer torque para tirar da inércia a planta. Já na terceira etapa, são abordados os elementos de *software*, de rede e supervisão do satélite. Nessa etapa também se encontram as especificações e o desenvolvimento do *software* de controle e supervisão.

3.1 Hardware

Após a modelagem apresentada no referencial teórico, é possível modelar um simulador de satélite de fabricação factível. O mancal a ar, a base de suporte do simulador do satélite, rodas de reação, motores *Brushless* (motor de corrente contínua sem escovas) e os outros elementos, podem ser vistos na figura 17.

Figura 17 – Desenho mecânico completo do simulador de satélite.



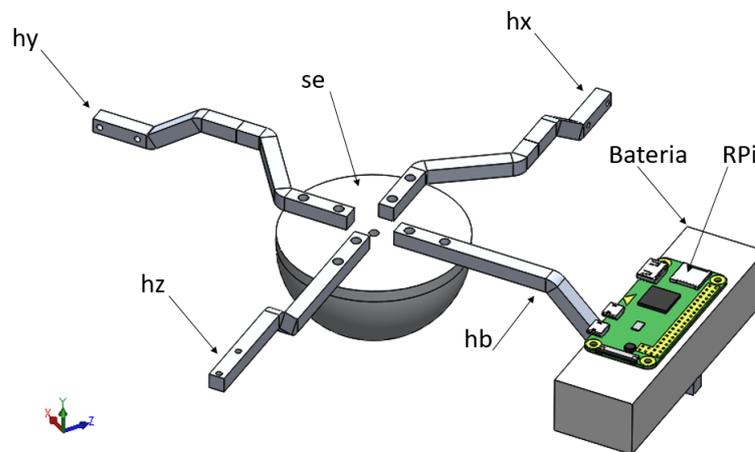
Fonte – Elaborado pelo Autor.

Como podemos ver na imagem anterior, o modelo sofre várias simplificações em relação a um satélite real, onde o modelo passa a ser de fácil implementação, possibilitando a validação dos conceitos propostos nesse trabalho. Dentre várias simplificações, a mais óbvia, o simulador não apresentará de forma direta os movimentos de translação descritos no seção 2.1.1. Ainda, existe uma limitação para os ângulos ϕ , θ e ψ que é um intervalo de $0 < \phi, \theta, \psi < 240^\circ$ devido a existência do mancal a ar e a simetria do corpo do satélite, como princípio para simplificar a modelagem matemática do mesmo. Os principais elementos do simulador de satélites, serão descritos de forma detalhada na sequência, dentre eles, o corpo do satélite.

3.1.1 Corpo do Simulador de Satélite

O elemento de dinâmica mais importante que deve ser dimensionado, é o corpo do satélite. É nele que são fixados todos os outros elementos que compõem o simulador de satélites. Para isso, optamos como material, o alumínio, pois possui menor densidade e resistência mecânica adequada para o projeto. Na figura abaixo podemos ver o corpo do simulador.

Figura 18 – Corpo do simulador de satélites.



Fonte – Elaborado pelo Autor.

Onde hx é a haste do motor que atua no eixo x, hy é a haste do motor que atua no eixo y, hz é a haste do motor que atua no eixo z, hb é a haste da bateria e se é a semiesfera que é a interface entre o corpo e o mancal a ar. Esse último, apresenta explicação na sequência. O projeto leva em consideração a distribuição de massa e os centros de massa, pois o simulador de satélite deve ficar o mais simétrico e em equilíbrio possível, facilitando a atuação do controle que prevê sua simetria.

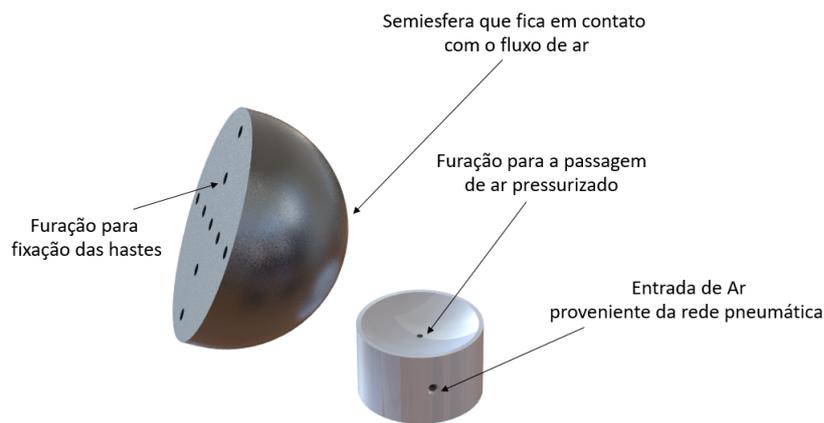
Através do *software* de desenho mecânico SolidWorks®, onde o projeto foi desenvolvido, conseguimos calcular o peso do satélite, que é de aproximadamente 1,25kg, contado com a bateria de polímero de Lítio e outro periféricos. Esse valor é necessário para o dimensionamento

das rodas de reação, que também será descrito na sequência. Ainda, com o mesmo *software*, conseguimos os valores de momento de inércia máximo entre os eixos, que é de aproximadamente $0,5\text{ kg m}^2$, o qual é usado para dimensionamento das rodas de reação.

3.1.2 Mancal a Ar

Um dos elementos mais importantes do simulador, é o mancal a ar. Ele possibilita a criação mínima de atrito entre a esfera e a região côncava, pois existe um fluxo de ar promovido por um sistema pneumático que cria uma camada de ar entre as duas partes. Com isso, a esfera fica com movimento praticamente livre de atritos dentro de uma região limitada pela geometria da esfera, o corpo do satélite, a massa do satélite e a pressão do sistema pneumático. O modelo mecânico do mancal com seus elementos podem ser vistos na figura 19.

Figura 19 – Desenho mecânico do mancal a ar.



Fonte – Elaborado pelo Autor.

O mancal precisou ser usinado, devido à precisão do encaixe entre a esfera e a região côncava, para que a camada de ar seja a mais homogênea possível, evitando turbulências e regiões com atrito maiores que em outras.

3.1.3 Motor DC e Rodas de Reação

Como foi descrito no referencial, os atuadores do satélite serão rodas de reação acopladas em motores de corrente contínua (DC). Na sequência, são descritos os cálculos das rodas de reação e a escolha dos motores DC.

3.1.3.1 Projeto das Rodas de Reação

Partindo do princípio da conservação do momento angular do corpo do satélite, precisamos dimensionar o momento angular da roda e, por consequência, os momentos de inércia delas.

Para isso, precisamos calcular o momento angular do satélite, que pode ser descrito como:

$$\vec{L}_s = 2 \left(I_{sat} + m \left(\frac{r_{sat}}{2} \right)^2 \right) \quad (61)$$

Cada roda é responsável por um terço do momento angular total do satélite, como pode ser visto na sequência:

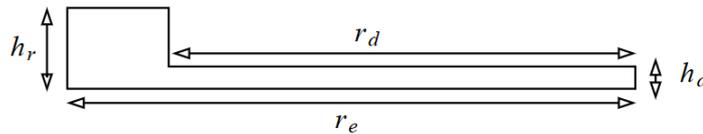
$$\vec{L}_{roda} = \frac{\vec{L}_s}{3} \quad (62)$$

Onde I_{sat} é o momento de inércia do satélite, m é a massa, r_{sat} é a distância dos motores em relação ao centro de massa. Podemos relacionar o momento angular com o momento de inércia das rodas e a velocidade angular, como podemos ver abaixo.

$$\vec{L}_{roda} = I_{roda} \vec{\omega}_{roda} \quad (63)$$

Conseguimos dimensionar o momento angular das rodas através da variação do momento de inércia ou rotação das rodas. Com o objetivo de controlar o satélite, varia-se a velocidade angular das rodas com momento de inércia fixo. Na figura 20, temos o desenho escolhido para o projeto das rodas, pois, concentramos mais massa na extremidade, contribuindo assim, para um maior momento de inércia.

Figura 20 – Geometria básica para o projeto de rodas de reação.



Fonte – Elaborado pelo Autor.

Essa geometria escolhida, possui momento de inércia que pode ser descrito pela seguinte equação:

$$\vec{I}_{roda} = \rho \frac{\pi}{2} (h_r (r_e^4 - r_d^4) + h_d r_d^4) \quad (64)$$

Onde \vec{I}_{roda} é o momento de inércia da roda, ρ é a densidade do material que compõe a roda, h_r é a espessura da borda, h_d é a espessura do disco, r_e é o raio total da roda e r_d é o raio do disco. Um conjunto de parâmetros que satisfazem as equações acima descritas à uma velocidade angular de $230,38 \text{ rad s}^{-1}$ (2200 RPM) pode ser visto na tabela 3.

Tabela 3 – Dimensões das rodas de reação.

Dimensão	Unidade
ρ	7860 kg m^{-3} (Aço SAE 1045)
h_r	10,5 mm
h_d	2 mm
r_e	37 mm
r_d	30 mm

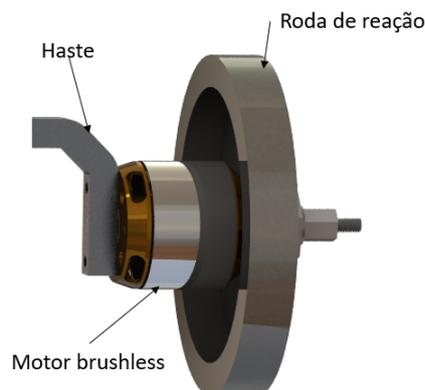
Fonte – Elaborado pelo Autor.

Com isso, concluímos o projeto da roda de reação, agora, passamos para a etapa do projeto do motor, que está na sequência.

3.1.3.2 Escolha dos Motores DC

O atuador do simulador de satélite é constituído de um conjunto de três motores *Brushless* dispostos um em cada eixo cartesiano. Juntamente com esses motores, são acopladas rodas de reação, que farão os movimentos de rotação do satélite. Como exibido na imagem 17, os motores e as hastes estão distribuídos de forma simétrica e afastados do centro de massa do satélite, para facilitar a movimentação promovida pelo torque dos motores. O modelo de motor escolhido foi o *Turnigy D2826-6 2200kV*, que é um motor usado em aeromodelos que possui relação de velocidade de 2200 RPM por Volt. Na imagem 21 podemos ver parte da haste, o motor e a roda de reação, todos acoplados.

Figura 21 – Desenho mecânico do conjunto motor-roda.



Fonte – Elaborado pelo Autor.

A roda de reação também foi usinada, pois é necessário uma roda com uma massa e formato específico, para que possa ser acoplar ao motor e o conjunto consiga realizar os movimentos desejados.

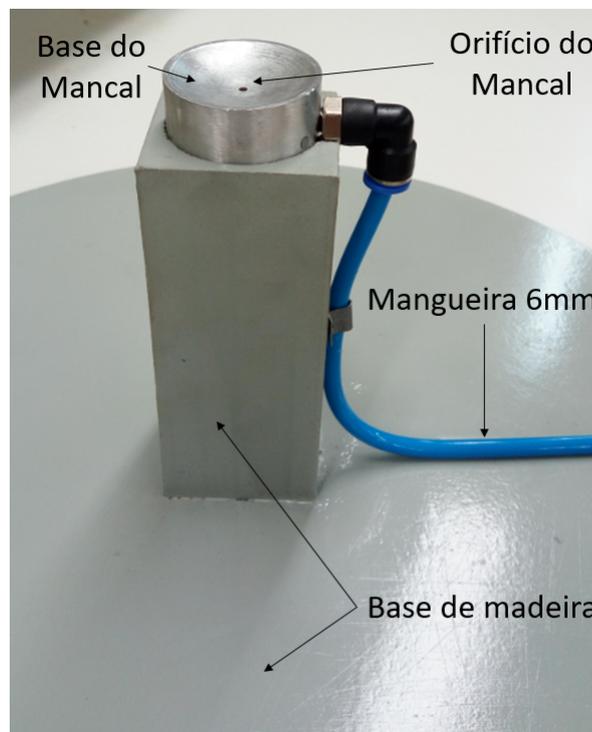
3.1.4 Elementos Eletrônicos

A placa de desenvolvimento e prototipação de sistemas embarcados Raspberry Pi Zero W (RPi) foi a escolhida, pois ela conta com os elementos mínimos de interfaceamento com os atuadores e outros periféricos. Além disso, ela possui conectividade *wireless* (sem fio), possibilitando a supervisão e controle do satélite sem fio. Como o trabalho terá o seu desenvolvimento baseado em desenvolvimento de *software* e adaptações do sistema operacional, uma placa com suporte, boa documentação e comunidade ativa, facilita o desenvolvimento.

Um outro elemento que é indispensável, é o acelerômetro, o qual desempenha o papel de sensor de realimentação da posições angulares do satélite. Sua comunicação com a RPi é dada através da interface I2C, que está descrita na seção de *software*. O giroscópio informará ao modelo a velocidade angular em graus por segundo, e o algoritmo calcula a posição através da integral da velocidade instantânea em cada eixo. Para isso, escolhemos o giroscópio MPU6050.

Ainda, para o acionamento dos motores *Brushless* serão utilizados ESC (*Electronic-Speed-Control* - Controlador eletrônico de velocidade) que serão conectados com a bateria, os motores e a RPi. Após a conclusão do dimensionamento e escolha dos materiais, foi possível fazer a fabricação mecânica e montagem do simulador satélite, o qual pode ser visto nas figuras 22, 23 e 24.

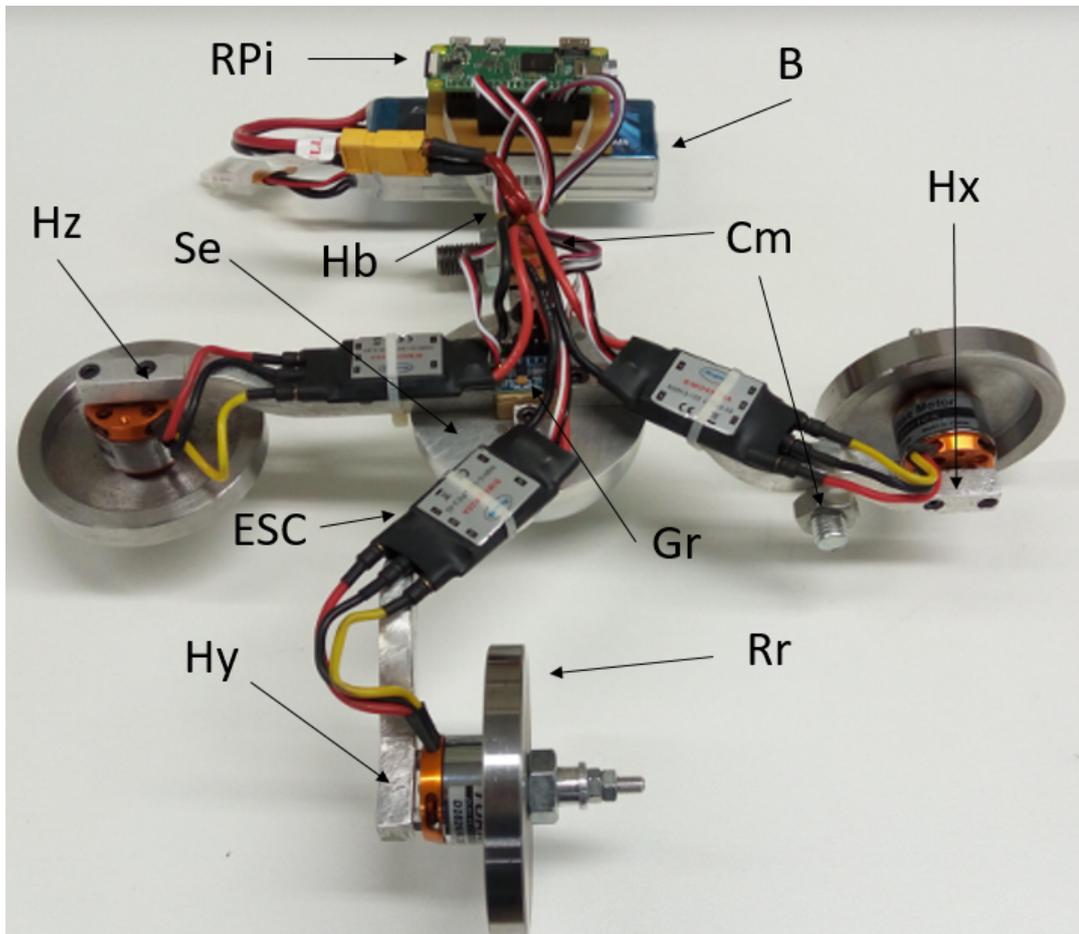
Figura 22 – Base do simulador de satélites prototipado.



Fonte – Elaborado pelo Autor.

Na imagem anterior, pode-se ver a base do simulador, feita em madeira, onde se encontram a região concava do mancal e as conexões pneumáticas. Já na figura 23, podemos ver as partes que compõem o corpo do simulador.

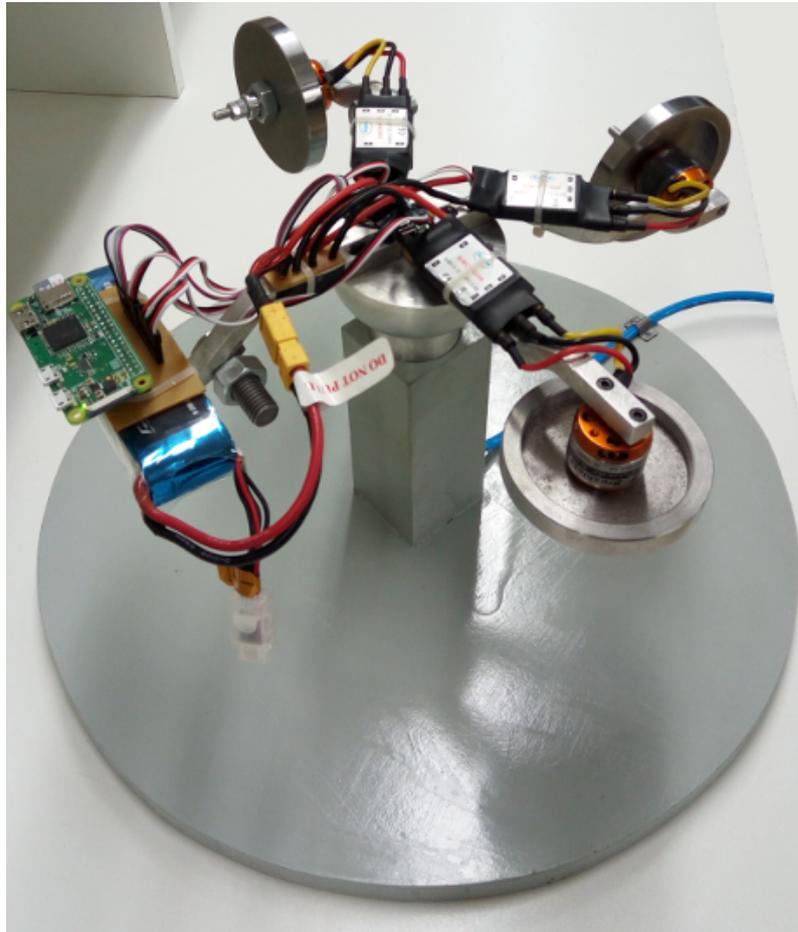
Figura 23 – Corpo do simulador de satélites prototipado.



Fonte – Elaborado pelo Autor.

Onde *Rpi* é a Raspberry Pi Zero W, *B* é bateria, *ESC* são os dispositivos de acionamento dos motores brushless, *Rr* são as rodas de reação, *Cm* são massas para compensação de massa do simulador, *Gr* é o acelerômetro/giroscópio, *Se* é a semiesfera de alumínio que compõe o mancal, *Hb* é a haste de alumínio da bateria, *Hx* é a haste do motor do eixo x, *Hy* é a haste do motor do eixo y e *Hz* é a haste do motor do eixo z. E por fim, a imagem completa do simulador de satélites, que pode ser vista na figura 24.

Figura 24 – Simulador de satélites prototipado.



Fonte – Elaborado pelo Autor.

Com isso, todos os principais elementos de hardware foram descritos, sendo a modelagem dos mesmos o próximo passo do desenvolvimento.

3.2 Sistemas de Controle

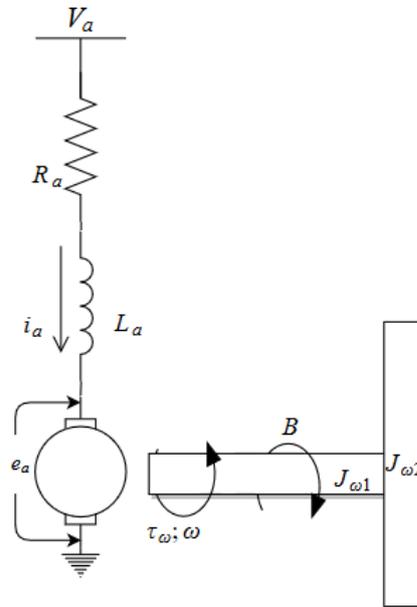
Após a descrição dos conceitos básicos de controle e do modelo mecânico do satélite, é possível desenvolver os modelos da planta e do sistema de controle. Ainda nessa seção, é descrita a modelagem do motor de corrente contínua sugerido na seção anterior e posteriormente a modelagem completa do simulador de satélites.

3.2.1 Modelo Motor DC

Após a escolha do tipo de motor que será utilizado como atuador no controle da atitude do satélite, se faz necessário a modelagem do mesmo, para que consigamos acoplar ao modelo do satélite, o torque que promoverá a variação no momento angular do satélite, e por consequência,

a posição angular. A figura 25 representa o modelo elétrico associado ao momento de inércia do rotor $J_{\omega 1}$ ao rotor e a da roda de reação $J_{\omega 2}$

Figura 25 – Modelo do motor DC.



Fonte – Elaborado pelo Autor.

Onde R_a , L_a , i_a e e_a são a resistência, a indutância, a corrente e a tensão de armadura, respectivamente, e_b é a tensão induzida e τ_ω é o torque do motor. A relação entre a corrente com a tensão de armadura pode ser vista na sequência (OGATA; HALL, 2010).

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a \quad (65)$$

Como existe a relação entre a constante de torque K_t e a tensão induzida, juntamente com a relação entre a constante de velocidade contra-eletromotriz K_w e a tensão da fonte V_a . Isso pode ser visto na sequência.

$$e_a = K_t \frac{d\theta}{dt} \quad (66)$$

e

$$e_b = K_w V_a \quad (67)$$

Temos que:

$$L_a \frac{di_a}{dt} + R_a i_a + K_t \frac{d\theta}{dt} = K_w V_a \quad (68)$$

Ainda, a relação que através do conceito de equilíbrio de torque, conseguimos relacionar o torque do motor τ_ω com os momentos de inércia do rotor e da roda de reação com a corrente da armadura da seguinte forma:

$$\tau_\omega = (J_{\omega 1} + J_{\omega 2}) \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} = K_t i_a \quad (69)$$

Onde B é o atrito viscoso. Como queremos relacionar a tensão da fonte com a velocidade angular, devemos manipular as equações 68 e 69 e aplicarmos a transformada de Laplace. O resultado dessas operações pode ser visto na sequência:

$$\frac{\omega(s)}{V_a(s)} = \frac{K_w K_t}{(R_a + sL_a)(s(J_{\omega 1} + J_{\omega 2}) + B) + K_w K_t} \quad (70)$$

Mas como nas equações 41, 42 e 43 é necessário a a derivada instantânea da velocidade, ou seja, a aceleração do motor, precisamos derivar a equação anterior, e ainda, podemos dizer que a soma dos momentos de inércia é representado somente por J , obtendo assim:

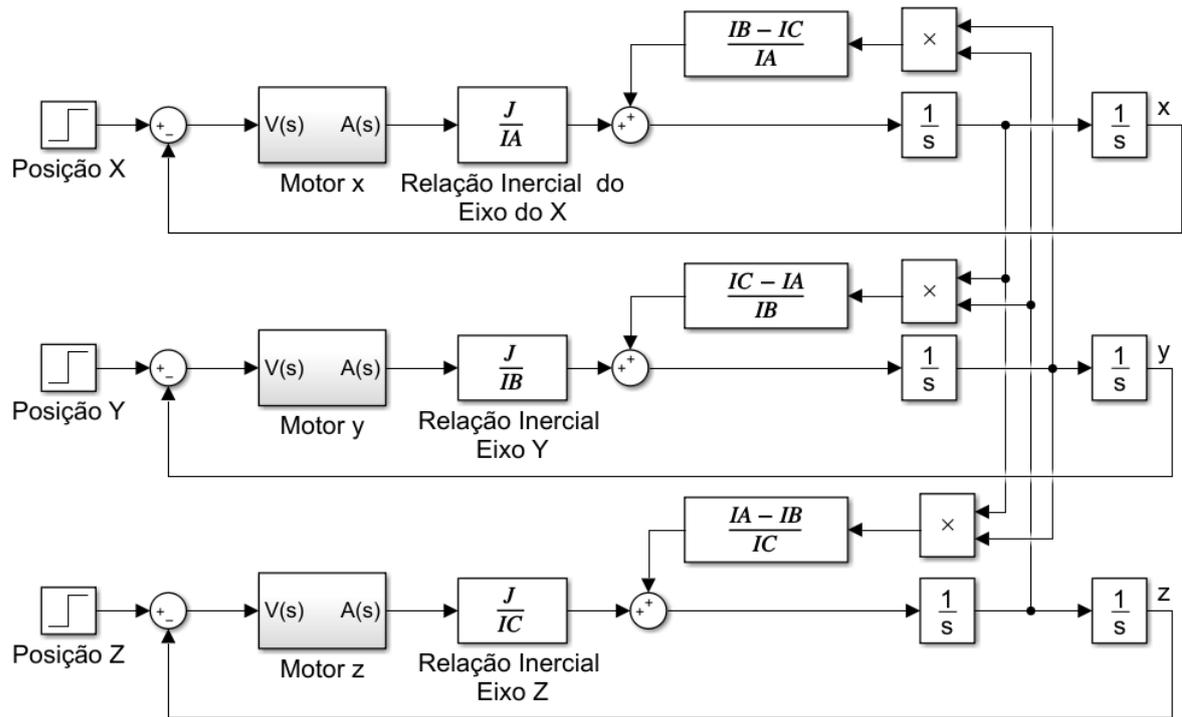
$$\frac{\dot{\omega}(s)}{V_a(s)} = \frac{K_w K_t s}{(R_a + L_a s)(J s + B) + K_w K_t} \quad (71)$$

Com isso, modelamos o atuador do satélite, o próximo passo para a completa modelagem, é acoplar esse modelo com o do satélite descrito na seção 2.1.1. Esses passos serão descritos na próxima subseção.

3.2.2 Modelo Completo do Satélite

Como em partes, todo o satélite já foi modelado até agora, nessa subseção acoplaremos todos os modelos em um único, que terá como variável de interesse a posição angular tridimensional definida pelos ângulos ψ, θ, ϕ , que são os ângulos em relação aos três eixos cartesianos. Para a modelagem completa, usamos as equações que descrevem o comportamento das rodas de reação, do motor cc e da própria dinâmica do satélite. Essa associação se dá em forma de diagrama de blocos já em malha fechada, que pode ser visto na figura 29.

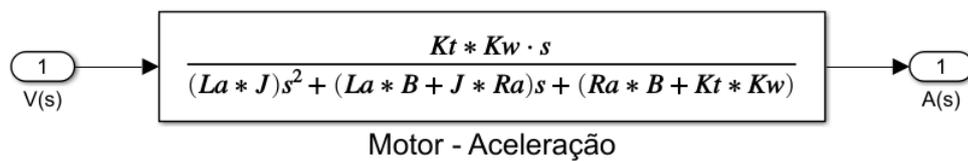
Figura 26 – Diagrama de blocos em malha fechada.



Fonte – Elaborado pelo Autor.

Onde o bloco subsistema Motor x, y e z, é constituído pela função de transferência da equação 72, que é inserida na ferramenta Matlab©da seguinte forma:

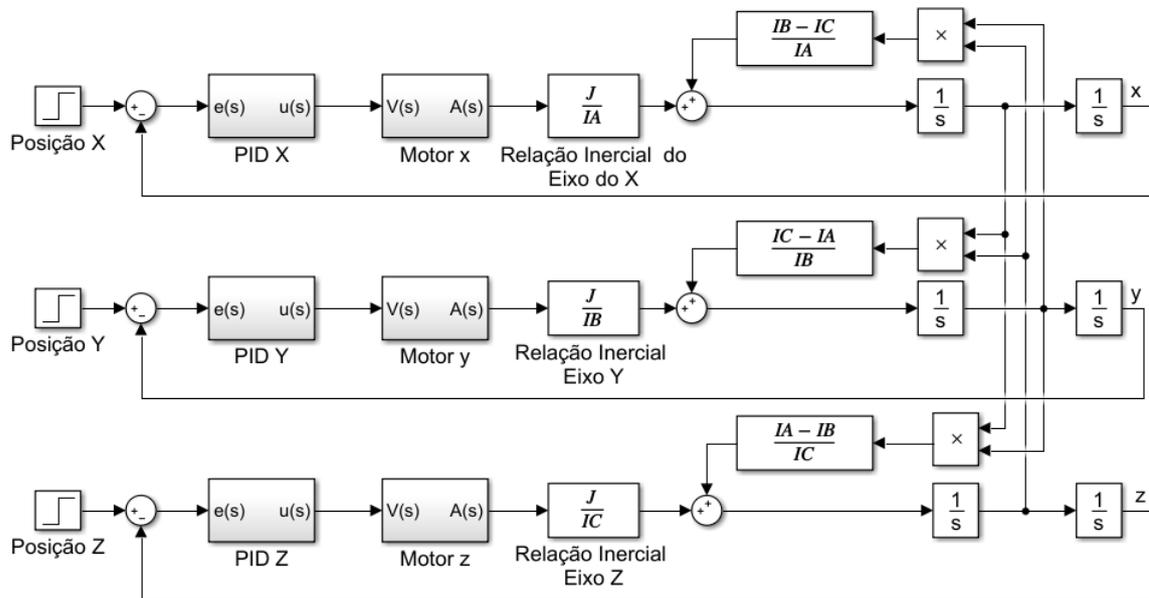
Figura 27 – Função de transferência dos motores.



Fonte – Elaborado pelo Autor.

Se fecharmos a malha em posição, usando um acelerômetro e usarmos um controlador PID, por consequência, conseguimos controlar a dinâmica do satélite em malha fechada. O diagrama de blocos com realimentação e com o controlador PID pode ser visto na figura 28.

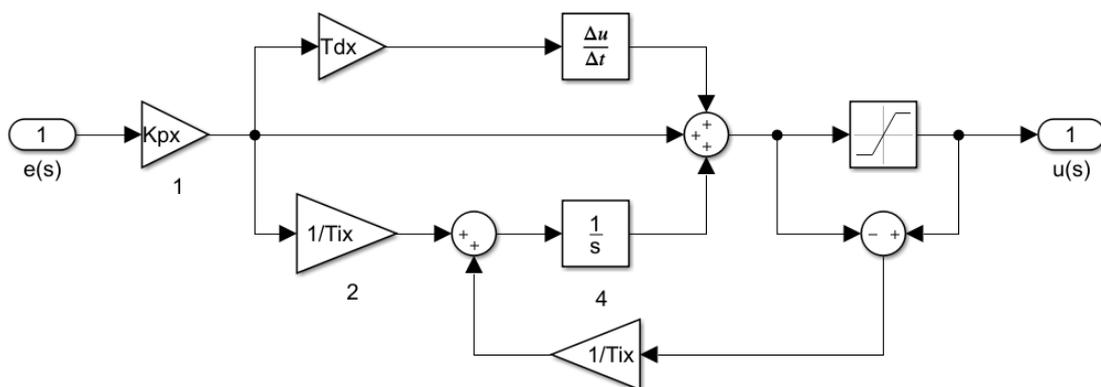
Figura 28 – Modelo em malha fechada com um controlador PID.



Fonte – Elaborado pelo Autor.

Onde o bloco subsistema PID é constituído pelos seguintes blocos:

Figura 29 – Digrama de blocos do controlador PID com Anti-Windup.



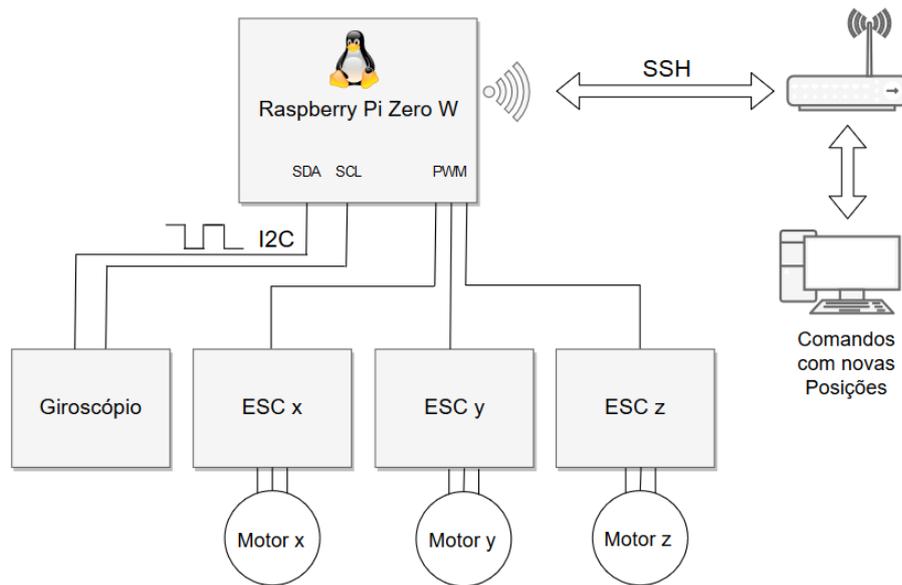
Fonte – Elaborado pelo Autor.

Esse último diagrama é usado para o desenvolvimento do controle em *software*, onde os parâmetros do controlador são estimados pelos diferentes tipos de sintonia descritos na revisão bibliográfica. Na sequência, serão descritos os métodos que serão utilizados para implementar esse modelo em Python e embarcar-lo na RPi.

3.3 Software

Após a modelagem, devemos implementar o modelo e configurar todas as interfaces com os atuadores e sistema supervisorio. A figura 30 descreve a forma que se dá a comunicação entre os periféricos, através da interface I2C para a RPi se comunicar com o acelerômetro/giroscópio, PWMs (*Pulse Width Modulation* - Modulação por Largura de Pulso) para a comunicação com os ESC e o protocolo SSH (*Secure Shell*) para a comunicação com o usuário.

Figura 30 – Representação geral do sistema.



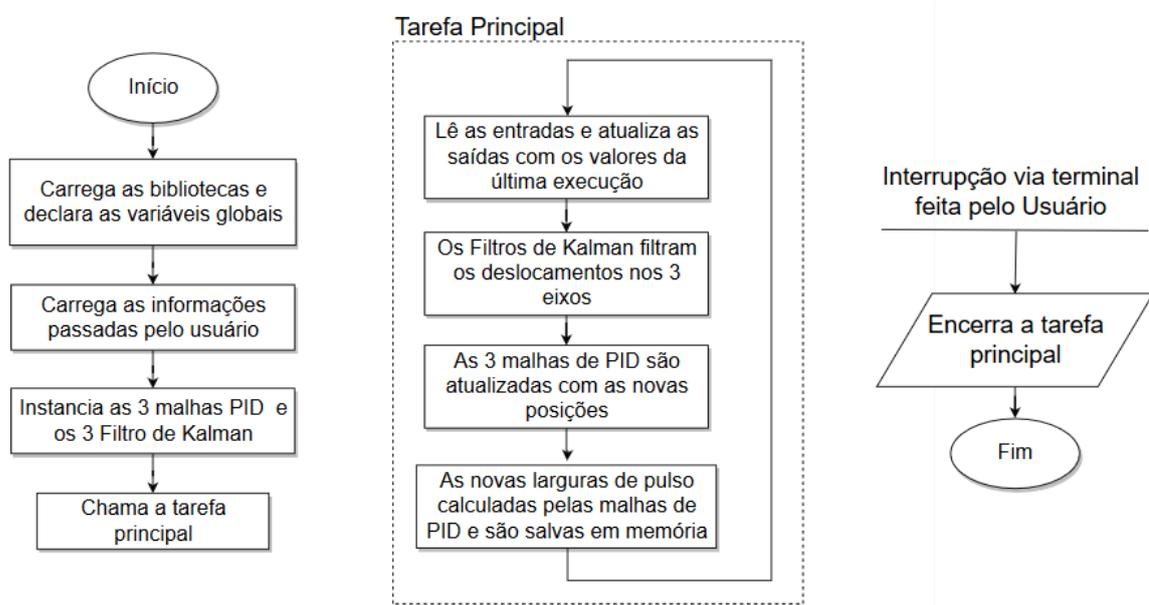
Fonte – Elaborado pelo Autor.

Após modelada a topologia geral do sistema, é possível dividir a parte de *software*, onde é descrita a implementação de cada etapa, começando com a implementação do controlador e da sua sintonia, que está na sequência.

3.3.1 Implementação do Controlador PID e Métodos de Sintonia

Nessa etapa, ocorre a implementação usando a linguagem de programação Python, a qual, possui diversas bibliotecas de controle, interfaceamento e protocolos de comunicação. O desenvolvimento foi orientado para o menor tempo possível de atualização das entradas e saídas, pois, como é uma linguagem *script* sendo executada por um sistema operacional, algumas tarefas poderiam ser executadas em baixa prioridade, criando um comportamento não determinístico do período de amostragem. Uma solução para esse problema, foi a criação de uma interrupção via relógio que chama uma função de atualização das entradas e saídas, além do cálculo do PID e do filtro de Kalman. Na figura 31 podemos ver o fluxograma do *script* de controle.

Figura 31 – Fluxograma do *software* de controle.



Fonte – Elaborado pelo Autor.

Após o desenvolvimento das malhas de controle e testes preliminares, ficou clara a grande influência dos sinais provenientes do giroscópio no resultado final, necessitando uma análise minuciosa para possibilitar o perfeito funcionamento do filtro de Kalman e, por consequência, controlar o simulador de satélites. Isso é abordado na sequência.

3.3.2 Análise dos Sinais e Filtragem

Ao fazer os primeiros testes, ficou perceptível o *bias* (deslocamento em relação ao zero) e a gama de ruídos que compõem os sinais amostrados. Para solucionar esse problema, e se os sinais possuem uma distribuição gaussiana, podemos usar a média aritmética de cada velocidade como *bias*. Os valores calculados pode ser vistos na tabela 4

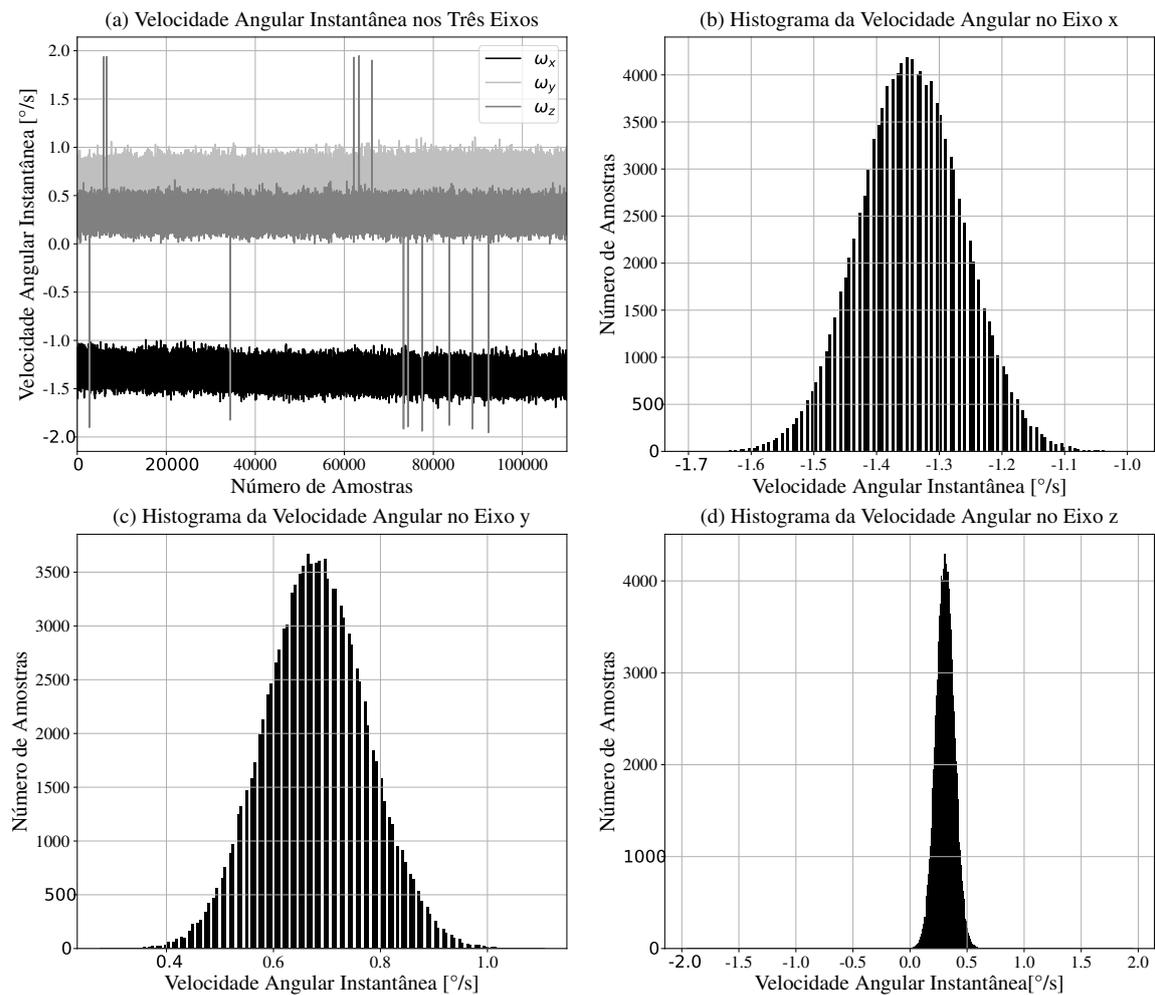
Tabela 4 – Correção do *bias* do giroscópio.

Eixo	Velocidade Angular ($\omega [^{\circ} s^{-1}]$)
x	-1.3441
y	0.6797
z	0.3071

Fonte – Elaborado pelo Autor.

Os sinais estão representados na figura 32 (a), onde temos mais de 100.000 amostras nos três eixos. Já nas imagens 32 (b), (c) e (d), podemos ver o histograma dos mesmos sinais.

Figura 32 – Características dos sinais amostrados pelo giroscópio.



Fonte – Elaborado pelo Autor.

Como podemos ver, os três sinais possuem uma distribuição característica muito próxima de uma normal, mesmo que no eixo z exista ruídos espúrios maiores que nos demais eixos, o que valida o uso do Filtro de Kalman e a escolha da média aritmética das amostras como *bias* para as amostras do giroscópio. Essa etapa corrigiu os problemas enfrentados, o próximo passo é sintonizar as malhas de controle, isso é descrito na sequência.

3.3.3 Métodos de Sintonia

O último passo da implementação do satélite, é a sintonia das malhas de controle, onde diferentes métodos foram utilizados para descobrir os parâmetros K_p , K_i e K_d . Para isso, usamos um método clássico de sintonia, que é o método do relé e por fim, o método automático usando redes neurais e regressão não-linear. Esses, serão descritos na sequência.

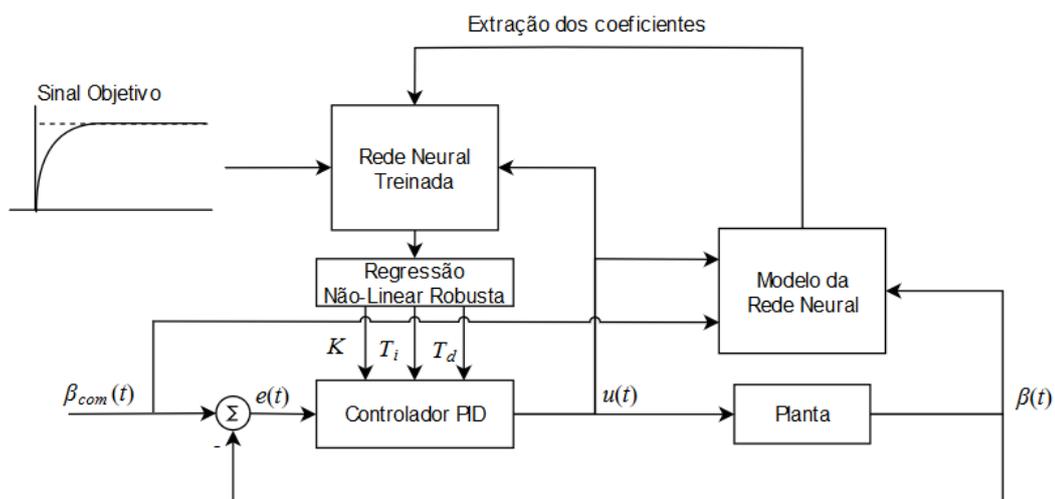
3.3.3.1 Método do Relé

A implementação desse método é simplesmente criarmos uma histerese sobre o erro, onde substituímos o sinal de controle por um valor fixo, fazendo o simulador de satélites oscilar ao tentar sair da região de histerese. Após a descoberta dos parâmetros que o levam à oscilação sustentada, basta usarmos a equação 55 e a tabela 2.

3.3.3.2 Método via RNA e Regressão Não-Linear

Por fim, o método proposto pelo presente trabalho, onde diferentes conceitos foram empregados para a sintonia automática de controladores PID. Os dois usados, foram o de RNAs e regressão não-linear robusta, de onde são extraídos os novo valores dos parâmetros. Na figura 33 podemos ver a topologia usada.

Figura 33 – Diagrama de blocos do sintonizador e otimizador.



Fonte – Elaborado pelo Autor.

O funcionamento dessa topologia é baseado na aprendizagem de máquina, onde uma RNA aprende o comportamento do satélite, tendo como entradas a posição atual (β), o valor de referência (β_{com}) e o sinal de controle, que nada mais é do que a largura de pulso entregue ao motor. Temos como saída da RNA, o sinal de controle, pois esse na prática, é um sinal modulado contendo a largura de pulso para o motor, onde a rede neural é mais assertiva devido ao comportamento não-linear, também característico de uma RNA. Após o treinamento da RNA, ela é usada para se criar uma nova base de dados, onde ela recebe como parâmetros um sinal que é uma sugestão de resposta ao degrau (Sinal Objetivo) fornecida pelo usuário, ou ainda, uma representação do sistema em forma de um sistema de primeira ordem, pois esse, apresenta características melhores do que o sistema original. Basicamente, o objetivo de se fornecer esse

sinal e usar como condições iniciais obtidas via método do relé, é orientar a regressão não-linear ao ponto ótimo ou algo próximo a ele. Os parâmetros usados na NRA podem ser vistos na tabela 5.

Tabela 5 – Parâmetros da RNA.

Elemento	Característica
Número de Neurônios	70
Número de Camadas	1
Tipo de RNA	Classificação
Número máximo de interações	400
Função de ativação	tanh

Fonte – Elaborado pelo Autor.

Por último, é feita a extração dos parâmetros do controlador via regressão, onde o antigo sinal de controle é modulado através da variação dos coeficientes do controlador PID discreto até possuir a forma de onda do novo sinal, que foi gerado pela rede neural. Essa regressão é feita usando como função de minimização, o próprio controlador PID discreto, como pode ser visto na equação da sequência:

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \sum_{n=0}^N e(t) dt - K_c T_d \frac{P[n] - P[n-1]}{dt} \quad (72)$$

Onde essa equação é análoga as equações do controlador PID discreto, descritas no referencial teórico, onde dt é a diferença de tempo entre as amostras, P é a variável de processo, $e(t)$ é o erro instantâneo. A partir dessa equação, podemos implementar uma função de minimização, que está no código 1.

Código 1 – Minimização da função do controlador PID

```

1 funcao(Kp, Ti, Td, n, dados_objetivo, y_original, soma_erro) :
2     erro = referencia - dados_objetivo[n]
3     se (n>0) :
4         soma_erro += erro
5         derivada = (dados_objetivo[n] - dados_objetivo[n-1])/dt
6         saida_controlador = Kp * erro + (Kp/Ti)*soma_erro*dt - Kp*Td*derivada
7     se nao :
8         saida_controlador = 0
9     n += 1
10    retorna saida_controlador - y_original

```

Onde n indexa as interações, $dado_teste$ é o valor pela da RNA, $y_original$ é saída do controlador não sintonizado, $dados_objetivo$ é o sinal da resposta objetivo. Com isso, conseguimos extrair os novos parâmetros do controlador.

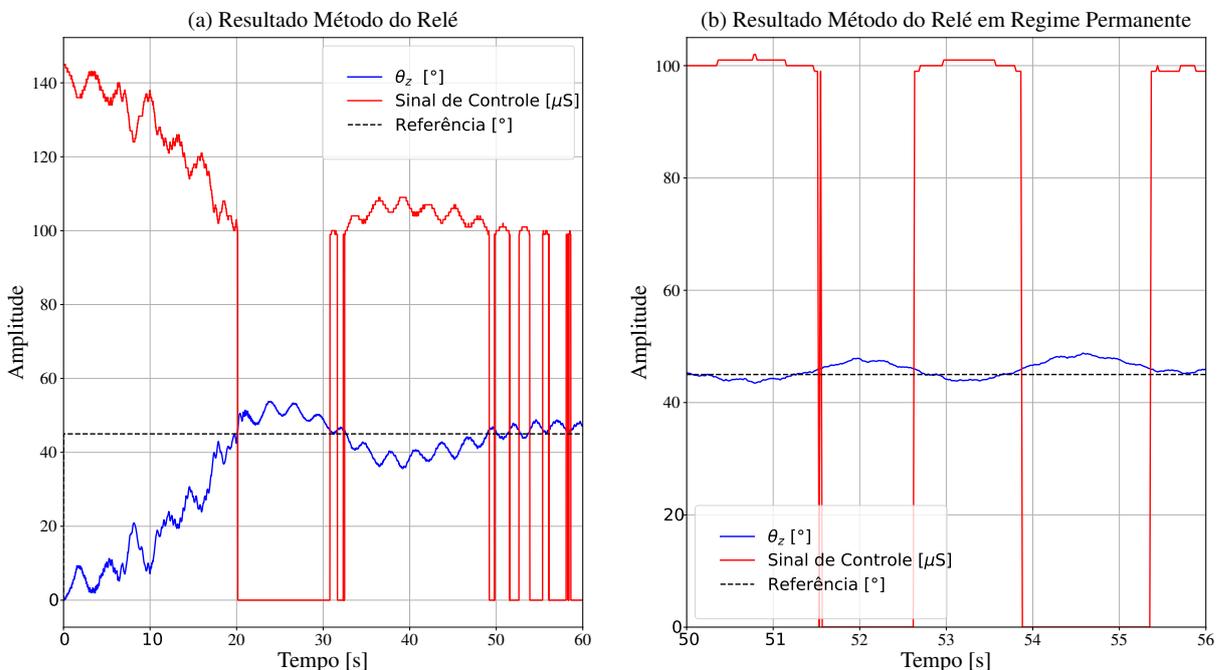
4 ANÁLISE DOS RESULTADOS

Nesse capítulo apresenta-se os principais resultados e as análises sobre eles, onde serão demonstrados os resultados pelos métodos clássicos de sintonia, o desenvolvido nesse trabalho e por fim, uma comparação entre todos.

4.0.1 Métodos Clássicos de Sintonia

Dentre eles, se encontram os métodos de sintonia de Ziegler-Nichols em malha fechada e o Método do Relé. O método de sintonia em malha aberta não pode ser aplicado nesse trabalho, pois em malha aberta, o sistema é instável, ou seja, ao colocarmos uma determinada largura de pulso e não modularmos ela, o simulador de satélite simplesmente ficará girando descontroladamente, pois não existem forças no eixo z para pararem o satélite, como acontece quando um objeto é acelerado no espaço na ausência de forças. Na figura 34 (a), podemos ver o resultado da aplicação do método do relé em um degrau de amplitude de 45° , valor esse, usado em todos os ensaios.

Figura 34 – Resultados do método do relé.



Fonte – Elaborado pelo Autor.

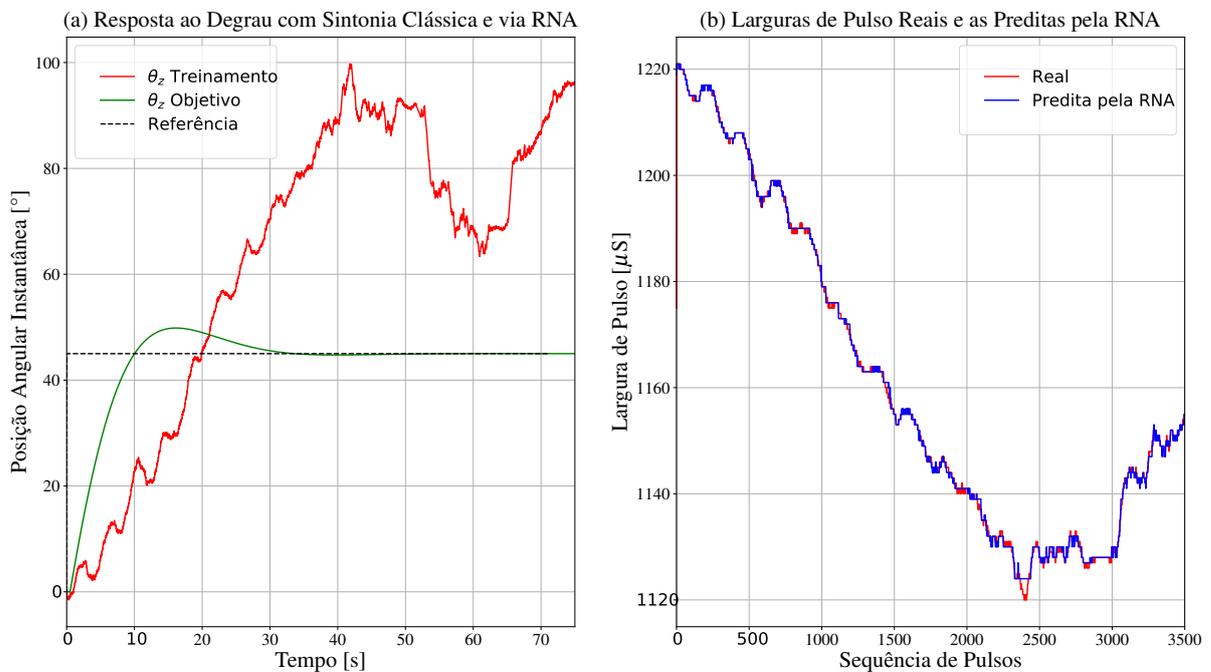
Já na figura 34 (b), podemos ver com maiores detalhes a resposta em regime do sistema, onde encontramos parâmetros do controlador PID, aplicando a equação 55 e tabela 2. Após

encontrar os valores para esse método, o próximo passo é usar o método com RNA e regressão não-linear.

4.0.2 Método de Sintonia Automático usando RNA e Regrão Não-Linear Robusta

Primeiro passo para a sintonia via RNA e regressão, é treina-la com dados válidos, onde ela aprende o comportamento do controlador e da planta. A criação dos dados para o treinamento foi realizado usando um controlador com simplesmente um ganho proporcional unitário, ou seja, a malha foi fechada usando um controlador PID com $P=1$, $I=0$ e $D=0$. O sinal usado para treinar a RNA (θ_z Treinamento) pode ser visto na figura 35 (a), juntamente com o sinal objetivo para a regressão (θ_z Objetivo)

Figura 35 – Saída da RNA treinada e o sinal real.



Fonte – Elaborado pelo Autor.

Ainda, podemos observar na figura 35 (a), onde podemos ver que o satélite oscila durante o regime transitório, como se acabasse retornando um pouco a cada novo avanço, isso é devido à rápida variação do sinal de controle que faz com que os motores perturbem o corpo com um torque contrário ao movimento, pois, a inércia do satélite é baixa e o atrito é desprezível devido ao mancal a ar. Já na figura 35 (b), podemos ver o sinal do controlador predito pela RNA, juntamente com o sinal real amostrado no simulador de satélites. Com esses últimos dados, é possível se calcular o RMSE (*Root Mean Square Error* - Erro quadrático médio) da seguinte

forma:

$$RMSE = \sqrt{\frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{T}} \quad (73)$$

Onde T o número de amostras, \hat{y}_t são os valores preditos pela RNA e y_t são os valores reais. Assim obtemos o seguinte RMSE para a rede treinada:

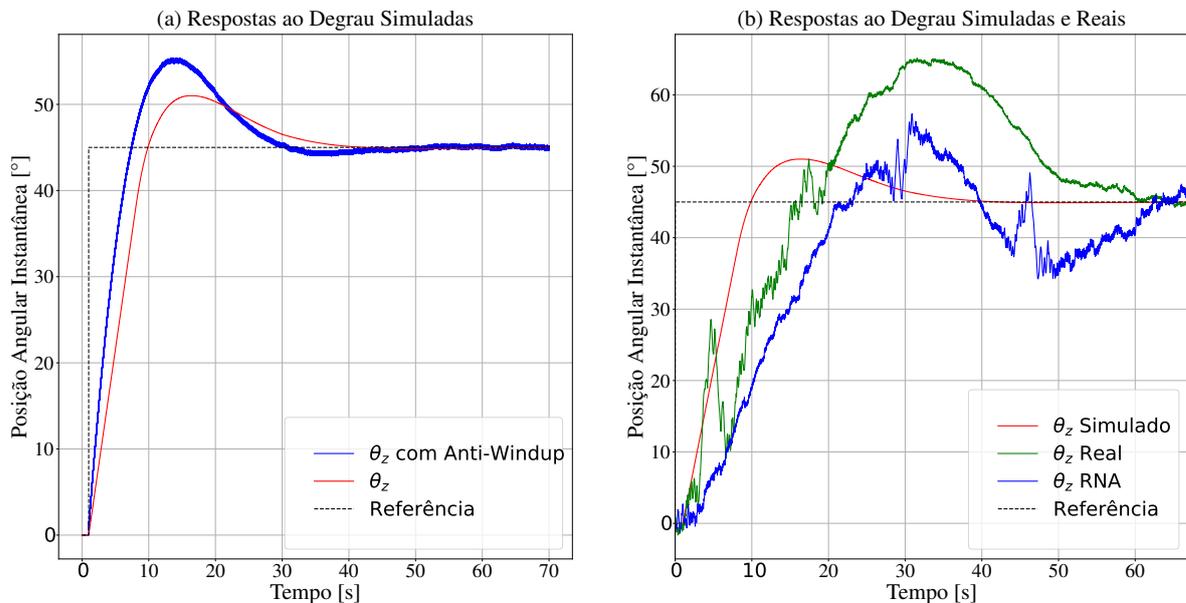
$$RMSE \approx 1.0112 \quad (74)$$

O que é muito satisfatório, indicando que a RNA conta com um bom número de neurônio, número de camadas e interações. Após esses passos, podemos comparar todos os resultados, para uma análise mais objetiva.

4.0.3 Respostas ao degrau

Após todos os métodos de sintonia aplicados, com foco no método do relé e no desenvolvido nesse trabalho, podemos comparar as respostas ao degrau. A figura 36 (a) representa os resultados das simulações feitas na ferramenta Simulink, onde podemos ver os resultados da sintonia do controlador PID simples e com Anti-Windup.

Figura 36 – Respostas ao degrau com sintonia via método do relé.



Fonte – Elaborado pelo Autor.

Já na figura 36 (b), podemos ver uma comparação entre o melhor resultado das simulações, ou seja, o controlador PID com Anti-Windup juntamente com o sinal real, também com

Anti-Windup e sintonizado via RNA e regressão não-linear. Já a tabela 6, representa os valores dos parâmetros do controlador das respostas da figura 36, anteriormente apresentada.

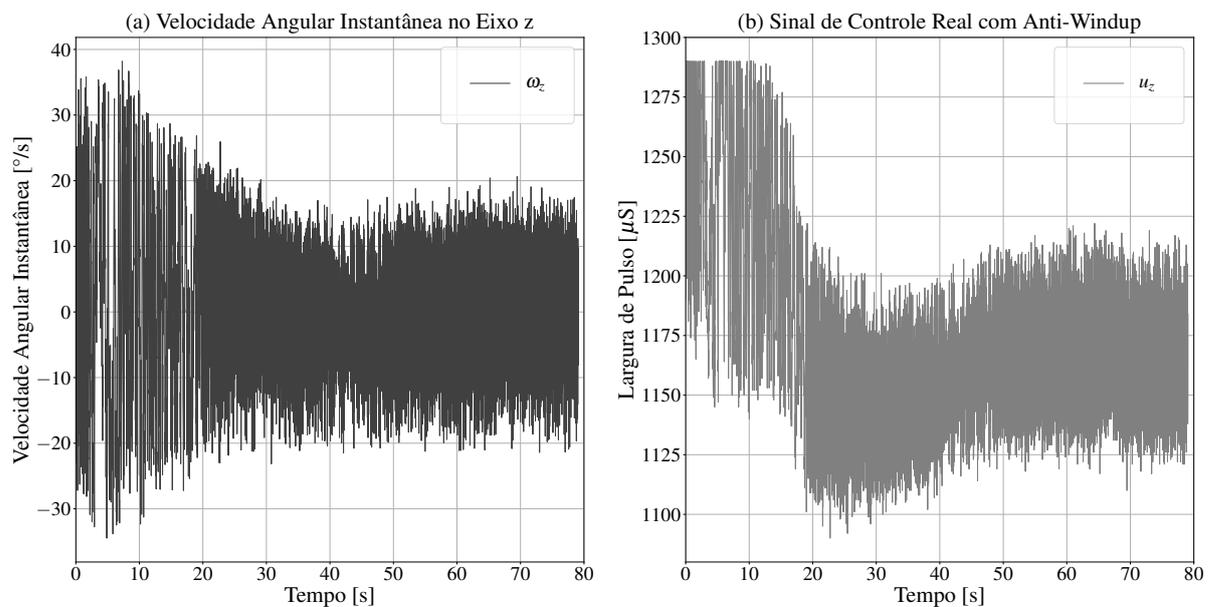
Tabela 6 – Parâmetros dos Controladores

Método	K_p	K_i	K_d
Método do Relé	1.2	10	2.5
RNA + Regressão Não-Linear	1.2	2	0.25

Fonte – Elaborado pelo Autor.

Como podemos ver, o sinal proveniente da simulação é mais rápido e com menor sobressinal. O sinal real com Anti-Windup apresentou o pior desempenho entre os três, pois teve o maior tempo de estabilização e sobressinal. O comportamento mais lento, maior sobressinal e mais ruidoso, se justifica pela presença de forças e perturbações que o modelo teórico não prevê. Alguns problemas práticos que fazem o comportamento se desviar em relação ao teórico, são: a não perfeita simetria do simulador de satélites; perturbações pelo fluxo de ar no mancal, vibração excessiva dos motores, o que dificulta a aquisição de dados. Essa última pode ser vista na figura 37 (a), onde podemos ver a velocidade angular em z da resposta real ao degrau, com um controlador PID com Anti-Windup. Percebe-se a grande presença de ruídos, muito mais do que pode ser visto na figura 32 (a), onde o simulador está estático.

Figura 37 – Velocidade angular instantânea e sinal de controle.



Fonte – Elaborado pelo Autor.

Por consequência, o sinal do controlador também apresentará grande quantidade de ruído,

o que pode ser visto na figura 37 (b), o que dificulta muito o controle. Sem a presença do filtro de Kalman, o controle não funcionaria de maneira adequada.

Desde o desenvolvimento mecânico, que exigiu a fabricação de todas as peças e além disso, ajustes para o perfeito funcionamento da dinâmica do simulador de satélites, os resultados foram funcionais e satisfatórios. O sistema de controle conseguiu manter uma performance que não influenciou nos resultados, mesmo que executado em um sistema operacional, que possui escalonamento de tarefas, o que pode tornar a execução das tarefas não periódicas em muitos casos. E por fim, os resultados do método de sintonia desenvolvido usando RNA e regressão não linear, que otimizou de forma também satisfatória a resposta em relação aos métodos clássicos de sintonia.

5 CONCLUSÃO

Na primeira parte do trabalho, foram abordados os princípios da dinâmica de um satélite, de técnicas clássicas e modernas para a sintonia de controladores. Foram descritos conceitos como rodas de reação, dinâmica de corpo rígido, controladores PID, método do relé, redes neurais artificiais, entre outros. O referencial descreve principalmente as possibilidades de sintonia de controladores e a modelagem de um satélite artificial.

A implementação do protótipo foi dividida em 3 partes, uma foi o desenvolvimento do *hardware*, outra o *software*, e ainda o sistema de controle, levando em consideração a melhora na resposta ao degrau e da dinâmica do satélite através da sintonia adequada do controlador. Na análise de resultados, pôde-se observar a melhora na resposta transitória do simulador de satélites, mesmo com perceptível presença de ruídos.

Ainda na análise de resultados, ficou clara a interferência nos resultados pela forte presença de ruídos nos sinais oriundos do giroscópio, provocados pela vibração dos motores e da própria física do giroscópio, pois, mesmo parado, apresenta ruídos estocásticos. Também é perceptível uma oscilação durante o movimento de rotação do satélite, esse movimento indesejado fica cada vez menor ao passo que o sinal de controle é menor, ou seja, uma rápida variação do sinal de controle, faz com que o corpo oscile devido à baixa inércia do simulador em comparação ao torque dos motores.

Através desse trabalho, ficou clara a possibilidade de se empregar técnicas modernas em controle, onde uma RNA aprendeu o comportamento de dinâmica de um simulador de satélites, sendo possível se aplicar em diversas outras plantas. Assim, nesse trabalho, foi possível atingir os objetivos propostos, salvo melhorias que podem ser desenvolvidas em trabalhos futuros.

Trabalhos futuros a partir desse podem ser desenvolvidos, principalmente com a substituição do giroscópio simples por um sensor inercial absoluto. Esse outro sensor conta com magnetômetro, barômetro, acelerômetro, giroscópio e uma unidade que processa os sinais e entrega as grandezas sem comprometer o processamento do controlador, além de uma ótima exatidão e blindagem maior ao ruídos provenientes dos motores. Ainda, é desejável se criar uma interface gráfica para facilitar o usuário operar e configurar o simulador de satélites, com o objetivo de aprendizagem de conceitos de controle.

REFERÊNCIAS BIBLIOGRÁFICAS

- ÅSTRÖM, K. J.; HÄGGLUND, T. *PID ontrollers: Theory, Design and Tuning*. [S.l.: s.n.], 1995. Citado 6 vezes nas páginas 26, 27, 31, 32, 33 e 35.
- Bong Wie. *Space vehicle dynamics and control*. [S.l.: s.n.], 2001. v. 37. 2077–2078 p. Citado 4 vezes nas páginas 24, 28, 29 e 30.
- BROWN, C. D. *Elements of Spacecraft Design*. [S.l.: s.n.], 2002. Citado na página 17.
- CHEN, J.; HUANG, T. C. Applying neural networks to on-line updated PID controllers for nonlinear process control. *Journal of Process Control*, 2004. Citado 2 vezes nas páginas 35 e 36.
- LEVINE, W. S. *The Control Handbook*. [S.l.]: CRC-Press, 1996. 83 p. Citado 6 vezes nas páginas 24, 25, 26, 30, 33 e 34.
- MAYOROV, N. *Robust nonlinear regression in scipy*. 2015. Disponível em: <https://scipy-cookbook.readthedocs.io/items/robust_regression.html#robust-nonlinear-regression-in-scipy>. Citado na página 37.
- OGATA, K.; HALL, P. *Modern Control Engineering*. [S.l.: s.n.], 2010. Citado 3 vezes nas páginas 26, 31 e 46.
- Riazoshams, H., Midi, H., & Ghilagaber, G. *Robust Nonlinear Regression with Applications Using R*. [S.l.: s.n.], 2018. Citado na página 36.
- SNIDER, R. E. *Attitude Control of a Satellite Simulator Using Reaction Wheels and a PID Controller*. Tese (Doutorado), 2010. Citado 5 vezes nas páginas 18, 19, 20, 26 e 27.
- ÜNAL, M. et al. Optimization of PID Controllers Using Ant Colony and Genetic Algorithms. 2013. Citado 2 vezes nas páginas 34 e 35.