

Bruno Caixeta Piazza - 11260551  
Gustavo Nunes Ribeiro - 11262212  
Murilo Camargo Marchioni - 11260717  
Vinícius Maia Neto - 11261785

**PME3481 - Controle e Aplicações**  
**Controle da atitude de um *CubeSat* 3U - PicSat**  
**Grupo 3**

São Paulo  
12 de julho de 2022

Bruno Caixeta Piazza - 11260551  
Gustavo Nunes Ribeiro - 11262212  
Murilo Camargo Marchioni - 11260717  
Vinícius Maia Neto - 11261785

**PME3481 - Controle e Aplicações**  
**Controle da atitude de um *CubeSat* 3U - PicSat**  
**Grupo 3**

Relatório contendo o controle da atitude de  
um *CubeSat* e análises por abordagens de  
Controle Moderno e Clássico.

Universidade de São Paulo  
Escola Politécnica  
Departamento de Engenharia Mecânica

Orientador: Prof. Dr. Agenor de Toledo Fleury

São Paulo  
12 de julho de 2022

# Listas de ilustrações

Figura 1 – Ilustrações do modelo a ser simulado . . . . .	5
Figura 2 – Módulo de sensores e atuadores iADCS-100. . . . .	7
Figura 3 – Sistema de Coordenadas Geocêntrico Equatorial Inercial (GEI). . . . .	8
Figura 4 – Sistema de Coordenadas Referente ao Corpo do Satélite (RCS). . . . .	8
Figura 5 – Representação do PicSat com os eixos principais de inércia evidenciados.	9
Figura 6 – Relação entre o eixo inercial, GEI, e o eixo móvel, RCS. . . . .	9
Figura 7 – Representação do Eixo de Euler. . . . .	10
Figura 8 – Modelo mecânico da roda de reação. . . . .	12
Figura 9 – Torques de perturbação aplicados ao <i>PicSat</i> . . . . .	13
Figura 10 – Resposta do primeiro quatérnio a pulso - Malha Aberta. . . . .	18
Figura 11 – Resposta em frequência - Malha Aberta. . . . .	19
Figura 12 – $\omega_2(0) = 0,15 \text{ rad/s}$ - regulador por Alocação de Polos. . . . .	22
Figura 13 – $\omega_1(0) = \omega_2(0) = \omega_3(0) = 0,15 \text{ rad/s}$ - regulador por Alocação de Polos.	23
Figura 14 – $\omega_2(0) = 0,15 \text{ rad/s}$ - regulador LQR. . . . .	24
Figura 15 – $\omega_1(0) = \omega_2(0) = \omega_3(0) = 0,15 \text{ rad/s}$ - regulador LQR. . . . .	25
Figura 16 – Estabilização do erro para observador de estado via Alocação de Polos.	26
Figura 17 – Estabilização do erro para observador de estado LQR. . . . .	27
Figura 18 – Evolução temporal do sistema - seguidor com referência constante. . .	29
Figura 19 – Evolução temporal dos ângulos de Euler - seguidor com referência constante. . . . .	30
Figura 20 – Evolução do estado do sistema ao longo do tempo. . . . .	31
Figura 21 – Torques de distúrbio ao longo do tempo . . . . .	32
Figura 22 – Fonte: autoria própria. . . . .	32
Figura 23 – Diagrama de polos e zeros para o sistema em malha aberta. . . . .	34
Figura 24 – Resposta do sistema frente a um degrau unitário - Malha Aberta. . .	34
Figura 25 – Sistema sob ganho crítico. . . . .	36
Figura 26 – Resposta do sistema ao degrau com PID sintonizado por <i>Ziegler-Nichols</i>	37
Figura 27 – Posição angular $q_3$ para degrau aplicado na entrada $T_{rr3}$ - alocação de polos . . . . .	39
Figura 28 – Posição angular $q_3$ para degrau aplicado na entrada $T_{rr3}$ - ITAE. . . . .	40
Figura 29 – Diagramas de lugar das raízes para a sintonia do ganho proporcional .	42
Figura 30 – Diagramas de lugar das raízes para a sintonia do ganho integrativo .	42
Figura 31 – Diagramas de lugar das raízes para a sintonia do ganho derivativo .	43
Figura 32 – Sobreposição das respostas a degrau com PID's sincronizados por lugar das raízes para sistemas com e sem compensador . . . . .	43
Figura 33 – Análise de estabilidade - <i>Ziegler-Nichols</i> . . . . .	44

Figura 34 – Análise de estabilidade - Alocação de Polos. . . . .	45
Figura 35 – Análise de estabilidade - ITAE. . . . .	45
Figura 36 – Análise de estabilidade - Lugar das Raízes. . . . .	46
Figura 37 – Seguidor de referência. . . . .	47
Figura 38 – Seguidor de referência. . . . .	47
Figura 39 – Comparação entre métodos de sintonia PID. . . . .	48
Figura 40 – Comparação de respostas a degrau de diferentes métodos de sintonia PID. . . . .	50
Figura 41 – Comparação entre os diagramas de bode para lugar das raízes com e sem ordem reduzida. . . . .	50

# **Lista de tabelas**

Tabela 1 – Parâmetros físicos da estrutura utilizados na modelagem simulação . . . . .	6
Tabela 2 – Polos alocados - regulador por Alocação de Polos. . . . .	21
Tabela 3 – Polos obtidos - regulador LQR. . . . .	24
Tabela 4 – Polos alocados para o Observador de Estado - regulador via Alocação de Polos. . . . .	26
Tabela 5 – Polos para o Observador de Estado - regulador LQR. . . . .	27
Tabela 6 – Polos alocados - PID. . . . .	38

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
<b>1.1</b>	<b>Revisão Bibliográfica</b>	<b>1</b>
<b>1.2</b>	<b>Objetivos</b>	<b>2</b>
<b>1.3</b>	<b>Metodologia</b>	<b>2</b>
1.3.1	Desenvolvimento do Modelo	2
1.3.2	Desenvolvimento do Controle	3
<b>1.4</b>	<b>Hipóteses Simplificadoras</b>	<b>3</b>
1.4.1	Restrição do estudo ao movimento rotacional do satélite	4
1.4.2	Momentos principais de inércia invariantes no tempo	4
1.4.3	O sistema de coordenadas referente ao corpo do satélite coincide com os eixos principais de inércia	4
1.4.4	O referencial inercial utilizado é o Sistema de Coordenadas Geocêntrico Equatorial Inercial (SCGEI)	4
1.4.5	Torque giroscópico das rodas de reação desprezível	4
<b>2</b>	<b>MODELO</b>	<b>5</b>
<b>2.1</b>	<b>Modelo físico</b>	<b>5</b>
2.1.1	Estrutura	5
2.1.2	Sensores e atuadores	6
<b>2.2</b>	<b>Sistemas de coordenadas</b>	<b>7</b>
2.2.1	Sistema de Coordenadas Geocêntrico Equatorial Inercial	7
2.2.2	Sistema de Coordenadas Referente ao Corpo do Satélite	8
2.2.3	Sistema de Coordenadas Referente aos Eixos Principais de Inércia	8
<b>2.3</b>	<b>Representação de atitude</b>	<b>9</b>
2.3.1	Ângulos de Euler	9
2.3.2	Eixo de Euler, Parâmetros Simétricos de Euler e Quaternios	10
<b>2.4</b>	<b>Modelo matemático</b>	<b>11</b>
2.4.1	Modelagem Cinemática	11
2.4.2	Modelagem Dinâmica	11
2.4.3	Modelagem de distúrbios	13
2.4.4	Sistema de equações do modelo	15
<b>2.5</b>	<b>Linearização</b>	<b>15</b>
<b>2.6</b>	<b>Representação em Espaço de Estados</b>	<b>16</b>
<b>2.7</b>	<b>Análise no domínio do tempo</b>	<b>17</b>
<b>2.8</b>	<b>Análise no domínio da frequência</b>	<b>18</b>

<b>3</b>	<b>CONTROLE MODERNO</b>	<b>20</b>
3.1	Análise de controlabilidade e observabilidade	20
3.2	Controlador por Alocação de Polos	21
3.3	Controlador Linear Quadrático	23
3.4	Observadores de estados	25
3.4.1	Observador de Estado para o regulador via Alocação de Polos	25
3.4.2	Observador de Estado para o Controlador Linear Quadrático	27
3.5	Seguidor com referência constante	28
3.6	Seguidor com referência móvel e perturbações	30
<b>4</b>	<b>CONTROLE CLÁSSICO</b>	<b>33</b>
4.1	Função de transferência em malha aberta	33
4.2	Análise de polos e zeros	34
4.3	Resposta em malha aberta	34
4.4	Sintonia PID	35
4.4.1	Métodos de Ziegler-Nichols	35
4.4.2	Método de alocação de polos	37
4.4.3	Método de controle ótimo - ITAE	39
4.4.4	Método do Lugar das Raízes ( <i>Root Locus</i> )	41
4.5	Análise da estabilidade relativa e resposta em frequência	44
4.5.1	Sistema sintonizado por <i>Ziegler-Nichols</i> (ganho crítico)	44
4.5.2	Sistema sintonizado por Alocação de Polos	44
4.5.3	Sistema sintonizado por controle ótimo ITAE	45
4.5.4	Sistema sintonizado por Lugar das Raízes	46
4.6	Seguidor de referência e rejeitador de distúrbio	46
4.6.1	Seguidor de referência	46
4.6.2	Rejeitador de distúrbio	47
4.7	Comparação dos controladores	48
4.8	Redução de ordem	49
<b>5</b>	<b>CONCLUSÃO</b>	<b>51</b>
	<b>REFERÊNCIAS</b>	<b>52</b>
<b>A</b>	<b>OBSERVADOR DE ESTADOS POR ALOCAÇÃO DE POLOS</b>	<b>53</b>
<b>B</b>	<b>SEGUIDOR DE REFERÊNCIA CONSTANTE: MATRIZES</b>	<b>54</b>
<b>C</b>	<b>SEGUIDOR DE REFERÊNCIA VARIÁVEL: MATRIZES DE DISTÚRBIOS E REFERÊNCIAS</b>	<b>55</b>
<b>D</b>	<b>ALOCAÇÃO DE POLOS: SOLUÇÃO ANALÍTICAS DOS GANHOS</b>	<b>56</b>

<b>E</b>	<b>MODELO MATEMÁTICO: CÓDIGOS DAS SIMULAÇÕES . . . . .</b>	<b>57</b>
E.1	Modelos cinemático e dinâmico . . . . .	57
E.2	Simulação em malha aberta . . . . .	59
<b>F</b>	<b>CONTROLE MODERNO: CÓDIGOS DAS SIMULAÇÕES . . . . .</b>	<b>62</b>
F.1	Controlador por alocação de polos . . . . .	62
F.2	Controlador linear quadrático . . . . .	64
F.3	Observador de estado para o regulador via alocação de polos . . . . .	66
F.4	Observador de estado para o controlador linear quadrático . . . . .	67
F.5	Seguidor com referência constante . . . . .	69
F.6	Seguidor com referência móvel e perturbações . . . . .	71
<b>G</b>	<b>CONTROLE CLÁSSICO: CÓDIGOS DAS SIMULAÇÕES . . . . .</b>	<b>74</b>
G.1	Sintonia PID: Métodos de Ziegler-Nichols . . . . .	74
G.2	Método de alocação de polos . . . . .	77
G.3	Método de controle ótimo - ITAE . . . . .	80
G.4	Método do Lugar das Raízes ( <i>Root Locus</i> ) . . . . .	82
G.5	Comparação entre métodos . . . . .	87
G.6	Redução de ordem . . . . .	92
G.7	Compensador . . . . .	92
<b>H</b>	<b>CÓDIGOS AUXILIARES . . . . .</b>	<b>94</b>
H.1	Conversão entre ângulos de Euler e quatérnios . . . . .	94

# 1 Introdução

*CubeSats* são satélites compostos de unidades cúbicas de massa inferior a 1,3 kg e dimensões padronizadas ( $10 \times 10 \times 10$ ) cm (CHIN et al., 2017) utilizados, principalmente, em missões científicas de baixo orçamento. Por conta de seu baixo custo de fabricação, sua concepção no início dos anos 2000 significou um ganho de acessibilidade às instituições científicas, como Universidades, à exploração espacial. Desse modo, o desenvolvimento de *CubeSats* acarretou uma maior flexibilidade no acesso a missões espaciais, privilégio antes, em geral, restrito à organizações com alto aporte financeiro, a exemplo da organização americana *National Aeronautics and Space Administration (NASA)*.

Além de relativamente baratos, a possibilidade de combinar mais de uma unidade cúbica padrão no projeto de um *CubeSat* os torna versáteis, de forma que uma ampla gama de missões diferentes podem ser realizadas por esta categoria de satélites. Um exemplo é o projeto PicSat, do *Laboratoire d'Études Spatiales et d'Instrumentation en Astrophysique (LESIA)* - lançado em 12 de janeiro de 2018 - cujos objetivos eram observar efeitos de trânsito<sup>1</sup> do exoplaneta Beta Pictoris b e de cometas no brilho da estrela  $\beta$  – *Pictoris* e testar um conceito tecnológico envolvendo o uso de fibra óptica na captação de luz solar (The PicSat Team, 2016).

Para que os *CubeSats* realizem suas missões com eficiência, é necessário garantir um controle adequado da orientação do veículo. Para tal, lança-se mão de artifícios como atuadores magnéticos e rodas de reação. Neste trabalho, serão estudados os aspectos cinemáticos e dinâmicos da atitude (orientação de um corpo rígido em relação a um referencial inercial) do *CubeSat* PicSat.

## 1.1 Revisão Bibliográfica

A fonte bibliográfica deste trabalho consiste principalmente de artigos, livros-texto, dissertações e teses. Os parâmetros da modelagem física da estrutura do satélite foram retirados de publicações<sup>2</sup> de Mathias Nowak (NOWAK, 2019b)(NOWAK, 2019a) contendo informações gerais da missão PicSat.

A tese de mestrado (CROWELL, 2011) foi uma importante base que auxiliou o grupo a modelar a cinemática do satélite em torno dos ângulos de Euler, bem como ter uma visão geral da representação do problema em espaço de estados e da linearização do sistema. O referido documento trata do desenvolvimento de um equipamento voltado para

<sup>1</sup> Redução da percepção do brilho de uma estrela causada pela passagem de um corpo entre a fonte luminosa (a própria estrela) e o observador (SANTOS; AMORIM, 2017)

<sup>2</sup> As produções de Nowak foram úteis também na introdução da missão PicSat ao grupo.

o ensaio e testes de componentes do Subsistema de Controle e Determinação de Atitude (*ADCS*, da sigla em inglês) de CubeSats no *Massachusetts Institute of Technology*.

Quanto à modelagem dinâmica da estrutura do *PicSat*, a principal referência foi a dissertação de Cauê Garcia Menegaldo ([MENEGALDO, 2020](#)), que se propõe a criar um simulador de controle de atitude e órbita para nanossatélites. Esse trabalho também foi importante base para o grupo na introdução do conceito de quatérnios. Por fim, a dissertação contém estudos fundamentais acerca do controle da orientação do *PicSat*, indicando diversos requisitos, a exemplo da estabilização de apontamento, e propondo ferramentas de Controle para cumpri-los.

Finalmente, o livro ‘‘*Modern control engineering*’’ de Katsuhiko Ogata ([OGATA, 2010](#)) serviu de base em diversas fases da modelagem matemática, simulação, análise e controle do sistema.

## 1.2 Objetivos

O objetivo do projeto, de início, é refinar as modelagens cinemática e dinâmica (já feitas na disciplina *PME 3380 - Modelagem de sistemas dinâmicos*) da atitude de um satélite CubeSat composto de três unidades-padrão (3U). Sendo assim, apesar de o movimento de um satélite ao longo de sua órbita possuir seis graus de liberdade - três de translação e três de rotação - será feito um desenvolvimento apenas acerca das suas coordenadas angulares.

Pretende-se, ainda, aplicar técnicas de controle moderno e clássico com o objetivo de, não somente, estabilizar o satélite, mas também, controlar sua orientação. Tais fatores são essenciais ao cumprimento da missão, haja vista que o acompanhamento do trânsito de  $\beta$  – *Pictoris* requer precisão de apontamento e, consequentemente, de controle da atitude.

## 1.3 Metodologia

Visando a alcançar os objetivos deste trabalho, seguiu-se a metodologia abordada na disciplina. Ou seja, foram elaborados os modelos físico e matemático do sistema (o satélite *PicSat*) e, posteriormente, realizaram-se simulações numéricas e discutiram-se seus resultados. Em seguida, desenvolveram-se diversos tipos de controle.

### 1.3.1 Desenvolvimento do Modelo

O modelo físico do satélite foi construído considerando o sistema composto apenas por um único corpo rígido de propriedades materiais constantes ao longo do tempo de

forma que as variáveis de entrada sejam torques aplicados ao satélite. Nesse contexto, foram escolhidas rodas de reação como atuadores do sistema de controle desenvolvido.

Estudaram-se três sistemas de coordenadas utilizados na modelagem matemática do sistema: o Sistema de Coordenadas Geocêntrico Equatorial Inercial (GEI), o Sistema de Coordenadas Referente ao Corpo do Satélite (RSC) e o Sistema de Coordenadas Referente aos Eixos Principais (REP). A partir deles, foi possível obter as equações cinemáticas e dinâmicas envolvendo o satélite.

Uma vez definidos os sistemas de eixos coordenados a serem utilizados, foi feita a modelagem cinemática do PicSat. Neste passo, expressou-se a velocidade angular em função dos Ângulos de Euler (precessão, nutação e rotação própria) no referencial referente ao corpo do satélite e fez-se a posterior conversão para quatérnios<sup>3</sup>.

Posteriormente, deduziram-se as equações que regem a dinâmica do veículo espacial. Foi aplicado o Teorema do Momento da Quantidade de Movimento Angular, obtendo-se a aceleração angular do satélite em função do torque aplicado nele.

De posse do sistema de equações diferenciais que regem a atitude do satélite, foi feita a linearização do sistema e posterior simulação dele no software *Matlab*. De posse dos resultados das simulações, resultados referentes a estabilidade foram analisados.

### 1.3.2 Desenvolvimento do Controle

Desenvolvido o modelo dinâmico do *Cubesat*, foram verificadas a controlabilidade e a observabilidade do sistema. Após constatado que o veículo espacial é tanto controlável quanto observável, fez-se uso da técnica de realocação de polos com o objetivo de torná-lo estável. Posteriormente, empregou-se o Controlador Linear Quadrático como uma alternativa de controlador. Ademais, foi desenvolvido um observador de estados para estimar o vetor de estados. Por fim, implementaram-se seguidores de referência, especialmente importantes no contexto analisado.

Dentro da abordagem de controle clássico, foram realizadas diferentes análises de estabilidade e características do sistema, em malha aberta e fechada, bem como foram desenvolvidos controladores PID pelos diferentes métodos estudados ao longo da disciplina.

## 1.4 Hipóteses Simplificadoras

Ao longo do trabalho, foi necessário, por vezes, impor condições aos modelos, a fim de torná-los manuseáveis. Seguem, nesta seção, as hipóteses simplificadoras adotadas.

<sup>3</sup> Número hipercomplexo composto por quatro parâmetros. Apesar de pouco ilustrativos em termos de análise, o uso de quatérnios possui utilidade por facilitar o trabalho algébrico da modelagem.

#### 1.4.1 Restrição do estudo ao movimento rotacional do satélite

Uma vez que a modelagem da atitude de um *CubeSat* é de elevada complexidade, optou-se por reduzir os graus de liberdade estudados do sistema. Portanto, todo o trabalho desenvolvido refere-se exclusivamente ao movimento rotacional do veículo.

#### 1.4.2 Momentos principais de inércia invariantes no tempo

Admite-se, por hipótese, que o satélite possa ser considerado como um corpo rígido e que sua operação não afeta a distribuição de massa, de modo que a matriz de inércia e os momentos principais de inércia sejam invariantes com o tempo.

#### 1.4.3 O sistema de coordenadas referente ao corpo do satélite coincide com os eixos principais de inércia

Caso o referencial associado ao corpo do satélite não seja coincidente com os eixos principais de inércia, a descrição dinâmica e cinemática estaria condicionada a matrizes de rotação  $[Q]_{p/s}$ , que relacionam os dois sistemas. Isso implica um acoplamento ainda maior das equações diferenciais em  $\phi$ ,  $\psi$  e  $\theta$ , dificultando-se a análise do problema. Desse modo, consideram-se alinhados os sistemas de coordenadas referidos.

#### 1.4.4 O referencial inercial utilizado é o Sistema de Coordenadas Geocêntrico Equatorial Inercial (SCGEI)

Conforme será explicitado posteriormente, utiliza-se o amplamente difundido Sistema de Coordenadas GEI, referência fundamental ao mapeamento por ângulos de Euler.

#### 1.4.5 Torque giroscópico das rodas de reação desprezível

Com o desenvolvimento do Teorema da Quantidade de Movimento Angular para as rodas de reação, notam-se dois termos: o torque de controle e o termo de perturbação, atrelado ao efeito giroscópico. Para o caso do *PicSat*, esse último termo é cerca de 100 vezes menor que o primeiro ([MENEGALDO, 2020](#)), de modo que será desconsiderado.

## 2 Modelo

Para a modelagem do problema proposto, são apresentadas duas frentes. A primeira consiste na apresentação do modelo físico, com os parâmetros do sistema, incluindo os sensores e atuadores. A segunda, na descrição dinâmica do *PicSat*, a qual servirá de base para a implementação do controle.

### 2.1 Modelo físico

O modelo físico tem como intuito analisar os componentes do PicSat e realizar as devidas simplificações para facilitar a modelagem matemática.

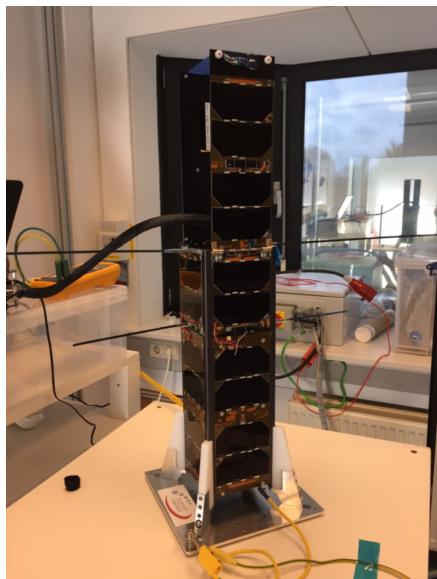
#### 2.1.1 Estrutura

Uma vez que a intenção do grupo é estudar a atitude do PicSat, optou-se por modelá-lo como um corpo único com o objetivo de obter as equações diferenciais que governam o sistema (a atitude do nanossatélite em termos de rotação).

A estrutura do PicSat comprehende a plataforma principal do satélite (as três unidades padrões de CubeSat) e todos os seus componentes internos, evidenciados na [Figura 1](#).

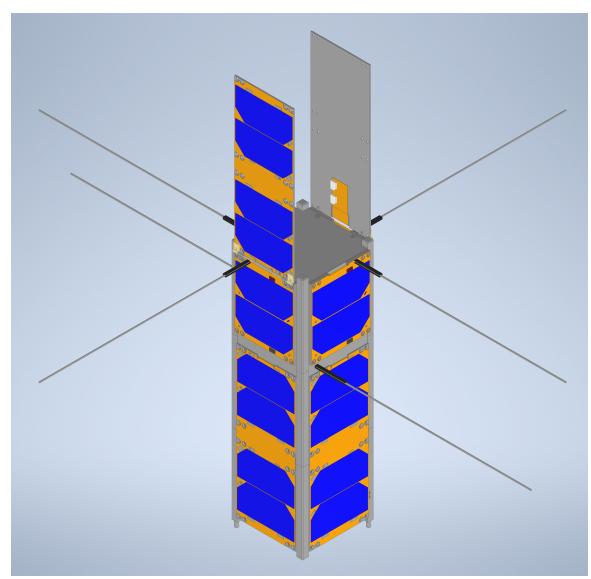
Figura 1 – Ilustrações do modelo a ser simulado

(a) Foto do PicSat em fase de testes



Fonte: ([MENEGALDO, 2020](#))

(b) Modelo em CAD do PicSat



Fonte: autoria própria.

Para efeito da modelagem física, considerou-se a estrutura como um corpo rígido de formato paralelepípedico e massa de 3,5 kg. Os parâmetros relevantes física e matematicamente para a modelagem deste componente estão evidenciados na [Tabela 1](#).

Tabela 1 – Parâmetros físicos da estrutura utilizados na modelagem simulação

Parâmetro	Valor	Fonte
Dimensão	3U (30 x 10 x 10) cm	(NOWAK, 2019a)
Massa	3,5 kg	(NOWAK, 2019b)
Matriz de Inércia	$\begin{bmatrix} 40 & 0,1 & 0,6 \\ 0,1 & 40 & -1 \\ 0,6 & -1 & 5 \end{bmatrix} 10^{-3} \text{ kg m}^2$	Professor Fábio Fialho <sup>1</sup>
Centro de Massa	(0;0;0)	Definição do referencial adotado

Fontes diversas (vide tabela).

### 2.1.2 Sensores e atuadores

A fim de se obter informações da dinâmica do sistema, são utilizados sensores. Ademais, para controlar a orientação, atuadores. A missão do PicSat contava com o módulo de controle de altitude iADCS-100, comercializado pela empresa *Hyperion Technologies*. Tal modelo é composto por:

- Três atuadores magnéticos, sendo o modelo MTQ 200.20 para os eixos X e Y e o modelo MTQ 200.10S para o eixo Z;
- Três rodas de reação modelo RW 210.60;
- Um sensor de estrelas modelo ST 200;
- Um magnetômetro de 3 eixos;
- Um girômetro de 3 eixos.

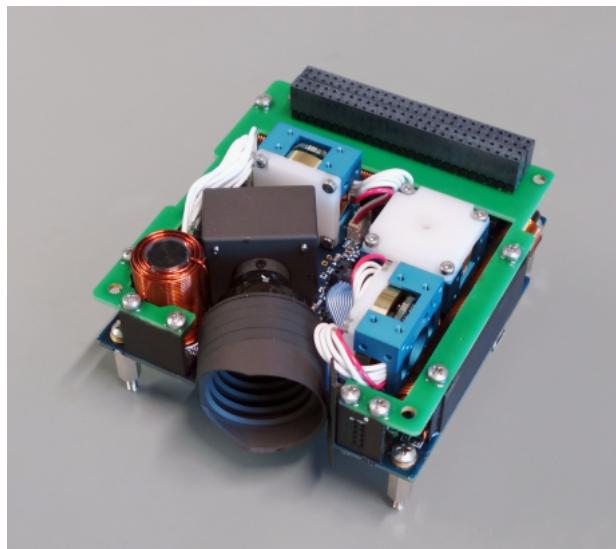
Desse modo, para o escopo deste trabalho, os atuadores magnéticos e as rodas de reação atuam sobre o sistema, fornecendo o torque de entrada. Deve-se ressaltar, no entanto, que os atuadores magnéticos apenas são ativados na fase de “*detumbling*”, isto é, quando o satélite possui uma alta velocidade angular, logo após seu lançamento, e precisa ser estabilizado. Sua análise encontra-se, portanto, fora da abordagem de controle da altitude para um apontamento, em que um satélite parte de uma posição de equilíbrio ou com

<sup>1</sup> Dados como a matriz de inércia de um satélite são confidenciais e, portanto, inacessíveis. O grupo agradece ao Professor Fábio Fialho, do departamento PTC da Escola Politécnica da USP, pelo fornecimento de uma aproximação da matriz de inércia original do PicSat (suficiente para viabilizar a modelagem e simulação do sistema).

velocidade angular reduzida. Já sobre as rodas de reação, foco do estudo a ser desenvolvido, deve-se ressaltar o torque de saturação de aproximadamente  $T_{rr} \approx 50 \text{ mN} \cdot \text{m}$ .

Em relação aos sensores, o magnetômetro de 3 eixos possibilita a obtenção do vetor campo magnético, ao passo que o sensor estelar permite a identificação de uma estrela com coordenadas conhecidas e, consequentemente, seu vetor posição. A partir desses vetores, é possível utilizar o método TRIAD (do inglês, *tri-axial attitude determination*) para determinar a atitude do satélite. Já os girômetros permitem a obtenção direta das velocidades angulares de rotação. A [Figura 2](#) ilustra o módulo utilizado.

Figura 2 – Módulo de sensores e atuadores iADCS-100.



Fonte: ([Berlin Space Technologies, 2021](#)).

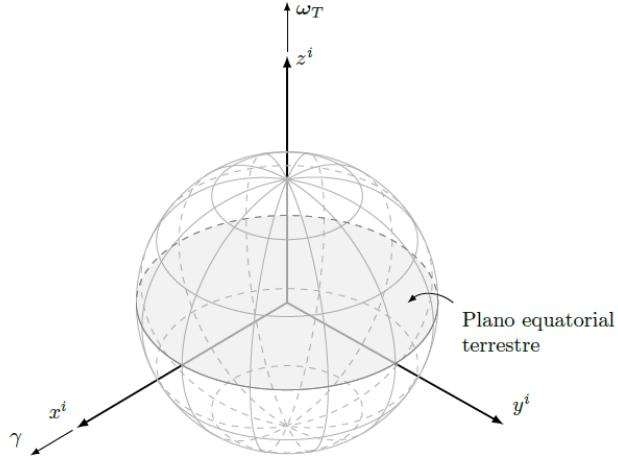
## 2.2 Sistemas de coordenadas

Para as descrições cinemática e dinâmica, será usado o sistema de coordenadas associado aos eixos principais de inércia, considerado coincidente com o referencial do satélite.

### 2.2.1 Sistema de Coordenadas Geocêntrico Equatorial Inercial

O Sistema de Coordenadas Geocêntrico Equatorial Inercial (GEI) é um dos sistemas inerciais (ou seja, sem aceleração) mais utilizados no ramo de Mecânica Celeste, uma vez que é definido em relação à Terra. A notação  $i$  indicará, no escopo deste trabalho, que a grandeza estará descrita no sistema GEI. Conforme ilustra a [Figura 3](#), o eixo  $x^i$  está na direção do equinócio de março J2000.0, representado pela letra  $\gamma$ , o eixo  $z^i$  está no sentido norte do eixo de rotação terrestre e o eixo  $y^i$  completa o sistema dextrogiro. A origem de GEI é o centro de massa da Terra, de modo que  $x^i y^i$  é um plano coincidente com o Equador.

Figura 3 – Sistema de Coordenadas Geocêntrico Equatorial Inercial (GEI).

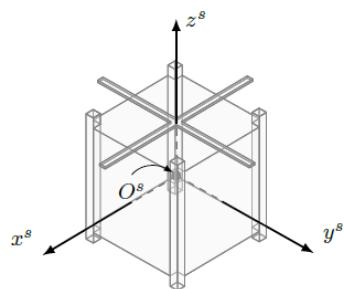


Fonte: ([MENEGALDO, 2020](#)).

## 2.2.2 Sistema de Coordenadas Referente ao Corpo do Satélite

Um sistema não-inercial utilizado é Sistema de Coordenadas Referente ao Corpo do Satélite (RCS), denotado por  $s$ . O sistema é solidário à espaçonave e definido convenientemente conforme as demandas da missão. Usualmente, define-se a origem como o centro geométrico do satélite e o eixo  $z^s$  apontando para a face do CubeSat que contém as antenas e/ou painéis solares. O eixo  $x^s$  é normal a uma das faces laterais e o eixo  $y^s$  completa o sistema dextrogiro. A [Figura 4](#) ilustra os eixos associados aos RCS.

Figura 4 – Sistema de Coordenadas Referente ao Corpo do Satélite (RCS).



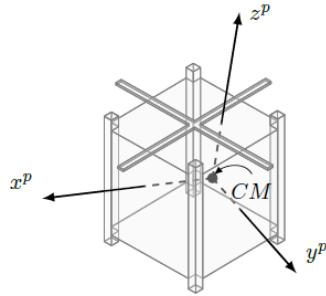
Fonte: ([MENEGALDO, 2020](#)).

## 2.2.3 Sistema de Coordenadas Referente aos Eixos Principais de Inércia

Para efeitos de simplificação da modelagem, é conveniente a representação em relação ao Sistema de Coordenadas Referente aos Eixos Principais de Inércia (REP), denotado por  $p$ , cuja origem é o baricentro do veículo. Adota-se  $z^p$  no sentido do maior momento principal de inércia,  $x^p$ , do menor e  $y^p$ , do intermediário. A [Figura 5](#) ilustra o REP em uma representação do PicSat. Conforme detalhado na [subseção 1.4.3](#), RCS e

REP são tomados como coincidentes e, portanto, passar-se-á a utilizar apenas a notação  $p$  para indicá-los.

Figura 5 – Representação do PicSat com os eixos principais de inércia evidenciados.



Fonte: ([MENEGALDO, 2020](#)).

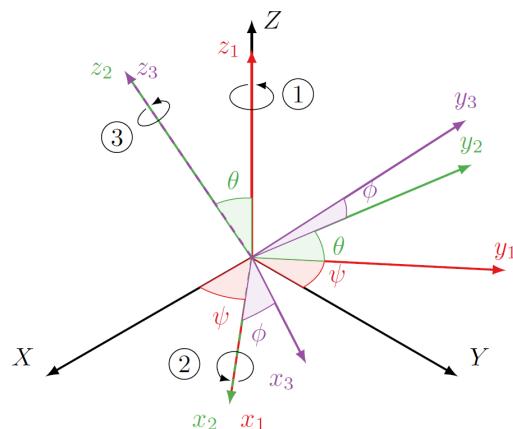
## 2.3 Representação de atitude

A atitude é definida como a orientação de um corpo rígido em relação a um referencial inercial. Diversas são as formas de representá-la, senda as abordadas na análise feita: ângulos de Euler, eixo de Euler e quatérnios.

### 2.3.1 Ângulos de Euler

Para a representação por ângulos de Euler, realizou-se a sequência clássica dos Ângulos de Euler  $Z X Z$  de rotação do RCS em torno do GEI: rotação de ângulo  $\psi$  em torno de  $Z$  (**ângulo de precessão**), rotação de  $\theta$  em torno de  $x_1$  (**ângulo de nutação**) e rotação de  $\phi$  em torno de  $z_2$  (**ângulo de rotação própria**). O resultado é ilustrado pela Figura 6.

Figura 6 – Relação entre o eixo inercial, GEI, e o eixo móvel, RCS.



Fonte: ([MENEGALDO, 2020](#)).

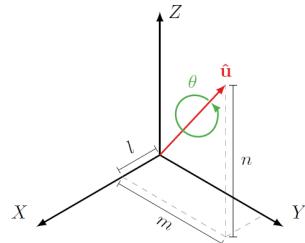
Desse modo, a sequência de rotações anteriormente proposta mapeia as rotações necessárias para se ir de GEI a RCS, cujas matrizes de cossenos diretores são

$$[Q]_{s/i} = [Q_{z_2}(\phi)][Q_{x_1}(\theta)][Q_{Z^i}(\psi)] \quad (2.1)$$

### 2.3.2 Eixo de Euler, Parâmetros Simétricos de Euler e Quatérnios

Leonhard Euler propôs que, de modo análogo ao fato de que a relação entre dois eixos distintos pode ser feita por três rotações elementares sucessivas, é possível relacionar dois sistemas de coordenadas por uma única rotação em torno de um único eixo, o qual passa pela origem comum desses sistemas, conforme a [Figura 7](#). O eixo  $\hat{u}$  é denominado Eixo de Euler e  $\theta$  indica a rotação realizada.

Figura 7 – Representação do Eixo de Euler.



Fonte: ([MENEGALDO, 2020](#)).

Uma abordagem similar que pode ser feita é denominada Parâmetros Simétricos de Euler, na qual não há dependência trigonométrica e, assim, torna-se conveniente a implementação de sucessivas rotações. Essa representação é denominada de quatérnios.

O quatérnio é um número hipercomplexo composto por quatro parâmetros,

$$\begin{aligned} \mathbf{q} &= q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} \\ &= q_0 + \mathbf{q}, \end{aligned} \quad (2.2)$$

onde  $q_0$  representa a parte escalar e  $\mathbf{q}$  a parte vetorial ou imaginária. Eles podem ser relacionados aos parâmetros  $\theta$  e  $\hat{u}$ , definidos acima, pelas equações

$$\begin{aligned} q_0 &= \cos \frac{\theta}{2} \\ \mathbf{q} &= \sin \frac{\theta}{2} \hat{u}. \end{aligned} \quad (2.3)$$

Por fim, a matriz de rotação  $[\mathbf{Q}]_{s/i} \equiv [Q]_{p/i}$  é dada, em função dos quatérnios, por

$$[\mathbf{Q}]_{p/i} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.4)$$

## 2.4 Modelo matemático

A descrição do movimento do *CubeSat* está associada a diversos conceitos atrelados ao objeto de estudo em si e ao meio no qual ele se encontra. Primeiramente, serão apresentados os sistemas de coordenadas necessários às descrições cinemática e dinâmica. Em seguida, os equacionamentos cinemático e dinâmico serão desenvolvidos. Por fim, o modelo matemático não linear será apresentado, com sua posterior linearização.

### 2.4.1 Modelagem Cinemática

Tendo em vista os diferentes métodos de representação da atitude, opta-se pelo desenvolvimento das equações diferenciais utilizando quatérnios. Tal escolha se deve aos fatos de os quatérnios não apresentarem singularidades, além de apresentarem custo computacional menor, quando comparado aos métodos que valem-se de funções trigonométricas. No entanto, a relação dos quatérnios com o mundo real é de difícil interpretação. Desse modo, o desenvolvimento matemático será feito em quatérnios, mas as análises serão concretizadas após transformação para ângulos de Euler.

Conforme (SIDI, 2012), a derivada temporal do quatérnio permite relacionar a dinâmica com a cinemática do satélite. Adotando  $\omega_{p/i}^p = [\omega_1 \ \omega_2 \ \omega_3]^\top$  como a velocidade angular do REP em relação ao GEI, tem-se,

$$\frac{d}{dt} \mathbf{q}_{p/i} = \frac{1}{2} [\Omega] \mathbf{q}_{p/i} \quad (2.5)$$

sendo,

$$[\Omega] = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \quad (2.6)$$

### 2.4.2 Modelagem Dinâmica

A modelagem dinâmica da espaçonave é obtida através da aplicação do Teorema da Quantidade de Movimento Angular (TQMA), com polo no baricentro do veículo, de modo a se obter

$$\mathbf{T}^p = [\mathbf{I}_v]^p \dot{\boldsymbol{\omega}}_{p/i}^p + \boldsymbol{\omega}_{p/i}^p \times ([\mathbf{I}_v]^p \boldsymbol{\omega}_{p/i}^p) \quad (2.7)$$

Na expressão acima, o sobrescrito  $p$  indica que todos os vetores estão expressos em coordenadas do REP.  $[\mathbf{I}_v]^p$  é a matriz diagonal de inércia do veículo no sistema REP,

$$[\mathbf{I}_v]^p = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \quad (2.8)$$

em que  $I_1 > I_2 > I_3$ .  $\mathbf{T}^p$  é o torque aplicado ao satélite, que pode ser associado aos atuadores ou a perturbações internas e externas do veículo, sintetizado pela expressão

$$\mathbf{T}^p = T_x \hat{i}_p + T_y \hat{j}_p + T_z \hat{k}_p \quad (2.9)$$

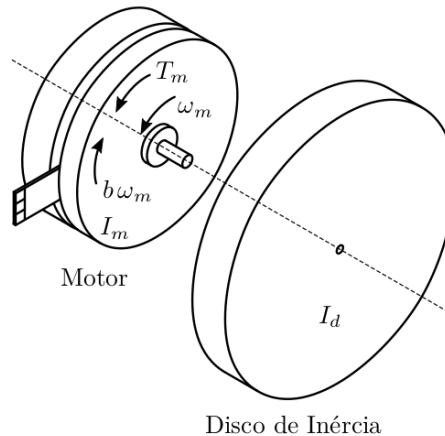
e  $\boldsymbol{\omega}_{p/i}^p$  é a velocidade angular do REP em relação ao GEI.

Desse modo, pode-se escrever o sistema de equações,

$$\begin{cases} T_x = I_3 \dot{\omega}_1 + (I_1 - I_2) \omega_2 \omega_3 \\ T_y = I_2 \dot{\omega}_2 + (I_3 - I_1) \omega_1 \omega_3 \\ T_z = I_1 \dot{\omega}_3 + (I_2 - I_3) \omega_1 \omega_2 \end{cases} \quad (2.10)$$

O torque de controle é aplicado pelas rodas de reação. Um esquema dos atuadores é explicitado na [Figura 8](#).

Figura 8 – Modelo mecânico da roda de reação.



Fonte: ([MENEGALDO, 2020](#))

O momento de inércia da roda de reação  $I_{rr}$  é dado pela soma dos momentos de inércia do motor  $I_m$  e do disco de inércia  $I_d$

$$I_{rr} = I_m + I_d. \quad (2.11)$$

Devido à relação direta da derivada do momento angular da roda de reação ( $\dot{h}_{rr}$ ) com o torque produzido por ela, como será apresentado adiante, são usadas as variáveis

$$\begin{cases} h_{rr} = I_{rr}\omega_{rr} \\ \dot{h}_{rr} = I_{rr}\dot{\omega}_{rr} \end{cases}. \quad (2.12)$$

Ao considerar a roda de reação como um corpo rígido, o torque gerado no corpo  $T_{rr}^p$  é resultado de dois torques distintos: um torque de controle  $\dot{H}_{rr}^p$  e um de perturbação  $\omega_{p/i}^p \times H_{rr}^p$ , sendo  $\omega_{p/i}^p$  a velocidade angular de REP em relação ao GEI, em razão do efeito giroscópico. Conforme a hipótese descrita na [subseção 1.4.5](#), aproxima-se:

$$\mathbf{T}_{rr}^p = -(\dot{\mathbf{H}}_{rr}^p + \boldsymbol{\omega}_{p/i}^p \times \mathbf{H}_{rr}^p) \approx -\dot{\mathbf{H}}_{rr}^p \quad (2.13)$$

Ainda, para modelar o torque resultante das rodas de reação sobre o *CubeSat*, é necessário definir a disposição geométrica dos atuadores. No caso, adota-se um alinhamento destes com os eixos principais do corpo do satélite, de modo que

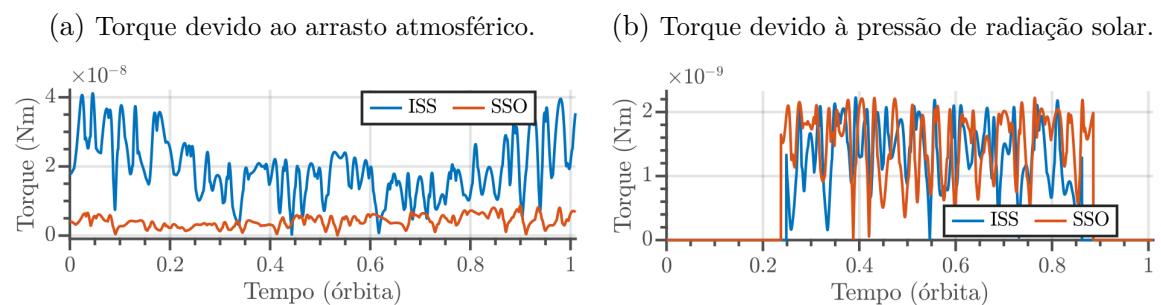
$$\mathbf{T}_{rr}^p = -\begin{bmatrix} I_{rr_1} \dot{\omega}_{rr_1}^{rr} \\ I_{rr_2} \dot{\omega}_{rr_2}^{rr} \\ I_{rr_3} \dot{\omega}_{rr_3}^{rr} \end{bmatrix} = -\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{h}_z \end{bmatrix}, \quad (2.14)$$

no qual  $h_x = h_{rr_1}$  refere-se ao momento angular da roda de reação alinhada ao eixo  $x$ . A roda  $rr_2$  foi considerada alinhada ao eixo  $y$  e  $rr_3$ , ao eixo  $z$ .

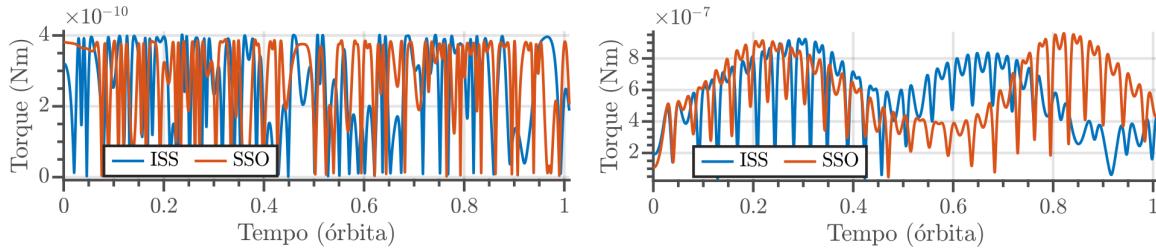
### 2.4.3 Modelagem de distúrbios

Conforme ([MENEGALDO, 2020](#)), as perturbações de altitude mais pronunciáveis numa órbita de baixa altitude (*Low Earth Orbit*, da sigla em inglês), tais como a do *PicSat*, são produzidas pelo arrasto atmosférico, pressão de radiação solar, gradiente de gravidade e dipolo magnético residual. A fim de se analisar qual o de maior relevância para implementação no modelo, usam-se as simulações feitas por Menegaldo, explicitadas na [Figura 9](#). As curvas em azul referem-se à órbita da Estação Espacial Internacional (ISS, da sigla em inglês), ao passo que as em laranja indicam uma órbita heliosíncrona, que é a do *PicSat*.

Figura 9 – Torques de perturbação aplicados ao *PicSat*.



(c) Torque devido ao gradiente de gravidade. (d) Torque devido ao dipolo magnético residual.



Fonte: ([MENEGALDO, 2020](#)).

Nota-se da [Figura 9d](#) que o distúrbio associado ao dipolo magnético residual é o fenômeno associado ao torque de perturbação com maior ordem de grandeza, com valores próximos a  $10^{-6} \text{ Nm}$ . O segundo fenômeno mais relevante, de arrasto atmosférico, acarreta um torque cerca de 20 vezes menor que aquele associado ao dipolo. Nesse sentido, justifica-se a opção por modelar o fenômeno do dipolo magnético residual.

Tal torque é oriundo da interação do campo magnético terrestre com um dipolo no satélite, o qual está associado à corrente de Foucault, histerese e outros momentos magnéticos que podem surgir devido a correntes circulantes nos componentes eletrônicos do *PicSat*. Denotando por  $\mathbf{m}_{mr}$  o dipolo magnético residual do satélite e  $\mathbf{T}_{mr}$  o torque causado pelo dipolo magnético residual, tem-se a [Equação 2.15](#),

$$\mathbf{T}_{mr}^p = \mathbf{m}_{mr}^p \times \mathbf{B}^p \quad (2.15)$$

em que  $\mathbf{m}_{mr} = [-0,14 \quad -0,12 \quad -19,32]^\top 10^{-3} \text{ A m}^2$  é o dipolo magnético residual obtido por meio de experimentos em laboratórios com o *PicSat*, valor extraído de ([MENEGALDO, 2020](#)).

Para o campo magnético terrestre,  $\mathbf{B}$ , adota-se um modelo simplificado, conforme a abordagem de ([CELANI, 2022](#)), na qual o dipolo magnético da Terra está alinhado ao seu eixo de rotação. Desse modo, obtém-se a [Equação 2.16](#)

$$\mathbf{B}^i(t) = \frac{\mu_{ad}}{a^3} \begin{bmatrix} \sin i \cos(\omega_s t + \phi_0) \\ -\cos i \\ 2 \sin i \sin(\omega_s t + \phi_0) \end{bmatrix}, \quad (2.16)$$

na qual  $i = 98^\circ$  é a inclinação da órbita do *PicSat*,  $a = 6920 \text{ km}$  é o raio da órbita,  $\phi_0 = 0$  é o ângulo de fase e  $\omega_s = 0,00116 \text{ rad/s}$  é a velocidade angular do satélite em sua órbita ao redor da Terra. Ademais,  $\mu_{ad} = 7,71 \cdot 10^{15} \text{ Wb m}$  é o módulo do dipolo inclinado, valor extraído de ([CELANI, 2022](#)).

Vale ressaltar que o campo magnético da [Equação 2.16](#) está expresso no sistema de coordenadas inercial e, para que o torque seja incluído na [Equação 2.10](#), deve ser convertido para coordenadas do Referencial dos Eixos Principais. Tal procedimento pode ser realizado com a matriz de rotação, em função dos quatérnios, conforme a [Equação 2.4](#).

#### 2.4.4 Sistema de equações do modelo

Ante o desenvolvimento explicitado, pode-se determinar o sistema de equações diferenciais que modelam o sistema

$$\dot{\omega}_1 = \frac{T_{p1} + (I_2 - I_3)\omega_2\omega_3 - \dot{h}_{rr1}}{I_1} \quad (2.17)$$

$$\dot{\omega}_2 = \frac{T_{p2} + (I_3 - I_1)\omega_1\omega_3 - \dot{h}_{rr2}}{I_2} \quad (2.18)$$

$$\dot{\omega}_3 = \frac{T_{p3} + (I_1 - I_2)\omega_1\omega_2 - \dot{h}_{rr3}}{I_3} \quad (2.19)$$

$$\dot{q}_0 = -\frac{q_1\omega_1 + q_2\omega_2 + q_3\omega_3}{2} \quad (2.20)$$

$$\dot{q}_1 = \frac{q_0\omega_1 + q_2\omega_3 - q_3\omega_2}{2} \quad (2.21)$$

$$\dot{q}_2 = \frac{q_0\omega_2 - q_1\omega_3 + q_3\omega_1}{2} \quad (2.22)$$

$$\dot{q}_3 = \frac{q_0\omega_3 + q_1\omega_2 - q_2\omega_1}{2} \quad (2.23)$$

em que  $T_p$  refere-se ao torque de perturbação associado ao dipolo magnético residual, conforme desenvolvido na [subseção 2.4.3](#).

## 2.5 Linearização

Como alternativa para a linearização do sistema, utilizou-se a expansão em série de Taylor.

Para uma função qualquer  $f(x)$  e um ponto de operação  $\bar{x}$ , define-se  $\delta x = x - \bar{x}$  como uma pequena variação ao redor do ponto de equilíbrio. Desprezando os termos de maior ordem, a expansão é representada na [Equação 2.24](#).

$$f(x) \approx f(\bar{x}) + \left. \frac{\partial f(x)}{\partial x} \right|_{x=\bar{x}} \delta x \quad (2.24)$$

Nesse trabalho, utiliza-se como ponto de equilíbrio os pontos  $\bar{\mathbf{q}} = [\bar{q}_0 \ \bar{q}_1 \ \bar{q}_2 \ \bar{q}_3]^T$  e  $\bar{\boldsymbol{\omega}} = [\bar{\omega}_1 \ \bar{\omega}_2 \ \bar{\omega}_3]^\top$ , que são referentes à atitude e velocidade angular do *CubeSat*.

Pela álgebra dos quatérnios, as variações de atitude  $\delta q$  e velocidade angular  $\delta \boldsymbol{\omega}$  são dadas por:

$$\delta q = \bar{q}^{-1} q_{p/i} \quad (2.25)$$

e

$$\delta \boldsymbol{\omega} = \omega_{p/i}^p - \bar{\boldsymbol{\omega}}. \quad (2.26)$$

Para pequenas variações de ângulo,  $\delta q_0 \approx 1$ , conforme a Equação 2.3. Desse modo,  $\delta \mathbf{q} \approx [1, \delta \mathbf{q}]^\top$  e as equações do modelo linear cinemático são

$$\frac{d}{dt} \mathbf{q} \approx \begin{bmatrix} 0 \\ -\frac{1}{2} [\mathbf{S}(\bar{\boldsymbol{\omega}})] \delta \mathbf{q} \end{bmatrix} + \frac{1}{2} [\mathbf{1}]_{3 \times 3} \delta \boldsymbol{\omega} \quad (2.27)$$

e, analogamente, do modelo dinâmico,

$$\dot{\boldsymbol{\omega}} \approx ([\mathbf{I}_v]^p)^{-1} [[\mathbf{S}([\mathbf{I}_v]^p \bar{\boldsymbol{\omega}})] - [\mathbf{S}(\bar{\boldsymbol{\omega}})] [\mathbf{I}_v]^p] \delta \boldsymbol{\omega} - ([\mathbf{I}_v]^p)^{-1} \dot{\mathbf{H}}_{rr}. \quad (2.28)$$

## 2.6 Representação em Espaço de Estados

Representa-se um sistema dinâmico e invariante no tempo, a partir da abordagem de espaços de estados, por

$$\begin{cases} \dot{\mathbf{x}} = [\mathbf{A}]\mathbf{x} + [\mathbf{B}_2]\mathbf{u} + [\mathbf{B}_1]\boldsymbol{\omega} \\ \mathbf{y} = [\mathbf{C}]\mathbf{x} + [\mathbf{D}]\mathbf{u} \end{cases}. \quad (2.29)$$

Assim, conforme o desenvolvimento feito na linearização, o vetor de estados  $\mathbf{x}$  e de entradas  $\mathbf{u}$  são representados pelas equações

$$\mathbf{x} = [\delta q_1 \ \delta q_2 \ \delta q_3 \ \delta \omega_1 \ \delta \omega_2 \ \delta \omega_3]^\top \quad (2.30)$$

$$\mathbf{u} = [\dot{h}_{rr1} \ \dot{h}_{rr2} \ \dot{h}_{rr3}]^\top \quad (2.31)$$

bem como o vetor de distúrbios

$$\boldsymbol{\omega} = [T_{p1} \ T_{p2} \ T_{p3}]^\top \quad (2.32)$$

Com o intuito de simular a operação do PicSat, são definidos ângulos de Euler em torno dos quais será desenvolvida a linearização e que serão convertidos em quatérnios. Ademais, na condição de equilíbrio, as velocidades angulares são nulas. Assim, tem-se:

$$\begin{cases} \bar{\phi} = 0 \\ \bar{\psi} = 0 \\ \bar{\theta} = 0 \end{cases} \implies \begin{cases} q_0 = 1 \\ q_1 = 0 \\ q_2 = 0 \\ q_3 = 0 \end{cases} \text{ e } \begin{cases} \bar{\omega}_1 = 0 \\ \bar{\omega}_2 = 0 \\ \bar{\omega}_3 = 0 \end{cases} \quad (2.33)$$

A matriz de estados  $[\mathbf{A}]$  e de entradas de controle  $[\mathbf{B}_2]$  são, portanto,

$$[\mathbf{A}] = \begin{bmatrix} 0 & 0 & 0 & 0,5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad [\mathbf{B}_2] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -24,9359 & 0 & 0 \\ 0 & -25,0400 & 0 \\ 0 & 0 & -201,5687 \end{bmatrix} \quad (2.34)$$

e, tendo em vista o desenvolvimento das equações apresentadas anteriormente, a matriz de entrada de distúrbios  $[\mathbf{B}_1]$  abaixo. A matriz de saídas  $[\mathbf{C}]$  é construída com base no desenvolvimento da [subseção 2.1.2](#). O sensor estelar e os magnetômetros fornecem constantemente o vetor de orientação do satélite. Constrói-se, portanto,

$$[\mathbf{B}_1] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 24,9359 & 0 & 0 \\ 0 & 25,0400 & 0 \\ 0 & 0 & 201,5687 \end{bmatrix} \quad [\mathbf{C}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.35)$$

e o vetor de saídas,

$$\mathbf{y} = [\delta q_1 \ \delta q_2 \ \delta q_3]^T \quad (2.36)$$

Por fim, a matriz de alimentação é nula,

$$[\mathbf{D}] = [[0]_{6 \times 3}] \quad (2.37)$$

## 2.7 Análise no domínio do tempo

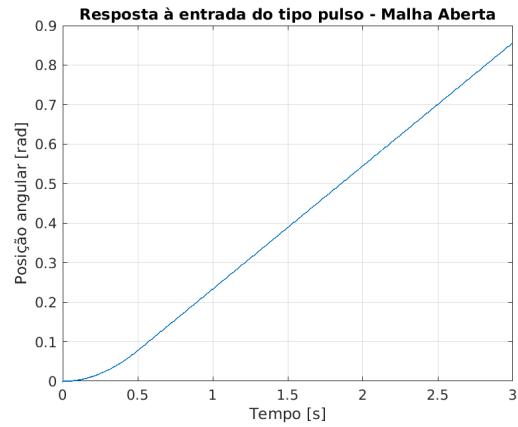
No ambiente do *Matlab*, foram feitas simulações das equações lineares com o intuito de analisar as respostas do sistema a um pulso. O pulso aplicado tem magnitude  $50 \text{ mN} \cdot \text{m}$  em relação ao primeiro eixo principal de inércia e duração de 0,5 segundo. Não é aplicado nenhum tipo de entrada em relação aos outros dois eixos principais de inércia.

É importante ressaltar que as simulações foram feitas utilizando quatérnios, de forma que os números obtidos como respostas não podem ser traduzidos diretamente para os ângulos de precessão, nutação e rotação própria de forma isolada. Ao longo do desenvolvimento do presente estudo, será feita uma conversão de quatérnios para ângulos, de forma que a análise seja simplificada.

Figura 10 – Resposta do primeiro quatérnio a pulso - Malha Aberta.

(a)  $\omega_1$ .

Fonte: autoria própria.

(b)  $q_1$ .

Fonte: autoria própria.

Percebe-se que, como esperado, a velocidade angular  $\omega_1$  varia por 0,5 segundo, estabilizando após esse intervalo de tempo, coerente com a aplicação do pulso. Como resposta a essa variação, o quatérnio  $q_1$  varia de forma quadrática no intervalo de 0 a 0,5 segundo e, em seguida, de forma linear, dado que  $\omega_1$  não volta a ser nulo. Isso se deve à ausência de forças externas atuando no sistema. Assim, ainda que cessada a entrada, a velocidade se mantém constante, conservando a energia do sistema.

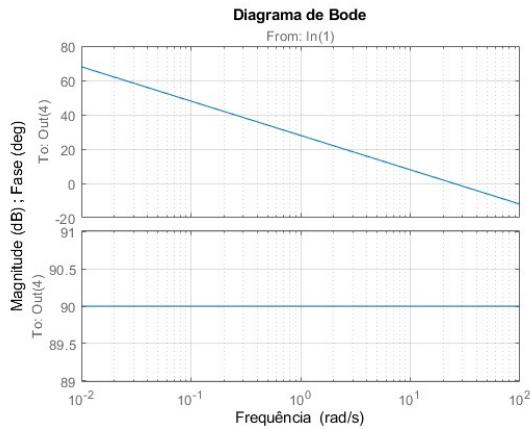
Para os outros dois quatérnios, como não há aplicação de entradas em relação aos demais eixos de inércia, não se verifica nenhum tipo de variação, nem nos quatérnios ( $q_2$  e  $q_3$ ), tampouco nas velocidades angulares dos eixos principais de inércia ( $\omega_2$  e  $\omega_3$ ), resultado esperado dado que os eixos são desacoplados.

## 2.8 Análise no domínio da frequência

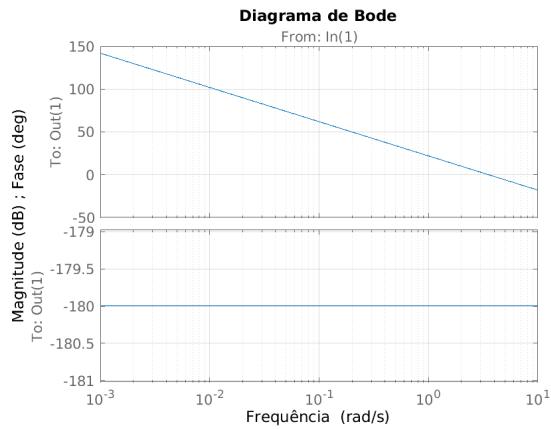
Com o objetivo de analisar a resposta do sistema linearizado a frente a uma variação periódica de sinais de entrada com diversas frequências, foram plotados 36 Diagramas de Bode (uma para combinação entre entradas e saídas).

Para cada um dos três quatérnios, a resposta é idêntica, tanto em termos de magnitude quanto em fase, fator associado à dinâmica do sistema. A análise, portanto, será limitada a somente uma representação gráfica da resposta para a primeira e quarta componentes do vetor de estados, conforme a [Figura 11](#).

Figura 11 – Resposta em frequência - Malha Aberta.

(a)  $\omega_1$ .

Fonte: autoria própria.

(b)  $q_1$ .

Fonte: autoria própria.

Observa-se no diagrama de magnitude que não há pico de ressonância e que a resposta de magnitude do quatérnio é mais intensa do que a resposta da taxa de variação do quatérnio. O resultado é típico de um sistema com zeros e polos na origem.

Para o diagrama da Figura 11a, a fase é constante e igual a  $-90^\circ$  e o ganho apresenta decréscimo de 20 dB por década. Tal característica está associada a uma Função de Transferência com um único polo na origem. Esse resultado é possível, mesmo que o sistema tenha 6 polos na origem, devido ao cancelamento de zeros e polos, conforme será explicitado no Capítulo 4.

Já o diagrama da Figura 11b apresenta fase constante e igual a  $-180^\circ$ , além de ganho com decréscimo de 40 dB por década. Esse é um comportamento de um sistema com dois polos na origem. Novamente, trata-se de um caso em que há cancelamento de zeros e polos.

# 3 Controle Moderno

A teoria de Controle Moderno surge, em meados da década de 1950, da necessidade de controlar sistemas complexos tais como mísseis e sondas espaciais, para os quais a Teoria de Controle Clássico eram limitadas. Algumas de suas principais características são o desenvolvimento em domínio do tempo, a utilização de sistemas MIMO (*Multiple Input Multiple Output*, da sigla em inglês) e a possibilidade de otimização de acordo com critérios definidos no projeto. A seguir, serão apresentados os conceitos e projetos de controladores associados a essa teoria.

## 3.1 Análise de controlabilidade e observabilidade

A primeira e fundamental etapa da teoria de Controle Moderno é a análise do sistema quanto a sua controlabilidade e observabilidade.

Para um sistema invariante no tempo  $\dot{\mathbf{x}} = [\mathbf{A}]\mathbf{x} + [\mathbf{B}]\mathbf{u}$ , a controlabilidade pode ser calculada através do Teorema Algébrico da Controlabilidade, que determina que um sistema é controlável se o posto da matriz de controlabilidade, definida na [Equação 3.1](#), for igual ao número de variáveis de estado do sistema, ou seja, a sua ordem ([FRIEDLAND, 2012](#)).

$$[\mathbf{Co}] = [[\mathbf{B}] \quad [\mathbf{A}][\mathbf{B}] \quad [\mathbf{A}]^2[\mathbf{B}] \quad \dots \quad [\mathbf{A}]^{n-1}[\mathbf{B}]] \quad (3.1)$$

Destaca-se que a matriz de controlabilidade possui  $n$  linhas e  $n \times m$  colunas, onde  $n$  representa o número de variáveis de estado e  $m$ , o número de entradas. Dessa forma, para o PicSat, a matriz de controlabilidade para as 6 variáveis de estado e 3 entradas foi calculada através do comando `ctrb()` implementado no *Matlab*.

Para determinar o posto da matriz  $[\mathbf{Co}]$ , utilizou-se o comando `rank()`, de tal forma que o valor obtido coincide com a ordem do sistema,  $n = 6$ . Portanto, conclui-se que o conjunto de três rodas de reação é suficiente para controlar a atitude do satélite.

De maneira análoga, para a análise de observabilidade de um sistema, é possível utilizar o teorema algébrico da observabilidade, que determina que o posto da matriz representada na [Equação 3.2](#) deve ser igual à ordem do sistema.

$$[\mathbf{Ob}] = [[\mathbf{C}]^\top \quad [\mathbf{A}]^\top[\mathbf{C}]^\top \quad ([\mathbf{A}]^2)^\top[\mathbf{C}]^\top \quad \dots \quad ([\mathbf{A}^{n-1}])^\top[\mathbf{C}]^\top] \quad (3.2)$$

No sistema de análise, a matriz de saídas é dada pela [Equação 2.35](#). Assim, calculase o posto da matriz com os valores numéricos, que é igual a ordem do sistema  $n = 6$ ,

assim como esperado. Uma vez determinado que o sistema é controlável e observável, é possível desenvolver métodos para controlá-lo.

## 3.2 Controlador por Alocação de Polos

O método de controle por alocação de polos fundamenta-se no conceito de que é possível alterar a dinâmica de um sistema, a partir da alocação de polos de malha fechada capazes de determinar performance dinâmica do sistema, para sistemas estáveis e instáveis. Para isso, utiliza-se uma matriz de ganhos  $[K]$  de realimentação negativa do vetor de estados.

Teoricamente, todo sistema pode ter qualquer resposta dinâmica desejada, desde que os polos sejam alocados corretamente. No entanto, na prática, o resultado pode não ser tangível, uma vez que determinados comportamentos podem elevar o sinal de entrada a níveis não alcançáveis pelos atuadores, de forma a saturá-los e a provocar instabilidade no sistema de malha fechada ([FRIEDLAND, 2012](#)).

O regulador tem como objetivo primordial levar um sistema para sua origem, isto é,  $x(t_f) = 0$ . Quanto maior for a distância até a origem do sistema, maior será a ação de controle. Assim, o problema do regulador resume-se a determinar a matriz de ganhos para realimentação do vetor de estados:  $\mathbf{u} = -[K]\mathbf{x}$ .

Para o nanossatélite estudado, os polos do sistema sem atuação são todos nulos, isto é, o sistema é estável, mas só se move enquanto houver momentos atuantes.

No caso do regulador projetado, espera-se uma resposta dinâmica rápida e com pouca oscilação, que são pontos importantes para determinar o cumprimento dos requisitos de um controle com apontamento estável. Para isso, são alocados os polos representados na [Tabela 2](#). Dessa forma, a dinâmica do sistema é alterada de acordo com os polos alocados, como representado na [Equação 3.3](#).

$$\dot{\mathbf{x}} = ([\mathbf{A}] - [\mathbf{B}_2][\mathbf{K}])\mathbf{x} + [\mathbf{B}_1]\boldsymbol{\omega} \quad (3.3)$$

Tabela 2 – Polos alocados - regulador por Alocação de Polos.

	<b>Valor</b>
Polo 1	-0,80 + 0,60i
Polo 2	-0,80 - 0,60i
Polo 3	-2,60 + 0,30i
Polo 4	-2,60 - 0,30i
Polo 5	-1,00
Polo 6	-0,80

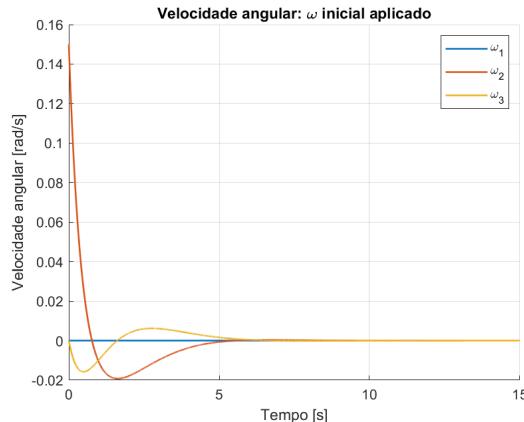
Fonte: autoria própria.

Os polos alocados foram definidos de forma interativa, ou seja, partindo de três pares de polos arbitrários, foram feitas simulações e, a depender dos resultados, ajustados os parâmetros para uma nova rodada de simulações. O processo se repetiu até que uma solução conveniente - amplitude de resposta baixa em relação à condição inicial imposta, tempo de convergência baixo (definido comparativamente durante o processo) e polos não muito distantes do eixo imaginário - foi alcançada.

Para uma variação nas condições iniciais aplicada à velocidade angular paralela à segunda roda de reação - eixo Y do satélite - observam-se as respostas temporais das variáveis de velocidade e posição angular representadas, respectivamente, nas figuras 12a e 12b.

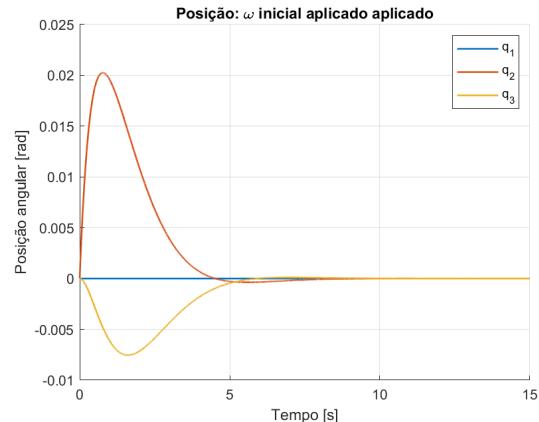
Figura 12 –  $\omega_2(0) = 0,15 \text{ rad/s}$  - regulador por Alocação de Polos.

(a) Velocidades angulares.



Fonte: autoria própria.

(b) Posições angulares.



Fonte: autoria própria.

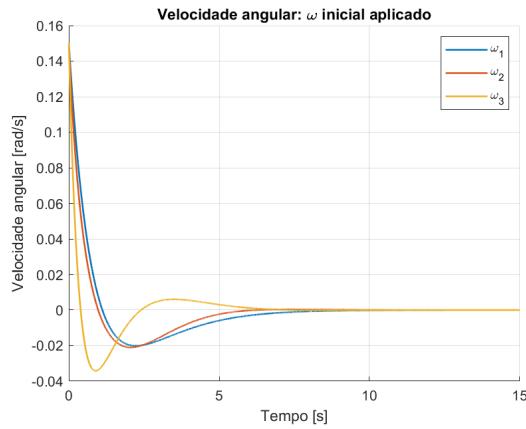
A partir da Figura 12a, observa-se que somente as velocidades angulares  $\omega_2$  e  $\omega_3$  são afetadas, de forma que somente as componentes  $q_2$  e  $q_3$  apresentam variação temporal. Esse comportamento indica um acoplamento entre estas duas componentes, causado pela mudança da dinâmica do sistema após a alocação de seus polos.

Outro ponto interessante de se notar é o fato do sistema voltar à sua posição de equilíbrio após 5 segundos de simulação com um pico secundário (mínimo das curvas de velocidade) da ordem de quatorze vezes menor que a condição inicial imposta, o que atesta a eficiência do regulador projetado.

Para o caso em que se aplica velocidade angular inicial de  $0,15 \text{ rad/s}$  em todas os eixos principais do satélite, observa-se que todas as componentes de velocidade e posição variam no tempo e retornam à origem do sistema, comportamento contido nas figuras 13a e 13b.

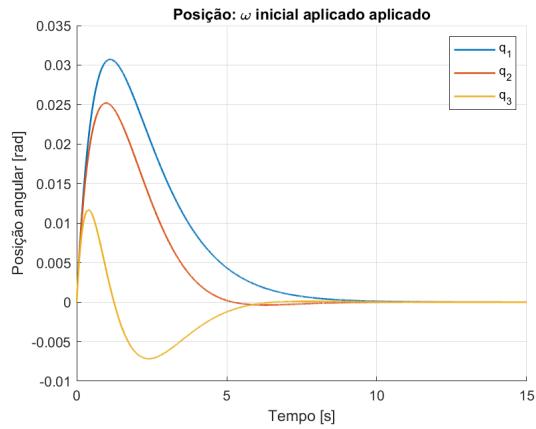
Figura 13 –  $\omega_1(0) = \omega_2(0) = \omega_3(0) = 0,15 \text{ rad/s}$  - regulador por Alocação de Polos.

(a) Velocidades angulares.



Fonte: autoria própria.

(b) Posições angulares.



Fonte: autoria própria.

### 3.3 Controlador Linear Quadrático

O Controlador Linear Quadrático insere-se no contexto da busca pelo controle ótimo e fundamenta-se na determinação de uma lei de controle que minimize um índice quadrático sujeito às restrições do sistema linear (FRIEDLAND, 2012). Sua característica fundamental é a ponderação entre a velocidade de resposta do sistema e o ganho necessário.

Para tanto, tal ponderação é feita de modo a minimizar a função de custo, contida na [Equação 3.4](#), por meio da escolha das matrizes  $[\mathbf{Q}]$  e  $[\mathbf{P}]$ .

$$J = \int_0^{t_f} (\mathbf{x}^\top [\mathbf{Q}] \mathbf{x} + \mathbf{u}^\top [\mathbf{P}] \mathbf{u}) dt \quad (3.4)$$

Novamente, a escolha dos parâmetros foi feita de forma interativa. Partiu-se de matrizes identidade para  $[\mathbf{Q}]$  e  $[\mathbf{P}]$  e, conforme os resultados das simulações, foram feitos ajustes.

Como mostrado na [Tabela 1](#), o menor eixo de inércia do satélite corresponde à terceira componente. Por conta disso, para uma correção similar, é necessário uma menor ação de controle neste eixo.

Dessa forma, como alternativa para reduzir o custo energético do sistema, foi aplicado um fator de 50 no terceiro número da diagonal principal da matriz  $P$ . Essa medida se mostrou eficiente em balancear as respostas do sistema entre suas componentes e evitar uma ação demasiadamente forte no eixo Z. O resultado obtido pode ser verificado na [Equação 3.5](#).

$$[\mathbf{Q}] = \begin{bmatrix} 0,1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,01 \end{bmatrix} \quad [\mathbf{P}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 50 \end{bmatrix} \quad (3.5)$$

A partir das matrizes  $[\mathbf{Q}]$  e  $[\mathbf{P}]$ , foram obtidos os polos explicitados na [Tabela 3](#).

Tabela 3 – Polos obtidos - regulador LQR.

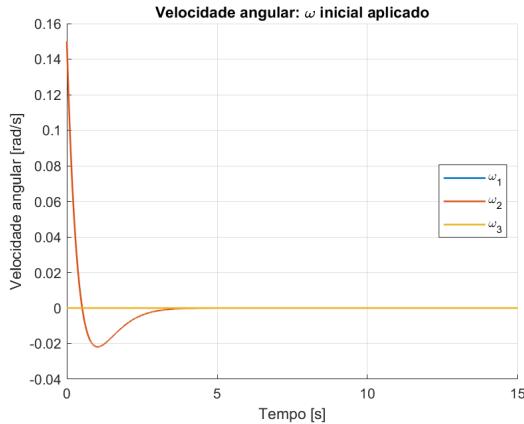
	Valor
Polos 1 e 2	$-1,88 \pm 0,65i$
Polos 3 e 4	$-2,07 \pm 0,47i$
Polos 5 e 6	$-1,88 \pm 0,64i$

Fonte: autoria própria.

Novamente, as simulações foram feitas variando as condições iniciais quanto às velocidades angulares. Inicialmente, somente  $\omega_2$  inicial foi alterada e os resultados podem ser vistos na [Figura 14](#).

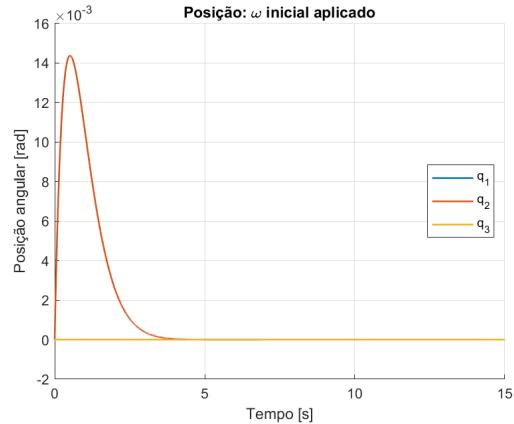
Figura 14 –  $\omega_2(0) = 0,15 \text{ rad/s}$  - regulador LQR.

(a) Velocidades angulares.



Fonte: Autoria própria

(b) Posições angulares.



Fonte: autoria própria

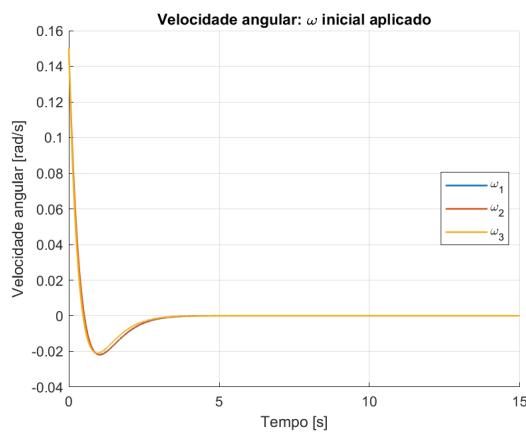
Ao analisar a [Figura 14](#), nota-se que, ao contrário do observado no caso de alocação de polos, apenas  $\omega_2$  e  $q_2$  são afetados pela condição inicial imposta ao sistema. Ou seja, as modificações na dinâmica do sistema não interferiram no não acoplamento das suas equações. Quanto ao desempenho, o método linear quadrático mostrou-se mais eficiente em

relação ao de alocação de polos tanto em termos de tempo de convergência - estabilizando-se pouco antes dos 5 segundos - quanto em picos de resposta - de  $14 \cdot 10^{-3} \text{ rad}$ .

Foi ainda analisado o caso em que se atribuiu a todas as componentes de velocidade angular do satélite um valor inicial diferente de zero. A Figura 15 mostra uma resposta quase idêntica para  $q_1$ ,  $q_2$  e  $q_3$ , atestando o sucesso do esforço em balancear as ações de controle ajustando os valores da matriz  $P$ . Ainda, nesta simulação, o desempenho do sistema se mostrou similar ao da Figura 14.

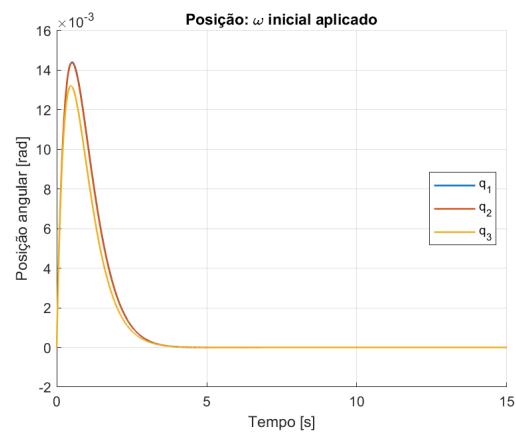
Figura 15 –  $\omega_1(0) = \omega_2(0) = \omega_3(0) = 0,15 \text{ rad/s}$  - regulador LQR.

(a) Velocidades angulares.



Fonte: autoria própria.

(b) Posições angulares.



Fonte: autoria própria.

## 3.4 Observadores de estados

Conforme explicitado na Equação 2.35, os sensores do sistema medem apenas 3 das 6 variáveis de estado. Desse modo, para executar o controle em malha fechada, deve-se reconstruir  $\mathbf{x}$  a partir de  $\mathbf{y}$ , desde que o sistema seja observável. O desenvolvimento da seção 3.1 indica que, mesmo medindo-se apenas as posições angulares, o sistema é observável. Podem ser estimadas, portanto, as velocidades angulares e se obter uma realimentação completa dos estados.

### 3.4.1 Observador de Estado para o regulador via Alocação de Polos

De modo análogo ao desenvolvimento anterior, pode-se utilizar a estratégia de Alocação de Polos para estimar a matriz de ganhos do observador,  $[\mathbf{K}_o]$ . É estruturada uma malha paralela à da planta, com blocos relativos ao observador de estados. Desse modo, seu principal papel é fechar a malha principal com um ganho suficiente para que o erro convirja a zero antes de o sistema atingir regime permanente.

Nesse sentido, um ponto fundamental é que os polos estejam à esquerda dos polos da planta, contidos na [Tabela 2](#). Uma estratégia adotada foi de deslocar os polos da planta por um fator  $-2$ , assegurando-se tal condição. A [Tabela 4](#) indica os polos alocados.

Tabela 4 – Polos alocados para o Observador de Estado - regulador via Alocação de Polos.

	Valor
Polos 1 e 2	$-2,80 \pm 0,60i$
Polos 3 e 4	$-4,60 \pm 0,30i$
Polo 5	-3,00
Polo 6	-2,80

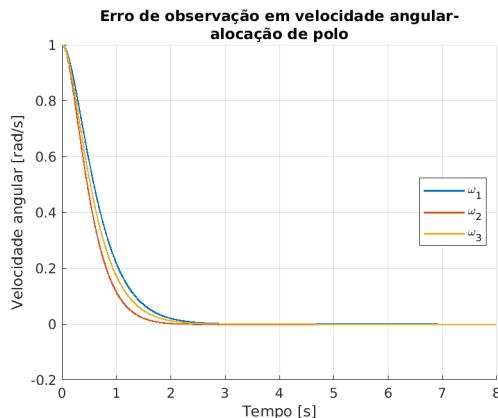
Fonte: autoria própria.

A matriz de ganhos obtida pode ser conferida no [Apêndice A](#). Feito isso, pode-se definir o erro estimado como sendo  $e = x - \hat{x}$ , em que  $\hat{x}$  é o valor estimado de  $x$ . Assim, avaliando-se se há convergência dos erros nas estimativas das velocidades angulares, bem como o tempo para que ela ocorra, conclui-se que os polos alocados constituem boas escolhas.

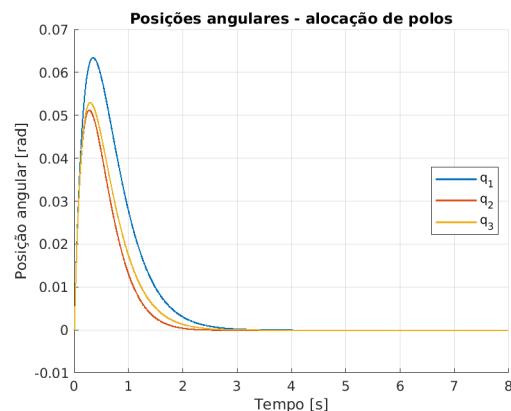
Para a simulação, utilizou-se um erro inicial de  $1 \text{ rad/s}$  em cada componente de velocidade angular, que são as variáveis não medidas por sensores. A evolução temporal do erro é dada pela [Figura 16a](#), ao passo que a [Figura 16b](#) indica as posições angulares.

Figura 16 – Estabilização do erro para observador de estado via Alocação de Polos.

(a) Erro associado às velocidades angulares.



(b) Posições angulares.



Fonte: autoria própria.

Nota-se que, com um tempo de aproximadamente 3 segundos, o erro estabiliza-se em 0. Tal valor é 2,5 vezes menor que o tempo de estabilização do regulador por Alocação de Polos, conforme os resultados da [Figura 12](#), ainda que o erro inicial imposto, de  $1 \text{ rad/s}$ , seja extremamente alto no contexto de um *CubeSat*. Para fins de comparação, na fase de *detumbling*, logo após o lançamento, o satélite atinge velocidades angulares próximas a  $1,2 \text{ rad/s}$ , conforme ([MENEGALDO, 2020](#)). Conclui-se, portanto que os polos do observador obtidos pelo método de Alocação de Polos são válidos.

### 3.4.2 Observador de Estado para o Controlador Linear Quadrático

Novamente, trata-se de uma otimização em que se deve minimizar a função de custo. Tomando-se procedimentos análogos aos já descritos na [seção 3.3](#), obtém-se que as matrizes da [Equação 3.6](#) são as que melhor se adéquam ao problema. Pode-se indicar, ainda, a matriz de ganhos obtida com o uso do comando *lqr()* do *Matlab*.

$$[\mathbf{Q}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 & 0 & 35 \end{bmatrix} \quad [\mathbf{P}] = \frac{1}{40} \cdot \begin{bmatrix} 1,5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix} \quad [\mathbf{K}_o] = \begin{bmatrix} 7,41 & 0 & 0 \\ 0 & 6,67 & 0 \\ 0 & 0 & 4,97 \\ 28,28 & 0 & 0 \\ 0 & 24,49 & 0 \\ 0 & 0 & 16,73 \end{bmatrix} \quad (3.6)$$

Os polos para o novo sistema também podem ser encontrados, explicitados na [Tabela 5](#). Nota-se que todos encontram-se à esquerda dos polos do regulador LQR desenvolvido, o que é esperado para um observador.

Tabela 5 – Polos para o Observador de Estado - regulador LQR.

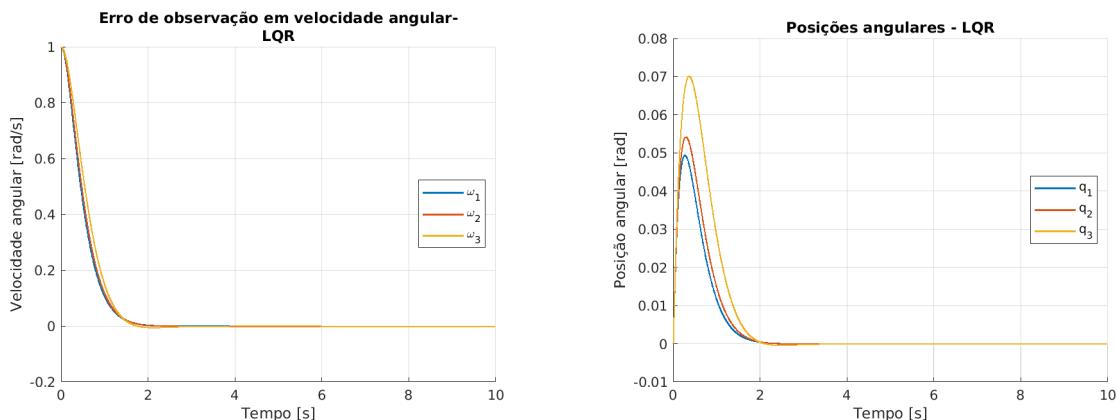
	Valor
Polos 1 e 2	$-3,71 \pm 0,64i$
Polos 3 e 4	$-3,34 \pm 1,06i$
Polos 5 e 6	$-2,49 \pm 1,48i$

Fonte: autoria própria.

Novamente, utilizando-se um erro inicial de  $1 \text{ rad/s}$  em cada componente de velocidade angular, isto é,  $e(0) = [0 \ 0 \ 0 \ 1 \ 1]^T$ , são obtidas as respostas dinâmicas contidas na [Figura 17](#).

Figura 17 – Estabilização do erro para observador de estado LQR.

(a) Erro associado às velocidades angulares. (b) Posições angulares.



Fonte: autoria própria.

Vê-se que as respostas praticamente não possuem *overshoot*, além de terem um tempo de acomodação da mesma ordem de grandeza do caso de alocação de polos, com tempo levemente inferior (2,5 segundos). Tal resultado é satisfatório, haja vista que o sistema regulado por LQR entra em regime permanente, conforme a [Figura 15b](#), em 5 segundos.

### 3.5 Seguidor com referência constante

Os seguidores de referência, projetados a partir de um regulador, têm por objetivo seguir uma trajetória ao longo do tempo, a partir de uma sinal de pré-alimentação. Para uma referência constante, do tipo degrau, a trajetória passa a ser um novo estado.

Nesse caso, o cálculo da ação de controle, para minimizar o erro entre o estado do sistema  $\mathbf{x}$  e o estado de referência  $\mathbf{x}_r$ , deve-se considerar o erro em regime permanente, que a entrada  $\mathbf{u} = -[\mathbf{K}](\mathbf{x} - \mathbf{x}_r)$  geraria. Assim, calcula-se o estado do sistema em regime permanente  $\mathbf{x}_{RP}$  e a ação de controle  $\mathbf{u}_{RP}$ , que é capaz de tornar nula a diferença entre  $\mathbf{x}$  e  $\mathbf{x}_r$ , resultando na ação de controle representada na [Equação 3.7](#).

$$\mathbf{u} = \mathbf{u}_{RP} - [\mathbf{K}](\mathbf{x} - \mathbf{x}_{RP}) \quad (3.7)$$

Assim, ao definir que  $\mathbf{x}_{RP}$  e  $\mathbf{u}_{RP}$  são proporcionais à trajetória de referência  $\mathbf{x}_r$ , determina-se o sistema de equações representado na [Equação 3.8](#).

$$\begin{cases} [0] = [\mathbf{A}][\mathbf{N}_x]\mathbf{x}_r + [\mathbf{B}][\mathbf{N}_u]\mathbf{x}_r \\ \mathbf{x}_r = [\mathbf{C}][\mathbf{N}_x]\mathbf{x}_r + [\mathbf{D}][\mathbf{N}_u]\mathbf{x}_r \end{cases} \quad (3.8)$$

Em que  $[\mathbf{N}_x]$  e  $[\mathbf{N}_u]$  determinam a nova lei de controle, para uma solução não trivial  $\mathbf{x}_r \neq 0$ , dada na [Equação 3.9](#).

$$\mathbf{u} = -[\mathbf{K}]\mathbf{x} + ([\mathbf{K}][\mathbf{N}_x] + [\mathbf{N}_u])\mathbf{x}_r \quad (3.9)$$

Como as variáveis de estado a seguirem a trajetória de referência são as três componentes de posição angular, ao resolver o sistema de equações a seguir

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{B}] \\ [\mathbf{C}] & [\mathbf{D}] \end{bmatrix} \begin{bmatrix} [\mathbf{N}_x] \\ [\mathbf{N}_u] \end{bmatrix} = \begin{bmatrix} [0]_{6 \times 6} \\ [\mathbf{I}]_{3 \times 6} \\ [0]_{3 \times 6} \end{bmatrix}, \quad (3.10)$$

obtém-se as matrizes representadas no [Apêndice B](#).

Assim, é calculado a entrada de alimentação prévia da malha de controle. Com o objetivo de validar a malha de controle feita, executa-se uma das principais atividades do

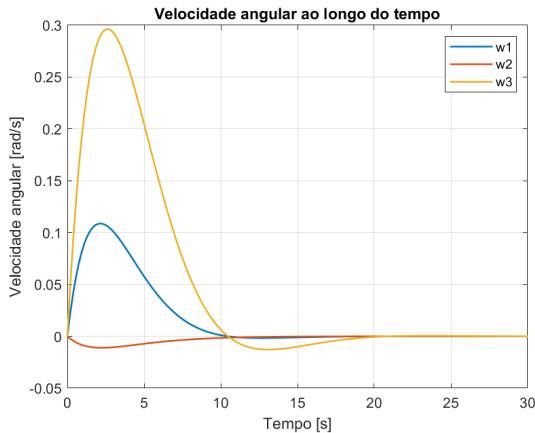
subsistema de controle de atitude de um satélite: o apontamento, ou seja, o seguimento de uma referência constante.

Para o primeiro caso estudado, deseja-se que o *CubeSat* aponte para uma direção descrita pelos ângulos de Euler  $\psi = \pi/4$  (precessão),  $\theta = \pi/6$  (nutação) e  $\phi = \pi/3$  (rotação própria) para a sequência de rotação “ZXZ”. Dado que o simulador elaborado trabalha com a descrição cinemática em parâmetros de Euler, o valor de referência passa a ser  $q = [q_0 \ q_1 \ q_2 \ q_3] = [0,5880 \ 0,2566 \ -0,0338 \ 0,7663]$ .

Uma vez simulado o sistema, obtém-se a entrada em regime permanente  $\mathbf{u}_{\text{RP}} = [-5,1 \cdot 10^{-3} \ -5,4 \cdot 10^{-4} \ -1,4 \cdot 10^{-3}]$ , como equacionada na [Equação 3.7](#). Sendo assim, a partir do regulador por alocação de polos elaborado, controla-se o sistema e levá-o para o estado de referência determinado, como ilustram as Figuras 18a e 18b, que contêm a evolução das velocidades e posições angulares ao longo do tempo, respectivamente.

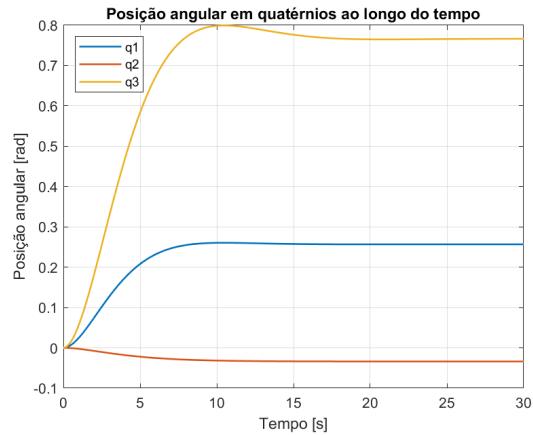
Figura 18 – Evolução temporal do sistema - seguidor com referência constante.

(a) Velocidades angulares .



Fonte: autoria própria.

(b) Posições angulares.

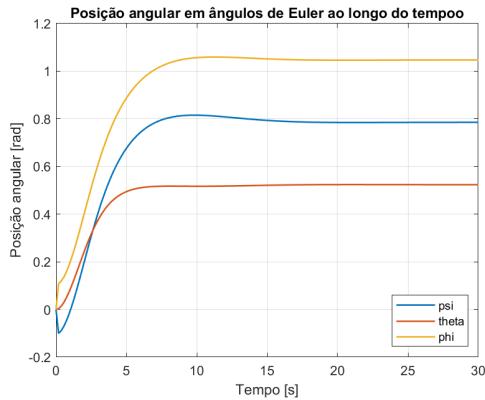


Fonte: autoria própria.

A partir da [Figura 18a](#), nota-se que a maior velocidade angular atingida foi de aproximadamente de  $\omega_3 \approx 0,3 \text{ rad/s}$ , sendo, portanto, compatível com a resposta desejada, uma vez valores reduzidos para as velocidades angulares e retorno lento para o ponto de equilíbrio são de interesse, visando maior penalização da ação de controle.

Nota-se que os valores dos parâmetros de Euler são estabilizados em torno de  $t = 20 \text{ s}$ , assim que as velocidades angulares do satélites são zeradas. Para melhor visualizar a progressão do movimento do satélite, é feita a conversão para ângulos de Euler, cuja representação é observada na [Figura 19](#). Observa-se que após um *overshoot* bastante reduzido, o sistema é estabilizado em torno da nova posição de equilíbrio, conforme determinado pela referência constante.

Figura 19 – Evolução temporal dos ângulos de Euler - seguidor com referência constante.



Fonte: autoria própria.

### 3.6 Seguidor com referência móvel e perturbações

Como explicitado pelo nome, o seguidor com referência variante e perturbações é capaz de acompanhar uma trajetória ao longo do tempo, bem como rejeitar perturbações. Assim, o sistema de equações passa a ser definido pela [Equação 3.11](#),

$$\begin{cases} \dot{\mathbf{x}} = [\mathbf{A}]\mathbf{x} + [\mathbf{B}_1]\mathbf{x}_w + [\mathbf{B}_2]\mathbf{u} \\ \dot{\mathbf{x}}_w = [\mathbf{A}_w]\mathbf{x}_w \\ \dot{\mathbf{x}}_r = [\mathbf{A}_r]\mathbf{x}_r \end{cases}, \quad (3.11)$$

em que os vetores  $\mathbf{x}_r$  e  $\mathbf{x}_w$  são referentes à trajetória de referência e às perturbações, respectivamente, assim como as matrizes  $[\mathbf{A}_r]$  e  $[\mathbf{A}_w]$ .

Dessa forma, define-se o vetor de variáveis exógenas como  $\mathbf{x}_{ex} = [\mathbf{x}_w \ \mathbf{x}_r]^\top$ . Em seguida, após determinar a matriz singular  $[\bar{\mathbf{C}}]$ , define-se a matriz  $[\mathbf{K}_{ex}]$  com base na equação [Equação 3.12](#).

$$[\mathbf{K}_{ex}] = \left[ [\bar{\mathbf{C}}] ([\mathbf{A}] - [\mathbf{B}_2][\mathbf{K}])^{-1} [\mathbf{B}_2] \right]^{-1} [\bar{\mathbf{C}}] ([\mathbf{A}] - [\mathbf{B}_2][\mathbf{K}])^{-1} [\mathbf{F}] \quad (3.12)$$

Por fim, definindo-se as matrizes a seguir,

$$[\mathbf{A}_y] = [[\mathbf{B}_1] \quad [\mathbf{B}_2][\mathbf{K}]] - [\mathbf{B}_2][\mathbf{K}_{ex}], \quad [\mathbf{A}_0] = \begin{bmatrix} [\mathbf{A}_w] & [\mathbf{0}]_{6 \times 6} \\ [\mathbf{0}]_{6 \times 6} & [\mathbf{A}_r] \end{bmatrix} \quad (3.13)$$

obtém-se o novo sistema de equações não forçadas a ser simulado, a partir do vetor de entradas  $\mathbf{u} = -[\mathbf{K}]\mathbf{e} - [\mathbf{K}_{ex}]\mathbf{x}_{ex} = -[\mathbf{K}](\mathbf{x} - \mathbf{x}_r) - [\mathbf{K}_{ex}]\mathbf{x}_{ex}$ .

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_{ex} \end{bmatrix} = \begin{bmatrix} ([\mathbf{A}] - [\mathbf{B}_2][\mathbf{K}]) & [\mathbf{A}_y] \\ [\mathbf{0}]_{12 \times 6} & [\mathbf{A}_0] \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{ex} \end{bmatrix} \quad (3.14)$$

Com o desenvolvimento da [Subseção 2.4.3](#), constituindo os torques de distúrbio o dipolo magnético residual e tomando-se as derivadas temporais das componentes de [Equação 2.16](#), pode-se construir a matriz e distúrbios  $[\mathbf{A}_w]$ . Ademais, considerando-se uma referência de posição angular constante, determina-se também a matriz de referências  $[\mathbf{A}_r]$ , representada no [Apêndice C](#).

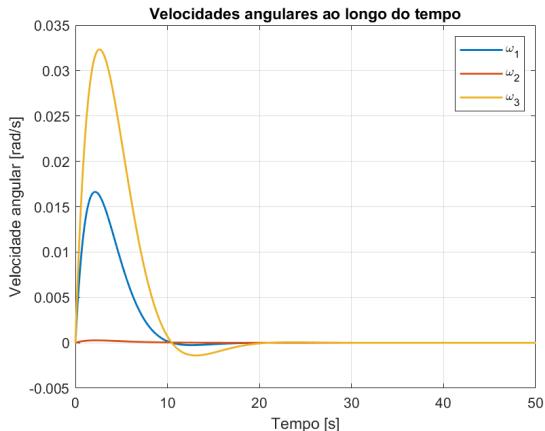
A próxima etapa consiste na definição das condições iniciais, bem como amplitude de oscilação do torque e os parâmetros da referência a ser seguida. Assim, define-se o vetor  $\bar{\mathbf{x}}_0$ :

$$\bar{\mathbf{x}}_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ A_{B1} \ A_{B2} \ A_{B3} \ 0 \ 0 \ 0 \ q_{r1} \ q_{r2} \ q_{r3} \ 0 \ 0 \ 0]^T \quad (3.15)$$

Com isso, tem-se, para os ângulos de Euler  $\psi = \pi/30$ ,  $\theta = \pi/40$  e  $\phi = \pi/50$ , ou seja,  $q = [0,9958 \ 0,0393 \ 0,0008 \ 0,0836]$ , e para as amplitudes de distúrbio  $A_{B1} = A_{B2} = A_{B3} = 5 \cdot 10^{-9}$ , os valores observados para as velocidades e posições angulares ao longo do tempo, representados nas Figuras 20a e 20b.

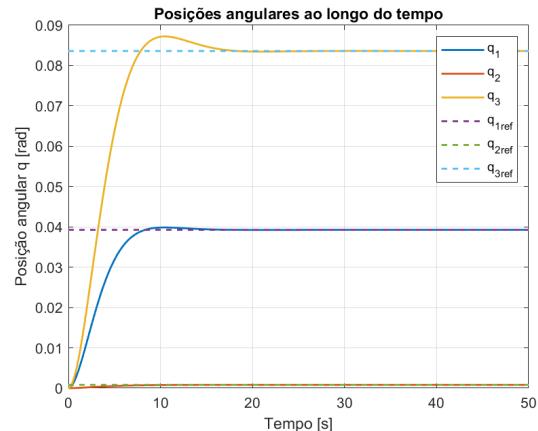
Figura 20 – Evolução do estado do sistema ao longo do tempo.

(a) Velocidade angular dos eixos principais.



Fonte: autoria própria.

(b) Quatérnios.



Fonte: autoria própria.

No caso, observa-se que tanto as velocidades quanto as posições angulares não apresentam graficamente oscilações em função dos distúrbios provocados pelo dipolo magnético. Essa constatação é corroborada com o gráfico de torques aplicados ao longo de duas órbitas completas, ilustrado na [Figura 22](#), que indica que os torques limites possuem ordem de grandeza de  $10^{-6}$ , enquanto os torques da roda de reação são da ordem de  $10^{-3}$ .

Figura 21 – Torques de distúrbio ao longo do tempo

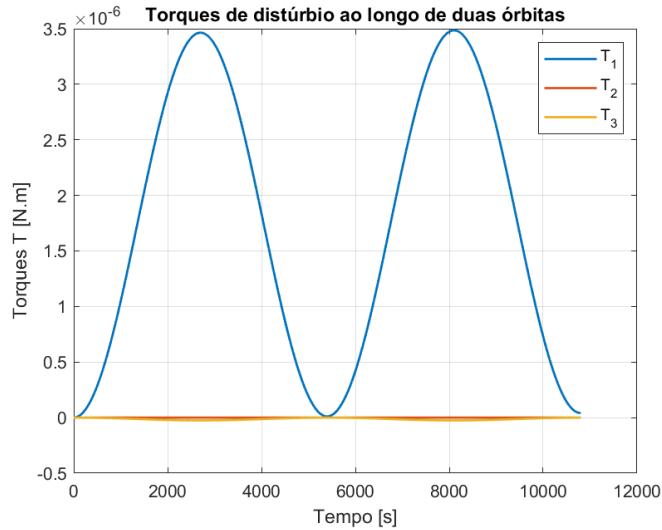


Figura 22 – Fonte: autoria própria.

Com isso, observa-se que, embora existam distúrbios no ambiental espacial para o qual o satélite é projetado, as suas contribuições para a atitude do satélite são, de forma geral, mínimas, como constado pelas análises gráficas. Além disso, nota-se que o problema de referência móvel não apresenta significado físico para o satélite, uma vez que as suas atividades principais são a estabilização e apontamento, ambas tarefas atingidas por referências constantes.

# 4 Controle Clássico

Controladores do tipo proporcional-integrativo-derivativo (PID) são amplamente utilizados no ramo industrial por sua simplicidade e seu baixo custo de implementação. Tal ferramenta, entretanto é compatível apenas com sistemas SISO. Neste capítulo, serão desenvolvidas técnicas de sintonia para controladores PID.

## 4.1 Função de transferência em malha aberta

As funções de transferência caracterizam relações entre as entradas e saídas de um sistema, de forma que é possível obter uma relação entre a função de resposta e a função de excitação para condições iniciais nulas. A [Equação 4.1](#) fornece a matriz de transferência, composta pelas funções de transferência em malha aberta, com  $G_{i,j}$  representando a função entre a saída  $i$  e a entrada  $j$ .

$$[\mathbf{G}] = [\mathbf{C}](s[\mathbf{I}] - [\mathbf{A}])^{-1}[\mathbf{B}_2] + [\mathbf{D}] \quad (4.1)$$

O sistema em análise é composto por 6 variáveis de saída  $\mathbf{y}^\top = [q_1 \ q_2 \ q_3 \ \omega_1 \ \omega_2 \ \omega_3]$  e por 3 variáveis de entrada  $\mathbf{u}^\top = [T_{rr1} \ T_{rr2} \ T_{rr3}]$ , de modo que a matriz de transferência  $\mathbf{G}$  é composta por 6 linhas e 3 colunas, como mostra a [Equação 4.2](#). Optou-se pela utilização dos torques das rodas de reação como variáveis entrada, de modo a se obter um sinal positivo nos numeradores das funções de transferência. Isso permite a análise de ganhos positivos ( $K > 0$ ) ao se desenvolver um sistema em malha fechada.

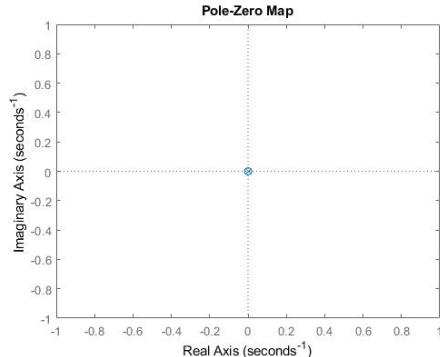
$$[\mathbf{G}] = \begin{bmatrix} \frac{q_1}{T_{rr1}} & \frac{q_1}{T_{rr2}} & \frac{q_1}{T_{rr3}} \\ \frac{q_2}{T_{rr1}} & \frac{q_2}{T_{rr2}} & \frac{q_2}{T_{rr3}} \\ \frac{q_3}{T_{rr1}} & \frac{q_3}{T_{rr2}} & \frac{q_3}{T_{rr3}} \\ \frac{\omega_1}{T_{rr1}} & \frac{\omega_1}{T_{rr2}} & \frac{\omega_1}{T_{rr3}} \\ \frac{\omega_2}{T_{rr1}} & \frac{\omega_2}{T_{rr2}} & \frac{\omega_2}{T_{rr3}} \\ \frac{\omega_3}{T_{rr1}} & \frac{\omega_3}{T_{rr2}} & \frac{\omega_3}{T_{rr3}} \end{bmatrix} \quad G_{33} = \frac{q_3}{T_{rr3}} = \frac{100,7843s^4}{s^6}. \quad (4.2)$$

Escolheu-se trabalhar com a função de transferência  $G_{33}$ . Essa escolha foi feita, pois  $q_3$  é a única componente dos quatérnios que depende somente de um dos ângulos de Euler, no caso, a rotação própria. Além disso, o  $q_3$  é a variável que mostra maior resposta em simulações, dado que está associada a uma rotação no eixo de menor momento de inércia. Ainda, uma vez que o sistema linearizado é desacoplado, apenas o torque  $T_{rr3}$  possui influência sobre a saída  $q_3$ , tornando-o candidato factível para entrada do sistema SISO.

## 4.2 Análise de polos e zeros

Em posse da função de transferência escolhida, não é difícil chegar à conclusão de que todos os seus zeros e polos são nulos - fato ilustrado na [Figura 23](#). Assim, fica evidente que o sistema é marginalmente estável.

Figura 23 – Diagrama de polos e zeros para o sistema em malha aberta.



Fonte: Autoria própria.

A função de transferência  $G_{33}$  indica que o sistema em questão é de ordem e tipo seis (denominador de com maior expoente seis e também seis polos em zero).

## 4.3 Resposta em malha aberta

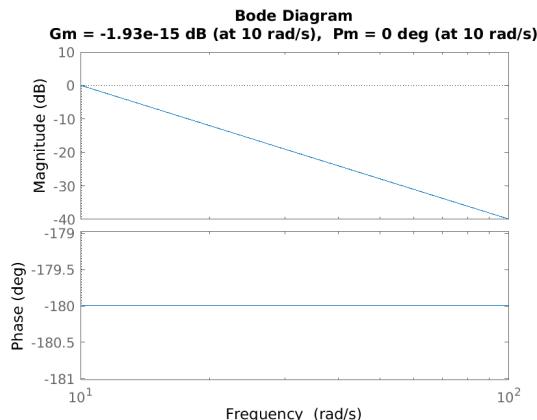
Na [Figura 24](#), vê-se um crescimento parabólico do quatérnio  $q_3$  como resposta à aplicação de um degrau unitário pela roda de reação analisada (a terceira). Ademais, o sistema apresenta ganho nulo em Malha Aberta, característica do sistema marginalmente estável, além de uma margem de fase nula.

Figura 24 – Resposta do sistema frente a um degrau unitário - Malha Aberta.

(a) Velocidade angular dos eixos principais.



(b) Quatérnios.



Fonte: autoria própria.

Fonte: autoria própria.

## 4.4 Sintonia PID

O controlador proporcional integral derivado (PID) é considerado a base da teoria de controle clássico, que se utiliza do erro observado entre o estado do sistema e a referência para dar comandos de controle à planta do sistema. Ele é composto por três partes distintas (proporcional, integral e derivativa), cada uma com um ganho de controle e com características próprias que, quando corretamente combinadas, conseguem atuar de forma satisfatória sobre o sistema.

O controlador proporcional (ganho de controle proporcional  $K_p$ ) gera uma ação de controle proporcional ao erro -  $u(t) = K_p e(t) \rightarrow U(s) = K_p E(s) \rightarrow \frac{U(s)}{E(s)} = K_p$ . Esse controlador não altera a ordem e o tipo do sistema, e também não introduz polos e zeros. Vale também lembrar que o controlador P não é capaz de corrigir erros em regime permanente e que um ganho de controle  $K_p$  excessivo pode criar uma instabilidade no sistema.

O controlador integral (ganho de controle integral  $K_i$ ) gera uma ação de controle proporcional a integral do erro -  $u(t) = K_i \int e(t)dt \rightarrow U(s) = K_i \frac{E(s)}{s} \rightarrow \frac{U(s)}{E(s)} = \frac{K_i}{s}$ . O controlador integral tem como principal característica a correção do erro em regime permanente, no entanto, o seu uso é ruim para o regime transitório, dado que aumenta o tipo e a ordem do sistema, introduzindo polos em zero e a estabilidade relativa.

Por fim, o controlador derivativo (ganho de controle derivativo  $K_d$ ) gera uma ação de controle proporcional a derivada (velocidade) do erro -  $u(t) = K_d \frac{de(t)}{dt} \rightarrow U(s) = K_d E(s) s \rightarrow \frac{U(s)}{E(s)} = K_d s$ . O controlador derivativo tem como principal característica o seu efeito antecipativo, ou seja, é capaz de aumentar a rapidez de resposta do controlador e amortecer o sistema. Além disso, introduz zeros no sistema, aumentando a estabilidade relativa. Como só atua em regime transitório ( $\frac{d e_{RP}(t)}{dt} = 0$ ), esse controlador nunca é usado sozinho.

Existem quatro principais métodos para se determinar os ganhos de controle  $K_p$ ,  $K_i$  e  $K_d$  mais adequados para cada sistema: Ziegler-Nichols, alocação de polos, controle ótimo e lugar da raízes. Cada método será analisado a seguir.

### 4.4.1 Métodos de Ziegler-Nichols

Os métodos de Ziegler-Nichols correspondem a duas formas distintas de sintonia do sistema: método da curva de reação e método do ganho limite.

O método da curva de reação é desenvolvido a partir da resposta ao degrau em malha aberta. A condição para a aplicação desse método é que a resposta ao degrau em malha aberta seja no formato de S, ou seja, a resposta tende a um valor final constante diferente do esperado. Como pode ser visto na [Figura 24](#), a resposta do sistema em estudo

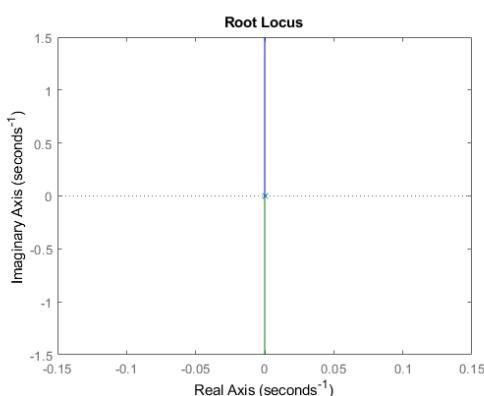
ao degrau não tem o formato necessário, de forma que não é possível aplicar corretamente o método da curva de reação de Ziegler-Nichols.

Já o método do ganho limite é desenvolvido a partir de uma malha fechada que coloca o sistema em oscilação harmônica. Ter o sistema oscilando de forma harmônica significa dizer que existe um ganho crítico  $K_{cr}$  que torna a planta marginalmente estável, ou seja, oscilando infinitamente ao redor de um valor. Como visto na [seção 4.3](#), em malha aberta, todos os polos da função de transferência estudada estão na origem (componentes real e imaginária nulas). Dessa forma, apesar de marginalmente estável, sua resposta a uma entrada em degrau não é oscilatória.

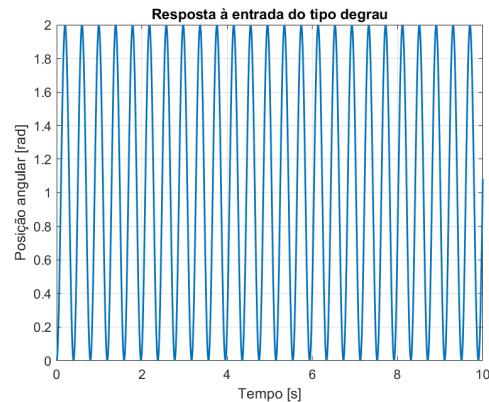
Por outro lado, a [Figura 25a](#) mostra que qualquer ganho  $K$  leva o sistema a produzir respostas harmônicas a um degrau unitário, uma vez que seu lugar das raízes é restrito ao eixo imaginário (componente real nula).

Figura 25 – Sistema sob ganho crítico.

- (a) Lugar das raízes para o sistema em malha aberta      (b) Resposta do sistema frente a um degrau quando utilizado  $K_{crítico} = 0,75$



Fonte: autoria própria.



Fonte: autoria própria.

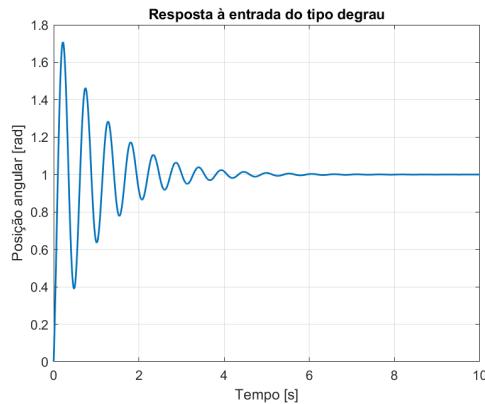
Nesse sentido, por definição, qualquer valor de  $K$  é válido como ganho crítico. Portanto, a escolha deste parâmetro foi feita de forma iterativa, visando uma resposta adequada (pouco *overshoot* e tempo baixo de estabilização), sem gerar polos de norma elevada e utilizar ganhos muito elevados ( $K_p$ ,  $K_i$  e  $K_d$ ). Tais precauções visam, principalmente, evitar a saturação do atuador estudado e a propagação de sinal ruidoso.

Ao fim, arbitrou-se  $K_{crítico} = 2,5$ , levando o sistema à resposta oscilatória ilustrada na [Figura 25b](#) com período crítico de  $P_{crítico} = 0,39$  s. Dessa forma, utilizando as equações de *Ziegler-Nichols* para a síntese de um controlador PID, chega-se em  $K_p = 1,5$ ,  $T_i = 0,195$  e  $T_d = 0,0488$ . A [Equação 4.3](#) mostra os ganhos obtidos a partir destes parâmetros.

$$\begin{cases} K_p = 1,5 \\ K_i = 7,6923 \\ K_d = 0,0731 \end{cases} \quad (4.3)$$

Na [Figura 26](#), é possível verificar que foi possível obter uma resposta com tempo de estabilização de 3,98 segundos - valor razoável para o sistema -, mas um *overshoot* de 70,86%, indicando que esta técnica de sintonização não é a mais adequada.

Figura 26 – Resposta do sistema ao degrau com PID sintonizado por *Ziegler-Nichols*



Fonte: Autoria própria.

#### 4.4.2 Método de alocação de polos

De forma análoga ao método utilizado para o controle moderno, o método é fundamentado na escolha de polos desejados para a malha fechada, para assim determinar os ganhos proporcional, integral e integrativo, que constituem o controlador PID. Assim, o procedimento matemático para a obtenção dos ganhos consiste em igualar as equações características do sistema em malha fechada e a desejada, de acordo com os polos alocados.

Dessa forma, a partir da função de transferência do sistema em malha fechada, dada na [Equação 4.4](#), define-se a equação característica, representada na [Equação 4.5](#).

$$T = \frac{G_C G}{1 + G_C G} \quad (4.4)$$

$$1 + G_C G = 0 \quad (4.5)$$

Assumindo-se  $G = N/D$  e manipulando-se a [Equação 4.5](#), obtém-se a [Equação 4.6](#).

$$Ds + (K_I + K_P s + K_D s^2) N = 0 \quad (4.6)$$

Substituindo-se os termos da função de transferência de malha aberta, dada na Equação 4.2, obtém-se uma equação de 7<sup>a</sup> ordem, ilustrada na Equação 4.7, sendo  $d$  e  $n$  os coeficiente das equações do denominador e numerador, respectivamente.

$$K_i n_0 + (d_0 + K_p n_0 + K_i n_1) s + (d_1 + K_d n_0 + K_p n_1 + K_i n_2) s^2 + (d_2 + K_d n_1 + K_p n_2 + K_i n_3) s^3 + (d_3 + K_d n_2 + K_p n_3 + K_i n_4) s^4 + (d_4 + K_d n_3 + K_p n_4) s^5 + (d_5 + K_d n_4) s^6 + d_6 s^7 = 0 \quad (4.7)$$

Além disso, a equação característica desejada assume o formato observado na Equação 4.8, sendo  $P$  os polos alocados.

$$(s + P_1)(s + P_2) \dots (s + P_7) = 0 \quad (4.8)$$

Ao igualar as Equações 4.7 e 4.8 para determinar  $K_p$ ,  $K_i$  e  $K_d$ , obtém-se um sistema de equações sobre-determinado, com 3 incógnitas e 7 equações. Assim, o método natural para resolução desse sistema seria o método dos mínimos quadrados.

No entanto, para a função de transferência estudada, apenas dois termos são não nulos:  $d_6$  e  $n_4$ . Com isso, o sistema de equações é reduzido a três equações, de forma que os ganhos podem ser calculados de forma analítica. As Equações D.1, D.2 e D.3, descritas no Apêndice D, determinam os ganhos conforme os polos alocados.

Dessa forma, como uma característica da função de transferência é cancelar polos, para que a alocação de polos seja satisfeita, são definidos 2 polos não nulos e 5 nulos, como ilustra a Tabela 6.

Tabela 6 – Polos alocados - PID.

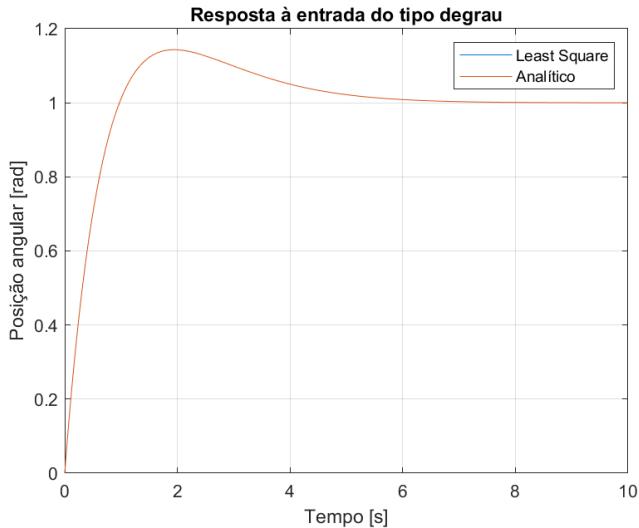
Polo	Valor
$p_1$ e $p_2$	$-1 \pm 0.3i$
$p_3, p_4, p_5, p_6$ e $p_7$	0

Fonte: autoria própria.

Com isso, são obtidos os ganhos observados na Equação 4.9. Nota-se que o integrativo é nulo, nesse caso, para os polos escolhidos, um controlador do tipo PD é suficiente. Destaca-se que os valores obtidos pelo método analítico é igual ao obtido pelo método dos mínimos quadrados.

$$\begin{cases} K_p = 0,01081 \\ K_i = 0 \\ K_d = 0,01984 \end{cases} \quad (4.9)$$

A resposta a uma entrada do tipo degrau para a função de transferência abordada é ilustrada na Figura 27.

Figura 27 – Posição angular  $q_3$  para degrau aplicado na entrada  $T_{rr3}$  - alocação de polos

Fonte: autoria própria.

Nesse caso, nota-se uma resposta bem comportada do sistema, com um *overshoot* único de menos de 20% do valor final e com tempo de estabilização em torno de 7 s.

#### 4.4.3 Método de controle ótimo - ITAE

À semelhança do Controle Moderno, em que o regulador LQR visa à otimização, o Controle Clássico também possui uma técnica de controle ótimo. Para se obter a sintonia ótima, são utilizados índices de desempenho, que avaliam os erros em simulação. Os mais comuns são do tipo integral, isto é, avaliam a integral do erro da simulação. Alguns dos mais relevantes são o ISE (*Integral Square Error*), IAE (*Integral Absolute Error*), ITSE (*Integral Time Square Error*) e ITAE (*Integral Time Absolute Error*), foco da análise a seguir e sintetizado pela expressão da [Equação 4.10](#).

$$ITAE = \int_0^T t |e(t)| dt \quad (4.10)$$

O índice ITAE atua na redução do erro em regime transitório e permanente. É necessário obter, *a priori*, a Função de Transferência de Malha Fechada com o controlador PID, de modo a se obter a equação característica do sistema em MF e, então, compará-la aos coeficientes para um ITAE com entrada degrau. Conforme o procedimento da [Equação 4.7](#), obtém-se uma equação característica da forma

$$s^4 (s^3 + K_d a s^2 + K_p a s + K_i a) \quad (4.11)$$

em que  $a = 100,7843$  é obtido a partir da [Equação 4.2](#). Assim, a segunda parcela pode ser igualada ao polinômio de 3º grau da tabela ITAE para uma entrada degrau, explicitado

na [Equação 4.12](#).

$$s^3 + 1,783\omega_n s^2 + 2,172\omega_n^2 s + \omega_n^3 \quad (4.12)$$

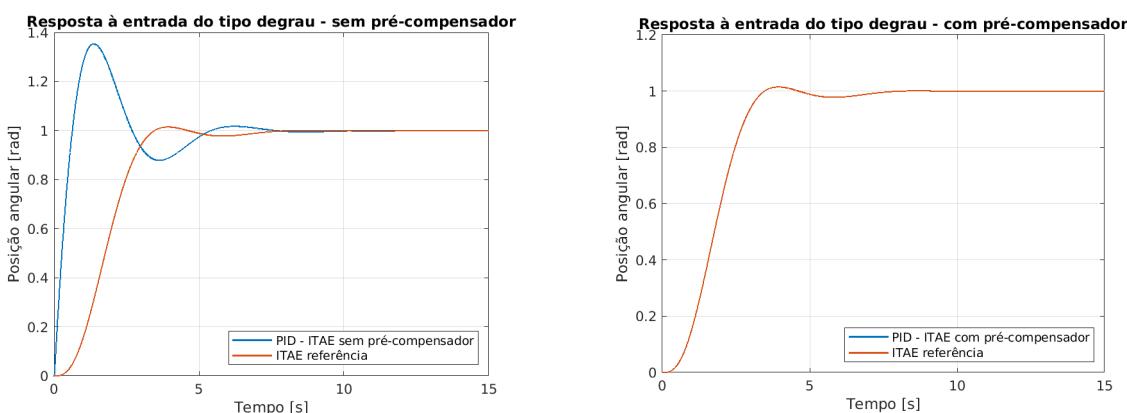
Por meio de testes preliminares, estipulou-se que o valor de frequência natural do sistema em Malha Fechada adequado seria  $\omega_n = 1,2 \text{ rad/s}$ . Assim, os valores de ganhos obtidos com a resolução do sistema estão sintetizados na [Equação 4.13](#).

$$\begin{cases} K_p = 0,0310 \\ K_i = 0,0171 \\ K_d = 0,0212 \end{cases} \quad (4.13)$$

Pôde-se simular, por fim, o sistema. Para tanto, aplicou-se uma entrada do tipo degrau ao PID sem pré-compensador, cujo resultado está na [Figura 28a](#). Posteriormente, construiu-se um filtro pré-compensador de modo a se cancelar os zeros do sistema introduzidos pela sintonia PID, com resposta mostrada na [Figura 28b](#). Em ambas as análises, incluiu-se uma curva denominada “ITAE referência”, que indica a função de transferência com coeficientes ótimos ITAE para MF e entrada do tipo degrau.

Figura 28 – Posição angular  $q_3$  para degrau aplicado na entrada  $T_{rr3}$  - ITAE.

(a) PID sintonizado sem pré-compensador.      (b) PID sintonizado com pré-compensador.



Fonte: autoria própria.

Para o gráfico da [Figura 28a](#), nota-se um *overshoot* de aproximadamente 40%, valor consideravelmente alto. Há, ainda, um tempo de estabilização de aproximadamente 10 segundos, que é ligeiramente superior ao obtido com a Alocação de Polos, conforme a [Figura 27](#). Quando comparado à referência, nota-se que o PID apresenta um *overshoot* muito maior, com mais oscilações e de maior amplitude.

Já com o uso do pré-compensador, há uma resposta de comportamento melhor que o descrito anteriormente. O *overshoot* é de apenas 8%, com tempo de estabilização próximo a 8 segundos. Ademais, nota-se a sobreposição das curvas do controlador modelado e do

controlador ótimo. Isso se deve ao fato de que o procedimento de igualar os coeficientes do polinômio ITAE da tabela aos coeficientes do problema resultou num problema determinado. Isto é, havia um número de incógnitas igual ao de equações, o que permitiu a obtenção do denominador ótimo. Assim, utilizando-se o procedimento de eliminar os zeros com o filtro, obteve-se o resultado indicado na [Figura 28b](#), considerado bastante satisfatório.

#### 4.4.4 Método do Lugar das Raízes (*Root Locus*)

Além do método de *Ziegler-Nichols* por ganho crítico, outro método de sintonia PID que utiliza o diagrama de lugar das raízes como principal ferramenta é o método que recebe o mesmo nome. Porém, enquanto o algoritmo de Ziegler e de Nichols utiliza a representação gráfica com o objetivo de identificar o ganho que leva a planta à marginalidade da sua estabilidade, este consiste na análise de sucessivos diagramas em malha aberta visando a escolha das constantes  $K_p$ ,  $K_i$  e  $K_d$  que fornecem os melhores resultados para o sistema, ou seja, maior estabilidade.

Para tanto, parte-se do lugar das raízes da função de transferência estudada e define-se o ganho proporcional do controlador. Uma vez definido  $K_p$ , ele é incorporado, de forma adequada, à função de transferência de malha aberta e segue-se para a análise do diagrama de lugar das raízes para a determinação do próximo ganho,  $K_i$ , e, posteriormente,  $K_d$ .

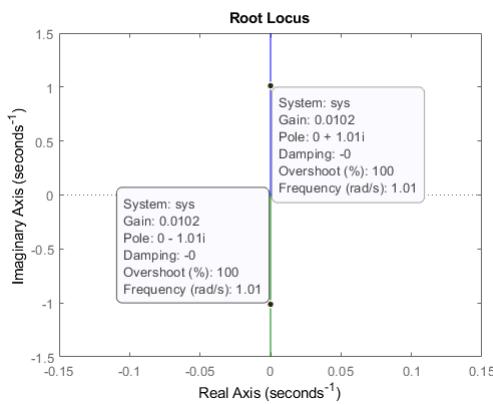
Com o objetivo de melhorar o desempenho do controlador, foi, ainda, utilizado um compensador de avanço, cuja função de transferência é  $\frac{3,7s+2,184}{s+2,118}$ . O efeito deste elemento, como ilustrado na [Figura 29](#), é, ao introduzir um polo e um zero à função de transferência, forçar a passagem do lugar das raízes dos polos dominantes por um ponto localizado no semi-plano de estabilidade. Isso melhora diretamente o desempenho do controlador proporcional.

A [Figura 29](#) evidencia a escolha das constantes de ganho proporcional para os controladores com e sem compensador acoplado. Para o caso sem compensador, uma vez que o lugar das raízes é restrito ao eixo imaginário, arbitrou-se  $K_p = 0,01$  com o objetivo de impor ao sistema polos  $0 \pm 1i$  (de norma não muito alta para evitar a necessidade de ganhos integrativo e derivativo excessivos).

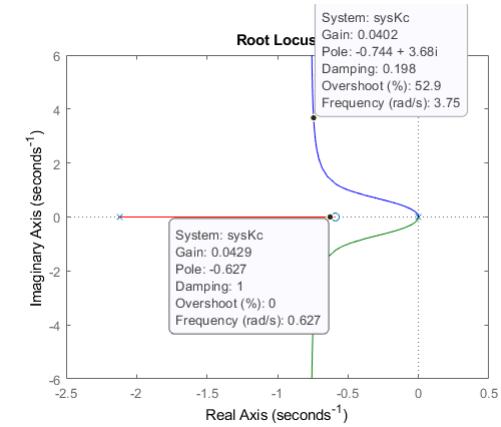
Quanto ao caso em que foi utilizado um pré-compensador de avanço, optou-se por um ganho  $K_p = 0,04$ , por garantir a maior distância possível dos polos dominantes do eixo imaginário e evitar que o polo do compensador fique muito a esquerda.

Figura 29 – Diagramas de lugar das raízes para a sintonia do ganho proporcional

(a) Sintonia sem o uso de compensador



(b) Sintonia com o uso de compensador



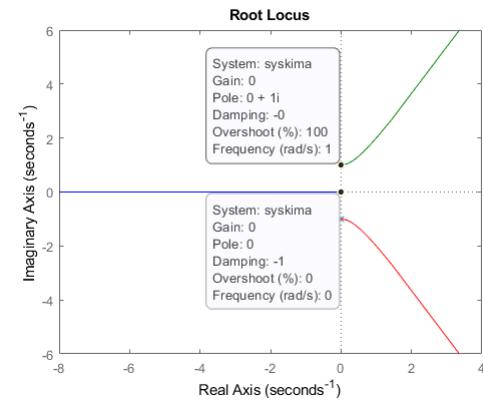
Fonte: autoria própria.

Na [Figura 30](#), é possível observar o lugar das raízes utilizado para a determinação de  $K_i$ . Nota-se que, para o sistema sem compensador, qualquer valor de  $K_i$  maior do que 0 o instabiliza. Portanto, optou-se por um ganho integrativo nulo.

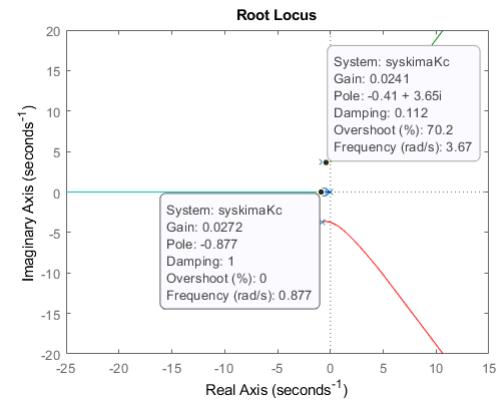
A [Figura 30b](#) mostra que, novamente, o compensador traz certa margem de estabilidade para o sistema. Nesse caso, o efeito deste componente se traduz na possibilidade de utilizar ganhos integrativos maiores do que 0 sem instabilizar a planta. Optou-se, portanto, por um  $K_i = 0,025$  - grande o suficiente para trazer o polo em 0 para o semiplano de estabilidade, mas não ao ponto dos outros ficarem com componentes reais positivas-.

Figura 30 – Diagramas de lugar das raízes para a sintonia do ganho integrativo

(a) Sintonia sem o uso de compensador



(b) Sintonia com o uso de compensador

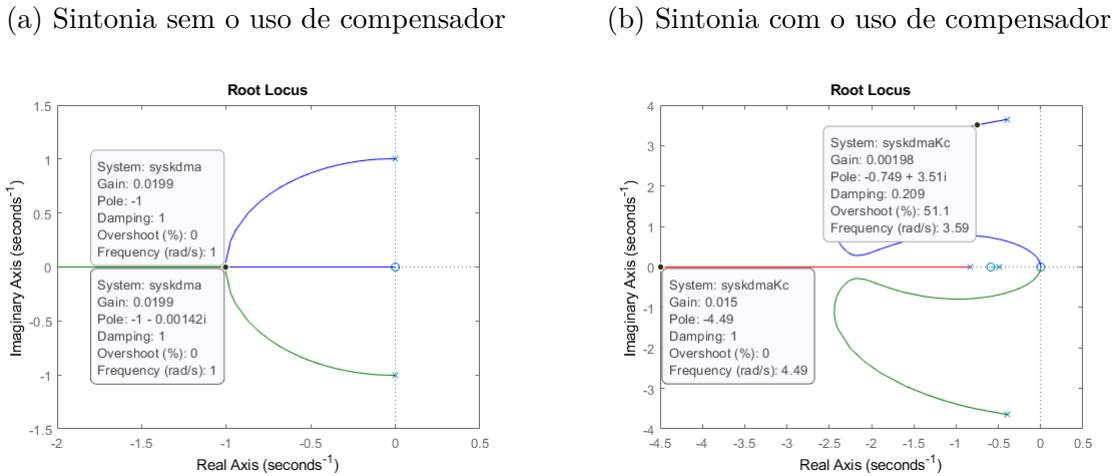


Fonte: autoria própria.

Finalmente, para a sintonia do ganho derivativo, no controlador sem compensador, escolheu-se o que garantia a maior distância dos dois polos dominantes do eixo imaginário.

Assim,  $K_d = 0,0199$ . Já para o sistema com compensador,  $K_d = 0,02$ . A Figura 31 ilustra os diagramas de lugar das raízes utilizados nessa etapa.

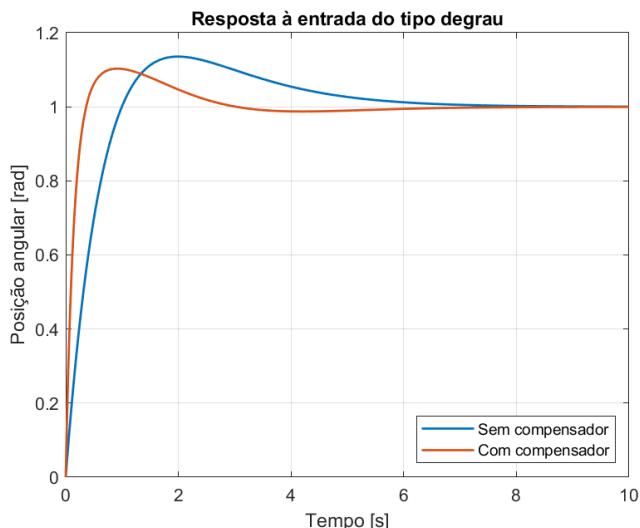
Figura 31 – Diagramas de lugar das raízes para a sintonia do ganho derivativo



Fonte: autoria própria.

Definidos todos os ganhos, é possível finalmente sintetizar os controladores PID's (PD no caso sem compensador) e compará-los. A Figura 32 mostra o desempenho deles frente a uma entrada do tipo degrau. Nota-se, pela curva laranja, que o uso do compensador permite ao sistema uma resposta com *overshoot* de 10,30% e tempo de estabilização de 2,48 s. Já a curva azul indica um resultado inferior do sistema sem compensador, com sobressinal de 13,55% e acomodação em 5,37 s.

Figura 32 – Sobreposição das respostas a degrau com PID's sincronizados por lugar das raízes para sistemas com e sem compensador



Fonte: autoria própria.

## 4.5 Análise da estabilidade relativa e resposta em frequência

A fim de se comparar o quanto estável um sistema é em comparação a outro, são utilizadas as ferramentas de margens de estabilidade, com margem de ganho e fase. Desse modo, podem-se equiparar os quatro controladores sintetizados na [seção 4.4](#). Em seguida, a resposta em frequência dos sintonizadores é analisada.

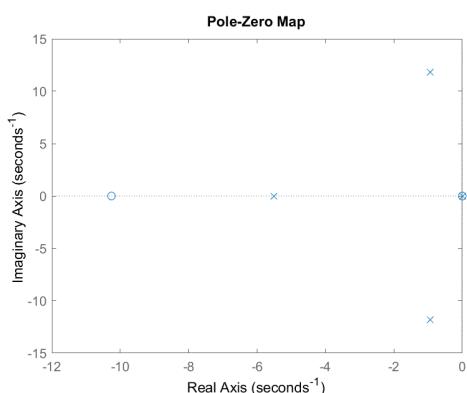
### 4.5.1 Sistema sintonizado por Ziegler-Nichols (ganho crítico)

A partir do diagrama de polos e zeros ilustrado na [Figura 33a](#) é possível verificar a presença de 3 polos estáveis e 4 marginalmente estáveis. Ainda, nota-se que o sistema não apresenta polos no semiplano à direita do eixo imaginário, atestando sua estabilidade.

Já a [Figura 33b](#) mostra que o sistema possui margem de ganho infinita, o que é um indicativo de boa estabilidade relativa. Verifica-se ainda uma margem de fase de 57,6 graus.

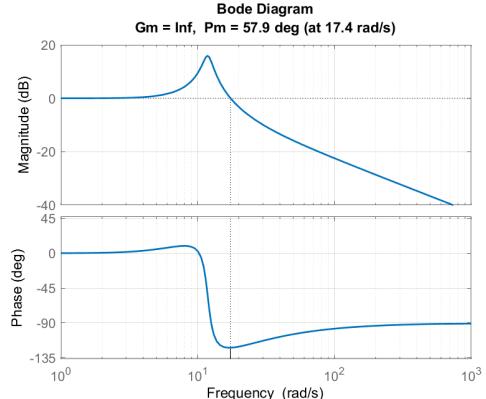
Figura 33 – Análise de estabilidade - *Ziegler-Nichols*.

(a) Mapa de polos e zeros.



Fonte: autoria própria.

(b) Diagrama de ganho e fase.



Fonte: autoria própria.

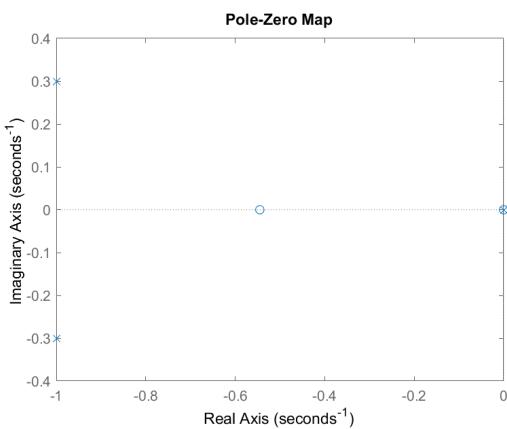
### 4.5.2 Sistema sintonizado por Alocação de Polos

A partir da [Figura 34a](#), é possível observar que o sistema obtido com o controlador PID (ou PD) por alocação de polos apresenta polos de malha aberta estáveis e marginalmente estáveis. Os polos alocados são naturalmente estáveis, como mostra o diagrama, ao passo que os polos nulos determinam a segunda condição descrita.

Ademais, em relação ao diagrama de fase e ganho, ilustrado na [Figura 34b](#), observa-se que a margem de ganho do sistema é infinita, indicando que o sistema apresenta uma boa estabilidade relativa, uma vez que a fase nunca atinge  $-180^\circ$ . Além disso, quando o ganho atinge valor nulo, há uma margem de aproximadamente  $41^\circ$ , indicando estabilidade, novamente.

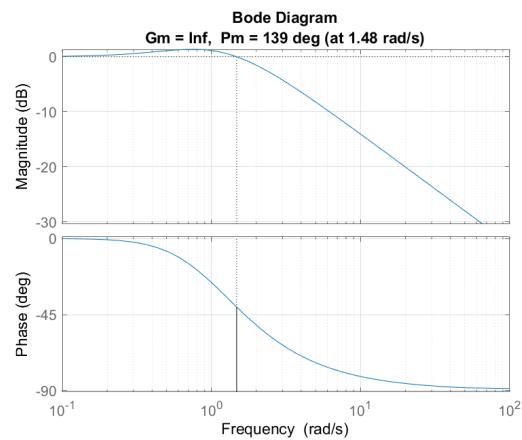
Figura 34 – Análise de estabilidade - Alocação de Polos.

(a) Mapa de polos e zeros.



Fonte: autoria própria.

(b) Diagrama de ganho e fase.



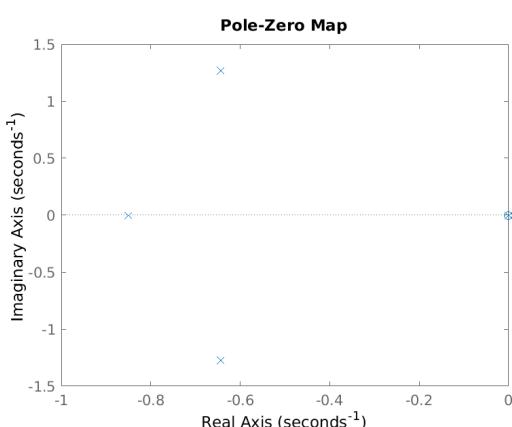
Fonte: autoria própria.

#### 4.5.3 Sistema sintonizado por controle ótimo ITAE

A análise a seguir é feita para o sistema com pré-compensador. Desse modo, um primeiro fator essencial é a ausência de zeros na Figura 35a, exceto os contidos na origem. Tal fator é esperado, tendo em vista que a concepção de tal filtro tem como pré-requisito o cancelamento de zeros introduzidos pelo PID, de modo a aproximar o sistema do controlador ótimo ITAE. Ademais, em relação aos polos, notam-se 4 na origem, além de outros 3 com parte real negativa, não havendo, portanto, polos instáveis. Em relação à Figura 35b, nota-se uma margem de ganho positiva, o que é um indicativo da estabilidade. Ademais, quando o ganho é nulo, a fase também é nula e, portanto, a margem de fase é  $PM = 0^\circ - (-180^\circ) = 180^\circ$ , valor também positivo que corrobora a estabilidade.

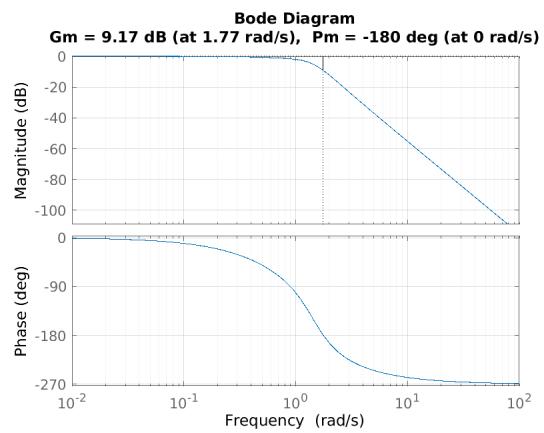
Figura 35 – Análise de estabilidade - ITAE.

(a) Mapa de polos e zeros.



Fonte: autoria própria.

(b) Diagrama de ganho e fase.



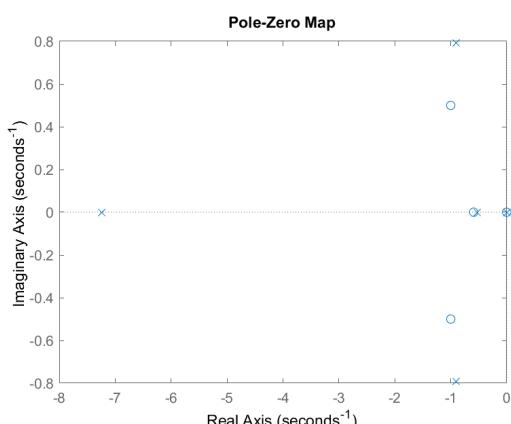
Fonte: autoria própria.

#### 4.5.4 Sistema sintonizado por Lugar das Raízes

Para o sistema com PID sintonizado por lugar das raízes, optou-se por analisar o caso de melhor desempenho, ou seja, com compensador. A [Figura 36a](#) mostra 4 polos estáveis, 4 na origem e nenhum instável. Além disso o diagrama de margens de ganho e fase ([Figura 36b](#)) indica margem de ganho infinita e de 154 graus para a fase, ambos indicando elevado grau de estabilidade relativa.

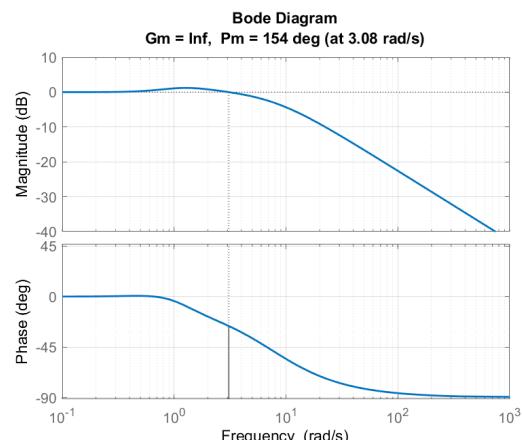
[Figura 36 – Análise de estabilidade - Lugar das Raízes.](#)

(a) Mapa de polos e zeros.



Fonte: autoria própria.

(b) Diagrama de ganho e fase.



Fonte: autoria própria.

### 4.6 Seguidor de referência e rejeitador de distúrbio

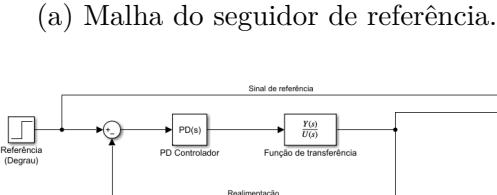
Para a abordagem de controle clássico, o objetivo determinado foi o controle da posição angular do satélite. Assim, realiza-se de forma análoga, a partir função de transferência que relaciona a componente de posição angular  $q_3$  com o torque proveniente da terceira roda de reação  $T_{rr3}$ .

Sendo assim, ao considerar que a principal função de um sistema de controle de atitude de um satélite é sua a estabilização e o apontamento, analisa-se a capacidade do controlador PID elaborado em cumprir essas atividades.

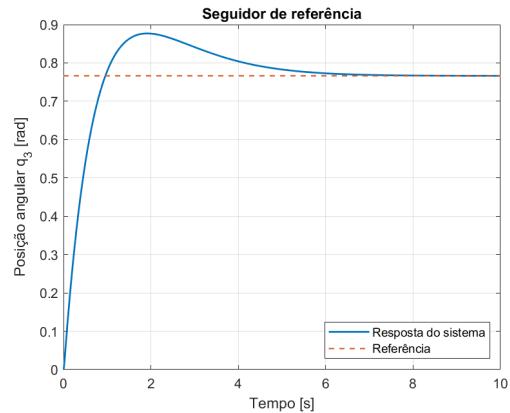
#### 4.6.1 Seguidor de referência

Inicialmente, estuda-se o problema de apontamento do satélite, para isso, determina-se uma referência constante a ser seguida. Portanto, considerando uma rotação própria  $\phi = \pi/3$ , obtém-se em quatérnios  $q_3 = 0,7663$ . Assim, através de uma entrada do tipo degrau, ilustrada na [Figura 37a](#), o sistema atinge o equilíbrio em torno da referência determinada em aproximadamente  $t = 7$  s, como mostra a [Figura 37b](#).

Figura 37 – Seguidor de referência.

(b) Posição angular  $q_3$ .

Fonte: autoria própria.



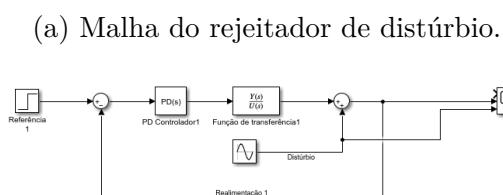
Fonte: autoria própria.

#### 4.6.2 Rejeitador de distúrbio

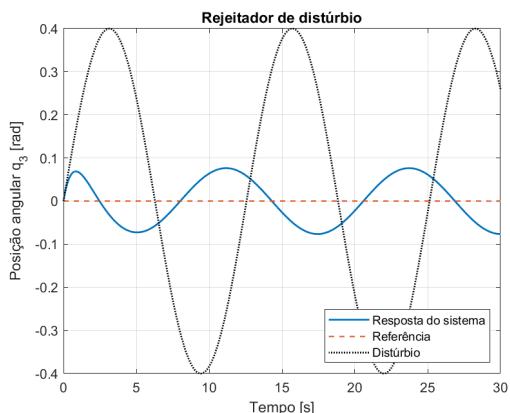
Para estudar a capacidade do controlador de rejeitar distúrbios, simulou-se a malha de controle elaborada sob à ação de um distúrbio senoidal. No caso, o sinal não apresenta qualquer significado físico, no entanto, julga-se importante analisar o sistema do ponto de vista de estabilidade, uma vez que o ambiente espacial impõe distintos torques ao satélite, enquanto atividades de apontamento estão sendo realizadas concomitantemente, seja para imageamento, seja para transmissão de dados de telemetria.

Assim, a partir da malha de controle observada na [Figura 38a](#), produziu-se os resultados ilustrados na [Figura 38b](#), para um função senoidal de amplitude 0,2 e frequência 0,4 rad/s. Com isso, foi possível observar que o sistema controlado foi capaz de reduzir a amplitude do distúrbio em menos de 1/4, um resultado bastante satisfatório, uma vez que os torques do ambiente espacial são de baixíssima ordem de grandeza.

Figura 38 – Seguidor de referência.

(b) Posição angular  $q_3$ .

Fonte: autoria própria.



Fonte: autoria própria.

## 4.7 Comparação dos controladores

É possível fazer uma análise comparativa quanto ao desempenho dos controladores sintetizados na [seção 4.4](#) a partir da [Figura 39a](#), que mostra a sobreposição de suas respostas a um degrau unitário.

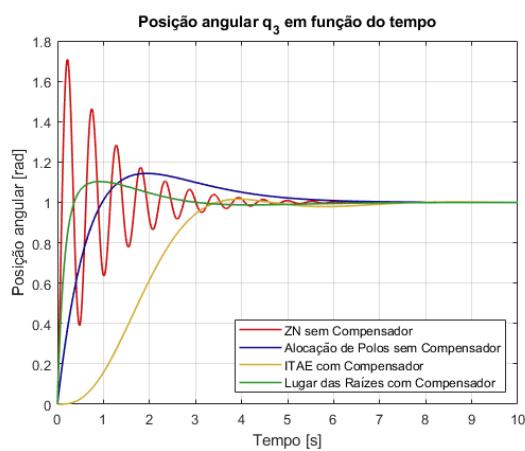
Quanto ao tempo de estabilização, destacam-se positivamente os métodos de sintonia de lugar das raízes e ITAE. Os dois controladores mais lentos, entretanto, acomodam-se com apenas 1 segundo de atraso.

Por outro lado, analisando o sobressinal dos resultados, é evidente que a sintonia por *Ziegler-Nichols*, com 70% de *overshoot*, é deficiente em relação aos demais. Os métodos de lugar das raízes e alocação de polos apresentam picos de magnitude semelhante (ambos da ordem de 15%). Finalmente, o controlador sintonizado por ITAE destaca-se por apresentar sobressinal quase nulo.

Levando em conta a pequena diferença entre os tempos de estabilização e o *overshoot* de pequena magnitude, conclui-se que a sintonia por ITAE foi a mais adequada para sintetizar um controlador para a função de transferência estudada. Outro ponto positivo desse controlador que pode ser identificado na [Figura 39a](#) é o fato de apresentar a menor derivada máxima. Ou seja, não há necessidade de aplicar grandes velocidades angulares no sistema para que ele responda ao degrau, o que reduz a demanda do atuador responsável pelo controle da componente  $q_3$ .

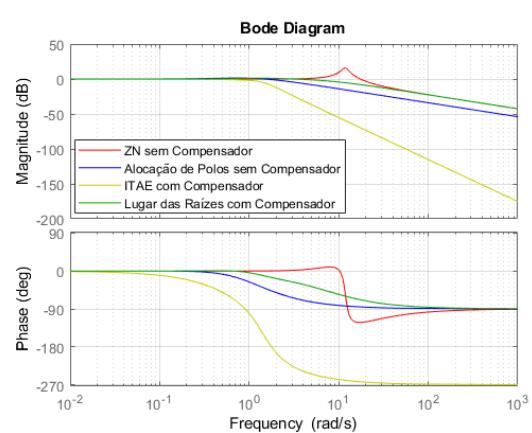
Figura 39 – Comparação entre métodos de sintonia PID.

(a) Resposta a um degrau unitário.



Fonte: autoria própria.

(b) Diagrama de bode.



Fonte: autoria própria.

Já a [Figura 39b](#) apresenta a sobreposição das respostas em frequência dos sistemas em malha fechada. Nota-se que os sistemas controlados por *Ziegler-Nichols*, alocação de polos e lugar das raízes, em altas frequências, comportam-se como de primeiro grau (decaimento de 90 graus para a fase e de 20 dBs por década para a magnitude). É

evidente ainda o pico de ganho na curva de *Ziegler-Nichols*, indicando a presença de polos sub-amortecidos, o que explica as altas oscilações vistas na [Figura 39a](#).

## 4.8 Redução de ordem

O principal objetivo ao reduzir a ordem de um sistema dinâmico é simplificá-lo a nível algébrico. Essa simplificação se traduz, no contexto de controle, na redução do tempo de processamento do sistema de controladores. O principal problema dessa técnica, entretanto, é a inerente adição de erros aos sinais. Dessa forma, é importante fazer uma análise minuciosa do custo-benefício ao reduzir a ordem de um sistema.

Para a função de transferência estudada, não faz sentido aplicar a redução de ordem. Isso pois, como constatado na [seção 4.2](#), seus polos são todos iguais, dificultando a análise de relevância que cada um possui no sistema e inviabilizando o procedimento pelo método da norma de Hankel.

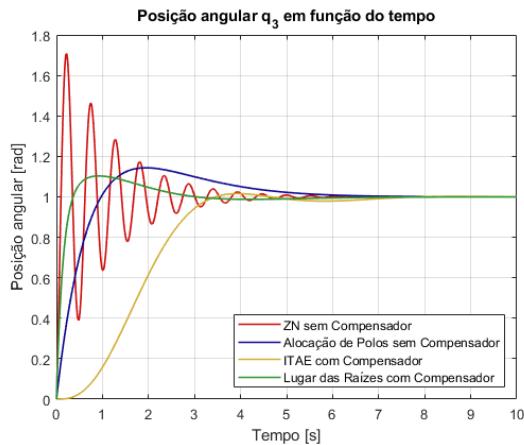
Por outro lado, com a síntese dos controladores PID's desenvolvida na [seção 4.4](#), os polos do sistema controlado ficam espalhados pelo semiplano real negativo. Dessa forma, a redução de ordem torna-se viável.

Uma tentativa de redução de ordem por norma de Hankel foi realizada para os controladores sintetizados com cada uma das metodologias de sintonia apresentadas na [seção 4.4](#) (para ITAE e lugar das raízes, foi considerado apenas o caso com pré-compensador). Em todos os casos, foi possível simplificar apenas um polo (simplificar mais polos geraria erros grandes a ponto de descaracterizar o sistema completamente).

A [Figura 40a](#) mostra as respostas a uma entrada do tipo degrau obtidas pelos sistemas originais, enquanto a [Figura 40b](#), pelos sistemas após a simplificação de um polo. Nota-se que o procedimento não gerou bons resultados no geral. Um erro de 20% em regime permanente foi produzido em 3 das 4 curvas na [Figura 40b](#). Entretanto, o sistema sintonizado por lugar das raízes não apresentou erros de grande magnitude.

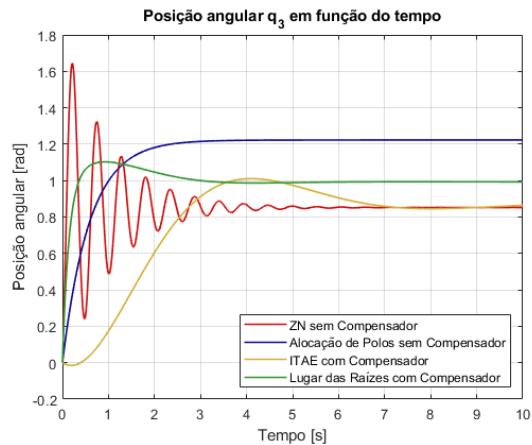
Figura 40 – Comparação de respostas a degrau de diferentes métodos de sintonia PID.

(a) Sem redução de ordem.



Fonte: autoria própria.

(b) Com redução de ordem.

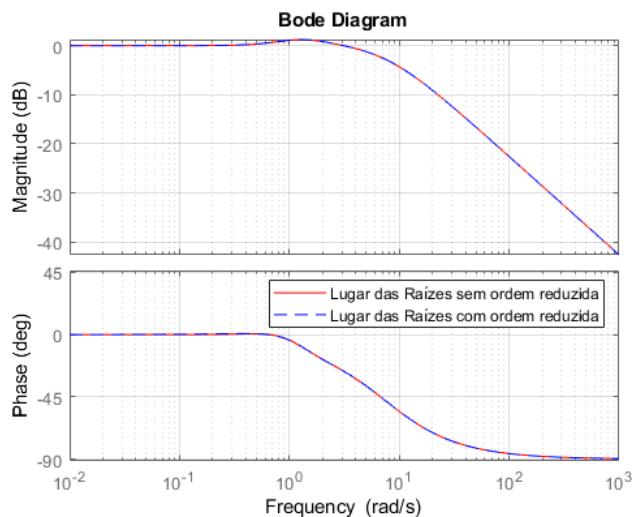


Fonte: autoria própria.

Na [Figura 41](#), verifica-se que, de fato, a redução de ordem no sistema controlado por lugar das raízes não o descaracteriza. Pelo contrário, fica evidente a sobreposição quase perfeita dos diagramas de bode tanto para fase quanto para magnitude por toda a banda de frequências analisadas.

Conclui-se, portanto, que a função de transferência obtida a partir do controle sintonizado por lugar das raízes possuía um polo de baixa influência. Assim, neste caso, verificado o baixo custo de fazê-lo (em termos de exatidão do sinal manuseado), é vantajoso reduzir a ordem do sistema.

Figura 41 – Comparação entre os diagramas de bode para lugar das raízes com e sem ordem reduzida.



Fonte: autoria própria.

## 5 Conclusão

O projeto de controle da atitude do *CubeSat* teve como ponto fundamental de partida a análise do sistema físico real, o PicSat, incluindo parâmetros físicos, sensores e atuadores. Em seguida, foi necessária a simplificação do modelo, utilizando-se hipóteses oriundas, sobretudo, da literatura.

Dispondo-se de um modelo físico e matemático robusto, foi possível utilizar as técnicas de controle moderno de modo a atuar sobre o sistema. Inicialmente, os métodos de alocação de polos e do Controlador Linear Quadrático mostraram-se satisfatórios quanto ao requisito de estabilizar o sistema ante a uma velocidade angular inicial do PicSat. No entanto, para a obtenção de alguns dos parâmetros essenciais à malha de controle, tais como os polos ou as matrizes  $[Q]$  e  $[P]$ , foram utilizados métodos de busca manuais, fato que não é almejável.

Ainda, dentro da abordagem de controle moderno, implementou-se o seguidor de referência constante, parte fundamental do controle de atitude de um satélite, uma vez que as atividades de apontamento são realizadas por meio desta técnica de controle linear. Além disso, projetou-se um rejeitador de distúrbios, com base nos torques externos do ambiente espacial, que tornou o sistema de controle mais robusto e estável.

Em seguida, estudou-se o sistema através da abordagem de controle clássico, que restringe o estudo do problema às suas funções de transferência. Com isso, em torno de uma das funções de transferência que relaciona uma entrada a uma posição, desenvolveu-se diferentes análises em malha aberta, para os polos e zeros, estabilidade, tipo, resposta a uma entrada do tipo degrau. Dessa forma, observou-se que a função escolhida, em razão das suas características, adequaria-se aos projetos de controle a serem sintetizados, com base nas margens de ganhos e fase disponíveis.

Desse modo, foram desenvolvidos diferentes controladores PID com base nos métodos de sintonização estudados: métodos de Ziegler e Nichols, Alocação de Polos, ITAE e Lugar das raízes; cada um com suas particularidades e limitações. Além disso, utilizou-se uma estratégia de mudança de fase para os dois últimos métodos, através dos pré-compensadores. Assim, foram desenvolvidas diferentes análises e comparações entre os controladores.

Por fim, analisou-se os diferentes modelos de malha fechada elaborados, através dos diagramas de fase e ganho e mapas de polos e zeros. Ademais, foi possível desenvolver o projeto do seguidor de referência e rejeitador de distúrbios, que produziu resultados adequados aos objetivos do sistema.

# Referências

- Berlin Space Technologies. *IADCS-100 [cubesat components]*. 2021. Disponível em: <<https://www.berlin-space-tech.com/portfolio/iadcs/>>. Citado na página 7.
- CELANI, F. Quaternion versus rotation matrix feedback for spacecraft attitude stabilization using magnetorquers. *Aerospace*, v. 9, p. 24, 01 2022. Citado na página 14.
- CHIN, J. et al. *CubeSat101: Basic Concepts and Processes for First-Time CubeSat Developers*. [S.l.]: NASA CubeSat Launch Initiative, 2017. Citado na página 1.
- CROWELL, C. Development and analysis of a small satellite attitude determination and control system testbed. p. 267, 11 2011. Citado na página 1.
- FRIEDLAND, B. *Control System Design: An Introduction to State-Space Methods*. Dover Publications, 2012. (Dover Books on Electrical Engineering). ISBN 9780486135113. Disponível em: <<https://books.google.com.br/books?id=9WRKZlaCnF8C>>. Citado 3 vezes nas páginas 20, 21 e 23.
- 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*. 160 f. Dissertação (Mestrado em Ciências) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2020. Citado 10 vezes nas páginas 2, 4, 5, 8, 9, 10, 12, 13, 14 e 26.
- NOWAK, M. Short life and abrupt death of PicSat, a small 3U CubeSat dreaming of exoplanet detection. 2019. Citado 2 vezes nas páginas 1 e 6.
- NOWAK, M. The 2017 conjunction of Beta Pictoris b : the life and death of Picsat, followed by a VLTI/GRAVITY observation of the re-emergence. 2019. Citado 2 vezes nas páginas 1 e 6.
- OGATA, K. *Modern Control Engineering*. Prentice Hall, 2010. (Instrumentation and controls series). ISBN 9780136156734. Disponível em: <<https://books.google.com.br/books?id=Wu5GpNAelzkC>>. Citado na página 2.
- SANTOS, W. C.; AMORIM, R. Descobertas de exoplanetas pelo método do trânsito. *Revista Brasileira de Ensino de Física*, v. 39, n. 2, 2017. Citado na página 1.
- SIDI, M. J. *Spacecraft dynamics and control: a practical engineering approach*. [S.l.]: Cambridge University Press, 2012. (Cambridge Aerospace Series, v. 7). ISBN 978-0-521-78780-2. Citado na página 11.
- The PicSat Team. *Mission Goals*. 2016. Disponível em: <<https://picsat.obspm.fr/science/objectives?locale=en>>. Citado na página 1.

# A Observador de Estados por Alocação de Polos

A matriz de ganhos obtida com a alocação de polos, com auxílio do *Matlab*, foi:

$$[\mathbf{K}_o] = \begin{bmatrix} 5,8 & 0 & 0 \\ 0 & 7,206 & -0,056 \\ 0 & -0,540 & 7,594 \\ 16,8 & 0 & 0 \\ 0 & 24,771 & 1,590 \\ 0 & -5,495 & 27,784 \end{bmatrix} \quad (\text{A.1})$$

## B Seguidor de referência constante: matrizes

Matrizes obtidas, com os valores numéricos:

$$[\mathbf{N}_x] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad [\mathbf{N}_u] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (\text{B.1})$$

## C Seguidor de referência variável: matrizes de distúrbios e referências

As matrizes de distúrbios e de referências com os valores numéricos obtidas são:

$$[\mathbf{A}_w] = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3,35 & 2,87 & 462,7 & 0 & 0 & 0 \\ 0 & -1,35 \cdot 10^{-6} & 0 & 0 & 0 & 0 \\ -0,024 & -0,021 & -3,35 & 0 & 0 & 0 \end{bmatrix} \quad [\mathbf{A}_r] = \begin{bmatrix} 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{C.1})$$

## D Alocação de polos: solução analíticas dos ganhos

$$K_p = \frac{1}{n_4} (P_1 P_2 + P_1 P_3 + P_2 P_3 + P_1 P_4 + P_2 P_4 + P_3 P_4 + P_1 P_5 + P_2 P_5 + P_3 P_5 + P_4 P_5 + P_2 P_6 + P_2 P_6 + P_3 P_6 + P_4 P_6 + P_5 P_6 + P_1 P_7 + P_2 P_7 + P_3 P_7 + P_4 P_7 + P_5 P_7 + P_6 P_7) \quad (\text{D.1})$$

$$K_i = \frac{1}{n_4} (-P_1 P_2 P_3 - P_1 P_2 P_4 - P_1 P_3 P_4 - P_2 P_3 P_4 - P_1 P_2 P_5 - P_1 P_3 P_5 - P_2 P_3 P_5 - P_1 P_4 P_5 - P_2 P_4 P_5 - P_3 P_4 P_5 - P_1 P_5 P_6 - P_2 P_5 P_6 - P_3 P_5 P_6 - P_4 P_5 P_6 - P_1 P_2 P_7 - P_1 P_3 P_7 - P_2 P_3 P_7 - P_1 P_4 P_7 - P_2 P_4 P_7 - P_3 P_4 P_7 - P_1 P_5 P_7 - P_2 P_5 P_7 - P_3 P_5 P_7 - P_4 P_5 P_7 - P_1 P_6 P_7 - P_2 P_6 P_7 - P_3 P_6 P_7 - P_4 P_6 P_7 - P_5 P_6 P_7) \quad (\text{D.2})$$

$$K_d = \frac{-P_1 - P_2 - P_3 - P_4 - P_5 - P_6 - P_7}{n_4} \quad (\text{D.3})$$

# E Modelo Matemático: Códigos das simulações

## E.1 Modelos cinemático e dinâmico

```

1 ClearAll["Global`*"]
2
3 (* Modelo do distúrbio do dipolo magnético*)
4 m = {\[Minus]0.14, \[Minus]0.12, \[Minus]19.32}*10^-3;
5 Qmi = {{q0[t]^2 + q1[t]^2 - q2[t]^2 - q3[t]^2,
6   2*(q1[t]*q2[t] + q0[t]*q3[t]),
7   2*(q1[t]*q3[t] - q0[t]*q2[t])}, {2*(q1[t]*q2[t] - q0[t]*q3[t]),
8   q0[t]^2 - q1[t]^2 + q2[t]^2 - q3[t]^2,
9   2*(q2[t]*q3[t] + q0[t]*q1[t])}, {2*(q1[t]*q3[t] + q0[t]*q2[t]),
10  2*(q2[t]*q3[t] - q0[t]*q1[t]),
11  q0[t]^2 - q1[t]^2 - q2[t]^2 + q3[t]^2}};
12 B = {B1[t], B2[t], B3[t]};
13 Tpp[t_] := Cross[m, Qmi.B];
14 w = {B1[t], B2[t], B3[t]};
15
16 (*Modelo dinâmico do satélite – RCS = REP*)
17 Ip = {{I1, 0, 0}, {0, I2, 0}, {0, 0, I3}};
18 wp[t_] := {wp1[t], wp2[t], wp3[t]};
19 Hrrp[t_] := {Hrrp1[t], Hrrp2[t], Hrrp3[t]};
20
21 TQMA := Ip.wp'[t] + wp[t]\[Cross](Ip.wp[t](*+Hrrp[t]*)) == -Hrrp'[t] +
22   Tpp[t]
23 sol1 = Solve[TQMA, {wp1'[t], wp2'[t], wp3'[t]}];
24
25 (*Modelo cinemático do satélite*)
26 q[t_] := {q0[t], q1[t], q2[t], q3[t]}
27 \[CapitalOmega][t_] := {{0, -wp1[t], -wp2[t], -wp3[t]},
28   {wp1[t], 0, wp3[t], -wp2[t]},
29   {wp2[t], -wp3[t], 0, wp1[t]},
30   {wp3[t], wp2[t], -wp1[t], 0}}
31 dq := q'[t] == 1/2*\[CapitalOmega][t].q[t]
32 sol2 = Solve[dq, {q0'[t], q1'[t], q2'[t], q3'[t]}]
33
34 f1 = q1'[t] /. sol2 [[1]];
35 f2 = q2'[t] /. sol2 [[1]];
36 f3 = q3'[t] /. sol2 [[1]];
37 f4 = wp1'[t] /. sol1 [[1]];
38 f5 = wp2'[t] /. sol1 [[1]];

```

```

39 f6 = wp3'[t] /. sol1 [[1]];
40 f = {f1, f2, f3, f4, f5, f6};
41
42 g1 = q1[t] /. sol2 [[1]];
43 g2 = q2[t] /. sol2 [[1]];
44 g3 = q3[t] /. sol2 [[1]];
45 g4 = wp1[t] /. sol1 [[1]];
46 g5 = wp2[t] /. sol1 [[1]];
47 g6 = wp3[t] /. sol1 [[1]];
48 g = {g1, g2, g3, g4, g5, g6};
49
50 x = {q1[t], q2[t], q3[t], wp1[t], wp2[t], wp3[t]};
51 u = {Hrrp1'[t], Hrrp2'[t], Hrrp3'[t]};
52
53 (*Espaço de estados – Não linearizado*)
54 A1 = ConstantArray[0, {Length[f], Length[x]}];
55
56 B1 = ConstantArray[0, {Length[f], Length[u]}];
57
58 C1 = ConstantArray[0, {Length[g], Length[x]}];
59
60 For[i = 1, i < Length[f] + 1, i++,
61   For[j = 1, j < Length[x] + 1, j++,
62     A1[[i, j]] = D[f[[i]], x[[j]]]
63   ]]
64 A1;
65 For[i = 1, i < Length[f] + 1, i++,
66   For[j = 1, j < Length[u] + 1, j++,
67     B1[[i, j]] = D[f[[i]], u[[j]]]
68   ]]
69 B1;
70 For[i = 1, i < Length[g] + 1, i++,
71   For[j = 1, j < Length[x] + 1, j++,
72     C1[[i, j]] = D[g[[i]], x[[j]]]
73   ]]
74 C1;
75 D1 = ConstantArray[0, {6, 3}];
76
77 (*Espaço de estado – Linearizado*)
78 A1lin = ConstantArray[0, {Length[f], Length[x]}];
79 B1lin = ConstantArray[0, {Length[f], Length[u]}];
80
81 For[i = 1, i <= Length[f], i++,
82   For[j = 1, j <= Length[x], j++ ,
83     A1lin[[i, j]] =
84       D[f[[i]], x[[j]]] /. {wp1[t] -> wp10, wp2[t] -> wp20,
85       wp3[t] -> wp30, q0[t] -> q00, q1[t] -> q10, q2[t] -> q20,

```

```

86 q3[t] -> q30}]];
87
88 For[i = 1, i <= Length[f], i++,
89 For[j = 1, j <= Length[u], j++ ,
90 B1lin[[i, j]] =
91 D[f[[i]], u[[j]]] /. {wp1[t] -> wp10, wp2[t] -> wp20,
92 wp3[t] -> wp30, q0[t] -> q00, q1[t] -> q10, q2[t] -> q20,
93 q3[t] -> q30, Hrrp1'[t] -> dHrrp10, Hrrp2'[t] -> dHrrp20,
94 Hrrp3'[t] -> dHrrp30]}
95
96 A1lin // MatrixForm
97 B1lin // MatrixForm;
98 C1 // MatrixForm;
99 D1 // MatrixForm;
```

```

1 (*Dados numéricos*)
2 IBRF = {{0.04, 0.0001, 0.0006}, {0.0001,
3 0.04, -0.001}, {0.0006, -0.001, 0.005}};
4 Ip = Eigenvalues[IBRF]
5 I1 = Ip [[1]];
6 I2 = Ip [[2]];
7 I3 = Ip [[3]];
8
9 (* Ponto de equilíbrio *)
10 wp10 = 0;
11 wp20 = 0;
12 wp30 = 0;
13 q00 = 1;
14 q10 = 0;
15 q20 = 0;
16 q30 = 0;
17 A1lin // MatrixForm
18 B1lin // MatrixForm
```

## E.2 Simulação em malha aberta

```

1 %% Sistema em Malha Aberta
2 clc
3 clear all
4 close all
5
6 %% Parâmetros do sistema
7 addpath('Matrizes/')
8
9 A = importdata('matrix_A1lin.txt');
10 B = -importdata('matrix_B1lin.txt');
11 C = importdata('matrix_C1.txt');
```

```
12 D = importdata('matrix_D1.txt');
13
14 dt = 0.005;
15 t_f = 3;
16
17 tstep = 0.5;
18 t = 0:dt:t_f;
19
20 %% Sistema CH2
21
22 sys = ss(A,B,C,0);
23
24 x0 = [0 0 0 0 0 0];
25
26 u = zeros(length(t),3);
27 u(1:tstep/dt,1) = 50e-3 * ones(tstep/dt,1);
28
29
30 [ys,ts,xbarra] = lsim(sys,u,t,x0);
31
32 figure(1);
33 plot(t,xbarra(:,1),LineWidth=1.3);
34 grid on
35 title ('Resposta à entrada do tipo pulso – Malha Aberta')
36 xlabel('Tempo [s]')
37 ylabel('Posição angular [rad]')
38 baseFileName = sprintf('Image_%s.png', "ma_step_q1_1");
39 fullFileName = fullfile ("Imagens/", baseFileName);
40 saveas(1, fullFileName);
41
42 figure(2);
43 plot(t,xbarra(:,4),LineWidth=1.3);
44 grid on
45 title ('Resposta à entrada do tipo pulso – Malha Aberta')
46 xlabel('Tempo [s]')
47 ylabel('Velocidade angular [rad/s]')
48 baseFileName = sprintf('Image_%s.png', "ma_step_q1_2");
49 fullFileName = fullfile ("Imagens/", baseFileName);
50 saveas(2, fullFileName);
51
52 % figure(3);
53 bodeplot(sys)
54
55 % bode_plot(A,B,C,D)
56
57 %% Sistema CH4
58
```

```
59
60 sys = ss(A,B,C,0);
61
62 x0 = [0 0 0 0 0];
63
64 u = zeros(length(t),3);
65 u(1:t_f/dt,3) = ones(t_f/dt,1);
66
67
68 [ys,ts,xbarra] = lsim(sys,u,t,x0);
69
70 figure(4);
71 plot(t,xbarra(:,3),LineWidth=1.3);
72 grid on
73 title ('Resposta à entrada do tipo degrau — Malha Aberta')
74 xlabel('Tempo [s]')
75 ylabel('Posição angular q_3 [rad]')
76 baseFileName = sprintf('Image_%s.png', "ma_degrau_q3");
77 fullFileName = fullfile ("Imagens/", baseFileName);
78 saveas(4, fullFileName);
79
80
81
82 num = [100.7843 0 0 0 0];
83 den = [1 0 0 0 0 0];
84 FT = tf(num,den);
85
86 figure(5);
87 margin(FT)
88 baseFileName = sprintf('Image_%s.png', "ma_degrau_margin");
89 fullFileName = fullfile ("Imagens/", baseFileName);
90 saveas(5, fullFileName);
```

# F Controle Moderno: Códigos das simulações

## F.1 Controlador por alocação de polos

```

1 clc
2 clear
3 close all
4 %% Parâmetros do sistema
5 addpath('Matrizes\'')
6 addpath("Imagens\Controle Moderno\")

7
8 passo=0.01;
9 t=0:passo:10;
10 u=ones(size(t));
11
12 A = importdata('matrix_A1lin.txt');
13 B = importdata('matrix_B1lin.txt');
14 C = importdata('matrix_C1.txt');
15 D = importdata('matrix_D1.txt');

16
17 %% Alocação dos polos
18 p = [-0.8+0.6i -0.8-0.6i -2.6+0.3i -2.6-0.3i -1. -0.8];
19 K = place(A,B,p);
20 B1 = -B*0.2;
21 sys_aloc = ss(A-B*K,B1,C,D);
22
23 %% Simulação do sistema
24 [y,t,x]=step(sys_aloc);
25
26 Torq1=-K*y(:,:,1)';
27 Torq2=-K*y(:,:,2)';
28 Torq3=-K*y(:,:,3)';
29 %% Plots
30 figure(1)
31 pzmap(sys_aloc)
32 grid on
33 baseFileName = sprintf('Image_%s.png', "aloc_pol");
34 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
35 saveas(1, fullFileName);
36
37 figure(3)
38 hold on
39 plot(t,y (:,4,2) ,LineWidth=1.20)
40 plot(t,y (:,5,2) ,LineWidth=1.20)

```

```
41 plot(t,y (:,6,2) ,LineWidth=1.20)
42 title ("Velocidade angular: degrau aplicado")
43 xlabel("Tempo [s]")
44 ylabel("Velocidade angular [rad/s]")
45 legend('w_1','w_2','w_3','Location','east')
46 hold off
47 grid on
48 baseFileName = sprintf('Image_%s.png', "aloc_vel");
49 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
50 saveas(3, fullFileName);
51
52 figure (4)
53 hold on
54 plot(t,y (:,1,2) ,LineWidth=1.20)
55 plot(t,y (:,2,2) ,LineWidth=1.20)
56 plot(t,y (:,3,2) ,LineWidth=1.20)
57 title ("Posição: degrau aplicado")
58 xlabel("Tempo [s]")
59 ylabel("Posição angular [rad]")
60 legend('q_1','q_2','q_3','Location', 'southeast')
61 hold off
62 grid on
63 baseFileName = sprintf('Image_%s.png', "aloc_pos");
64 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
65 saveas(4, fullFileName);
66
67 figure (5)
68 plot(t,Torq1(1,:))
69 hold on
70 plot(t,Torq2(2,:))
71 plot(t,Torq3(3,:))
72 ylabel('Torque N.m')
73 legend('T_1','T_2','T_3','Location','east')
74 hold off
75 grid on
76
77 figure (3)
78 hold on
79 plot(t,y (:,4,1) ,LineWidth=1.20)
80 plot(t,y (:,5,2) ,LineWidth=1.20)
81 plot(t,y (:,6,3) ,LineWidth=1.20)
82 title ("Velocidade angular: degrau aplicado")
83 xlabel("Tempo [s]")
84 ylabel("Velocidade angular [rad/s]")
85 legend('w_1','w_2','w_3','Location','east')
86 hold off
87 grid on
```

```

88 baseFileName = sprintf('Image_%s.png', "aloc_vel");
89 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
90 saveas(3, fullFileName);
91
92 figure (4)
93 hold on
94 plot(t,y (:,1,1) ,LineWidth=1.20)
95 plot(t,y (:,2,2) ,LineWidth=1.20)
96 plot(t,y (:,3,3) ,LineWidth=1.20)
97 title ("Posição: degrau aplicado")
98 xlabel("Tempo [s]")
99 ylabel("Posição angular [rad]")
100 legend('q_1','q_2','q_3','Location', 'southeast')
101 hold off
102 grid on
103 baseFileName = sprintf('Image_%s.png', "aloc_pos");
104 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
105 saveas(4, fullFileName);

```

## F.2 Controlador linear quadrático

```

1 clc
2 clear all
3 close all
4 %% Parâmetros do sistema
5 addpath('Matrizes\' )
6
7 A = importdata('matrix_A1lin.txt');
8 B = importdata('matrix_B1lin.txt');
9 C = importdata('matrix_C1.txt');
10 D = importdata('matrix_D1.txt');
11
12 %% Matrizes de ganho
13 Q=[.1 0 0 0 0 0    %Aumenta a ação de controle em q1
14     0 .1 0 0 0 0    %Aumenta a ação de controle em q2
15     0 0 .1 0 0 0    %Aumenta a ação de controle em q3
16     0 0 0 .01 0 0   %Diminui a amplitude da velocidade, mas aumenta a duração do sinal em q1
17     0 0 0 0 .01 0   %Diminui a amplitude da velocidade, mas aumenta a duração do sinal em q2
18     0 0 0 0 0 0.01];
19 P = [1 0 0
20       0 1 0
21       0 0 50];
22
23 K = lqr(A,B,Q,P);
24 B1 = -B*0.2;
25 sys_lqr = ss(A-B*K,B1*0,C,D);
26

```

```
27 %% Simulação do sistema
28 [y,t,x]=step(sys_lqr);
29
30 Torq1=-K*y(:,:,1)';
31 Torq2=-K*y(:,:,2)';
32 Torq3=-K*y(:,:,3)';
33
34 %% Plots
35
36 figure(1)
37 pzmap(sys_lqr)
38 grid on
39 baseFileName = sprintf('Image_%s.png', "lqr_pol");
40 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
41 saveas(1, fullFileName);
42
43 figure(2)
44 step(sys_lqr,5)
45
46 figure(3)
47 hold on
48 plot(t,y (:,4,1) ,LineWidth=1.20)
49 plot(t,y (:,5,2) ,LineWidth=1.20)
50 plot(t,y (:,6,3) ,LineWidth=1.20)
51 title ("Velocidade angular: degrau aplicado")
52 xlabel("Tempo [s]")
53 ylabel("Velocidade angular [rad/s]")
54 legend('omega_1','omega_2','omega_3','Location','east')
55 hold off
56 grid on
57
58 figure(4)
59 hold on
60 plot(t,y (:,1,1) ,LineWidth=1.20)
61 plot(t,y (:,2,2) ,LineWidth=1.20)
62 plot(t,y (:,3,3) ,LineWidth=1.20)
63 title ("Posição: degrau aplicado")
64 xlabel("Tempo [s]")
65 ylabel("Posição angular [rad]")
66 legend('q_1','q_2','q_3','Location','east')
67 hold off
68 grid on
69
70 figure(5)
71 plot(t,Torq1(1,:),LineWidth=1.20)
72 hold on
73 plot(t,Torq2(2,:),LineWidth=1.20)
```

```

74 plot(t,Torq3(3,:),LineWidth=1.20)
75 title ("Torque: degrau aplicado")
76 ylabel('Torque [N.m]')
77 xlabel('Tempo [s]')
78 legend('T_1','T_2','T_3','Location','east')
79 hold off
80 grid on

```

### F.3 Observador de estado para o regulador via alocação de polos

```

1 clc
2 clear all
3 close all
4
5 addpath('Matrizes\'')
6 addpath("Imagens\Controle Moderno\")

7
8 A = importdata('matrix_A1lin.txt');
9 B2 = importdata('matrix_B1lin.txt');
10 C = [eye(3) zeros(3,3)];
11
12 Co = ctrb(A',C');
13 rank(Co);
14 Ob = obsv(A,C);
15 rank(Ob);
16
17 % polos alocados
18 p = [-0.8+0.6i -0.8-0.6i -2.6+0.3i -2.6-0.3i -1. -0.8];
19
20 k = 2;
21 p_obs = p - k;
22
23 K_obs = transpose(place(A',C',p_obs));
24
25 O = (A-K_obs*C);
26
27 B1 = -B2;
28
29 sys_obs_aloc = ss(O,B1,C,0);
30
31
32
33 dt = 0.001;
34 tf = 8;
35 t = 0:dt:tf;
36
37 u=zeros(length(t),3)*0;

```

```

38 u(1:5000,2)=ones(5000,1)*0.0;
39
40 e0 = [0 0 0 1 1 1];
41
42 [ys,ts,xs] = lsim(sys_obs_aloc,u,t,e0);
43
44
45 figure(2)
46 hold on
47 plot(t, xs (:,4) ,LineWidth=1.20)
48 plot(t, xs (:,5) ,LineWidth=1.20)
49 plot(t, xs (:,6) ,LineWidth=1.20)
50 title ("[ Erro de observação em velocidade angular – ", " alocação de polo"])
51 xlabel("Tempo [s]")
52 ylabel("Velocidade angular [rad/s]")
53 legend('omega_1','omega_2','omega_3','Location','east')
54 hold off
55 grid on
56 saveas(2, "obs_aloc_vel.png");
57
58 figure(3)
59 hold on
60 plot(t, xs (:,1) ,LineWidth=1.20)
61 plot(t, xs (:,2) ,LineWidth=1.20)
62 plot(t, xs (:,3) ,LineWidth=1.20)
63 title ("Posições angulares – alocação de polos")
64 xlabel("Tempo [s]")
65 ylabel("Posição angular [rad]")
66 legend('q_1','q_2','q_3','Location', 'east')
67 hold off
68 grid on
69 saveas(3, "obs_aloc_pos.png");

```

## F.4 Observador de estado para o controlador linear quadrático

```

1 clc
2 clear all
3 close all
4
5 addpath('Matrizes\'')
6 addpath("Imagens\Controle Moderno\")

7
8 A = importdata('matrix_A1lin.txt');
9 B2 = importdata('matrix_B1lin.txt');
10 B1 = -B2;
11 C = [eye(3) zeros(3,3)];
12

```

```

13 % Definição das matrizes Q e P
14
15 Q_obs = 1*[1 0 0 0 0 0    %Aumenta a ação de controle em q1
16          0 1 0 0 0 0    %Aumenta a ação de controle em q2
17          0 0 1 0 0 0    %Aumenta a ação de controle em q3
18          0 0 0 30 0 0   %Diminui a amplitude da velocidade, mas aumenta a duração do sinal
19          em q1
20          0 0 0 0 30 0   %Diminui a amplitude da velocidade, mas aumenta a duração do sinal
21          em q2
22          0 0 0 0 0 35];
23 P_obs = 1/40*[1.5 0 0
24          0 2 0
25          0 0 5];
26 % Q_obs = 5000*[.20 0 0 0 0 0  %Aumenta a ação de controle em q1
27 % 0 .15 0 0 0 0    %Aumenta a ação de controle em q2
28 % 0 0 .05 0 0 0    %Aumenta a ação de controle em q3
29 % 0 0 0 .25 0 0   %Diminui a amplitude da velocidade, mas aumenta a duração do sinal em q1
30 % 0 0 0 0 .25 0    %Diminui a amplitude da velocidade, mas aumenta a duração do sinal em q2
31 % 0 0 0 0 0 .35];
32 % P_obs = 10*[1 0 0
33 % 0 1 0
34 % 0 0 5];
35 K_obs = lqr(A',C',Q_obs,P_obs);
36 K_obs = K_obs';
37
38 O = (A-K_obs*C);
39
40 sys_obs_lqr = ss(O,B1,C,0);
41
42 pole(sys_obs_lqr)
43
44 dt = 0.001;
45 tf = 10;
46 t = 0:dt:tf;
47
48 u=zeros(length(t),3);
49 u(1:5000,2)=ones(5000,1)*0.0;
50
51 e0 = [0 0 0 1 1 1];
52
53 [ys,ts,xs] = lsim(sys_obs_lqr,u,t,e0);
54
55
56 figure(2)
57 hold on

```

```

58 plot(t, xs (:,4) ,LineWidth=1.20)
59 plot(t, xs (:,5) ,LineWidth=1.20)
60 plot(t, xs (:,6) ,LineWidth=1.20)
61 title (["Erro de observação em velocidade angular -","LQR"])
62 xlabel("Tempo [s]")
63 ylabel("Velocidade angular [rad/s]")
64 legend('omega_1','omega_2','omega_3','Location','east')
65 hold off
66 grid on
67 saveas(2, "obs_lqr_vel.png");
68
69 figure(3)
70 hold on
71 plot(t, xs (:,1) ,LineWidth=1.20)
72 plot(t, xs (:,2) ,LineWidth=1.20)
73 plot(t, xs (:,3) ,LineWidth=1.20)
74 title ("Posições angulares - LQR")
75 xlabel("Tempo [s]")
76 ylabel("Posição angular [rad]")
77 legend('q_1','q_2','q_3','Location','east')
78 hold off
79 grid on
80 saveas(3, "obs_lqr_pos.png");

```

## F.5 Seguidor com referência constante

```

1 %% Seguidor de referência constante
2
3 clc
4 clear
5 close
6
7 %% Importa matrizes
8
9 addpath('Matrizes\')
10 addpath("Imagens\Controle Moderno\")
11
12 A = importdata('matrix_A1lin.txt');
13 B = importdata('matrix_B1lin.txt');
14 C = importdata('matrix_C1.txt');
15 D = importdata('matrix_D1.txt');
16
17 %% Define a referência constante
18 x_ref = [pi/4 pi/6 pi/3 0 0 0];
19 seq = 'ZXZ';
20 q_ref = angle2quat(x_ref(1),x_ref(2),x_ref(3),seq);
21 xq_ref = [q_ref(2:4),x_ref(4:6)];

```

```

22
23 %% Regulador por alocação de polos
24 p1=-0.3+0.3i;
25 p2=-0.4+0.3i;
26 p = [p1 conj(p1) p2 conj(p2) -0.5 -0.4];
27 K = place(A,B,p);
28 F = A-B*K;
29
30 %% Cálculo da entrada em regime permanente
31 Gamma = [A,B;C,D];
32 B_ls = [zeros(6,6);eye(3,6);zeros(3,6)];
33 N = linsolve(Gamma,B_ls);
34 Nx = N(1:6,:);
35 Nu = N(7:9,:);
36 u_rp = (Nu+K*Nx)*xq_ref;
37 B2 = B;
38 B2(4:6,:) = B2(4:6,:).*u_rp;
39
40 %% Simulação do sistema
41 sys_aloc = ss(F,B2,C,D);
42 [y,t,x]=step(sys_aloc,30);
43
44 q0 = q_ref(1);
45 q1 = y (:,1,1) ;
46 q2 = y (:,2,2) ;
47 q3 = y (:,3,3) ;
48 w1 = y (:,4,1);
49 w2 = y (:,5,2);
50 w3 = y (:,6,3);
51
52 [psi,theta,phi] = quat2eulang(q0,q1,q2,q3,seq);
53
54 figure(1)
55 plot(t,w1,LineWidth=1.20)
56 hold on
57 plot(t,w2,LineWidth=1.20)
58 plot(t,w3,LineWidth=1.20)
59 grid on
60 title ("Velocidade angular ao longo do tempo")
61 xlabel("Tempo [s]")
62 ylabel("Velocidade angular [rad/s]")
63 legend('w1','w2','w3')
64 hold off
65
66 baseFileName = sprintf('Image_%s.png', "seg_cte_w_t");
67 fullFileName = fullfile ("Imagens\Controle Moderno\", baseFileName);
68 saveas(1, fullFileName);

```

```

69
70 figure(2)
71 plot(t,q1,LineWidth=1.20)
72 hold on
73 plot(t,q2,LineWidth=1.20)
74 plot(t,q3,LineWidth=1.20)
75 grid on
76 title ("Posição angular em quatérnios ao longo do tempo")
77 xlabel("Tempo [s]")
78 ylabel("Posição angular [rad]")
79 legend('q1', 'q2', 'q3', 'Location', 'northwest')
80 hold off
81
82 baseFileName = sprintf('Image_%s.png', "seg_cte_quat_t");
83 fullFileName = fullfile ("Imagens\Controle Moderno\", baseFileName);
84 saveas(2, fullFileName);
85
86 figure(3)
87 plot(t,psi,LineWidth=1.20)
88 hold on
89 plot(t,theta,LineWidth=1.20)
90 plot(t,phi,LineWidth=1.20)
91 grid on
92 title ("Posição angular em ângulos de Euler ao longo do tempoo")
93 xlabel("Tempo [s]")
94 ylabel("Posição angular [rad]")
95 legend('psi', 'theta', 'phi', 'Location', 'southeast')
96 hold off
97
98 baseFileName = sprintf('Image_%s.png', "seg_cte_euler_t");
99 fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
100 saveas(3, fullFileName);

```

## F.6 Seguidor com referência móvel e perturbações

```

1 %% Rejeitador de distúrbio
2 clc
3 clear all
4 close all
5
6 %% Tempo
7 dt = 0.1;
8 tf = 50;
9 t = 0:dt:tf;
10
11 %% Planta
12 addpath("Matrizes/")

```

```

13
14 % Define a referência constante
15 x_r = [pi/30 pi/40 pi/50 0 0 0];
16 seq = 'ZXZ';
17 q_r = angle2quat(x_r(1),x_r(2),x_r(3),seq);
18 xq_r = [q_r(2:4),x_r(4:6)];
19
20 A = importdata('matrix_Allin.txt');
21 B = importdata('matrix_B1lin.txt');
22 B1 = -B;
23 C = [eye(3) zeros(3,3)];
24 Aw = importdata('Aw.txt');
25 sysma=ss(A,B,C,0);
26
27 p1 = -0.3+0.3i;
28 p2 = -0.4+0.3i;
29 p = [p1 conj(p1) p2 conj(p2) -0.5 -0.4];
30 K = place(A,B,p);
31
32 %% Seguidor de referência variável
33 % Condições iniciais
34 xbarra0 = [0;0;0;
35 0;0;0;
36 5e-9;5e-9;5e-9;
37 0;0;0;
38 xq_r(1);xq_r(2);xq_r(3);
39 0; 0; 0];
40
41 Ar=A;
42 Ad = Aw;
43 B1 = [B1 zeros(6,3)];
44
45 %matriz singular (sistema quadrado)
46 Cbarra = [1 0 0 0 0 0;
47 0 1 0 0 0 0;
48 0 0 1 0 0 0];
49
50
51 %% Desenvolvimento
52 F = [B1 (A-Ar)];
53 F1 = A-B*K;
54 Kex = inv(Cbarra*inv(F1)*B)*Cbarra*inv(F1)*F;
55 A0 = [Ad zeros(6,6); zeros(6,6) Ar];
56 Ay = [B1 B*K]-B*Kex;
57 Abarra = [F1 Ay; zeros(12,6) A0];
58 C1 = eye(size(Abarra));
59 Bbarra = zeros(18,1);

```

```
60
61 sys2 = ss(Abarra,Bbarra,C1,0);
62
63 u=zeros(length(t),1);
64
65 [ys,ts,xbarra] = lsim(sys2,u,t,xbarra0);
66 % xbarra = step(sys2,t);
67
68 figure(1);
69 plot(t,xbarra (:,7) ,t,xbarra (:,8) ,t,xbarra (:,9) ,LineWidth=1.30);
70 grid on
71 title ('Torques de distúrbio ao longo de duas órbitas')
72 legend('T_1','T_2','T_3');
73 xlabel('Tempo [s]');
74 ylabel('Torques T [N.m]');
75
76 % baseFileName = sprintf('Image_%s.png', "seg_var_dist");
77 % fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
78 % saveas(1, fullFileName);
79
80
81 figure (2);
82 plot(t,xbarra (:,1) ,t,xbarra (:,2) ,t,xbarra (:,3) ,LineWidth=1.3);
83 hold on
84 plot(t,xbarra (:,13) ,t,xbarra (:,14) ,t,xbarra (:,15) , 'LineStyle' , '—',LineWidth=1.3);
85 grid on
86 title ('Posições angulares ao longo do tempo')
87 legend('q_1','q_2','q_3','q_1_{ref}','q_2_{ref}','q_3_{ref}');
88 xlabel('Tempo [s]');
89 ylabel('Posição angular q [rad]');
90
91 % baseFileName = sprintf('Image_%s.png', "seg_var_q_t");
92 % fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
93 % saveas(2, fullFileName);
94
95
96 figure (3);
97 plot(t,xbarra (:,4) ,t,xbarra (:,5) ,t,xbarra (:,6) ,LineWidth=1.3);
98 grid on
99 title ('Velocidades angulares ao longo do tempo')
100 legend('omega_1','omega_2','omega_3');
101 xlabel('Tempo [s]');
102 ylabel('Velocidade angular [rad/s]');
103
104 % baseFileName = sprintf('Image_%s.png', "seg_var_w_t");
105 % fullFileName = fullfile ("Imagens\Controle Moderno", baseFileName);
106 % saveas(3, fullFileName);
```

# G Controle Clássico: Códigos das simulações

## G.1 Sintonia PID: Métodos de Ziegler-Nichols

```

1 % Sintonia por ZIEGLER–NICHOLS
2 clc
3 clear all
4 close all
5 %% Parâmetros do sistema
6 addpath('Matrizes\')
7
8 A = importdata('matrix_Alin.txt');
9 B = -importdata('matrix_B1lin.txt');
10 C = importdata('matrix_C1.txt');
11 D = importdata('matrix_D1.txt');
12
13
14 sample_rate = 100;
15 t = 0:1/sample_rate:10;
16 n = length(t);
17
18 %% Construção da função de transferência de malha aberta (q3->Rot Própria)
19 [Num,Den] = ss2tf(A,B,C,D,3);
20 Num=Num(3,:);
21
22 sys=tf(Num,Den);
23 N=tf(Num,1);
24 D=tf(Den,1);
25 s=tf([1 0],1) ;
26
27
28 %% Determinação do ganho crítico
29 rlocus(sys)
30 kcrit=2.5 %Utilizar o mais adequado a partir do lugar das raízes anterior
31 Gkcrit=pid(kcrit,0,0);
32 a=series(Gkcrit,sys);
33 syskcrit=feedback(a,1)
34
35 %% Determinação do período de oscilação harmônica
36 figure(2)
37 step(syskcrit ,t)
38 ycrit=step(syskcrit,t)
39
40 [~,peaklocs] = findpeaks(ycrit);

```

```

41 Pcrit = t(round(mean(diff(peaklocs))));  

42  

43 %% Determinação do controlador  

44 Ti=Pcrit/2;  

45 Td=.125*Pcrit;  

46  

47 kp=.6*kcrit;  

48 ki=kp/Ti;  

49 kd=Td*kp;  

50  

51 GcZN=pid(kp,ki,kd);  

52 a=series(GcZN,sys);  

53 sysZN=feedback(a,1)  

54  

55 y=step(sysZN,t);  

56 stepinfo(sysZN)  

57  

58  

59  

60 %% Plots oficiais  

61 close all  

62  

63 figure (1)  

64 plot(t,y, 'linewidth' ,1.3)  

65 grid on  

66 title ('Resposta à entrada do tipo degrau')  

67 xlabel('Tempo [s]')  

68 ylabel('Posição angular [rad]')  

69 baseFileName = sprintf('Image_%s.png', "ZN_step");  

70 fullFileName = fullfile ("Imagens\", baseFileName);  

71 saveas(1, fullFileName);  

72  

73 figure (2)  

74 bode(sysZN)  

75 grid on  

76 baseFileName = sprintf('Image_%s.png', "Bode_ZN");  

77 fullFileName = fullfile ("Imagens\", baseFileName);  

78 Fh = gcf;                                % Handle To Current Figure  

79 Kids = Fh.Children;                      % Children  

80 AxAll = findobj(Kids,'Type','Axes');     % Handles To Axes  

81 Ax1 = AxAll(1);                          % First Set Of Axes  

82 LinesAx1 = findobj(Ax1,'Type','Line');   % Handle To Lines  

83 LinesAx1(2).LineWidth = 1.3;             % Set LineWidth  

84 Ax2 = AxAll(2);                          % Second Set Of Axes  

85 LinesAx2 = findobj(Ax2,'Type','Line');   % Handle To Lines  

86 LinesAx2(2).LineWidth = 1.3;             % Set LineWidth  

87 saveas(2, fullFileName);

```

```

88
89 figure(3)
90 margin(sysZN)
91 grid on
92 baseFileName = sprintf('Image_%s.png', "Margem_ZN");
93 fullFileName = fullfile ("Imagens\", baseFileName);
94 Fh = gcf;                                % Handle To Current Figure
95 Kids = Fh.Children;                      % Children
96 AxAll = findobj(Kids,'Type','Axes');      % Handles To Axes
97 Ax1 = AxAll(1);                          % First Set Of Axes
98 LinesAx1 = findobj(Ax1,'Type','Line');    % Handle To Lines
99 LinesAx1(2).LineWidth = 1.3;              % Set LineWidth
100 Ax2 = AxAll(2);                         % Second Set Of Axes
101 LinesAx2 = findobj(Ax2,'Type','Line');    % Handle To Lines
102 LinesAx2(2).LineWidth = 1.3;
103 saveas(3, fullFileName);
104
105 %
106 % figure(1)
107 % Closed_Logarithmic_Nyquist(syskdmf)
108
109 % figure(3)
110 % Closed_Logarithmic_Nyquist(syskdmfKc)
111
112 figure(4)
113 pzmap(sysZN)
114 baseFileName = sprintf('Image_%s.png', "PZ_ZN");
115 fullFileName = fullfile ("Imagens\", baseFileName);
116 saveas(4, fullFileName);
117
118 figure(5)
119 plot(t, ycrit , 'linewidth' ,1.3)
120 grid on
121 title ('Resposta à entrada do tipo degrau')
122 xlabel('Tempo [s]')
123 ylabel('Posição angular [rad]')
124 baseFileName = sprintf('Image_%s.png', "ZN");
125 fullFileName = fullfile ("Imagens\", baseFileName);
126 saveas(5, fullFileName);
127
128
129 %% Plot malha aberta
130 yma=step(sys,t);
131 figure(6)
132 plot(t,yma,'linewidth' ,1.3)
133 grid on
134 title ('Resposta à entrada do tipo degrau')

```

```

135 xlabel('Tempo [s]')
136 ylabel('Posição angular [rad]')
137 baseFileName = sprintf('Image_%s.png', "MA-step");
138 fullFileName = fullfile ("Imagens\", baseFileName);
139 saveas(6, fullFileName);
140
141 rlocus(sys)

```

## G.2 Método de alocação de polos

```

1 %% Sintonia por Lugar das Raízes (Executar célula por célula)
2 clc
3 clear
4 close all
5 %% Parâmetros do sistema
6
7 A = importdata('matrix_Allin.txt');
8 B = -importdata('matrix_B1lin.txt');
9 C = importdata('matrix_C1.txt');
10 D = importdata('matrix_D1.txt');
11 t = 0:0.01:10;
12
13 %% Construção do sistema via espaço de estados
14 sysSS=ss(A,B,C,D);
15
16 %% Construção da função de transferência de malha aberta (q3->Rot Própria)
17 [Num,Den] = ss2tf(A,B,C,D,3);
18 Num=Num(3,:);
19 sys=tf(Num,Den);
20 N=tf(Num,1);
21 D=tf(Den,1);
22 s=tf([1 0],1) ;
23
24 %% Escolha dos polos
25 p1 = -1 + 0.3i;
26 p2 = conj(p1);
27 p3 = 0;
28 p4 = 0;
29 p5 = 0;
30 p6 = 0;
31 p7 = 0;
32
33 %% Cálculo dos ganhos pelo método dos mínimos quadrados
34
35 Num = flip(Num);
36 Den = flip(Den);
37

```

```

38 a = [Num(1) 0 0;...
39     Num(2) Num(1) 0;...
40     Num(3) Num(2) Num(1);...
41     Num(4) Num(3) Num(2);...
42     Num(5) Num(4) Num(3);...
43     0 Num(5) Num(4);...
44     0 0 Num(5)];
45
46
47 b = [(p1*p2*p3*p4*p5*p6*p7)*Den(7);...
48     (p1*p2*p3*p4*p5*p6 + p2*p3*p4*p5*p6*p7)*(Den(7)-Den(1));...
49     (p1*p2*p3*p4*p5 + p2*p3*p4*p5*p6 + p3*p4*p5*p6*p7)*(Den(7)-Den(2));...
50     (p1*p2*p3*p4 + p2*p3*p4*p5 + p3*p4*p5*p6 + p4*p5*p6*p7)*(Den(7)-Den(3));...
51     (p1*p2*p3 + p2*p3*p4 + p3*p4*p5 + p4*p5*p6 + p5*p6*p7)*(Den(7)-Den(4));...
52     (p1*p2 + p2*p3 + p3*p4 + p4*p5 + p5*p6 + p6*p7)*(Den(7)-Den(5));...
53     (p1 + p2 + p3 + p4 + p5 + p6 + p7)*(Den(7)-Den(6))];
54
55 Ks = lsqr(a,b);
56 ki_lsqr = norm(Ks(1));
57 kp_lsqr = norm(Ks(2));
58 kd_lsqr = norm(Ks(3));
59
60 %% Cálculo dos ganhos de forma analítica
61 ki_ana = (1/Num(5))*(-p1*p2*p3 - p1*p2*p4 - p1*p3*p4 - p2*p3*p4 - p1*p2*p5 -...
62     p1*p3*p5 - p2*p3*p5 - p1*p4*p5 - p2*p4*p5 - p3*p4*p5 - p1*p2*p6 - ...
63     p1*p3*p6 - p2*p3*p6 - p1*p4*p6 - p2*p4*p6 - p3*p4*p6 - p1*p5*p6 - ...
64     p2*p5*p6 - p3*p5*p6 - p4*p5*p6 - p1*p2*p7 - p1*p3*p7 - p2*p3*p7 - ...
65     p1*p4*p7 - p2*p4*p7 - p3*p4*p7 - p1*p5*p7 - p2*p5*p7 - p3*p5*p7 - ...
66     p4*p5*p7 - p1*p6*p7 - p2*p6*p7 - p3*p6*p7 - p4*p6*p7 - p5*p6*p7);
67 kp_ana = (1/Num(5))*(p1*p2 + p1*p3 + p2*p3 + p1*p4 + p2*p4 + p3*p4 + p1*p5 + p2*p5 +...
68     p3*p5 + p4*p5 + p1*p6 + p2*p6 + p3*p6 + p4*p6 + p5*p6 + p1*p7 + ...
69     p2*p7 + p3*p7 + p4*p7 + p5*p7 + p6*p7);
70 kd_ana = (-p1 - p2 - p3 - p4 - p5 - p6 - p7)/Num(5);
71
72 %% Simulação
73
74 Gc_lsqr = pid(kp_lsqr,ki_lsqr,kd_lsqr);
75 u_lsqr = series(Gc_lsqr,sys);
76 sys_lsqr = feedback(u_lsqr,1);
77 y_lsqr = step(sys_lsqr,t);
78
79 Gc_ana = pid(kp_ana,ki_ana,kd_ana);
80 u_ana = series(Gc_ana,sys);
81 sys_ana = feedback(u_ana,1);
82 y_ana = step(sys_ana,t);
83
84 figure(1)

```

```

85 pzmap(sys_ana)
86
87 figure(2)
88 pzmap(sys_lsqr)
89
90 figure(3)
91 plot(t,y_lsqr)
92 hold on
93 plot(t,y_ana)
94 grid on
95 title('Resposta à entrada do tipo degrau')
96 xlabel('Tempo [s]')
97 ylabel('Posição angular [rad]')
98 legend("Least Square","Analítico")
99
100 baseFileName = sprintf('Image_%s.png', "aloc_class");
101 fullFileName = fullfile ("Imagens\", baseFileName);
102 saveas(3, fullFileName);

```

```

1 ClearAll["Global`*"]
2 (* Função de transferência de malha aberta *)
3 Den = d0 + d1*s + d2*s^2 + d3*s^3 + d4*s^4 + d5*s^5 + d6*s^6;
4 Num = n0 + n1*s + n2*s^2 + n3*s^3 + n4*s^4;
5
6 (* Equação característica de malha fechada do sistema*)
7 eq1 := Den*s + (Ki + Kp*s + Kd*s^2)*Num;
8 Collect[eq1, s];
9
10 (* Equação característica de malha fechada desejada *)
11 eq2 := (s - p1)*(s - p2)*(s - p3)*(s - p4)*(s - p5)*(s - p6)*(s - p7);
12 Collect[eq2, s];
13
14 (* Dados *)
15 d0 = 0; d1 = 0; d2 = 0; d3 = 0; d4 = 0; d5 = 0; d6 = 1;
16 n0 = 0; n1 = 0; n2 = 0; n3 = 0; n4 = 100.784345078673;
17
18 (* Cálculo dos ganhos *)
19 a = CoefficientList [Collect[eq1, s], s];
20 b = CoefficientList [Collect[eq2, s], s];
21 Sol1 = Solve[{a [[5]] == b [[5]], a [[6]] == b [[6]],
22      a [[7]] == b [[7]]}, {Ki, Kp, Kd}];
23 Ki = Ki /. Sol1 [[1]];
24 Kp = Kp /. Sol1 [[1]];
25 Kd = Kd /. Sol1 [[1]];
26
27 (* Polos definidos *)
28 p1 = -1 + 0.3 I;
29 p2 = Conjugate[p1];

```

```

30 p3 = 0;
31 p4 = 0;
32 p5 = 0;
33 p6 = 0;
34 p7 = 0;
35
36 (* Resultados *)
37 Echo[Kp, "Ganho proporcional Kp: "];
38 Echo[Ki, "Ganho integrativo Ki: "];
39 Echo[Kd, "Ganho integrativo Kd: "];

```

### G.3 Método de controle ótimo - ITAE

```

1 %% Sintonia por ITAE (Executar célula por célula)
2 clc
3 clear all
4 close all
5 %% Parâmetros do sistema
6 addpath('Matrizes/');
7
8 A = importdata('matrix_A1lin.txt');
9 B = -importdata('matrix_B1lin.txt');
10 C = importdata('matrix_C1.txt');
11 D = importdata('matrix_D1.txt');
12 t = 0:0.005:15;
13
14 %% Construção do sistema via espaço de estados
15 sysSS=ss(A,B,C,D);
16
17 %% Construção da função de transferência de malha aberta (q3->Rot Própria)
18 [Num,Den] = ss2tf(A,B,C,D,3);
19 Num=Num(3,:);
20 sys=tf(Num,Den);
21 N=tf(Num,1);
22 D=tf(Den,1);
23 s=tf([1 0],1) ;
24
25
26 %% Polinômio ITAE
27
28 wn = 1.2; %valor arbitrado de omega_n
29
30 tab = [1 1.783*wn 2.172*wn^2 wn^3]; %tabela de coeficientes
31
32 a = Num(3);
33 kd = tab(2)/a;
34 kp = tab(3)/a;

```

```

35 ki = tab(4)/a;
36
37 itae_ref = tf([wn^3], tab);
38
39 %% Sintonia sem pré-compensador
40 Gkd = pid(kp,ki,kd);
41 a_t = series(Gkd,sys);
42 sys_itae = feedback(a_t,1);
43
44 y_sys = step(sys_itae,t);
45 y_itae = step(itae_ref,t);
46
47 figure(10)
48 pzmap(sys);
49
50 %% Sintonia com pré-compensador
51
52 [num, den] = tfdata(sys_itae);
53
54 num_comp = num{1}(4:8);
55 den_comp = den{1}(1:8);
56
57 FT_comp = tf(num_comp, den_comp);
58
59 y_sys_comp = step(FT_comp,t);
60
61
62 %% Plots
63
64 figure(1)
65 plot(t,y_sys,LineWidth=1.3)
66 hold on
67 plot(t,y_itae,LineWidth=1.3)
68 grid on
69 title('Resposta à entrada do tipo degrau — sem pré-compensador')
70 xlabel('Tempo [s]')
71 ylabel('Posição angular [rad]')
72 legend("PID — ITAE sem pré-compensador", "ITAE referência", Location="southeast")
73 baseFileName = sprintf('Image_%s.png', 'itae_sem_comp');
74 fullFileName = fullfile ("Imagens/", baseFileName);
75 saveas(1, fullFileName);
76
77 figure(2)
78 plot(t,y_sys_comp,LineWidth=1.3)
79 hold on
80 plot(t,y_itae,LineWidth=1.3)
81 grid on

```

```

82 title ('Resposta à entrada do tipo degrau — com pré-compensador')
83 xlabel('Tempo [s]')
84 ylabel('Posição angular [rad]')
85 legend("PID — ITAE com pré-compensador","ITAE referência",Location="southeast")
86 baseFileName = sprintf('Image_%s.png', "itae_com_comp");
87 fullFileName = fullfile ("Imagens/", baseFileName);
88 saveas(2, fullFileName);
89
90 %% Step Info
91 stepinfo(sys_itae)
92 stepinfo(FT_comp)
93
94 %% Diagramas
95
96 % Mapa de polos e zeros
97 figure(3)
98 pzmap(FT_comp)
99 baseFileName = sprintf('Image_%s.png', "itae_pole_map");
100 fullFileName = fullfile ("Imagens/", baseFileName);
101 saveas(3, fullFileName);
102
103 % Margem de ganho e fase
104 figure(4)
105 margin(FT_comp)
106 % hold on
107 % margin(sys_itae)
108 grid on
109 baseFileName = sprintf('Image_%s.png', "itae_margin");
110 fullFileName = fullfile ("Imagens/", baseFileName);
111 saveas(4, fullFileName);
112
113 figure(5)
114 rlocus(FT_comp)
115
116 figure(6)
117 rlocus(sys_itae)

```

## G.4 Método do Lugar das Raízes (*Root Locus*)

```

1 % Sintonia por Lugar das Raízes (Executar célula por célula)
2 clc
3 clear all
4 close all
5 %% Parâmetros do sistema
6 addpath('Matrizes\' )
7
8 A = importdata('matrix_A1lin.txt');

```

```
9 B = -importdata('matrix_B1lin.txt');
10 C = importdata('matrix_C1.txt');
11 D = importdata('matrix_D1.txt');
12
13 t = 0:0.01:10;
14 %% Construção do sistema via espaço de estados
15 sysSS=ss(A,B,C,D);
16
17 %% Construção da função de transferência de malha aberta (q3->Rot Própria)
18 [Num,Den] = ss2tf(A,B,C,D,3);
19 Num=Num(3,:);
20
21 sys=tf(Num,Den);
22 N=tf(Num,1);
23 D=tf(Den,1);
24 s=tf([1 0],1) ;
25
26 % [Gc,Kc]=compensador2P(-0.5+1i,Den)
27 % sys=series(Gc,sys)
28
29
30 figure (1)
31 step(sys)
32 grid on
33
34 figure (2)
35 pzmap(sys)
36 %% Sintonia proporcional
37
38 figure (3)
39 rlocus(sys)
40
41 %% Sintonia proporcional (continuação)
42 kp=0.01 %Utilizar o mais adequado a partir do lugar das raízes anterior
43 Gkp=pid(kp,0,0);
44 a=series(Gkp,sys);
45 syskpmf=feedback(a,1)
46
47 figure (4)
48 step(syskpmf);
49
50 figure (5)
51 pzmap(syskpmf);
52
53
54 %% Sintonia integrativa
55 syskima=(N/(s*(D+(kp*N))));
```

```
56
57 figure(6)
58 rlocus(syskima)
59
60 %% Sintonia integrativa (continuação)
61 ki=0 %Utilizar o mais adequado a partir do lugar das raizes anterior
62 Gki=pid(kp,ki,0);
63 a=series(Gki,sys);
64 syskimf=feedback(a,1)
65
66 figure(7)
67 step(syskimf);
68
69 figure(8)
70 pzmap(syskimf);
71
72 %% Sintonia derivativa
73 syskdma=((N*(s^2))/(s*D+(kp*s*N)+(ki*N)));
74
75 figure(9)
76 rlocus(syskdma)
77
78 %% Sintonia derivativa (continuação)
79 kd=0.0199 %Utilizar o mais adequado a partir do lugar das raizes anterior
80 Gkd=pid(kp,ki,kd);
81 a=series(Gkd,sys);
82 syskdmf=feedback(a,1)
83
84 figure(10)
85 [yLR]=step(syskdmf,t);
86 stepinfo(syskdmf)
87
88 figure(11)
89 pzmap(syskdmf);
90 %% Com compensador
91 close all
92 [Gc,Kc]=compensador2P(-0.5+1i,Den);
93 sysKc=series(Gc,sys);
94 Numcomp=tf(Gc.Numerator{1},1);
95 Dencomp=tf(Gc.Denominator{1},1);
96
97 figure(12)
98 step(sysKc)
99 grid on
100
101 figure(13)
102 pzmap(sysKc)
```

```
103
104 %% Sintonia proporcional
105 figure(14)
106 rlocus(sysKc)
107
108 %% Sintonia proporcional (continuação)
109 kpKc=0.04 %Utilizar o mais adequado a partir do lugar das raizes anterior
110 GkpKc=pid(kpKc,0,0);
111 a=series(GkpKc,sysKc);
112 syskpmfKc=feedback(a,1)
113
114 figure(15)
115 step(syskpmfKc);
116
117 figure(16)
118 pzmap(syskpmfKc);
119
120
121 %% Sintonia integrativa
122 syskimaKc=(Numcomp*N/(s*(Dencomp*D+(kpKc*Numcomp*N)))); 
123
124 figure(17)
125 rlocus(syskimaKc)
126
127 %% Sintonia integrativa (continuação)
128 kiKc=0.025 %Utilizar o mais adequado a partir do lugar das raizes anterior
129 GkiKc=pid(kpKc,kiKc,0);
130 a=series(GkiKc,sysKc);
131 syskimfKc=feedback(a,1)
132
133 figure(18)
134 step(syskimfKc);
135
136 figure(19)
137 pzmap(syskimfKc);
138
139 %% Sintonia derivativa
140 syskdmaKc=((Numcomp*N*(s^2))/(s*Dencomp*D+(kpKc*s*Numcomp*N)+(kiKc*Numcomp*N)));
141
142 figure(20)
143 rlocus(syskdmaKc)
144
145 %% Sintonia derivativa (continuação)
146 kdKc=0.02 %Utilizar o mais adequado a partir do lugar das raizes anterior
147 GkdKc=pid(kpKc,kiKc,kdKc);
148 a=series(GkdKc,sysKc);
149 syskdmfKc=feedback(a,1)
```

```

150
151 figure(21)
152 [yLRKc]=step(syskdmfKc,t);
153 stepinfo(syskdmfKc)

154
155 figure(22)
156 pzmap(syskdmfKc);

157
158
159
160 %% Plots oficiais
161 close all
162
163 figure(23)
164 plot(t,yLR,'linewidth',1.3)
165 hold on
166 plot(t,yLRKc,'linewidth',1.3)
167 grid on
168 title('Resposta à entrada do tipo degrau')
169 xlabel('Tempo [s]')
170 ylabel('Posição angular [rad]')
171 legend('Sem compensador','Com compensador','location','southeast')
172 baseFileName = sprintf('Image_%s.png', "LR");
173 fullFileName = fullfile ("Imagens\", baseFileName);
174 saveas(23, fullFileName);

175
176 figure(24)
177 bode(syskdmf,syskdmfKc)
178 grid on
179 legend('Sem compensador','Com compensador','location','northeast')
180 baseFileName = sprintf('Image_%s.png', "Bode_LR");
181 fullFileName = fullfile ("Imagens\", baseFileName);
182 saveas(24, fullFileName);
183 grid on
184 baseFileName = sprintf('Image_%s.png', "Margem_ZN");
185 fullFileName = fullfile ("Imagens\", baseFileName);
186 Fh = gcf;                                % Handle To Current Figure
187 Kids = Fh.Children;                      % Children
188 AxAll = findobj(Kids,'Type','Axes');      % Handles To Axes
189 Ax1 = AxAll(1);                          % First Set Of Axes
190 LinesAx1 = findobj(Ax1,'Type','Line');    % Handle To Lines
191 LinesAx1(2).LineWidth = 1.3;              % Set LineWidth
192 Ax2 = AxAll(2);

193
194
195 figure(25)
196 % margin(syskdmf)

```

```

197 % hold on
198 margin(syskdmfKc)
199 grid on
200 %legend('Sem compensador','Com compensador','location','northeast')
201 grid on
202 Fh = gcf;                                % Handle To Current Figure
203 Kids = Fh.Children;                      % Children
204 AxAll = findobj(Kids,'Type','Axes');      % Handles To Axes
205 Ax1 = AxAll(1);                          % First Set Of Axes
206 LinesAx1 = findobj(Ax1,'Type','Line');    % Handle To Lines
207 LinesAx1(2).LineWidth = 1.3;              % Set LineWidth
208 Ax2 = AxAll(2);                          % Second Set Of Axes
209 LinesAx2 = findobj(Ax2,'Type','Line');    % Handle To Lines
210 LinesAx2(2).LineWidth = 1.3;              % Set LineWidth
211 Ax2 = AxAll(2);
212 baseFileName = sprintf('Image_%s.png', "Margem_LR");
213 fullFileName = fullfile ("Imagens\", baseFileName);
214 saveas(25, fullFileName);
215
216 figure(26)
217 pzmap(syskdmf)
218 baseFileName = sprintf('Image_%s.png', "PZ_LR");
219 fullFileName = fullfile ("Imagens\", baseFileName);
220 saveas(26, fullFileName);
221
222 figure(27)
223 pzmap(syskdmfKc)
224 baseFileName = sprintf('Image_%s.png', "PZ_LR_comp");
225 fullFileName = fullfile ("Imagens\", baseFileName);
226 saveas(27, fullFileName);

```

## G.5 Comparação entre métodos

```

1 %% Comparaçao entre controladores PID
2
3 clc
4 clear
5 close all
6
7 %% Parâmetros do sistema
8 addpath('Matrizes\')
9
10 A = importdata('matrix_A1lin.txt');
11 B = -importdata('matrix_B1lin.txt');
12 C = importdata('matrix_C1.txt');
13 D = importdata('matrix_D1.txt');
14

```

```
15 sample_rate = 100;
16 t = 0:1/sample_rate:10;
17 n = length(t);
18
19 %% Construção da função de transferência de malha aberta (q3->Rot Própria)
20
21 [Num,Den] = ss2tf(A,B,C,D,3);
22 Num=Num(3,:);
23
24 sys=tf(Num,Den);
25 N=tf(Num,1);
26 D=tf(Den,1);
27 s=tf([1 0],1) ;
28
29 %% Método ZN sem compensador
30
31 % Determinação do ganho crítico
32 kcrit = 2.5; %Utilizar o mais adequado a partir do lugar das raízes anterior
33 Gkcrit=pid(kcrit,0,0);
34 a=series(Gkcrit,sys);
35 syskcrit=feedback(a,1);
36
37 % Determinação do período de oscilação harmônica
38 ycrit=step(syskcrit,t);
39
40 [~,peaklocs] = findpeaks(ycrit);
41 Pcrit = t(round(mean(diff(peaklocs))));
```

42

```
43 % Determinação do controlador
44 Ti=Pcrit/2;
45 Td=.125*Pcrit;
46
47 kp=.6*kcrit;
48 ki=kp/Ti;
49 kd=Td*kp;
50
51 GcZN=pid(kp,ki,kd);
52 a=series(GcZN,sys);
53 sysZN=feedback(a,1);
54
55 y_ZN=step(sysZN,t);
56
57 %% Método de alocação de polos
58
59 % Escolha dos polos
60 p1 = -1 + 0.3i;
61 p2 = conj(p1);
```

```

62 p3 = 0;
63 p4 = 0;
64 p5 = 0;
65 p6 = 0;
66 p7 = 0;
67
68 Num1 = flip(Num);
69 Den1 = flip(Den);
70
71 % Cálculo dos ganhos de forma analítica
72 ki_ana = (1/Num1(5))*(-p1*p2*p3 - p1*p2*p4 - p1*p3*p4 - p2*p3*p4 - p1*p2*p5 - ...
73   p1*p3*p5 - p2*p3*p5 - p1*p4*p5 - p2*p4*p5 - p3*p4*p5 - p1*p2*p6 - ...
74   p1*p3*p6 - p2*p3*p6 - p1*p4*p6 - p2*p4*p6 - p3*p4*p6 - p1*p5*p6 - ...
75   p2*p5*p6 - p3*p5*p6 - p4*p5*p6 - p1*p2*p7 - p1*p3*p7 - p2*p3*p7 - ...
76   p1*p4*p7 - p2*p4*p7 - p3*p4*p7 - p1*p5*p7 - p2*p5*p7 - p3*p5*p7 - ...
77   p4*p5*p7 - p1*p6*p7 - p2*p6*p7 - p3*p6*p7 - p4*p6*p7 - p5*p6*p7);
78
79 kp_ana = (1/Num1(5))*(p1*p2 + p1*p3 + p2*p3 + p1*p4 + p2*p4 + p3*p4 + p1*p5 + p2*p5 + ...
80   p3*p5 + p4*p5 + p1*p6 + p2*p6 + p3*p6 + p4*p6 + p5*p6 + p1*p7 + ...
81   p2*p7 + p3*p7 + p4*p7 + p5*p7 + p6*p7);
82
83 kd_ana = (-p1 - p2 - p3 - p4 - p5 - p6 - p7)/Num1(5);
84
85 Gc_ana = pid(kp_ana,ki_ana,kd_ana);
86 u_ana = series(Gc_ana,sys);
87 sys_ana = feedback(u_ana,1);
88 y_aloc = step(sys_ana,t);
89
90 %% Método ITAE
91
92 % Polinômio ITAE
93
94 wn = 1.2; %valor arbitrado de omega_n
95
96 tab = [1 1.783*wn 2.172*wn^2 wn^3]; %tabela de coeficientes
97
98 a = Num(3);
99 kd = tab(2)/a;
100 kp = tab(3)/a;
101 ki = tab(4)/a;
102
103 itae_ref = tf(wn^3, tab);
104
105 % Sintonia sem pré-compensador
106 Gkd = pid(kp,ki,kd);
107 a_t = series(Gkd,sys);
108 sys_itae = feedback(a_t,1);

```

```

109
110 y_scomp_itae = step(sys_itae,t);
111 y_itae = step(itae_ref,t);
112
113 % Sintonia com pré-compensador
114
115 [num, den] = tfdata(sys_itae);
116
117 num_comp = num{1}(4:8);
118 den_comp = den{1}(1:8);
119
120 FT_comp = tf(num_comp, den_comp);
121
122 y_comp_itae = step(FT_comp,t);
123
124 %% Método lugar das raízes
125
126 % Com compensador
127 close all
128 [Gc,Kc]=compensador2P(-0.5+1i,Den);
129 sysKc=series(Gc,sys);
130 Numcomp=tf(Gc.Numerator{1},1);
131 Dencomp=tf(Gc.Denominator{1},1);
132
133 % Sintonia proporcional
134 kpKc=0.04; %Utilizar o mais adequado a partir do lugar das raizes anterior
135 GkpKc=pid(kpKc,0,0);
136 a=series(GkpKc,sysKc);
137 syskpmfKc=feedback(a,1);
138
139
140 % Sintonia integrativa
141 syskimaKc=(Numcomp*N/(s*(Dencomp*D+(kpKc*Numcomp*N)))); 
142
143 % Sintonia integrativa (continuação)
144 kiKc=0.025; %Utilizar o mais adequado a partir do lugar das raizes anterior
145 GkiKc=pid(kpKc,kiKc,0);
146 a=series(GkiKc,sysKc);
147 syskimfKc=feedback(a,1);
148
149 % Sintonia derivativa
150 syskdmaKc=((Numcomp*N*(s^2))/(s*Dencomp*D+(kpKc*s*Numcomp*N)+(kiKc*Numcomp*N)));
151
152 % Sintonia derivativa (continuação)
153 kdKc=0.02; %Utilizar o mais adequado a partir do lugar das raizes anterior
154 GkdKc=pid(kpKc,kiKc,kdKc);
155 a=series(GkdKc,sysKc);

```

```

156 syskdmfKc=feedback(a,1);
157
158 [y_comp_rl]=step(syskdmfKc,t);
159
160
161 %% Gráficos finais
162 %Cores RGB (Vermelho Ferrari, Azul Marinho, Ouro metálico, Verde escuro,
163 %Violeta, Laranja)
164 col=[[207 14 14];[18 10 143];[212 175 55]; [50 150 50]; [102 0 161];...
165 [242 79 0]]./255;
166
167 % h=plot();
168 % h.Color=col(j,:); %Sendo j o número da cor desejada
169
170 figure(1)
171 ZN = plot(t,y_ZN,LineWidth=1.2);
172 ZN.Color=col(1,:);
173 hold on
174 aloc=plot(t,y_aloc,LineWidth=1.2);
175 aloc.Color=col(2,:);
176 itae=plot(t,y_comp_itae,LineWidth=1.2);
177 itae.Color=col(3,:);
178 rl=plot(t,y_comp_rl,LineWidth=1.2);
179 rl.Color=col(4,:);
180 grid on
181 title ("Posição angular q_3 em função do tempo")
182 xlabel("Tempo [s]")
183 ylabel("Posição angular [rad]")
184 legend('ZN sem Compensador','Alocação de Polos sem Compensador',...
185 'ITAE com Compensador','Lugar das Raízes com Compensador','Location','southeast')
186 hold off
187
188 baseFileName = sprintf('Image_%s.png', "comparacao");
189 fullFileName = fullfile ("Imagens\", baseFileName);
190 saveas(1, fullFileName);
191
192 figure(2)
193 ZNb = bodeplot(sysZN,'-r');
194 hold on
195 alocb=bodeplot(sys_ana,'-y');
196 itaeb=bodeplot(FT_comp,'-b');
197 rlb=bodeplot(syskdmfKc,'-g');
198 grid on
199 legend('ZN sem Compensador','Alocação de Polos sem Compensador',...
200 'ITAE com Compensador','Lugar das Raízes com Compensador','Location','best')
201 hold off

```

## G.6 Redução de ordem

```

1 function sysred=ReducaoDeOrdem(sys)
2 %% Verificação dos polos não dominantes
3 [sysb,g]=balreal(sys);
4 g'
5
6 %% Definição dos polos a serem excluídos
7 poldel = input('Quais polos devem ser deletados? Vetor com o índice deles (ordem que aparece em g) ');
8
9 %% Redução da ordem
10 sysred=modred(sysb,poldel,'del')

```

## G.7 Compensador

```

1 function [C,K]=compensador2P(LR,D)
2 %% Setup das variáveis
3 P=roots(D);
4 %Polo 1
5 Pr1=real(P(1));
6 Pr2=real(P(2));
7 %Polo 2
8 Pi1=imag(P(1));
9 Pi2=imag(P(2));
10 %Ponto de interesse =
11 LRr=real(LR);
12 LRI=imag(LR);
13
14 %% Cálculo do avanço da fase
15 phi=(180-abs(atand(LRI/(LRr-Pr1))))+(180-abs(atand(LRI/(LRr-Pr2))))-180;
16
17 %% Cálculo de ângulos importantes
18 theta=abs(atand(LRI/LRr));
19 bicetriz=(180-theta)/2;
20 angaux=bicetriz-(90-theta);
21
22 %% Cálculo do zero
23 Zc=LRr-(tand(angaux-phi/2)*LRI);
24
25 %% Cálculo do polo
26 Pc=LRr-(tand(angaux+phi/2)*LRI);
27
28 %% Cálculo do ganho
29 K=abs(polyval(D,LR)*polyval([1 -Pc],LR)/polyval([1 -Zc],LR));
30 % D(end-1)=D(end-1)+1;
31 % D(end)=D(end)+Pc

```

32 %% FT do compensador

33 C=tf(K\*[1 -Zc],[1 -Pc]);

# H Códigos Auxiliares

## H.1 Conversão entre ângulos de Euler e quatérnios

```

1 function q = eulang2quat(ang1,ang2,ang3,rot_seq)
2     %% Euler Angles to Direct Cosine Matrix (DCM)
3     DCM1 = rotation_matrix(rot_seq(1),ang1);
4     DCM2 = rotation_matrix(rot_seq(2),ang2);
5     DCM3 = rotation_matrix(rot_seq(3),ang3);
6     DCM = DCM3*DCM2*DCM1;
7
8     %% DCM to Quaternion – Sheppard's Method
9     % For this step, it's used the Sheppard's Method
10    %%
11    % 1. Find the largest value between $q_0^2, \, q_1^2, \, q_2^2, \, q_3^2$
12    q0_2 = 1/4*(1 + trace(DCM));
13    q1_2 = 1/4*(1 + 2*DCM(1,1) - trace(DCM));
14    q2_2 = 1/4*(1 + 2*DCM(2,2) - trace(DCM));
15    q3_2 = 1/4*(1 + 2*DCM(3,3) - trace(DCM));
16    q_2 = [q0_2 q1_2 q2_2 q3_2];
17    %%
18    % 2. Compute the remaining EPs using
19    [M,I] = max(q_2);
20    if I == 1
21        q0 = sqrt(q0_2);
22        q1 = (DCM(2,3)-DCM(3,2))/(4*q0);
23        q2 = (DCM(3,1)-DCM(1,3))/(4*q0);
24        q3 = (DCM(1,2)-DCM(2,1))/(4*q0);
25    end
26    if I == 2
27        q1 = sqrt(q1_2);
28        q0 = (DCM(2,3)-DCM(3,2))/(4*q1);
29        q2 = (DCM(1,2)+DCM(2,1))/(4*q1);
30        q3 = (DCM(3,1)+DCM(1,3))/(4*q1);
31    end
32    if I == 3
33        q2 = sqrt(q2_2);
34        q0 = (DCM(3,1)-DCM(1,3))/(4*q2);
35        q1 = (DCM(1,2)+DCM(2,1))/(4*q2);
36        q3 = (DCM(2,3)+DCM(3,2))/(4*q2);
37    end
38    if I == 4
39        q3 = sqrt(q3_2);
40        q0 = (DCM(1,2)-DCM(2,1))/(4*q3);

```

```

41     q1 = (DCM(3,1)+DCM(1,3))/(4*q3);
42     q2 = (DCM(2,3)+DCM(3,2))/(4*q3);
43 end
44 q = [q0 q1 q2 q3];
45 end

```

```

1 function [Z1,Y1,Z2] = quat2eulang(q0,q1,q2,q3,seq)
2 %% Quaternion to Euler Angles (DCM)
3 L = length(q2);
4 q = [ones(L,1)*q0,q1,q2,q3];
5 [Z1,Y1,Z2] = quat2angle(q,seq);
6 end

```

```

1 %% Rotation Matrix
2
3 %% Description
4 % The function returns the rotation matrix that gives the new set of
5 % coordinates as a function of the old ones
6 %% Calling Syntax
7 % function [matrix] = rotation_matrix(axis,angle)
8 %
9 %% I/O Variables
10 % |IN Integer| *axis*: _axis of rotation_
11 %
12 % |IN Double| *angle*: _angle of rotation (rad)_ $\theta$%
13
14 %% Example
15 % axis = 1;
16 % angle = 30;
17 %
18 % [matrix] = rotation_matrix(axis,angle);
19
20 %% Version Control
21 % 1.0; Bruno Caixeta Piazza; 2021/12/01 ; First issue.
22 %
23
24 %%
25 function [matrix] = rotation_matrix(axis,angle)
26 if axis == 1 || axis == 'X'
27     matrix = [1,0,0;0,cos(angle),sin(angle);0,-sin(angle),cos(angle)];
28 elseif axis == 2 || axis == 'Y'
29     matrix = [cos(angle),0,-sin(angle);0,1,0; sin(angle),0,cos(angle)];
30 elseif axis == 3 || axis == 'Z'
31     matrix = [cos(angle),sin(angle),0;-sin(angle),cos(angle),0;0,0,1];
32 end
33 end

```