

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA AEROESPACIAL

Tarso Kraemer Sarzi Sartori

**MODELAGEM MATEMÁTICA E CONTROLE DE UM SISTEMA DE
GUIMBAIS DE 3 GRAUS DE LIBERDADE**

Santa Maria, RS
2019

Tarso Kraemer Sarzi Sartori

MODELAGEM MATEMÁTICA E CONTROLE DE UM SISTEMA DE GUIMBAIS DE 3 GRAUS DE LIBERDADE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Aeroespacial da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Aeroespacial**.

ORIENTADOR: Prof. André Luís da Silva

Santa Maria, RS
2019

Tarso Kraemer Sarzi Sartori

MODELAGEM MATEMÁTICA E CONTROLE DE UM SISTEMA DE GUIMBAIS DE 3 GRAUS DE LIBERDADE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Aeroespacial da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Aeroespacial**.

Aprovado em 8 de julho de 2019:

André Luís da Silva, Dr. (UFSM)
(Presidente/Orientador)

Pedro Paglione, Dr. (ITA)

Lucas Vizzotto Bellinaso, Dr. (UFSM)

Santa Maria, RS
2019

RESUMO

MODELAGEM MATEMÁTICA E CONTROLE DE UM SISTEMA DE GUIMBAIS DE 3 GRAUS DE LIBERDADE

AUTOR: Tarso Kraemer Sarzi Sartori

ORIENTADOR: André Luís da Silva

Guimbais são suportes que permitem a rotação de um objeto em torno de um conjunto de eixos. Atualmente, guimbais são utilizados em diversas áreas, desde navegação inercial à estabilização óptica de equipamentos de filmagem. Este trabalho propõe a modelagem de um sistema de guimbais de 3 graus de liberdade (GDL) para reprodução de movimentos de rotação e seu controle através de técnicas lineares e não-lineares. O modelo servirá como base para futuros trabalhos que envolvam os guimbais que equipam os laboratórios do curso de Engenharia Aeroespacial da Universidade Federal de Santa Maria (UFSM).

Palavras-chave: Guimbais. Controle. Modelagem.

ABSTRACT

MODELING AND CONTROL OF A SYSTEM OF GIMBALS WITH 3 DEGREE OF FREEDOM

AUTHOR: Tarso Kraemer Sarzi Sartori

ADVISOR: André Luís da Silva

Gimbals are supports that allow the rotation of an object around a set of axis. Currently, gimbals are used in differently areas, since inertial navigation, until optical stabilization of equipments of filming. This work propose a model of a system of gimbals with 3 degree of freedom (DOF) for reproduction of rotation movements and the control through linear and non-linear techniques. The model will serve to further works that involve the gimbals that equip the laboratories of the course of Aerospace Engineering of Federal University of Santa Maria (UFSM).

Keywords: Gimbal. Model. Control.

LISTA DE FIGURAS

Figura 1.1 – Ilustração de um sistema de guimbais com 3 GDL.	8
Figura 1.2 – Modelo em CAD dos guimbais.	9
Figura 1.3 – Sistema de guimbais.	10
Figura 1.4 – Esquema do funcionamento do sistema de guimbais com os servomotores.	10
Figura 3.1 – Sequência de rotação 3-2-1.	14
Figura 3.2 – Circuito equivalente do motor.	21
Figura 3.3 – Ensaio de torque.	26
Figura 3.4 – Sistema regulador ótimo.	29
Figura 3.5 – Diagrama de blocos para rastreio de referência.	31
Figura 3.6 – Diagrama de blocos para inversão dinâmica.	34
Figura 4.1 – Dinâmica dos guimbais.	38
Figura 4.2 – Simulação do modelo dos guimbais.	39
Figura 4.3 – Modelo do motor no Simulink.	40
Figura 4.4 – Comparação servomotor sem carga.	41
Figura 4.5 – Comparação servomotor com carga do guimbal 3.	42
Figura 4.6 – Comparação servomotor com carga do guimbal 3 mais barra.	42
Figura 4.7 – União dos modelos.	43
Figura 4.8 – Modelo em Simulink da Inversão Dinâmica.	45
Figura 4.9 – Simulação da Inversão Dinâmica sem o modelo dos motores.	46
Figura 4.10 – Simulação da Inversão Dinâmica com o modelo dos motores.	47
Figura 4.11 – Controles inversão dinâmica.	48
Figura 4.12 – Modelo em Simulink do controle LQR.	49
Figura 4.13 – Servomotores.	50
Figura 4.14 – Matrizes A e B linearizadas.	50
Figura 4.15 – Matrizes de ponderação para LQR.	51
Figura 4.16 – Matrizes de ganho para LQR.	52
Figura 4.17 – Simulação do LQR	52
Figura 4.18 – Controles LQR.	53
Figura 4.19 – Modelo em Simulink do CME.	54
Figura 4.20 – Simulação do CME.	55
Figura 4.21 – Controles CME.	56

LISTA DE TABELAS

Tabela 3.1 – Dados dos servomotores.	24
Tabela 3.2 – Resumo dos parâmetros calculados e medidos.	27

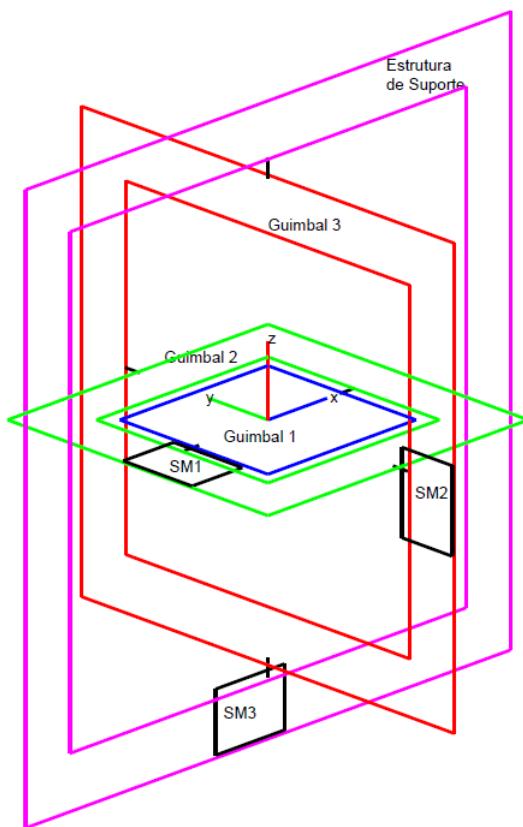
SUMÁRIO

1	INTRODUÇÃO	8
1.1	OBJETIVOS E MOTIVAÇÃO	11
2	METODOLOGIA	12
3	REVISÃO BIBLIOGRÁFICA.....	13
3.1	MODELAGEM MATEMÁTICA DO SISTEMA DE GUIMBAIS COM 3 GRAUS DE LIBERDADE	13
3.2	MODELAGEM DOS SERVOMOTORES	20
3.3	REGULADOR LINEAR QUADRÁTICO COM REALIMENTAÇÃO DE SAÍDA E RASTREADOR DE REFERÊNCIA	27
3.3.1	Regulador linear quadrático ótimo com realimentação de saída	28
3.3.2	Rastreamento de referência	30
3.4	LINEARIZAÇÃO	32
3.5	INVERSÃO DINÂMICA PARA SISTEMAS NÃO LINEARES	33
3.6	CONTROLE POR MAPEAMENTO EXPONENCIAL	35
4	RESULTADOS	37
4.1	IMPLEMENTAÇÃO DO MODELO DOS GUIMBAIS	37
4.2	IMPLEMENTAÇÃO DO MODELO DOS SERVOMOTORES	40
4.3	UNIÃO DO MODELO DO SERVOMOTOR E DOS GUIMBAIS	43
4.4	IMPLEMENTAÇÃO DO CONTROLE POR INVERSÃO DINÂMICA	45
4.5	IMPLEMENTAÇÃO DO CONTROLE LQR	48
4.6	IMPLEMENTAÇÃO DO CONTROLE PELO CME	53
5	CONCLUSÃO	57
	REFERÊNCIAS BIBLIOGRÁFICAS	58
	ANEXO A – CÓDIGO PARA DEDUÇÃO DAS EQUAÇÕES DE DINÂMICA DO SISTEMA DE GUIMBAIS E DAS MATRIZES PARA A INVERSÃO DINÂMICA..	59
	ANEXO B – PROGRAMA PARA CÁLCULO DO CONTROLE NO EQUILÍBRIO	63
	ANEXO C – PROGRAMA DE LINEARIZAÇÃO	64
	ANEXO D – PROGRAMA PARA O CÁLCULO DOS GANHOS	65
	ANEXO E – PROGRAMA PARA GERAR CONTROLE COM CME	66
	ANEXO F – CÓDIGO DA S-FUNCTION SF_DINAMICA	67
	ANEXO G – CÓDIGO DA S-FUNCTION SF_CME	68

1 INTRODUÇÃO

Guimbais são suportes que permitem a rotação de um objeto em torno de um conjunto de eixos. Um sistema de três guimbais pode ser montado de modo que um objeto gire livremente em torno dos seus três eixos de rotação. A Figura 1.1 ilustra um sistema de guimbais com três graus de liberdade (GDL).

Figura 1.1 – Ilustração de um sistema de guimbais com 3 GDL.



Fonte: Autor.

Como pode ser visto na Figura 1.1, há três guimbais acoplados e ainda uma moldura externa para sustentar o conjunto. Cada guimbal gira em torno de seu eixo de rotação movido através de servomotores. O objetivo do conjunto é rotacionar o guimbal interno, o qual representa um corpo rígido com três graus de liberdade para rotação. Qualquer orientação no espaço pode ser obtida através de rotações sucessivas e independentes de cada guimbal, compondo o movimento.

Atualmente, guimbais são utilizados em diversas áreas, desde navegação à estabilização óptica de equipamentos de filmagem. No setor aeroespacial, uma utilização de guimbais é em motores foguete, como em NASA (2018). Estes são dispostos sobre pares

de guimbais, o que permite controlar a tração do motor para ser distribuída em diferentes eixos. Além disso podem ser usados em mísseis táticos, como em Smith et al. (1999).

Os guimbais são muito utilizados em sistemas de imageamento de aeronaves remotamente pilotadas (ARPs). Nesse caso, estabilizam as imagens que a câmera a bordo da aeronave produz, suavizando o movimento relativo entre a câmera e a aeronave, para que a imagem fique nítida. Também são utilizados da mesma maneira quando embarcados em satélites que capturam imagens. Além disso, a maioria dos ARP's são equipados com pilotos automáticos que utilizam os dados de uma Unidade de medida Inercial (do inglês IMU) para aquisição dos dados de altitude. Os dados medidos pela IMU podem conter erros, os quais devem ser mensurados para que não interfiram na missão. Com o objetivo de quantificar esses erros, pode-se utilizar uma plataforma de guimbais que simulam o voo de uma aeronave, o qual promove dados precisos da altitude da aeronave que podem ser comparados com os dados da IMU, como feito em Carter e Ali (2008).

A UFSM conta com sistemas didáticos de guimbais elaborados pelos próprios alunos em conjunto com os professores. A representação do aparato, em Computer Aided Design (CAD), está ilustrada na Figura 1.2. O aparelho didático está ilustrado na Figura 1.3.

Figura 1.2 – Modelo em CAD dos guimbais.

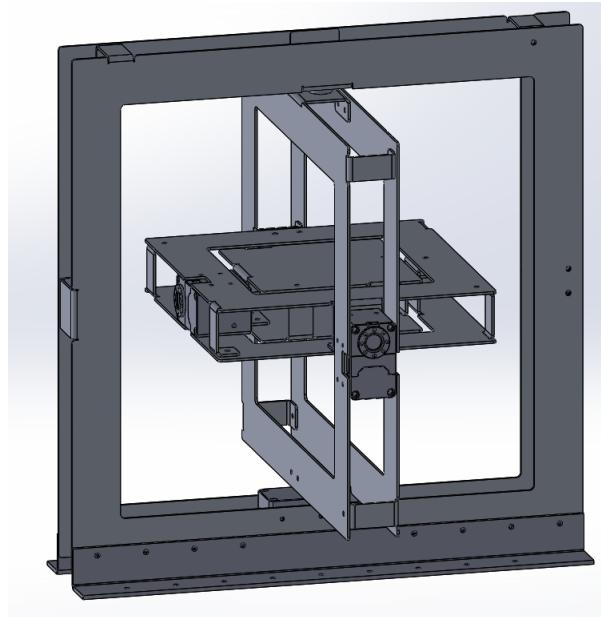


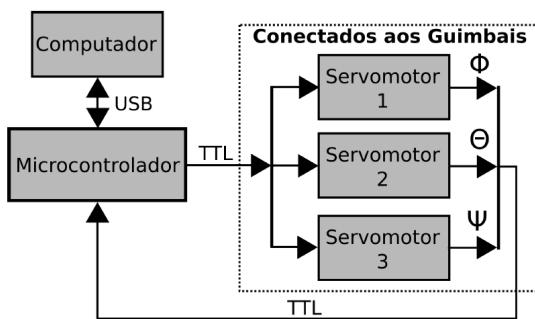
Figura 1.3 – Sistema de guimbais.



Os guimbais da Figura 1.3 são compostos de duas chapas de alumínio unidas por parafusos. As mãos francesas servem para fixar o sistema, fornecendo um melhor suporte estrutural. Os servomotores giram cada guimbal individualmente e ficam acoplados no interior do guimbal subsequente.

O funcionamento do sistema está representado na Figura 1.4.

Figura 1.4 – Esquema do funcionamento do sistema de guimbais com os servomotores.



Pode-se ver na Figura 1.4 que cada servomotor adquire e recebe um comando de um microcontrolador, que por sua vez envia e recebe dados de um computador. Os servomotores são comandados para atingir a referência lhes informada e retornam os ângulos de rotação com respeito a seus eixos. Os servomotores que equipam o sistema são os Dynamixel XM430-W210-T da empresa ROBOTIS, os quais são muito utilizados em projetos de robótica devido a agrupar em uma única unidade um motor elétrico, um *encoder*

absoluto para medição precisa de ângulos e um *drive* de comando e controle. O controle desses servomotores é feito através de um controle Proporcional Integral Derivativo próprio vindo de fábrica. Entretanto esse não prevê a carga imposta por cada guimbal. Se fez necessário obter um modelo experimental para o sistema, para que se possa implementar controles mais precisos.

Nesse projeto será desenvolvido um modelo matemático para o sistema de guimbais com 3 graus de liberdade, assim como a apresentação de um modelo para os servomotores Dynamixel XM430-W210-T e a aplicação de leis de controle lineares e não lineares nos sistemas.

1.1 OBJETIVOS E MOTIVAÇÃO

Esse projeto servirá para estabelecer um modelo matemático para os sistemas de guimbais da UFSM e para aplicação de técnicas de controle moderno, tanto linear quanto não linear no modelo.

O objetivo geral do trabalho é

- Realizar a modelagem matemática de um sistema de guimbais de 3 graus de liberdade para rotação e aplicar técnicas de controle lineares e não lineares no modelo.

Como objetivos específicos, pode-se citar:

- Apresentar o modelo dos motores Dynamixel XM430-W210-T;
- Aplicar as técnicas de controle: Regulador Linear Quadrático (LQR), Inversão dinâmica e Controle por mapeamento exponencial (CME) para controlar o torque aplicado a cada guimbal para atingir uma referência angular.

2 METODOLOGIA

Para a determinação de um modelo matemático para o sistema de guimbais de três graus de liberdade, recorreu-se aos conteúdos de física e matemática vistos nas disciplinas do curso de Engenharia Aeroespacial, abordados nos livros das referências Tewari (2007) e em Cannon (2012). Esses livros apresentam a base dos conteúdos avançados de física e matemática para diversas aplicações, as quais puderam ser adaptadas para o sistema de guimbais.

O modelo dos servomotores Dynamixel XM430-W210-T foi feito com base no trabalho de Máximo, Ribeiro e Afonso (2017) que estabelece um modelo para um servomotor da Dynamixel, o MX-28AT, com base em experimentos. Em Sartori e Fabbrin (2019) foi feito um modelo para esses servomotores com base em uma função de transferência. Esse modelo foi adaptado para poder ser utilizado em conjunto com os guimbais.

A teoria para a implementação das técnicas de controle LQR e Inversão Dinâmica foi obtida do livro Lewis, Stevens e Johson (2007), o qual trata sobre modelagem, simulação e controle de aeronaves. Para o CME, utilizou-se como base a referência Castro (2015), a qual é uma tese de doutorado sobre o assunto.

Para obter-se informações sobre os motores, recorreu-se aos manuais em ROBOTIS (2018).

Para obter-se mais informações sobre guimbais e suas aplicações, foi utilizado o trabalho feito por Smith et al. (1999), no qual realiza o controle sobre um sistema de guimbais de dois graus de liberdade. O trabalho de Carter e Ali (2008) foi utilizado como referência para uma aplicação do sistema de guimbais, onde implementa-se um sistema *Hardware in the loop* para calibração e testes de unidades de medidas iniciais, o qual pode ser aplicado ao sistema. O exposto em NASA (2018) mostra uma aplicação de um sistema de guimbais para dirigir o bocal de um foguete para direcionar o fluxo de saída dos gases.

3 REVISÃO BIBLIOGRÁFICA

3.1 MODELAGEM MATEMÁTICA DO SISTEMA DE GUIMBAIS COM 3 GRAUS DE LIBERDADE

Para o iniciar a modelagem matemática do sistema de guimbais é necessário estabelecer sistemas de referência para organizar o movimento. A Figura 1.1 mostra a configuração inicial do sistema, bem como um sistema de referência para o guimbal interno. Definir-se-ão quatro sistemas de referência. O sistema de referência 1, anexado ao guimbal 1 (interno), um sistema 2 anexado ao guimbal 2 (intermediário), um sistema 3 anexado ao guimbal 3 (externo) e um sistema inercial anexado à moldura. No instante inicial, todos estão alinhados conforme mostrado na Figura 1.1.

O sistema de referência para o guimbal 1, possui os eixos x_1 , y_1 e z_1 , solidários ao guimbal. O eixo x_1 aponta ao longo do eixo de rotação do guimbal 1. y_1 é ortogonal a x_1 e contido no plano do guimbal. O eixo z_1 é perpendicular ao plano. O ângulo para a rotação do guimbal 1 em torno de x_1 é ϕ . O sistema de referência para o guimbal 2, possui os eixos x_2 , y_2 e z_2 , solidários ao guimbal. O eixo y_2 aponta ao longo do eixo de rotação do guimbal 2. x_2 é ortogonal a y_2 e contido no plano do guimbal. O eixo z_2 é perpendicular ao plano. O ângulo para a rotação do guimbal 2 em torno de y_2 é θ . O sistema de referência para o guimbal 3, possui os eixos x_3 , y_3 e z_3 , solidários ao guimbal. O eixo z_3 aponta ao longo do eixo de rotação do guimbal 3. y_3 é ortogonal a z_3 e contido no plano do guimbal. O eixo x_3 é perpendicular ao plano. O ângulo para a rotação do guimbal 3 em torno de z_3 é ψ . O sistema de coordenadas iniciais possui os eixos x_I , y_I e z_I .

Utilizando os sistemas de referência definidos anteriormente, são necessários estabelecer vínculos cinemáticos. Analisando a Figura 1.1, pode-se ver que os eixos z_3 e z_I são sempre paralelos, assim como y_3 paralelo a y_I , e também x_2 paralelo a x_1 , não importando a rotação. Através desses vínculos é possível estabelecer matrizes de rotação para transformar os vetores de um sistema a outro. Essas matrizes derivam das matrizes elementares de rotação apresentadas em Tewari (2007), as quais podem ser definidas como:

Matriz de transição do sistema inercial para sistema 3:

$$\mathbf{C}_3^I = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Matriz de transição do sistema 3 para sistema 2:

$$C_2^3 = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.2)$$

Matriz de transição do sistema 2 para sistema 1:

$$C_1^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.3)$$

É possível transitar entre sistemas mais distantes através de rotações sucessivas, as quais são dadas pela multiplicação das matrizes elementares.

Matriz de transição do sistema inercial para sistema 2:

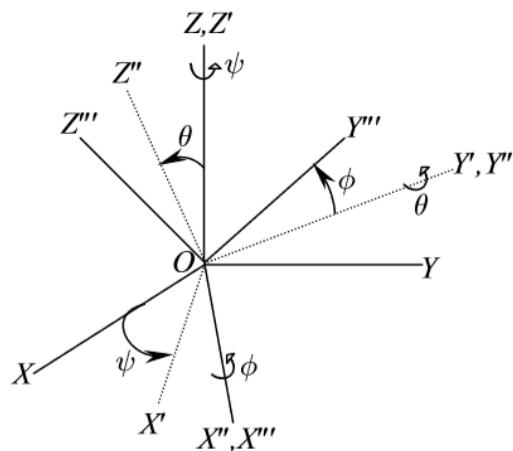
$$C_2^I = C_2^3 C_3^I \quad (3.4)$$

Matriz de transição do sistema inercial para sistema 1:

$$C_1^I = C_1^2 C_2^3 C_3^I \quad (3.5)$$

A Figura 3.1 mostra a sequência de rotação 3-2-1, comumente utilizada no setor aeroespacial e que será adotada ao decorrer do trabalho.

Figura 3.1 – Sequência de rotação 3-2-1.



Fonte: Tewari (2007)

Na Figura 3.1, o sistema XYZ representa o sistema inercial. Multiplicado pela matriz (3.1), chega-se no sistema $X'Y'Z'$ que representa o sistema do guimbal 3. Multiplicando agora pela matriz (3.2), chega-se no sistema $X''Y''Z''$, coincidente com o sistema

do guimbal 2. Para chegar ao sistema do guimbal 1, basta multiplicar pela equação (3.3), o qual está ilustrado como $X'''Y'''Z'''$.

Definidos os sistemas de referência e as matrizes para transitar de um sistema a outro, é necessário definir as velocidades angulares relativas entre cada guimbal, representadas através da variação dos ângulos de Euler, definidos anteriormente, de forma:

- $\dot{\phi}$: Velocidade angular relativa do guimbal 1 com respeito ao guimbal 2.
- $\dot{\theta}$: Velocidade angular relativa do guimbal 2 com respeito ao guimbal 3.
- $\dot{\psi}$: Velocidade angular relativa do guimbal 3 com respeito ao referencial inercial.

De posse das velocidades angulares relativas, é possível definir as velocidades angulares de cada guimbal escritas em seus respectivos referenciais, as quais representam as equações da cinemática dos guimbais.

Velocidade angular do guimbal 1 com respeito a 2 escrita no referencial 1:

$$\vec{\omega}_{2_1}^1 = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (3.6)$$

Velocidade angular do guimbal 2 com respeito a 3 escrita no referencial 2:

$$\vec{\omega}_{3_2}^2 = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \quad (3.7)$$

Velocidade angular do guimbal 3 com respeito ao referencial inercial escrita no referencial 3:

$$\vec{\omega}_{I_3}^3 = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3.8)$$

Como as Leis de Newton são válidas para referenciais iniciais e como os referenciais 1, 2 e 3 são não iniciais, é necessário escrever as velocidades angulares em relação ao referencial inercial. A equação para o guimbal 3 já está pronta, equação (3.8), pois já está medida em relação ao referencial inercial. A equação para o guimbal 2 deve ser adaptada. A velocidade angular do guimbal 2 com respeito ao referencial inercial será a soma das velocidades angulares do guimbal 3 com relação ao inercial e da velocidade angular de 2 com relação a 3, ambas escritas no referencial 2, ou seja:

$$\vec{\omega}_{I_2}^2 = \vec{\omega}_{3_2}^2 + \vec{\omega}_{I_2}^3 \quad (3.9)$$

O primeiro termo do lado direito da equação (3.9) é a equação (3.7). O segundo

termo pode ser obtido fazendo uma rotação para sair do sistema 3 e ir para o 2, através da multiplicação das equações (3.2) e (3.8), obtendo-se

$$\vec{\omega}_{I_2}^2 = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_2^3 \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3.10)$$

A equação de cinemática para o guimbal 1 será a soma da velocidade angular do guimbal 1 com respeito a 2 com a velocidade angular de 2 com respeito a 3 e de 3 com respeito ao referencial inercial, todas escritas no referencial 1.

$$\vec{\omega}_{I_1}^1 = \vec{\omega}_{2_1}^1 + \vec{\omega}_{3_1}^2 + \vec{\omega}_{I_1}^3 \quad (3.11)$$

O primeiro termo do lado direito da equação (3.11) é a equação (3.6). O segundo é a equação (3.7) rotacionada para o referencial 1, através da multiplicação pela equação (3.3). O terceiro termo é a equação (3.8) rotacionada primeiramente pela multiplicação da equação (3.2) e após outra rotação pela equação (3.3), resultando em

$$\vec{\omega}_{I_1}^1 = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + C_1^2 \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_1^2 C_2^3 \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (3.12)$$

As equações (3.12), (3.10) e (3.8) determinam a cinemática dos guimbais 1, 2 e 3 respectivamente.

A dinâmica do sistema pode ser definida através das Leis de Newton. Segundo Cannon (2012) a quantidade de movimento angular \vec{H} de um corpo com relação a um ponto fixo P é definida como sendo:

$$\vec{H} = \int \vec{r} \times \vec{v} dm \quad (3.13)$$

Em que \vec{r} é o vetor posição de um diferencial de massa dm e \vec{v} é a velocidade linear instantânea desse, ambos com respeito a P.

Para um corpo rígido, a velocidade linear é dada por

$$\vec{v} = \vec{r} \times \vec{\omega} \quad (3.14)$$

Em que $\vec{\omega}$ é a velocidade angular do corpo rígido com respeito a um referencial não girante.

Substituindo (3.14) em (3.13) chega-se em

$$\vec{H} = \int \vec{r} \times (\vec{r} \times \vec{\omega}) dm \quad (3.15)$$

A equação (3.15) pode ser simplificada de forma

$$\vec{H} = \mathbf{J} \vec{\omega} \quad (3.16)$$

Em que \mathbf{J} é a matriz de inércia, que para um sistema cartesiano com origem em P, com os eixos ortogonais x, y e z, é definida como

$$\mathbf{J} = \begin{bmatrix} \int y^2 + z^2 dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int x^2 + z^2 dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int x^2 + y^2 dm \end{bmatrix} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \quad (3.17)$$

A segunda Lei de Newton para rotação diz que o somatório dos momentos atuantes em um corpo é igual a variação da quantidade de movimento angular desse corpo no tempo, com respeito a um referencial inercial, ou seja:

$$\sum \vec{M} = \frac{d\vec{H}}{dt} \quad (3.18)$$

É assumida a hipótese de que os torques atuantes nos guimbais são aplicados nos respectivos eixos de rotação os quais passam pelo centro de massa.

Os referencias dos guimbais 1, 2 e 3 são não inerciais. Devido a isso, para determinar a derivada da quantidade de movimento angular de cada guimbal, é necessário utilizar a relação de derivada para referenciais não inerciais:

$$\frac{d\vec{V}}{dt}_{Inercial} = \frac{d\vec{V}}{dt}_{NãoInercial} + \vec{\omega} \times \vec{V} \quad (3.19)$$

Em que \vec{V} é um vetor qualquer e $\vec{\omega}$ é a velocidade angular do sistema não inercial.

Analizando o guimbal 1, utilizando a equação (3.16), sua quantidade de movimento angular em relação ao centro de massa, escrita no referencial 1 será

$$\vec{H}_1 = \mathbf{J}_1 \vec{\omega}_{I_1}^1 \quad (3.20)$$

Em que \mathbf{J}_1 é a matriz de inércia do guimbal 1 em relação ao seu centro de massa. O momento resultante atuando no guimbal 1, segundo a equação (3.18), é

$$\vec{M}_1 = \frac{d\vec{H}_1}{dt} \quad (3.21)$$

O referencial é não inercial, portanto deve-se usar a equação (3.19) para fazer a derivada.

$$\vec{M}_1 = \frac{d(\mathbf{J}_1 \vec{\omega}_{I_1}^1)}{dt} + \vec{\omega}_{I_1}^1 \times (\mathbf{J}_1 \vec{\omega}_{I_1}^1) \quad (3.22)$$

Para o guimbal 2, a quantidade de movimento angular total será a soma das quantidades de movimento angular do guimbal 2 com o guimbal 1, de forma que

$$\vec{H}_2 = \mathbf{J}_1 \vec{\omega}_{I_1}^1 + \mathbf{J}_2 \vec{\omega}_{I_2}^2 \quad (3.23)$$

Em que \mathbf{J}_2 é a matriz de inércia do guimbal 2. O momento resultante no guimbal 2 será então

$$\vec{M}_2 = \frac{d\vec{H}_2}{dt} \quad (3.24)$$

Novamente é necessário utilizar a equação (3.19), pois o referencial 2 é não inercial. Além disso, é necessário escrever todos os termos no referencial 2, utilizando as matrizes de rotação apresentadas anteriormente, o que resulta

$$\vec{M}_2 = \mathbf{C}_1^{2T} \left(\frac{d(\mathbf{J}_1 \vec{\omega}_{I_1}^1)}{dt} \Big|_1 + \vec{\omega}_{I_1}^1 \times \mathbf{J}_1 \vec{\omega}_{I_1}^1 \right) + \frac{d(\mathbf{J}_2 \vec{\omega}_{I_2}^2)}{dt} \Big|_2 + \vec{\omega}_{I_2}^2 \times (\mathbf{J}_2 \vec{\omega}_{I_2}^2) \quad (3.25)$$

A quantidade de movimento angular total do guimbal 3 será a soma das quantidades de movimento angulares dos guimbais 1, 2 e 3, ou seja

$$\vec{H}_3 = \mathbf{J}_1 \vec{\omega}_{I_1}^1 + \mathbf{J}_2 \vec{\omega}_{I_2}^2 + \mathbf{J}_3 \vec{\omega}_{I_3}^3 \quad (3.26)$$

Em que \mathbf{J}_3 é a matriz de inércia do guimbal 3. O momento resultante no guimbal 3 será dado pela derivada da equação do momento angular, passando todos os termos para o referencial 3 com as matrizes de rotação, chegando-se em

$$\vec{M}_3 = \mathbf{C}_2^{3T} \mathbf{C}_1^{2T} \vec{B}_1 + \mathbf{C}_2^{3T} \vec{B}_2 + \vec{B}_3 \quad (3.27)$$

Em que os vetores \vec{B}_1 , \vec{B}_2 e \vec{B}_3 são

$$\begin{aligned} \vec{B}_1 &= \frac{d(\mathbf{J}_1 \vec{\omega}_{I_1}^1)}{dt} \Big|_1 + \vec{\omega}_{I_1}^1 \times (\mathbf{J}_1 \vec{\omega}_{I_1}^1) \\ \vec{B}_2 &= \frac{d(\mathbf{J}_2 \vec{\omega}_{I_2}^2)}{dt} \Big|_2 + \vec{\omega}_{I_2}^2 \times (\mathbf{J}_2 \vec{\omega}_{I_2}^2) \\ \vec{B}_3 &= \frac{d(\mathbf{J}_3 \vec{\omega}_{I_3}^3)}{dt} \Big|_3 + \vec{\omega}_{I_3}^3 \times (\mathbf{J}_3 \vec{\omega}_{I_3}^3) \end{aligned}$$

Anteriormente foi assumido que o centro de massa de cada guimbal era coincidente com o respectivo eixo de rotação, no entanto, é necessário ainda considerar o momento gerado pela gravidade, devido ao desalinhamento entre o centro de massa de cada guimbal

e seu eixo de rotação. Para isso, define-se os vetores:

$$\vec{r}_1 = \begin{bmatrix} r_{1x} \\ r_{1y} \\ r_{1z} \end{bmatrix} \quad (3.28)$$

$$\vec{r}_2 = \begin{bmatrix} r_{2x} \\ r_{2y} \\ r_{2z} \end{bmatrix} \quad (3.29)$$

$$\vec{r}_3 = \begin{bmatrix} r_{3x} \\ r_{3y} \\ r_{3z} \end{bmatrix} \quad (3.30)$$

Os vetores \vec{r}_1 , \vec{r}_2 e \vec{r}_3 indicam a posição do centro de massa dos guimbais 1, 2 e 3, respectivamente, em relação ao meio do seu eixo de rotação escrito em seu próprio sistema de referência.

É necessário definir o vetor gravidade:

$$\vec{g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (3.31)$$

Em que g é o módulo da aceleração gravitacional local. Esse vetor está escrito no sistema de referência inercial.

O momento gravitacional agindo no guimbal 1, escrito no seu sistema de referência, será

$$\vec{M}_{1g} = m_1 \vec{r}_1 \times (\mathbf{C}_1^I \vec{g}) \quad (3.32)$$

Analogamente, os momentos gravitacionais para o guimbal 2 e 3, escritos em seus sistemas serão

$$\vec{M}_{2g} = m_2 \vec{r}_2 \times (\mathbf{C}_2^I \vec{g}) \quad (3.33)$$

$$\vec{M}_{3g} = m_3 \vec{r}_3 \times (\mathbf{C}_3^I \vec{g}) \quad (3.34)$$

Em que m_1 , m_2 e m_3 são as massas dos guimbais 1, 2 e 3 respectivamente.

Os momentos atuantes nos eixos de giro dos guimbais 1, 2 e 3 são dados pelos Equações (3.35), (3.36) e (3.37) respectivamente.

$$\vec{M}_{eixo_1} = \begin{bmatrix} \tau_1 - b_1 \dot{\phi} \\ M_{y1} \\ M_{z1} \end{bmatrix} \quad (3.35)$$

$$\vec{M}_{eixo_2} = \begin{bmatrix} M_{x2} \\ \tau_2 - b_2 \dot{\theta} \\ M_{z2} \end{bmatrix} \quad (3.36)$$

$$\vec{M}_{eixo_3} = \begin{bmatrix} M_{x3} \\ M_{y3} \\ \tau_3 - b_3 \dot{\psi} \end{bmatrix} \quad (3.37)$$

Em que τ_1 , τ_2 e τ_3 são os torques provenientes dos motores e b_1 , b_2 e b_3 são as constantes de atrito dos eixos dos guimbais 1, 2 e 3, respectivamente. A multiplicação das constantes de atrito viscoso pelas velocidades angulares representam o torque de resistência devido ao atrito viscoso. Os outros termos nas equações (3.38), (3.39) e (3.40) são as reações de apoio.

O momento total atuante no guimal 1, dado pela Equação (3.22), será a soma do momento no eixo de rotação do guimal 1, fornecido pela Equação (3.35), com o momento gravitacional nesse guimal, pela Equação (3.32), resultando na Equação (3.38). Da mesma forma, para o guimal 2, o momento total expresso por (3.25) será a soma de (3.33) com (3.36), resultando na Equação (3.39). Para o 3, o momento total dado por (3.27) será a soma de (3.34) com (3.37), o que resulta na Equação (3.40).

$$\vec{M}_1 = \vec{M}_{1g} + \begin{bmatrix} \tau_1 - b_1 \dot{\phi} \\ M_{y1} \\ M_{z1} \end{bmatrix} \quad (3.38)$$

$$\vec{M}_2 = \vec{M}_{2g} + \begin{bmatrix} M_{x2} \\ \tau_2 - b_2 \dot{\theta} \\ M_{z2} \end{bmatrix} \quad (3.39)$$

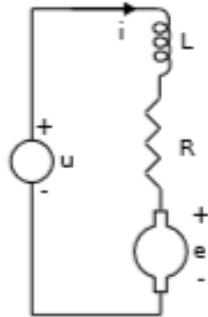
$$\vec{M}_3 = \vec{M}_{3g} + \begin{bmatrix} M_{x3} \\ M_{y3} \\ \tau_3 - b_3 \dot{\psi} \end{bmatrix} \quad (3.40)$$

3.2 MODELAGEM DOS SERVOMOTORES

Como mencionado anteriormente, os motores utilizados no sistema de guimbais serão os Dynamixel XM430-W210T da empresa coreana ROBOTIS. Para obter o modelo dos motores, tomou-se como base o exposto em Máximo, Ribeiro e Afonso (2017). Nesse trabalho foram determinados diversos parâmetros para o modelo do motor e caixa de redução do servomotor MX-28AT, baseado em um circuito equivalente ilustrado na Figura

3.2.

Figura 3.2 – Circuito equivalente do motor.



Fonte: Máximo, Ribeiro e Afonso (2017)

Em que u é a tensão média aplicada na malha, i é a corrente, R é a resistência equivalente da armadura, L é a indutância da armadura, e é chamada de força eletromotriz induzida. Fazendo a análise das tensões na malha, chega-se em:

$$u = Ri + L \frac{di}{dt} + e \quad (3.41)$$

A força eletromotriz induzida pode ser modelada como:

$$e = N \frac{\omega}{K_w}$$

Em que K_w é constante de força eletromotriz induzida do motor, N é a relação entre engrenagens do motor e ω é a velocidade angular do disco de carga do motor.

Como a indutância de armadura é muito pequena, foi assumido como sendo nula. Com isso, a corrente de armadura pode ser definida, com a equação (3.41), como

$$i = \frac{1}{R} (u - \frac{N}{K_w} \omega) \quad (3.42)$$

O torque eletromagnético visto no eixo do motor pode ser definido como

$$\tau_m = K_t i \quad (3.43)$$

Em que K_t é a constante de torque do motor. Devido ao sistema de engrenagens, as variáveis no eixo do motor podem ser convertidas para o eixo da carga multiplicando-se por um fator $N\eta$, em que η é a eficiência do conjunto de engrenagens. O torque no eixo da carga será

$$\tau_l = N\eta\tau_m \quad (3.44)$$

A relação entre as constantes de torque e de força eletromotriz é dada por

$$K_t = \frac{1}{K_w} \quad (3.45)$$

Ainda, fazendo a análise dos torques atuantes tanto no eixo do motor, quanto no eixo de carga, pode-se chegar nas Equações (3.46) e (3.47) respectivamente.

$$J_0 \dot{\omega}_m = \tau_m - \frac{\tau_l}{N\eta} - \tau_{fm} \quad (3.46)$$

$$J_l \dot{\omega} = \tau_l - \tau_{fl} \quad (3.47)$$

Em que ω_m é a velocidade angular do eixo do motor, sendo $\omega_m = N\omega$. J_0 é o momento de inércia do rotor mais engrenagens visto no eixo do motor e J_l é momento de inércia da carga conectada ao disco de carga do servomotor. τ_{fm} e τ_{fl} são respectivamente os torques de fricção nos eixos do motor e de carga. No trabalho, foi assumido que τ_{fl} é nulo e τ_{fm} é igual a $b_0\omega_m$, em que b_0 é a constante de atrito viscoso do conjunto rotor do motor e engrenagens visto no eixo do motor.

Através de medidas experimentais, notou-se que as medições de corrente pelo motor eram bastante imprecisas, portanto foi necessário prosseguir através do controle por PWM (*Pulse Width Modulation*).

Com as Equações (3.46), (3.47) e (3.41), é possível chegar na função de transferência que relaciona a tensão de entrada com a velocidade angular de saída, que é dada pela Equação (3.48).

$$\omega(s) = \frac{N\eta K_t}{R(J_l + J_{me})s + b_{me}R + \frac{K_t N^2 \eta}{K_w}} U(s) \quad (3.48)$$

J_{me} é o momento de inércia do rotor do motor, das engrenagens e do disco de conexão de carga, visto pelo eixo de carga. J_{me} se relaciona com J_0 pela seguinte expressão:

$$J_{me} = N^2 \eta J_0 \quad (3.49)$$

b_{me} é a constante de atrito viscoso do conjunto rotor do motor e engrenagens visto no eixo de carga, relacionado com b_0 por:

$$b_{me} = N^2 \eta b_0 \quad (3.50)$$

Essa função de transferência é feita considerando que o eixo de rotação é inercial, ou seja, que a carcaça do motor está fixa e a inércia rotativa é composta pelo momento de inércia do conjunto rotor, eixo e engrenagens visto no eixo da carga, mais o momento de inércia da carga acoplada a tal eixo de saída do servomotor, sendo válida quando a

carcaça do motor está fixa em um referencial inercial. Ou seja, o modelo de função de transferência acima não vale quando o motor gira junto com os guimbais. Também, este modelo não vale quando há saturação do torque e/ou da velocidade angular, visto que, conceitualmente, o modelo de função de transferência é linear e não leva em conta essas não linearidades. Mesmo assim, este modelo linear idealizado é válido para ensaios em bancada do servomotor. Como será explicado posteriormente, o modelo foi modificado para poder ser usado.

A fonte de alimentação do servomotor fornece uma tensão contínua (V_s idealmente 12V). O microcontrolador contido no servomotor modula a tensão de entrada do motor a partir da técnica PWM, podendo gerar tensões de comando positivas ou negativas de forma:

$$u = \frac{V_s}{N_{pwm}} u_{pwm}$$

Onde u_{pwm} é um comando discreto (inteiro), que varia entre $-N_{pwm}$ e $+N_{pwm}$.

Alguns dados do servomotor fornecidos no site do fabricante ROBOTIS (2018) estão ilustrados na tabela 3.1.

Tabela 3.1 – Dados dos servomotores.

Parâmetro	Valor	Descrição (condição de alimentação)
N_{PWM}	885	Número de níveis do PWM que comanda a geração da tensão (positivo ou negativo)
N	212,6	-
V_s	12 V	Tensão de suprimento do servomotor
$\tau_{l,s}$	2,7 N.m	Torque na carga para velocidade nula (11,1 V, 2,1A)
	3 N.m	Torque na carga para velocidade nula (12 V, 2,3 A)
	3,7 N.m	Torque na carga para velocidade nula (14,8 V, 2,7 A)
$\omega_{max,nl}$	70 rpm (7,33 rad/s)	Velocidade máxima sem carga (tensão de alimentação: 11,1 V)
	77 rpm (8,06 rad/s)	Velocidade máxima sem carga (tensão de alimentação: 12 V)
	95 rpm (9,95 rad/s)	Velocidade máxima sem carga (tensão de alimentação: 14,8 V)

Os dados do fabricante não são suficientes para modelar completamente o motor. Os parâmetros faltantes, K_t , R , η , b_{me} e J_{me} foram estimados experimentalmente em Sartori e Fabbrin (2019). Nesse trabalho foram feitos diversos experimentos para determinar os parâmetros das equações do modelo. As condições desses ensaios foram:

- Tensão de alimentação do motor: 12V;
- Comando por PWM (*Pulse Width Modulation*);
- Recepção de dados via USB usando uma placa U2D2 do Dynamixel;
- Programa em Python para comandar e ler dados dos motores;

- A carga mecânica é somente a inércia do rotor, engrenagens e disco de conexão à carga;¹
- Foram ensaiados apenas dois servomotores.

Primeiramente, foi determinada uma relação linear entre o comando de PWM aplicado e a tensão produzida no motor (medida diretamente através de um multímetro), expressa na equação (3.51).

$$u = 0,01279 \cdot u_{PWM} - 0,002875 \quad (3.51)$$

Onde u_{pwm} varia entre $-N_{pwm}$ e $+N_{pwm}$, sendo esta a origem da saturação do servomotor.

Em seguida, foi medida a resistência de armadura R do motor diretamente também com um multímetro.

A constante de torque K_t foi encontrada através da Equação (3.42) e da (3.45). Foram impostos diversos comandos de PWM para os motores através do programa em Python, que retornava as velocidades angulares, corrente, deslocamento angular e tempo medidas pelo próprio servomotor. Com as medidas adquiridas e a tensão calculada por (3.51) foi possível encontrar K_t , isolando-o em (3.42).

A eficiência do conjunto de engrenagens η foi calculada com base em um experimento utilizando uma mola. A mola foi ensaiada para obter-se a constante de elasticidade. Em seguida foi montado o experimento onde o motor atingia o torque de *stall* para determinada tensão, ilustrado na Figura 3.3

¹Em um dos ensaios, foi inserida uma inércia extra, de modo a obter mais dados para verificar o momento de inércia do conjunto do rotor do motor, engrenagens e disco de carga. Além de melhorar a amostragem de dados do regime transitório.

Figura 3.3 – Ensaio de torque.



Fonte: Sartori e Fabbrin (2019)

Na Figura 3.3, o elemento indicado pelo número 1 representa a barra de ferro aço-plada para aumentar a rigidez do guimbal. Os elementos marcados com os números 2 e 3 mostram respectivamente a mola e o servomotor ensaiados. O guimbal é identificado pelo número 4. Como o guimbal é feito de alumínio, possui alta flexibilidade, o que inviabilizou a medição de torque pela mola. Com o aumento da rigidez provocado pela barra, o deslocamento linear sentido pela mola, através da rotação do guimbal, pode ser assumido como proporcional ao ângulo de rotação do eixo de carga do servomotor.

Foi controlada a corrente do motor e o deslocamento da mola até atingir o torque de *stall*, podendo ser calculado pela Equação (3.52).

$$\tau_{l_{stall}} = K_x \Delta x l \quad (3.52)$$

Em que Δx é o deslocamento da mola e l é o braço de alavanca, que no caso era 120mm. O torque de *stall* é aquele no qual o torque do servomotor cancela o da mola, mantendo o guimbal em equilíbrio.

Foram traçados gráficos de torque por corrente. A inclinação dessas curvas forneceu um K_{tcarga} na carga médio. Para encontrar-se a eficiência, foi utilizada a equação (3.53)

$$\eta = \frac{K_{tcarga}}{N K_t} \quad (3.53)$$

As constantes de atrito viscoso foram determinadas através dos valores em regime

permanente das velocidades angulares experimentais. b_0 , constante de atrito viscoso do rotor do motor e do conjunto de engrenagens, escrita em função da velocidade de rotação do rotor do motor, pode ser calculada pela Equação (3.54).

$$b_0 = \frac{K_t i_{ss}}{\omega_{ss} N} - \frac{K_t^2}{R} \quad (3.54)$$

J_0 foi determinado com base nas constantes de tempo dos dados de velocidade angular pelo tempo do motor. A constante de tempo de uma resposta ao degrau unitário de uma função de transferência de primeira ordem para um sistema linear e invariante no tempo pode ser obtida observando o tempo para que atinja $1 - e^{-1}$ (aproximadamente 63%) do valor em regime permanente, em que e é o número de Euler.

A constante de tempo da função de transferência de primeira ordem da Equação (3.48) é definida pela Equação (3.55).

$$\tau = \frac{R(J_l + J_{me})}{b_{me}R + K_t^2 N^2 \eta} \quad (3.55)$$

Substituindo os valores encontrados de constantes de tempo em (3.55) pode-se encontrar J_{me} , o qual está relacionado com J_0 , momento de inércia do rotor mais engrenagens visto no eixo do motor por (3.49)

A tabela 3.2 mostra os parâmetros calculados e medidos.

Tabela 3.2 – Resumo dos parâmetros calculados e medidos.

Parâmetro	Valor
R	4.6Ω
K_t	$0.007145 Nm/A$
η	76.97%
b_0	$3.0447 \cdot 10^{-7} Nms$
J_0	$2.6395 \cdot 10^{-7} kg.m^2$

3.3 REGULADOR LINEAR QUADRÁTICO COM REALIMENTAÇÃO DE SAÍDA E RASTREADOR DE REFERÊNCIA

A teoria do controle apresentada nesta seção primeiramente mostrará como usar técnicas modernas para projetar sistemas de aumento de estabilidade e pilotos automáticos. Segundo Lewis, Stevens e Johson (2007), isso pode ser atingido regulando-se certos estados da planta para zero, enquanto obtém-se respostas satisfatórias em malha fechada. Isso envolve estabilizar o sistema alocando os polos em malha fechada nos locais desejados. Uma das maneiras de fazer isso é através da implementação de um regulador linear

quadrático com realimentação de saída, o qual constituirá o sistema de aumento de estabilidade e após fazer o projeto do rastreador, para seguimento de referência.

3.3.1 Regulador linear quadrático ótimo com realimentação de saída

Assumindo um sistema linear representado no espaço de estados por:

$$\dot{\vec{x}} = \mathbf{A} \vec{x} + \mathbf{B} \vec{u} \quad (3.56)$$

$$\vec{y} = \mathbf{C} \vec{x} \quad (3.57)$$

Em que \mathbf{A} e \mathbf{B} são as matrizes linearizadas do sistema, $\vec{x} = \vec{x}(t) \in \mathbb{R}^n$ é o vetor de estados, $\vec{u} = \vec{u}(t) \in \mathbb{R}^m$ é o vetor de controle e $\vec{y} \in \mathbb{R}^p$ é o vetor de saída escolhida a matriz \mathbf{C} .

O vetor de controle pode ser escrito, utilizando a realimentação de saída, de forma:

$$\vec{u} = -\mathbf{K} \vec{y} \quad (3.58)$$

Em que \mathbf{K} é a matriz $m \times p$ de ganhos por realimentação de saída a ser determinada.

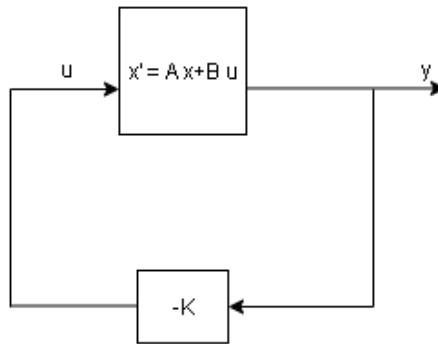
Segundo Lewis, Stevens e Johson (2007), o objetivo da regulação de estados é conduzir a partir de qualquer condição inicial para o estado nulo, ou seja, fazer com que após qualquer perturbação a resposta do estado converja para o equilíbrio, assim garantindo estabilidade. Isso pode ser garantido escolhendo uma lei de controle que minimize uma função de custo de forma:

$$J = \frac{1}{2} \int_0^{\infty} (\vec{x}^T \mathbf{Q} \vec{x} + \vec{u}^T \mathbf{R} \vec{u}) dt \quad (3.59)$$

Em que \mathbf{Q} e \mathbf{R} são matrizes hermitianas definidas positivas. Essas matrizes determinam a importância relativa do estado e o consumo da energia pelos controles. O problema consiste em determinar a matriz de ganhos \mathbf{K} que minimizam o índice de desempenho J para encontrar a lei de controle, equação (3.58), ótima para qualquer estado inicial.

O diagrama de blocos mostrando a configuração ótima está indicado na figura 3.4.

Figura 3.4 – Sistema regulador ótimo.



Fonte: Autor.

O procedimento para o cálculo da matriz de ganhos K é determinado no domínio do tempo. Basicamente, para encontrar a matriz de ganhos, é necessário resolver o seguinte conjunto de equações acopladas:

$$0 = A_c^T P + P A_c + C^T K^T R K C + Q \quad (3.60)$$

$$0 = A_c S + S A_c^T + X \quad (3.61)$$

$$0 = R K C S C^T - B^T P S C^T \quad (3.62)$$

Em que A_c é a matriz de dinâmica em malha fechada e X são definidos por:

$$A_c = A - B K C$$

$$X = \vec{x}(0) \vec{x}^T(0)$$

P é uma matriz positiva definida auxiliar, a qual deve satisfazer a equação (3.60), enquanto S é uma matriz simétrica $n \times n$ de multiplicadores de Lagrange que precisa ser determinada de modo a satisfazer (3.61). As equações (3.60) e (3.61) são equações de Lyapunov e (3.62) é uma equação para a matriz de ganhos K .

Para um caso particular em que as saídas são os próprios estados, ou seja, C é igual a uma matriz identidade, é possível encontrar a matriz de ganhos ótima resolvendo a

equação reduzida de Riccati:

$$A^T P + PA - PBR^{-1}BP^T + Q = 0 \quad (3.63)$$

Deve-se resolver a Equação (3.63) para P e substituir o resultado em

$$K - T^{-1}B^T P = R^{-1}BP \quad (3.64)$$

Em que T é definido por $R = T^T T$.

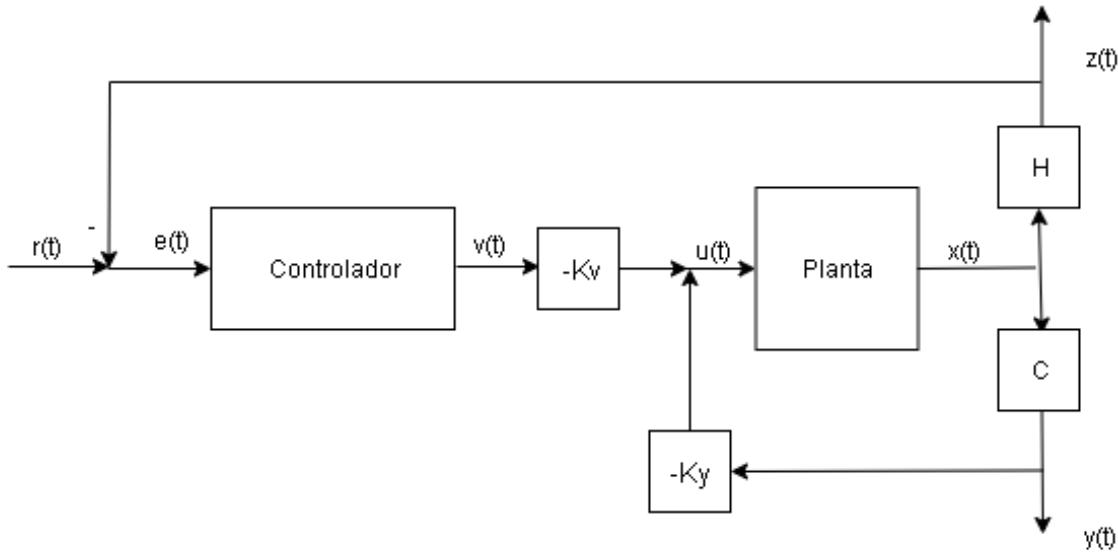
Segundo Ogata (2011), se $A - BK$ for estável, existirá a matriz P definida positiva que satisfaça a equação (3.63). Substituindo essa matriz em (3.64), encontra-se a matriz de ganhos ótima.

3.3.2 Rastreamento de referência

O tipo de controle desenvolvido na seção anterior tem como objetivo realizar o aumento da estabilidade do sistema, o qual será incorporado a essa seção. Entretanto, esse tipo de controle faz com que o sistema retorne para o equilíbrio após uma perturbação. Nessa seção será apresentado um método para que o sistema siga uma referência diferente do ponto de equilíbrio.

O diagrama de blocos do sistema com rastreador de referência está ilustrado na Figura 3.5.

Figura 3.5 – Diagrama de blocos para rastreio de referência.



Fonte: Autor.

Recorrendo novamente ao espaço de estados representado pelas equações (3.56) e (3.57), pode-se ainda definir

$$\vec{z} = \mathbf{H} \vec{x} \quad (3.65)$$

\vec{z} é chamado de *performance output*, o qual deve rastrear uma determinada referência.

Ainda é necessário incluir a dinâmica do controlador, a qual pode ser definida como:

$$\dot{\vec{w}} = \mathbf{F} \vec{w} + \mathbf{G} \vec{e} \quad (3.66)$$

$$\vec{v} = \mathbf{D} \vec{w} + \mathbf{J} \vec{e} \quad (3.67)$$

Em que $\vec{w}(t)$ é um vetor de estados, $\vec{v}(t)$ é a saída do controlador, e a entrada é igual ao erro de rastreio, definido como

$$\vec{e}(t) = \vec{r}(t) - z(t) \quad (3.68)$$

Onde $\vec{r}(t)$ é a referência a ser rastreada. As matrizes \mathbf{F} , \mathbf{G} , \mathbf{D} , \mathbf{J} são escolhidas previamente segundo o tipo de controlador a ser usado.

O vetor de controle pode ser escrito de forma

$$\vec{u} = -\mathbf{K}_y \vec{y} - \mathbf{K}_v \vec{v} \quad (3.69)$$

Em que as matrizes de ganhos K_y e K_v devem ser determinadas para chegar-se num resultado satisfatório de rastreio de $\vec{r}(t)$. As equações do diagrama de blocos da figura 3.5 podem ser escritas na forma aumentada como sendo:

$$\begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{w}} \end{bmatrix} = \mathbf{A}_a \begin{bmatrix} \vec{x} \\ \vec{w} \end{bmatrix} + \mathbf{B}_a \vec{u} + \begin{bmatrix} \vec{0} \\ \mathbf{G} \end{bmatrix} \vec{r} \quad (3.70)$$

$$\begin{bmatrix} \vec{y} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \vec{0} \\ -\mathbf{JH} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{w} \end{bmatrix} + \begin{bmatrix} \vec{0} \\ \mathbf{J} \end{bmatrix} \vec{r} \quad (3.71)$$

Em que \mathbf{A}_a e \mathbf{B}_a são as matrizes aumentadas do sistema dadas por:

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{A} & \vec{0} \\ -\mathbf{GH} & \mathbf{F} \end{bmatrix}$$

$$\mathbf{B}_a = \begin{bmatrix} \mathbf{B} \\ \vec{0} \end{bmatrix}$$

É importante ressaltar que todos os termos das matrizes das equações (3.70) e (3.71) são conhecidos. O que resta é determinar as matrizes de ganhos \mathbf{K} e \mathbf{L} da Equação (3.69), os quais podem ser encontrados com um procedimento análogo ao utilizado na seção anterior.

3.4 LINEARIZAÇÃO

Considerando um sistema não linear de forma:

$$\dot{X} = f(X, U) \quad (3.72)$$

Em que X é o vetor de estados e U é o vetor de controle.

Segundo Lewis, Stevens e Johson (2007) é possível linearizar o sistema expresso na Equação (3.72) através da expansão em série de Taylor em torno de um ponto de equilíbrio $f(X_e, U_e)$, com apenas as derivadas de primeira ordem, de forma:

$$\dot{X} = f(X_e, U_e) + \frac{\partial f}{\partial X} \delta X + \frac{\partial f}{\partial U} \delta U \quad (3.73)$$

As derivadas parciais denotam as matrizes Jacobianas e as perturbações são definidas como

$$\delta X = X - X_e$$

$$\delta U = U - U_e$$

O ponto de equilíbrio é obtido de forma

$$0 = \dot{X} = f(X_e, U_e)$$

A Equação (3.73) pode ser reescrita na forma mais conhecida

$$\dot{x} = \mathbf{A}x + \mathbf{B}u$$

Em que x e u representam as perturbações em torno do ponto de equilíbrio.

A implementação numérica pode ser feita definindo-se um valor pequeno ϵ para perturbar o ponto de equilíbrio. As matrizes \mathbf{A} e \mathbf{B} podem ser encontradas por

$$\mathbf{A} = \frac{f(X_e + n_1\epsilon, U_e) - f(X_e - n_1\epsilon, U_e)}{2\epsilon}$$

$$\mathbf{B} = \frac{f(X_e, U_e + n_2\epsilon) - f(X_e, U_e - n_2\epsilon)}{2\epsilon}$$

Em que n_1 e n_2 são vetores coluna de elementos unitários com o número de linhas iguais ao número de estados e de controles respectivamente.

3.5 INVERSÃO DINÂMICA PARA SISTEMAS NÃO LINEARES

As equações para os sistemas de guimbais são altamente não lineares, portanto é adequado utilizar um tipo de controle como a inversão dinâmica estendida a sistemas não lineares.

Considerando que o sistema pode ser escrito na forma de espaço de estados não linear:

$$\dot{x} = f(x) + g(x)u \quad (3.74)$$

$$y = h(x) \quad (3.75)$$

Com os estados $x(t) \in \mathbb{R}^n$, o controle $u(t) \in \mathbb{R}^m$ e a saída $y(t) \in \mathbb{R}^p$. É assumido que o sistema é quadrado, ou seja, u e y possuem as mesmas dimensões.

Para que o sistema siga a referência $r(t)$ é necessário definir o erro de rastreio:

$$e(t) = r(t) - y(t) \quad (3.76)$$

Derivando a saída com respeito aos estados obtém-se:

$$\dot{y} = \frac{\delta h}{\delta x} \dot{x} = \frac{\delta h}{\delta x} f(x) + \frac{\delta h}{\delta x} g(x)u = F(x) + G(x)u \quad (3.77)$$

Definindo então a entrada de controle:

$$u = G^{-1}(x)[-F(x) + \dot{r} + v] \quad (3.78)$$

Onde v é uma variável auxiliar a ser definida. Substituindo (3.78) em (3.77), chega-se na dinâmica do erro dada por:

$$\dot{e} = -v \quad (3.79)$$

Segundo Lewis, Stevens e Johson (2007), qualquer técnica linear, incluindo técnicas robustas como LQR, pode ser usada para selecionar $v(t)$ que estabilize esse sistema linear com p polos na origem. Uma forma conveniente é:

$$v = Ke \quad (3.80)$$

Em que K é uma matriz de ganhos definida positiva. A lei de controle fica, substituindo (3.80) em (3.78):

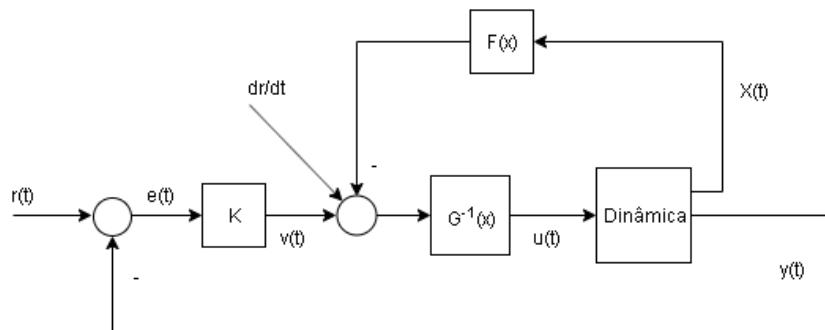
$$u = G^{-1}(x)[-F(x) + \dot{r} + Ke] \quad (3.81)$$

Como mostrado na figura 3.6, essa estrutura de controle requer realimentação completa de estados.

A dinâmica completa do sistema em malha fechada pode ser escrita como

$$\dot{x} = [I - gG^{-1} \frac{\delta h}{\delta x}]f(x) + gG^{-1}[\dot{r} + v] \quad (3.82)$$

Figura 3.6 – Diagrama de blocos para inversão dinâmica.



Fonte: Autor.

A lei de controle $u(t)$ foi escolhida para fazer o sistema de $v(t)$ para $y(t)$ ser simplesmente um sistema linear com p polos na origem. Isso é realizado através da linearização na realimentação interna de estados, a qual agora é não linear. Após, é feita uma malha externa para o rastreio, completando o projeto.

3.6 CONTROLE POR MAPEAMENTO EXPONENCIAL

Segundo Castro (2015), o controle por mapeamento exponencial (CME) unifica um método heurístico para abandonar a necessidade de um modelo e para definir seus parâmetros iniciais com um mapeamento exponencial unidimensional baseado na inferência da lógica nebulosa para lidar com uma classe de não-linearidades. Esse controle é baseado no método *bang-bang* e na solução de camada limite. É necessário determinar dois parâmetros iniciais heuristicamente. O primeiro é a faixa de erro, e_r , determinado aplicando-se uma entrada máxima de controle em uma planta estável e verificando a saída em regime permanente e o tempo para atingi-la. O segundo é a forma da função de entrada de controle, u_s , avaliando-se a forma da saída para a mesma situação. Ambos são determinados a partir da resposta de uma planta estável em malha aberta, quando são aplicadas as máximas ações de controle possíveis. Para plantas instáveis ou dinamicamente estáveis, u_s deve ser derivado a partir do comportamento da planta antes da instabilidade. A ideia é que o sistema atinja erro nulo seguindo o mesmo caminho que levou para atingir o seu limite, o que é seu caminho natural.

A implementação do CME é bem simples e começa determinando-se o erro de rastreio:

$$e = y - y_r \quad (3.83)$$

Em que y é a saída e y_r é a referência.

Após, é determinada uma linha de comutação s_{CME} :

$$s_{CME} = \frac{e}{e_r} + \dot{e} \quad (3.84)$$

Onde e_r é determinado como sendo o erro aceitável antes da aplicação da ação de controle máxima. Para sistemas de baixa ordem ou não oscilatórios, \dot{e} é aproximadamente zero. O s_{CME} é condicionado a:

- $s_{CME} = -1$, se $s_{CME} < -1$
- $s_{CME} = s_{CME}$, se $-1 \leq s_{CME} \leq 1$
- $s_{CME} = 1$, se $s_{CME} > 1$

Com isso uma função exponencial pode ser definida:

$$u_e = sign(s_{CME})((1 - |s_{CME}|)^{\frac{1}{2}u_s} - 1) \quad (3.85)$$

Em que $-10 \leq u_s \leq 10$ é o parâmetro que altera a exponencial de uma forma côncava para convexa.

A lei de controle será

$$u_{CME} = \frac{u_{max} - u_{min}}{2}(u_e - 1) + u_{max} \quad (3.86)$$

Em que u_{min} e u_{max} são as ações de controle mínima e máxima respectivamente.

4 RESULTADOS

Foram implementados programas em Matlab/Simulik e no Wolfram Mathematica para realizar as simulações e testar os modelos numericamente. Para os controles, ao realizar as integrações das equações de dinâmica, o vetor de estados definido foi:

$$\vec{X} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (4.1)$$

O vetor de controle é:

$$\vec{U} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (4.2)$$

4.1 IMPLEMENTAÇÃO DO MODELO DOS GUIMBAIS

As matrizes de inércia obtidas do modelo em CAD da figura 1.2, feito no SolidWorks, com relação a seus respectivos sistemas de referência são:

$$\mathbf{J}_1 = \begin{bmatrix} 0.00010991279 & 0 & 0.00000178791 \\ 0 & 0.00011489779 & 0 \\ 0.00000178791 & 0 & 0.00018335837 \end{bmatrix} kg \cdot m^2 \quad (4.3)$$

$$\mathbf{J}_2 = \begin{bmatrix} 0.0006598474 & 0.00004422168 & 0.00000188856 \\ 0.00004422168 & 0.00088063987 & 0.00000340106 \\ 0.00000188856 & 0.00000340106 & 0.0014616661 \end{bmatrix} kg \cdot m^2 \quad (4.4)$$

$$\mathbf{J}_3 = \begin{bmatrix} 0.00436924626 & 0 & 0 \\ 0 & 0.00193939665 & 0 \\ 0 & 0 & 0.00254130956 \end{bmatrix} kg \cdot m^2 \quad (4.5)$$

As coordenadas das posições dos centros de massa dos guimbais com respeito aos respectivos centros dos eixos de giro, escritas em seus próprios sistemas de referência

são:

$$\vec{r}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} m \quad (4.6)$$

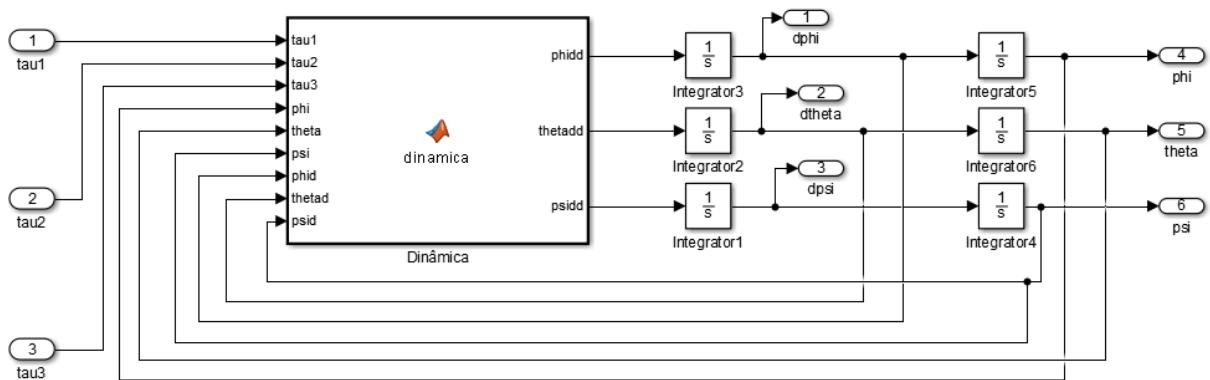
$$\vec{r}_2 = \begin{bmatrix} -21.45 \cdot 10^{-3} \\ 0 \\ 0.09 \cdot 10^{-3} \end{bmatrix} m \quad (4.7)$$

$$\vec{r}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} m \quad (4.8)$$

Para obter as equações diferenciais para os ângulos de Euler, foi necessário utilizar o software Wolfram Mathematica. Essa equações diferenciais podem ser obtidas isolando-se as acelerações angulares de ϕ , θ e ψ nas Equações (3.38), (3.39) e (3.40). Essas expressões são muito longas para serem mostradas, mas o programa está no Anexo A.

As equações diferenciais obtidas foram implementadas em Matlab/Simulink para poderem ser integradas, fornecendo as posições e velocidades angulares de cada guimbal. O modelo em Simulink está ilustrado na Figura 4.1.

Figura 4.1 – Dinâmica dos guimbais.



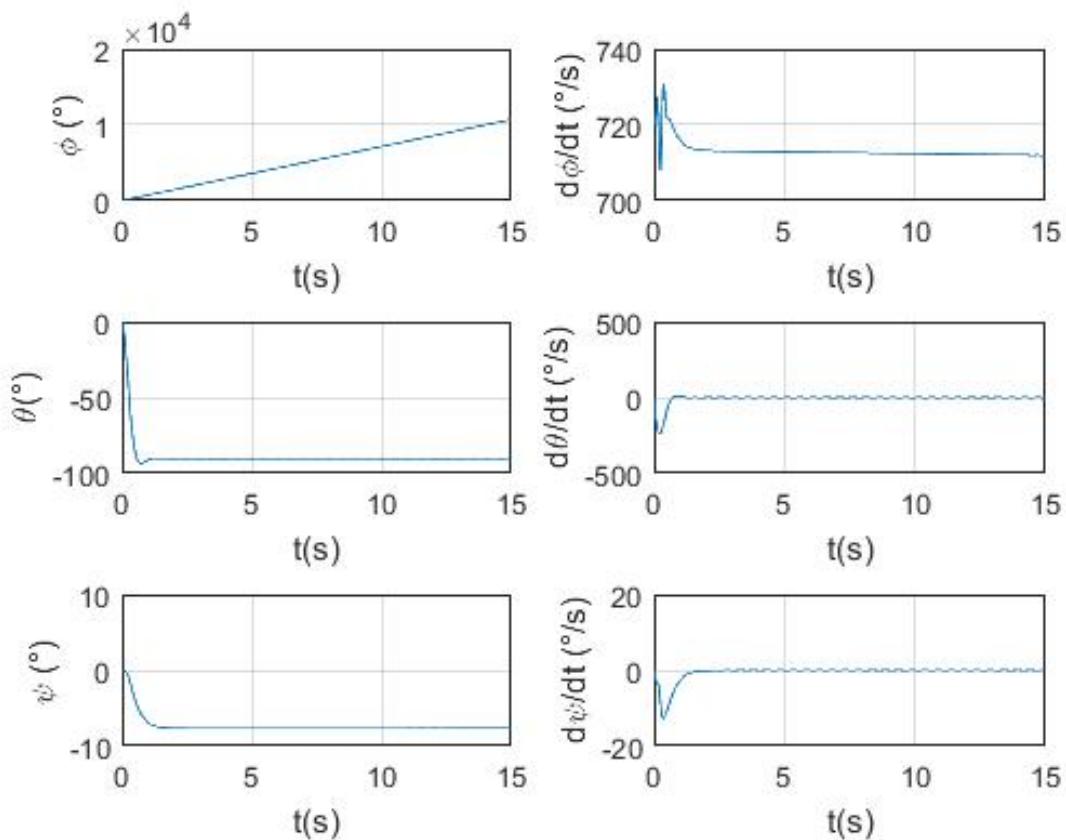
Fonte: Autor.

Entra-se com os torques e a dinâmica fornece as acelerações angulares dos guimbais. Faz-se uma primeira integração para obter-se as velocidades angulares e após outra para obter-se os ângulos de Euler. O sistema é realimentado com esses resultados.

Uma simulação do modelo em seu estado inicial, conforme a Figura 1.1, sujeita

apenas à força da gravidade está ilustrada na Figura 4.2. Foram adotados valores arbitrários para as constantes de atrito viscoso, sendo $b_1 = 0 \text{ Nms/rad}$, $b_2 = 0.01 \text{ Nms/rad}$ e $b_3 = 0.01 \text{ Nms/rad}$. O guimbal 1 é posto para girar sem atrito com velocidade inicial de 120 rpm.

Figura 4.2 – Simulação do modelo dos guimbais.



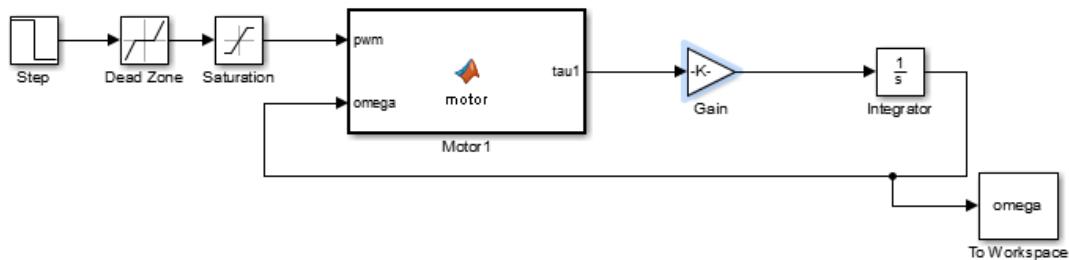
Fonte: Autor.

Analisando a Figura 4.2 e o sistema da Figura 1.1, os gráficos estão coerentes com o esperado. O guimbal 2 tende a tombar em 90 graus devido à ação da gravidade. No guimbal 3, pode-se observar um movimento, o qual provém da precessão. O eixo de giro do guimbal 3 é perpendicular tanto ao do guimbal 2 quanto do 1. O guimbal 1 gira sem atrito. A quantidade de movimento angular está alinhada ao seu eixo de giro. Há um torque aplicado ao sistema devido à gravidade. O efeito é mais pronunciado no guimbal 2. Esse torque age em um eixo perpendicular a vetor quantidade de movimento, o que provoca o movimento de precessão, no qual o sistema começa a girar em um terceiro eixo, perpendicular a ambos.

4.2 IMPLEMENTAÇÃO DO MODELO DOS SERVOMOTORES

Com os dados obtidos anteriormente foi elaborado um modelo em Simulink para os motores serem implementados junto ao sistema de guimbais. A ilustração do modelo está na figura 4.3.

Figura 4.3 – Modelo do motor no Simulink.



Inicia-se com um comando de PWM na forma de degrau. Inseriu-se um atraso no sinal, para representar o tempo que o comando leva para sair do computador até chegar ao circuito do motor. Foi adicionada uma zona morta, devido a falta de resposta do motor para tensões muito baixas. Foi observado que ocorriam para PWM entre -8 e 8. Ainda, uma saturação foi introduzida para PWM maior que 885 ou menor que -885, como especificado ROBOTIS (2018).

O bloco **Matlab Function** contém as equações (3.51) para converter o comando de PWM para tensão e (3.42) para gerar a corrente. Com essa grandeza é possível encontrar o torque produzido pelo motor, pela equação (3.43). Converte-se, então, o torque para o eixo de carga com a multiplicação pelo fator $N\eta$. A saída é o torque no eixo de carga do motor descontando o atrito no eixo de carga, de forma:

$$\tau_l = N\eta K_t i - b_0 N^2 \eta \omega \quad (4.9)$$

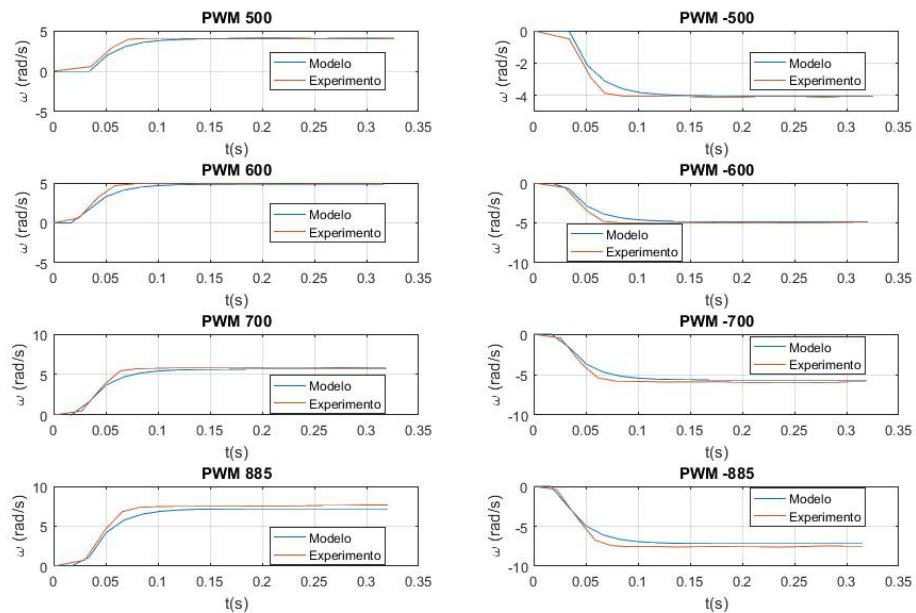
Na implementação do modelo do motor junto ao respectivo guimbal, o torque τ_l da Equação (4.9) é a variável de controle que alimenta cada entrada da Figura 4.1. Mas, para testar o motor num primeiro momento, é preciso um modelo simplificado. Assim sendo, nesta primeira implementação do modelo do motor, inseriu um modelo do momento de inércia do conjunto rotor+engrenagens+disco de carga, mais inércia de carga, situação similar àquela para a qual a função de transferência da Equação (3.48) é obtida. Assim, na figura 4.3, o ganho após o bloco do motor é simplesmente o inverso de um momento de inércia, sobre o qual é aplicado o torque de saída, obtendo-se uma aceleração angular,

para após fazer a integração e obter-se a velocidade angular do motor, que é necessária para a realimentação.

Quando o modelo de motor é integrado ao guimbal, o modelo de inércia do conjunto rotor+engrenagens+disco de carga deve ser somado ao momento de inércia do respectivo guimbal em torno do respectivo eixo de giro.

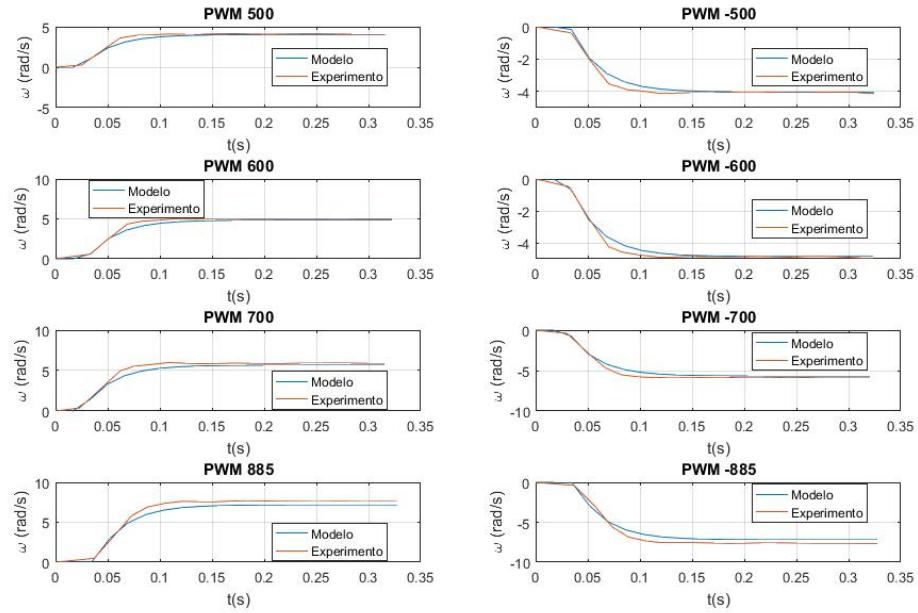
As simulações do modelo estão comparadas aos dados experimentais obtidos em Sartori e Fabbrin (2019), nas Figuras 4.4, 4.5 e 4.6. Foram comparados os dados experimentais com a simulação, tentando-se reproduzir, na simulação, a mesma condição de ensaio, que foi a aplicação de degraus de PWM com as amplitudes indicadas, inserindo-se diferentes cargas ao eixo de saída do motor.

Figura 4.4 – Comparação servomotor sem carga.



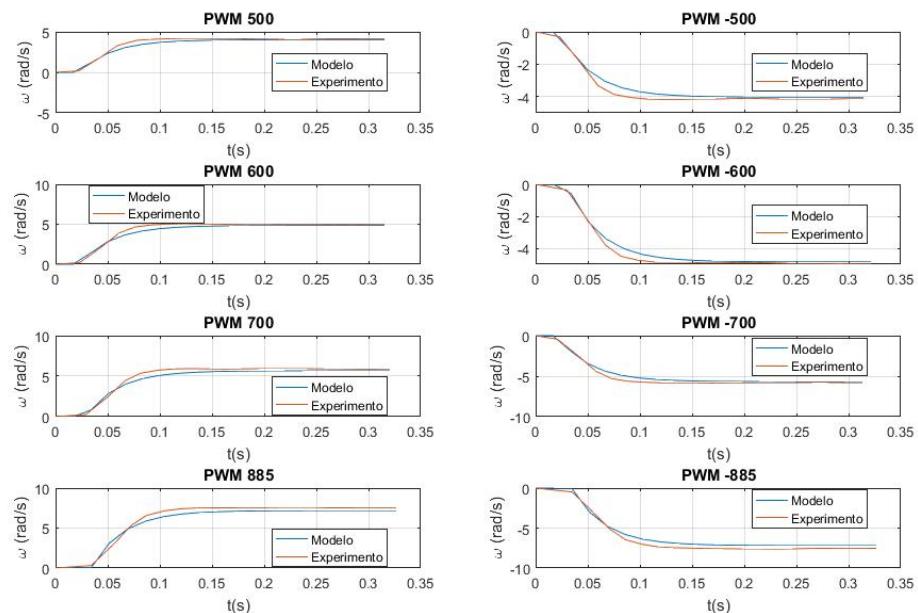
Fonte: Autor.

Figura 4.5 – Comparação servomotor com carga do guimbal 3.



Fonte: Autor.

Figura 4.6 – Comparação servomotor com carga do guimbal 3 mais barra.



Fonte: Autor.

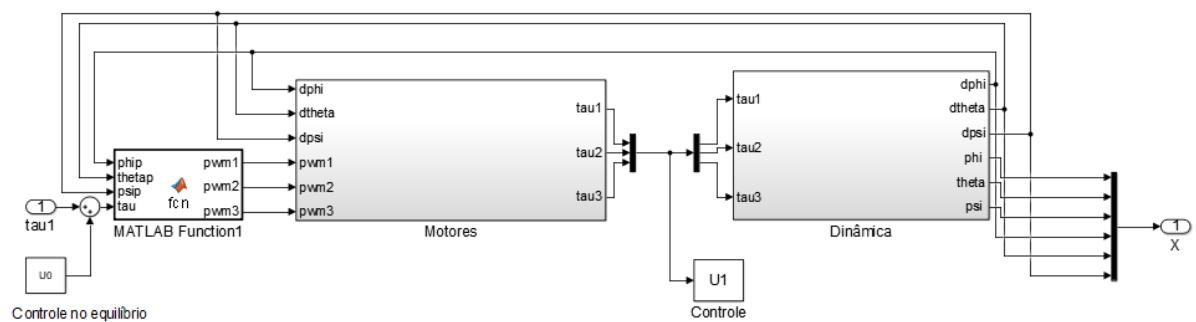
Pode-se ver que o modelo atinge o regime permanente praticamente ao mesmo

tempo que o experimental. Apenas para valores maiores de PWM há uma diferença entre os valores em regime permanente em todas as simulações, com ou sem carga. O regime transitório das respostas é um pouco diferente. Essas diferenças podem ser ocasionadas pelo modelo simplificado baseado em uma função de transferência de primeira ordem não levando em conta a indutância do motor e também devido a erros de medida.

4.3 UNIÃO DO MODELO DO SERVOMOTOR E DOS GUIMBAIS

Para unir os dois modelos, dinâmica dos motores e dinâmica dos guimbais, foi criado um programa em Simulink ilustrado na Figura 4.7.

Figura 4.7 – União dos modelos.



Fonte: Autor.

O bloco **Motores** possui o diagrama da Figura 4.3, sem fazer a integração, pois é necessária a saída ser os torques de cada motor. As entradas são os comandos de PWM e as velocidades angulares pela realimentação.

O bloco **Dinâmica** possui o sistema da Figura 4.1, com as equações diferenciais da dinâmica dos guimbais. Para incluir o momento de inércia do conjunto rotor+engrenagens+disco de carga ao modelo integrado motor+guimbais é preciso somar esse no eixo de giro de cada guimal, o que foi feito dentro da própria dinâmica. Entra-se com os torques e as saídas são as velocidades angulares e os ângulos.

O bloco **MATLAB Function 1** converte um comando de tensão para comando PWM. Entra-se com os torques desejados para cada motor e as velocidades angulares pela realimentação. O torque desejado virá das leis de controle e está no eixo da carga, podendo ser convertido em corrente pelas equações (3.43) e (4.9) de forma:

$$i_{desejado} = \frac{\tau}{N\eta K_t}$$

Essa corrente precisa ser convertida em uma tensão e essa convertida em comando de PWM. Para isso, pode-se usar a Equação (3.41). Para converter a tensão em comando de PWM, pode-se utilizar a Equação (3.51), o que resulta em

$$u_{PWM} = \frac{Ri_{desejado} + NK_t\omega_l}{K_{PWM}}$$

Em que ω_l é a velocidade angular no eixo de carga e K_{PWM} é o coeficiente angular da Equação (3.41).

O bloco **Controle no equilíbrio** contém o vetor controle no equilíbrio, o qual é obtido igualando-se as equações da dinâmica a zero e resolvendo o sistema de equações para os controles. Isso é feito devido a que os guimbais sofrem a ação da gravidade e como em sua configuração inicial estão dispostos segundo a Figura 1.1 não tombem. Esse vetor de controle no equilíbrio calculado é:

$$\vec{U}_0 = \begin{bmatrix} 0 \\ 0.051 \\ 0 \end{bmatrix} Nm \quad (4.10)$$

O programa para o cálculo do equilíbrio está no Anexo B.

A soma do controle de equilíbrio é necessária quando se trabalha com o controle CME e o LQR, pois os mesmos não preveem um controle aditivo somado ao controle realimentado. Por outro lado, como o controle de inversão dinâmica utiliza uma "cópia" do modelo do sistema, e este modelo, deduzido anteriormente, contém o torque gravitacional, este torque já estará presente no controle, não sendo necessário somar o controle de equilíbrio, que é basicamente o torque no guimbal 2 necessário para compensar o torque de gravidade na condição de equilíbrio.

A seguir são feitas as implementações das técnicas de controle e suas simulações. Foram feitas simulações sem o modelo dos motores, apenas limitando o torque máximo segundo os dados do fabricante, e com o modelo. Vale destacar que o torque máximo previsto no manual do fabricante é diferente daquele que se obteve a partir da modelagem do motor, inserindo-se o limite de PWM e os demais parâmetros dinâmicos. Assim sendo, o modelo que se obtém limitando somente o torque vai ser "mais otimista" que o real, pois o modelo com base em limitação de PWM tem menor torque máximo.

4.4 IMPLEMENTAÇÃO DO CONTROLE POR INVERSÃO DINÂMICA

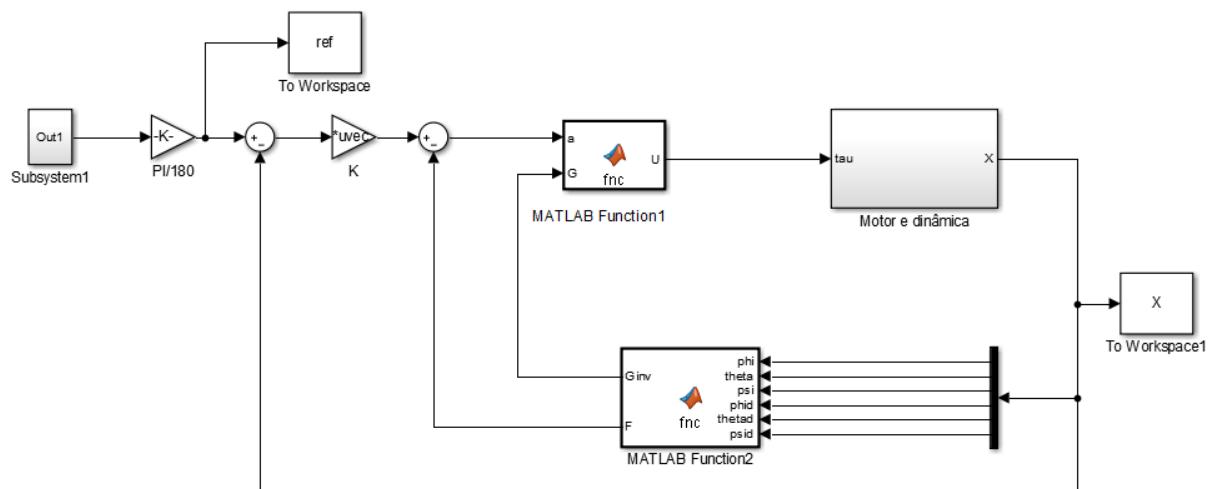
O controle por inversão dinâmica foi implementado a partir do diagrama de blocos da Figura 3.6. Para encontrar as matrizes $G(x)$ e $F(x)$ da lei de controle (3.81), utilizou-se a Equação (3.77). Para encontrar a matriz $G(x)$, derivou-se a Equação (3.77) com respeito a cada controle, assim eliminando a parcela de $F(x)$. Para encontrar $F(x)$, simplesmente foram zerados os controles, eliminando a parcela de $G(x)$. O procedimento foi feito no programa do Anexo A, implementado no Wolfram Mathematica.

Foi feito primeiramente uma simulação sem o modelo dos motores, apenas limitando o torque máximo que poderia ser aplicado segundo os dados do fabricante, Tabela 3.1. Em seguida, adicionou-se o modelo dos motores em um programa em Matlab e Simulink. A matriz de ganhos utilizada em ambos os casos foi determinada empiricamente, ou seja, testando matrizes até a resposta ficar adequada, sendo:

$$\mathbf{K} = \begin{bmatrix} 1000 & 0 & 0 & 100 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 100 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 100 \end{bmatrix}$$

O modelo feito em Simulink está ilustrado na Figura 4.8.

Figura 4.8 – Modelo em Simulink da Inversão Dinâmica.



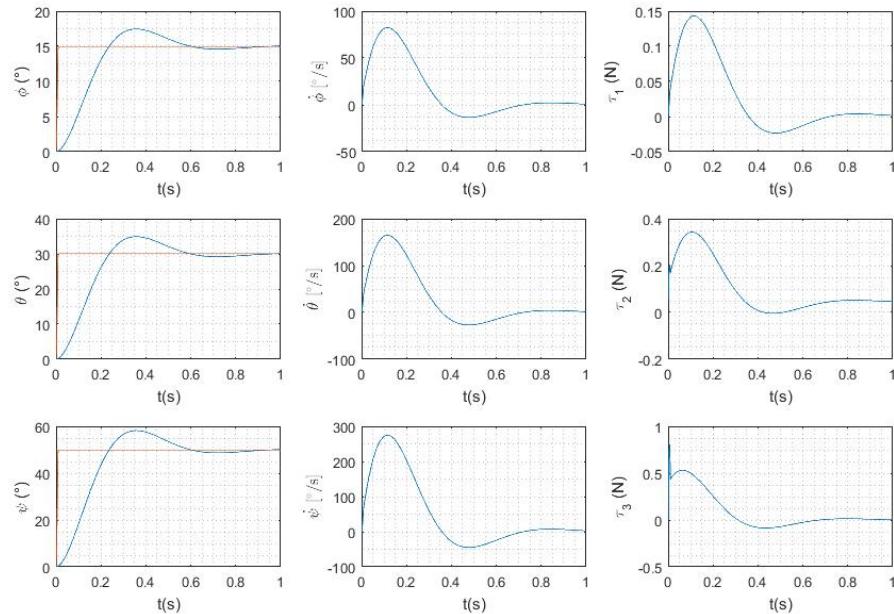
Fonte: Autor.

O bloco **Subsystem1** recebe as referências para os ângulos em graus que são logo convertidos em radianos. O bloco **K** possui a matriz de ganhos. **Motor e Dinâmica** possui o diagrama da Figura 4.7, enquanto o bloco **MATLAB Function2** possui as equações de $F(x)$ e $G(x)$ deduzidas no programa do Apêndice A. O bloco **MATLAB Function1** apenas

realiza a soma para chegar-se na lei de controle da Equação (3.81).

As respostas das simulações sem e com o modelo dos motores para a inversão dinâmica estão ilustradas nas Figuras 4.9 e 4.10. Os ângulos de referência para ϕ , θ e ψ foram respectivamente 15° , 30° e 50° , enquanto a referência para as velocidades angulares é nula.

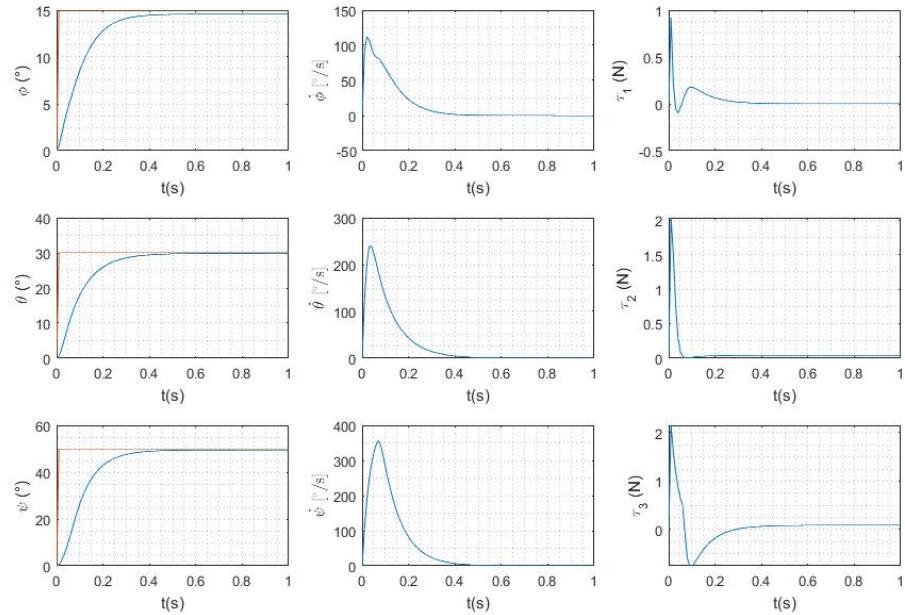
Figura 4.9 – Simulação da Inversão Dinâmica sem o modelo dos motores.



Fonte: Autor.

Pode-se ver na Figura 4.9 que o sistema atinge a referência rapidamente, mas possui sobressinal. O torque do guimbal 2 estabiliza em um valor pequeno, mas diferente de zero, para compensar a ação da gravidade.

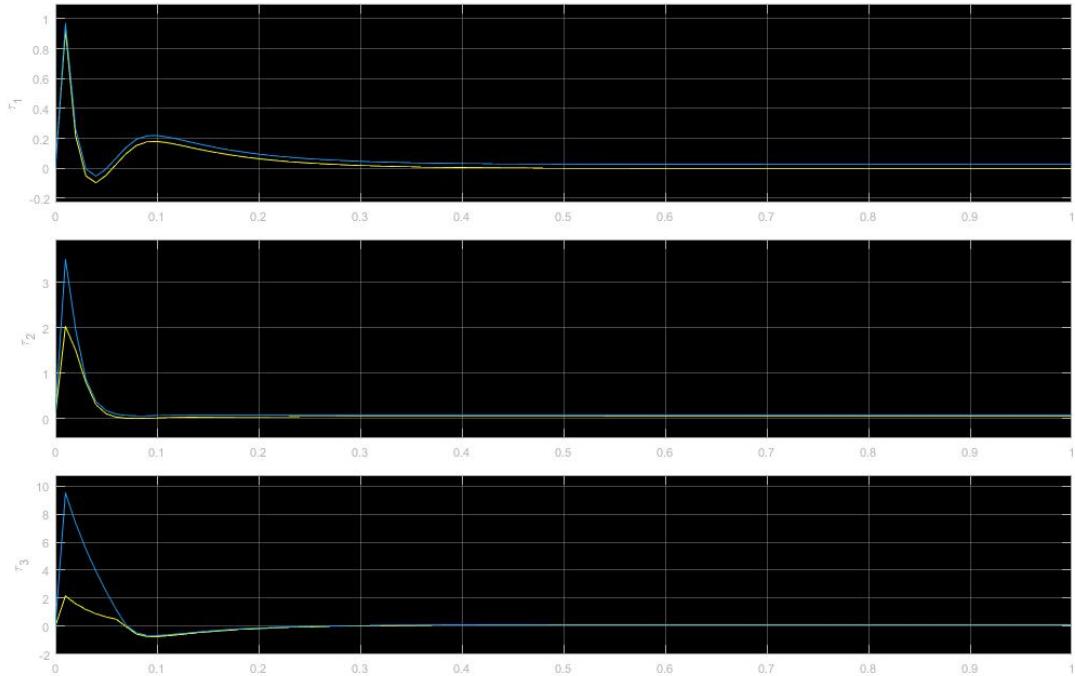
Figura 4.10 – Simulação da Inversão Dinâmica com o modelo dos motores.



Fonte: Autor.

Como pode ser observado na Figura 4.10, não há sobressinal nem oscilação relevante nos controles incluindo o modelo dos servomotores, pois esses adicionam um amortecimento a mais. A Figura 4.11 mostra a comparação entre o controle calculado pela inversão dinâmica (em azul) e após passar pelo motor (em amarelo).

Figura 4.11 – Controles inversão dinâmica.



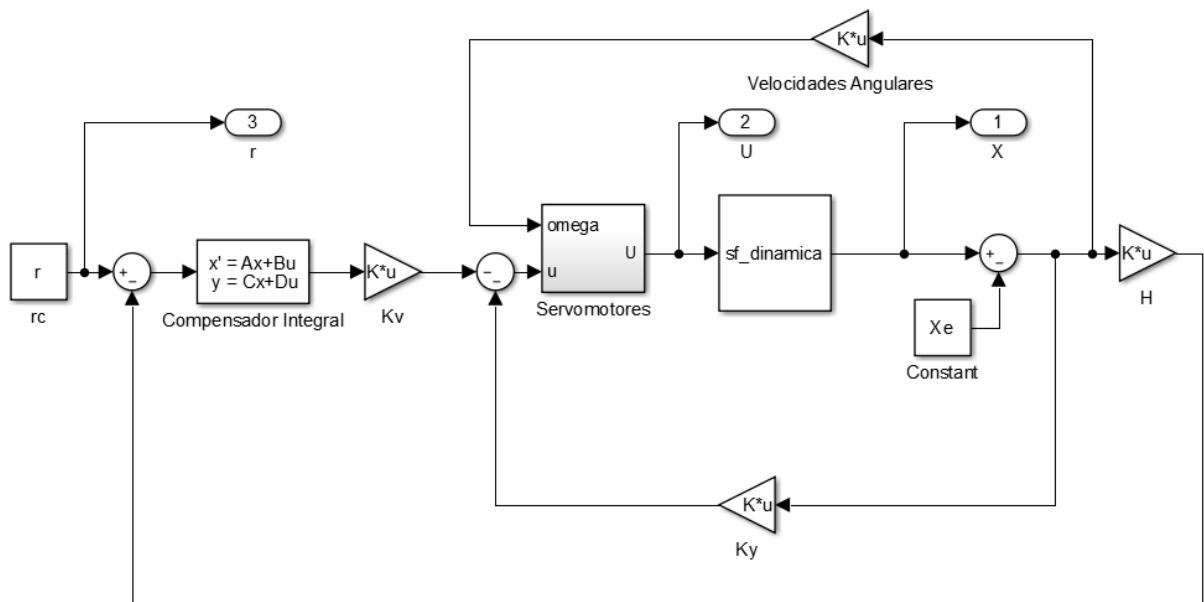
Fonte: Autor.

A Figura 4.11 mostra que o controle calculado pela inversão dinâmica (em azul) é bastante atenuado no regime transitório para os torques 2 e 3, devido a saturação dos motores (em amarelo). Mesmo assim, a resposta fica boa, sem sobressinal e oscilação, evidenciando robustez no projeto do controle.

4.5 IMPLEMENTAÇÃO DO CONTROLE LQR

Para implementar-se o controle por LQR seguiu-se o diagrama de blocos da Figura 3.5. Esse diagrama foi feito no Simulink segundo a Figura 4.12.

Figura 4.12 – Modelo em Simulink do controle LQR.



Fonte: Autor.

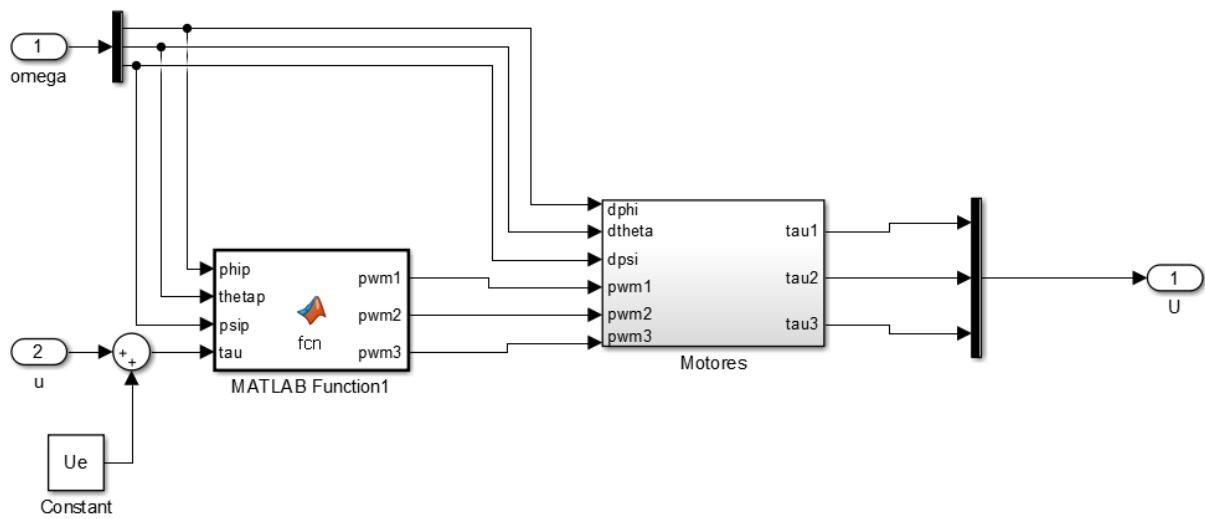
O bloco **r** gera uma referência para os ângulos. A Figura 4.13 mostra o conteúdo do bloco **Servomotores**, em que a **Matlab Function1** e **Motores** são os mesmos da Figura 4.7. Entra-se com o torque calculado pela lei de controle, o qual é convertido em comando de PWM que vai para o modelo dos motores e gera o torque real. São realimentadas as velocidades angulares.

O bloco de espaço de estados **Compensador** possui as matrizes das Equações (3.66) e (3.67) para um compensador integral. F é uma matriz de zeros 3×3 , assim como J , G e D do compensador são matrizes identidade 3×3 .

A matriz H é feita de forma a rastrear apenas os três primeiros estados, sendo:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Figura 4.13 – Servomotores.



Fonte: Autor.

O projeto das matrizes de ganho K_y e K_v , devido a esta ser uma técnica linear, foi feito com as matrizes linearizadas do sistema em torno do ponto de equilíbrio (estados todos nulos e controle no equilíbrio). O código para a linearização das equações está no Anexo C e segue o processo descrito na seção 3.4. As equações linearizadas estão ilustradas na Figura 4.14:

Figura 4.14 – Matrizes A e B linearizadas.

```

A =
0 0 0 1.0000 0 0
0 0 0 0 1.0000 0
0 0 0 0 0 1.0000
0.0000 0.0000 0 -909.8185 -0.0013 0.3886
-0.0000 0.2149 0 -0.0013 -100.4485 0.0816
0.0000 -0.0002 0 0.3886 0.0816 -23.8875

B =
1.0e+03 *
0 0 0
0 0 0
0 0 0
9.0982 0.0000 -0.0039
0.0000 1.0045 -0.0008
-0.0039 -0.0008 0.2389

```

Fonte: Autor.

A matriz C é uma matriz identidade 6×6 . As matrizes de ganho K_y e K_v foram calculadas segundo o procedimento descrito na seção 3.3.2. Como foi escolhido um compensador integral e a matriz C é uma identidade, é possível utilizar a função *lqr* do Matlab, a qual resolve a equação de Riccati, Equação (3.63). Entra-se com as matrizes aumentadas do sistema, Equação (3.70), e as matrizes de ponderação Q e R expressas na Figura 4.15, que foram determinadas empiricamente. O programa para o cálculo dos ganhos está ilustrado no Anexo D.

Figura 4.15 – Matrizes de ponderação para LQR.

```

Q =

```

1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	0	100	0
0	0	0	0	0	0	0	0	100


```

R =

```

0.1000	0	0
0	0.1000	0
0	0	0.1000

Fonte: Autor.

A matriz Q é feita dessa forma, pois as matrizes aumentadas do sistema englobam tanto a dinâmica do compensador quanto a dos guimbais. As seis primeiras linhas correspondem aos parâmetros para os estados da dinâmica dos guimbais, enquanto as três últimas para o compensador. O retorno do programa será uma matriz de ganhos aumentada em que a se pode obter K_y separando todas as linhas e colunas de 1 a 6 e K_v com todas as linhas e colunas restantes dessa matriz, chegando-se:

Figura 4.16 – Matrizes de ganho para LQR.

$K_y =$

$$\begin{array}{cccccc} 4.0745 & -0.0000 & 0.0004 & 0.0044 & -0.0000 & 0.0001 \\ -0.0000 & 4.3158 & 0.0007 & -0.0000 & 0.0364 & 0.0001 \\ 0.0004 & 0.0007 & 4.9319 & 0.0001 & 0.0001 & 0.1265 \end{array}$$

$K_v =$

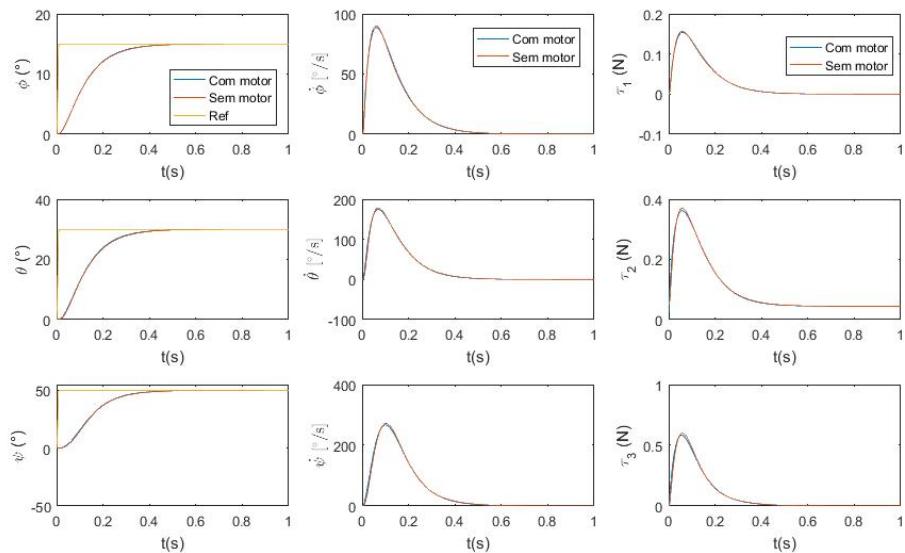
$$\begin{array}{ccc} -31.6228 & -0.0000 & -0.0000 \\ 0.0000 & -31.6228 & -0.0000 \\ 0.0000 & 0.0000 & -31.6228 \end{array}$$

Fonte: Autor.

O bloco **sf_dinamica** é uma *S-Function* do Simulink. Esse recurso recebe os controles e devolve os estados após fazer a integração das equações da dinâmica. A configuração desse recurso foi impondo 6 estados contínuos (os ângulos e as velocidades angulares), 3 entradas (torques) e 6 saídas (estados). A dinâmica do sistema foi adquirida das equações diferenciais obtidas no Anexo A. O código desse bloco está no Anexo F.

Primeiramente fez-se uma simulação sem o modelo dos motores, apenas limitando o torque máximo e após a simulação com o modelo. As respostas estão ilustradas na Figura 4.17. Os ângulos de referência para ϕ , θ e ψ foram respectivamente 15° , 30° e 50° .

Figura 4.17 – Simulação do LQR

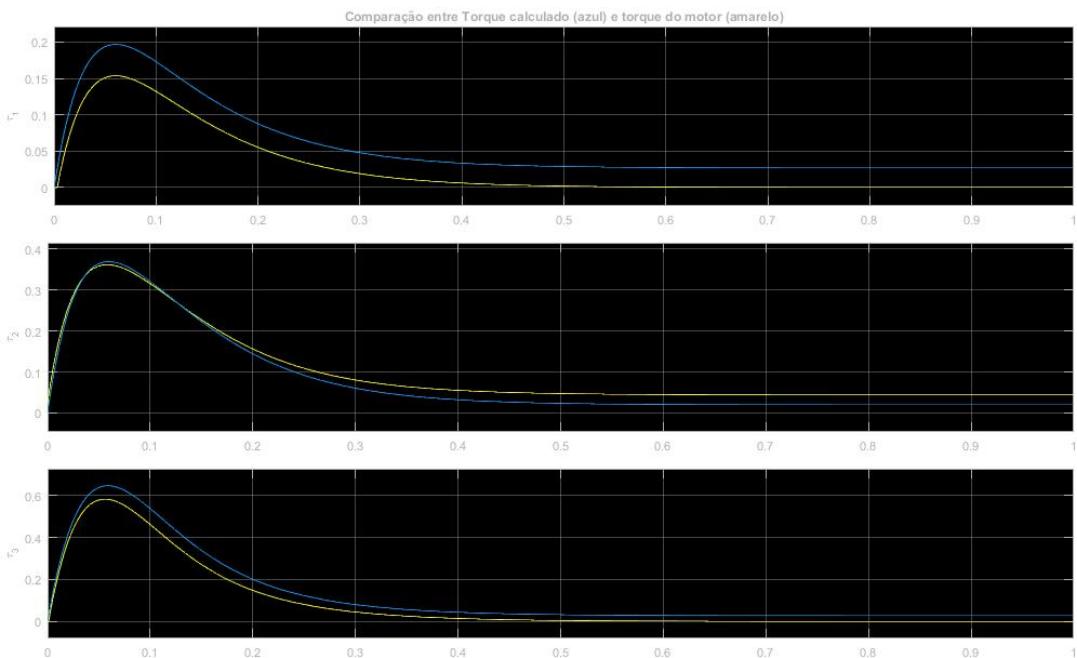


Fonte: Autor.

Pode-se ver que na simulação da Figura 4.17, que o sistema atinge rapidamente as referências, sem oscilações nem sobressinal. Percebe-se que as simulações ficaram muito parecidas. Há uma pequena diferença, mas quase imperceptível. É possível ver que o torque para o guimbal 2 não se estabiliza em 0, isso devido a ter que compensar a ação da gravidade.

A Figura 4.18 mostra em azul o controle ideal calculado pelo LQR e em amarelo o controle após passar pelos motores.

Figura 4.18 – Controles LQR.



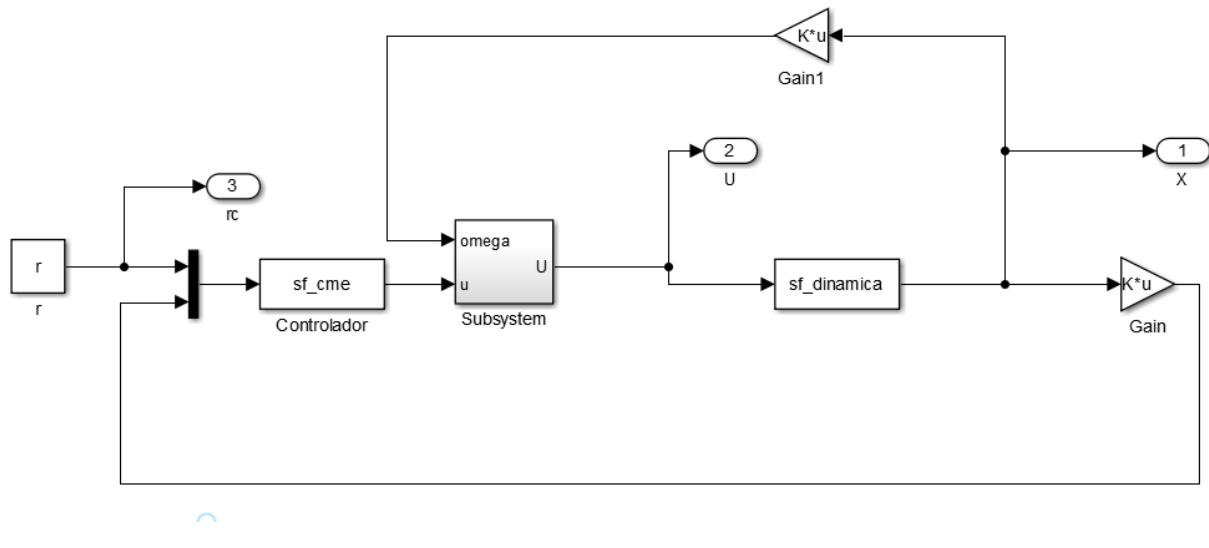
Fonte: Autor.

Como visto na Figura 4.18, o controle calculado pelo LQR (em azul) é levemente atenuado e defasado quando passa pelos motores (em amarelo), entretanto a interferência na resposta é pequena.

4.6 IMPLEMENTAÇÃO DO CONTROLE PELO CME

O CME foi implementado seguindo o procedimento descrito na seção 3.6. Primeiramente foi feito um programa sem o modelo do motor, apenas limitando o torque máximo. O programa para o cálculo da ação de controle está no Anexo E. Em seguida foram inseridos os servomotores, segundo o diagrama da Figura 4.19.

Figura 4.19 – Modelo em Simulink do CME.



Fonte: Autor.

O bloco **r** recebe as referências para os ângulos, enquanto **Ks** possui um vetor de ganho definido empiricamente. A matriz utilizada foi:

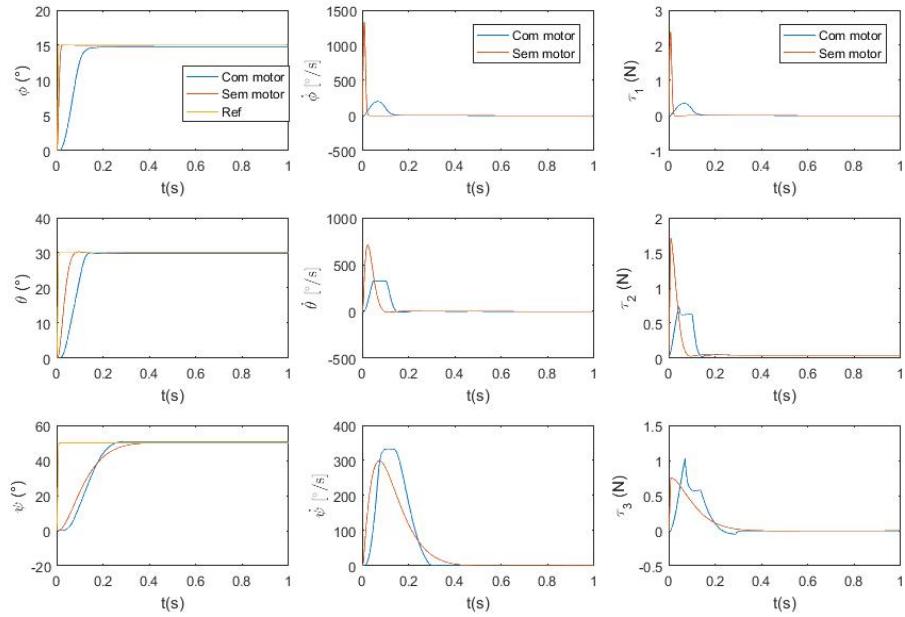
$$\vec{K}_s = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & 1 \end{bmatrix}^T \quad (4.11)$$

Essa matriz fornece os parâmetros para o erro aceitável e_r (posições 1, 3 e 5 no vetor \vec{K}_s) e os parâmetros para u_s que alteram a forma da função exponencial (posições 2, 4 e 6).

O bloco **Subsystem** possui o diagrama da Figura 4.13. A *S-Function* **sf_dinamica** é a mesma do LQR. Incluiu-se mais uma *S-Function* denominada **sf_cme** para fazer o controle pelo CME. Nesse recurso, foram zerados os estados, pois não há dinâmica, colocadas 6 entradas (os 3 primeiros estados e suas 3 referências) e 3 saídas (torques). O código dessa está no Anexo G. Esse recurso utiliza também a função para montar o controle que está no Anexo E.

A Figura 4.20 mostra os resultados das simulações. Os ângulos de referência para ϕ , θ e ψ foram respectivamente 15° , 30° e 50° .

Figura 4.20 – Simulação do CME.



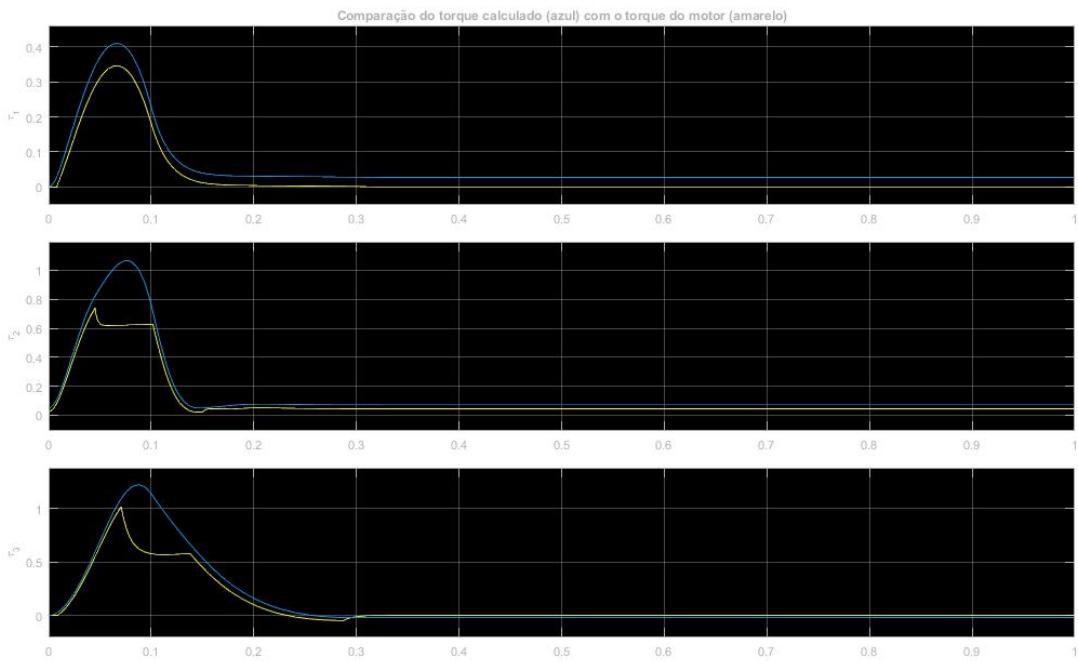
Fonte: Autor.

Pela Figura 4.20 pode-se ver que o sistema atinge as referências para os ângulos rapidamente, não havendo sobressinal nem oscilação nas respostas. Novamente o controle do guimbal 2 se estabiliza em um valor pequeno diferente de zero por causa da gravidade.

É interessante notar a saturação imposta pelo motor, bastante pronunciada nas respostas de ϕ e θ . Como é possível ver, a resposta com o modelo dos motores demora um pouco mais para chegar na referência, justamente devido a essa interferência do motor. Não ocorreu o mesmo caso com ψ . Isso provavelmente deve-se a que mesmo sem o motor, a resposta não chegou na velocidade de saturação, como nos outros casos, na qual ultrapassou bastante. Os torques também sofrem a saturação.

A Figura 4.21 mostra a comparação entre as respostas dos torques calculados e dos torques após a passagem pelos motores.

Figura 4.21 – Controles CME.



Fonte: Autor.

A Figura 4.21 mostra a saturação dos controles. Em azul é a lei de controle calculada e em amarelo é quando passa pelo motor. Pode-se ver que há uma diferença considerável entre as respostas, principalmente nos torques 2 e 3 no regime transitório. Há uma pequena diferença no torque de acomodação.

5 CONCLUSÃO

Nesse trabalho foram estabelecidos modelos matemáticos tanto para um sistema de guimbais com 3 graus de liberdade, quanto para um servomotor Dynamixel XM430-W210-T e para o estudo e aplicação de formas de controle linear, como o LQR, e não linear, como o CME e a Inversão Dinâmica. Pode-se ver nas simulações que o modelo dos motores pode tornar a resposta um pouco diferente dos valores teóricos, devido a não estar disponível a velocidade e torques calculados teoricamente. Isso aproxima bastante da realidade as simulações. Todos os controles se comportaram bem quando colocados os motores nas simulações. O CME mostrou-se um controle bem agressivo nas simulações, sofrendo a saturação quando em conjunto com os motores. A inversão dinâmica até mesmo melhorou sua resposta ao se inserir os motores, ficando sem o sobressinal. O LQR foi o que menos sofreu influência quando impostos os servomotores.

Não houve um estudo aprofundado das matrizes de ganhos para os controles, devido a isso, muitas melhorias nas respostas ainda podem ser feitas. Além disso, os parâmetros de atrito viscoso dos eixos devido aos rolamentos foram arbitrários, pois não foi possível encontrá-los até o presente trabalho, mas tem pouca influência, pois o dos motores já é bem grande.

Esse trabalho serve como um guia para futuros trabalhos com os sistemas de guimbais da UFSM, assim como qualquer outro. Ainda, é possível ver uma aplicação dos conteúdos vistos durante a graduação em Engenharia Aeroespacial das mais diversas áreas.

REFERÊNCIAS BIBLIOGRÁFICAS

- CANNON, R. H. J. **Dynamics of Physical Systems.** [S.I.]: Dover Publications, 2012.
- CARTER, L.; ALI, K. Hardware in the loop simulator for imu calibration. In: THE 12TH WORLD MULTI-CONFERENCE ON SYSTEMICS, CYBERNETICS AND INFORMATICS, 2008, Orlando. **Anais eletrônicos...** Orlando: WMSCI, 2008.
- CASTRO, H. **Exponential Mapping Function for Nonlinear Control.** 2015. Tese (Doutorado) — Aeronautics Technological Institute, São José dos Campos, Brasil, 2015.
- LEWIS, F.; STEVENS, B.; JOHSON, E. N. **Aircraft Control and Simulation.** Boston: Birkhäuser, 2007.
- MÁXIMO, M. R. O. A.; RIBEIRO, C. H. C.; AFONSO, R. J. M. Modeling of a position servo used in robotics applications. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, XIII., 2017, Porto Alegre-RS. **Anais eletrônicos...** São Paulo, 2017.
- NASA. **Guimbaled thrust.** 2018. Acessado em 25 NOV 2018. Disponível em: <<https://spaceflightsystems.grc.nasa.gov>>.
- OGATA, K. **Engenharia de Controle Moderno.** São Paulo: Pearson Prentice Hall, 2011.
- ROBOTIS. **Dynamixel XM430-W210T e-manual.** 2018. Acessado em 06 abr 2018. Disponível em: <http://support.robotis.com/en/product/actuator/dynamixel_x/xm_series/xm430-w210.htm>.
- SARTORI, T. K. S.; FABBRIN, J. A. **PROCESSO DE MONTAGEM E MODELAGEM DE SERVOMOTORES DE UM SISTEMA DE GUIMBAIS COM 3 GRAUS DE LIBERDADE.** 2019. Monografia (Trabalho de CPIO IV) — Curso de Graduação em Engenharia Aeroespacial, Universidade Federal de Santa Maria, Santa Maria, 2019.
- SMITH, B. J. et al. Sliding mode control in a two axis gimbal system. **Proc. IEEE Aerospace Conf.**, v. 5, p. 457–470, 1999.
- TEWARI, A. **Atmospheric and space flight dinamycs - Modelling and simulation with MATLAB and Simulink.** Boston: Birkhäuser, 2007.

ANEXO A – CÓDIGO PARA DEDUÇÃO DAS EQUAÇÕES DE DINÂMICA DO SISTEMA DE GUIMBAIS E DAS MATRIZES PARA A INVERSÃO DINÂMICA.

```

(*Matrizes elementares*)
C21 = {{1, 0, 0}, {0, Cos[\[Phi][t]], Sin[\[Phi][t]]}, {0, -Sin[\[Phi][t]], Cos[\[Phi][t]]}}; (*do 2 para o 1*)
C32 = {{Cos[\[Theta][t]], 0, -Sin[\[Theta][t]]}, {0, 1, 0}, {Sin[\[Theta][t]], 0, Cos[\[Theta][t]]}};
CI3 = {{Cos[\[Psi][t]], Sin[\[Psi][t]], 0}, {-Sin[\[Psi][t]], Cos[\[Psi][t]], 0}, {0, 0, 1}};
CI2 = C32.CI3;

CI1 = C21.C32.CI3;

(*Velocidades angulares*)
\[Omega]121 = {{D[\[Phi][t], t]}, {0}, {0}}; (*1 com respeito a 2 escrita em 1*)
\[Omega]232 = {{0}, {D[\[Theta][t], t]}, {0}};
\[Omega]3I3 = {{0}, {0}, {D[\[Psi][t], t]}};

(*Velocidades angulares com respeito ao inercial escritas nos \
respectivos sistemas*)
\[Omega]2I2 = \[Omega]232 + C32.\[Omega]3I3;
\[Omega]1I1 = \[Omega]121 + C21.\[Omega]232 + C21.C32.\[Omega]3I3;

(*Matrizes de inercia de cada guimbal em seu proprio sistema de eixos \
de referencia (guimbais simetricos*)*
J1 = {{Jxx1, Jxy1, Jxz1}, {Jxy1, Jyy1, Jyz1}, {Jxz1, Jyz1, Jzz1}};
J2 = {{Jxx2, Jxy2, Jxz2}, {Jxy2, Jyy2, Jyz2}, {Jxz2, Jyz2, Jzz2}};
J3 = {{Jxx3, Jxy3, Jxz3}, {Jxy3, Jyy3, Jyz3}, {Jxz3, Jyz3, Jzz3}};

(*Operador da matriz anti simetrica do produto vetorial*)
S[x_] := {{0, -x[[3, 1]], x[[2, 1]]}, {x[[3, 1]], 0, -x[[1, 1]]}, {-x[[2, 1]], x[[1, 1]], 0}};

(*Derivadas dos momentos angulares com respeito ao inercial escritas \
nos proprios sistemas - regra para derivadas em referenciais não \
iniciais*)
H1dI = J1.D[\[Omega]1I1, t] + S[\[Omega]1I1].J1.\[Omega]1I1;
H2dI = J2.D[\[Omega]2I2, t] + S[\[Omega]2I2].J2.\[Omega]2I2 +

```

```

Transpose[C21].H1dI;
H3dI = J3.D[\[Omega]3I3, t] + S[\[Omega]3I3].J3.\[Omega]3I3 +
Transpose[C32].H2dI;

(* Momentos gerados pela gravidade, devido ao desalinhamento entre o \
CM de cada guimbal e o respectivo eixo de rotação *)
r1 = {{r1x}, {r1y}, {r1z}}; (* CM do guimbal 1 em relação ao meio de \
seu eixo de rotação escrito em seu próprio sistema de referência *)
r2 = {{r2x}, {r2y}, {r2z}}; (* CM do guimbal 2 em relação ao meio de \
seu eixo de rotação escrito em seu próprio sistema de referência *)
r3 = {{r3x}, {r3y}, {r3z}}; (* CM do guimbal 3 em relação ao meio de \
seu eixo de rotação escrito em seu próprio sistema de referência *)
gv = {{0}, {0}, {-g}}; (* Vetor gravidade escrito no sistema de \
referência inercial. Convencionou-se que, na modelagem do guimbal, o \
eixo z é positivo para cima *)
(* Momento gravitacional sobre o guimbal 1 escrito em seu próprio \
sistema de referência *)
M1g = m1*S[r1].CI1.gv;
(* Momento gravitacional sobre o guimbal 2 escrito em seu próprio \
sistema de referência *)
M2g = m2*S[r2].CI2.gv;
(* Momento gravitacional sobre o guimbal 3 escrito em seu próprio \
sistema de referência *)
M3g = m3*S[r3].CI3.gv;
(*Momentos atuantes no guimbal 1 calculados em relacao ao seu centro \
de massa e escritos em seu proprio sistema de referencia.tau1 eh \
dado,My1 e Mz1 sao incognitas*)
M1 = {{tau1 - b1*D[\[Phi][t], t]}, {My1}, {Mz1}} + M1g;
(*Momentos atuantes no guimbal 2 calculados em relacao ao seu centro \
de massa e escritos em seu proprio sistema de referencia.tau2 eh \
dado,Mx2 e Mz2 sao incognitas*)
M2 = {{Mx2}, {tau2 - b2*D[\[Theta][t], t]}, {Mz2}} + M2g;
(*Momentos atuantes no guimbal 3 calculados em relacao ao seu centro \
de massa e escritos em seu proprio sistema de referencia.tau3 eh \
dado,Mx3 e My3 sao incognitas*)
M3 = {{Mx3}, {My3}, {tau3 - b3*D[\[Psi][t], t]}} + M3g;

(*Segunda lei de Newton para rotação*)
res = Solve[{M1 == H1dI, M2 == H2dI,

```

```

M3 == H3dI}, {(\[Phi]\^{\prime }\^{\prime })[
  t], (\[Theta]\^{\prime }\^{\prime })[t], (\[Psi]\^{\prime }\^{\prime })[t],
  My1, Mz1, Mx2, Mz2, Mx3, My3}];
sol = {(\[Phi]\^{\prime }\^{\prime })[t], (\[Theta]\^{\prime }\^{\prime })[
  t], (\[Psi]\^{\prime }\^{\prime })[t], My1, Mz1, Mx2, Mz2, Mx3,
  My3} /. res;
ddphi = sol[[1, 1]];
ddtheta = sol[[1, 2]];
ddpsi = sol[[1, 3]];

(*Inversão dinâmica*)
g11 = \!\
  \*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_1\)\)ddphi\)\ /. {Derivative[
    1][\[Phi]][t] -> dphi, Derivative[1][\[Theta]][t] -> dtheta,
    Derivative[1][\[Psi]][t] -> dpsi, \[Phi][t] -> phi, \[Theta][t] ->
    theta, \[Psi][t] -> psi}];

g12 = \!\
  \*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_2\)\)ddphi\)\ /. {Derivative[
    1][\[Phi]][t] -> dphi, Derivative[1][\[Theta]][t] -> dtheta,
    Derivative[1][\[Psi]][t] -> dpsi, \[Phi][t] -> phi, \[Theta][t] ->
    theta, \[Psi][t] -> psi}];

g13 = \!\
  \*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_3\)\)ddphi\)\ /. {Derivative[
    1][\[Phi]][t] -> dphi, Derivative[1][\[Theta]][t] -> dtheta,
    Derivative[1][\[Psi]][t] -> dpsi, \[Phi][t] -> phi, \[Theta][t] ->
    theta, \[Psi][t] -> psi}];

f1 = ddphi /. {\tau_1 -> 0, \tau_2 -> 0,
  \tau_3 -> 0} /. {Derivative[1][\[Phi]][t] -> dphi,
  Derivative[1][\[Theta]][t] -> dtheta,
  Derivative[1][\[Psi]][t] -> dpsi, \[Phi][t] -> phi, \[Theta][t] ->
  theta, \[Psi][t] -> psi};

g21 = \!\
  \*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_1\)\)ddtheta\)\ /. {Derivative[
    1][\[Phi]][t] -> dphi, Derivative[1][\[Theta]][t] -> dtheta,
    Derivative[1][\[Psi]][t] -> dpsi, \[Phi][t] -> phi, \[Theta][t] ->
    theta, \[Psi][t] -> psi}];

g22 = \!\
  \*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_2\)\)ddtheta\)\ /. {Derivative[
    1][\[Phi]][t] -> dphi, Derivative[1][\[Theta]][t] -> dtheta,
    Derivative[1][\[Psi]][t] -> dpsi, \[Phi][t] -> phi, \[Theta][t] ->
    theta, \[Psi][t] -> psi}];

```

```

Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

g23 = \!\
(*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_3\)\)ddtheta\)] /.
{Derivative[1][\[\Phi\]][t] -> dphi, Derivative[1][\[\Theta\]][t] -> dtheta,
Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

f2 = ddtheta /. {tau1 -> 0, tau2 -> 0,
tau3 -> 0} /. {Derivative[1][\[\Phi\]][t] -> dphi,
Derivative[1][\[\Theta\]][t] -> dtheta,
Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

g31 = \!\
(*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_1\)\)ddpsi\)] /.
{Derivative[1][\[\Phi\]][t] -> dphi, Derivative[1][\[\Theta\]][t] -> dtheta,
Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

g32 = \!\
(*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_2\)\)ddpsi\)] /.
{Derivative[1][\[\Phi\]][t] -> dphi, Derivative[1][\[\Theta\]][t] -> dtheta,
Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

g33 = \!\
(*SubscriptBox[\(\(\[PartialD]\), \(\(\tau_3\)\)ddpsi\)] /.
{Derivative[1][\[\Phi\]][t] -> dphi, Derivative[1][\[\Theta\]][t] -> dtheta,
Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

f3 = ddpsi /. {tau1 -> 0, tau2 -> 0,
tau3 -> 0} /. {Derivative[1][\[\Phi\]][t] -> dphi,
Derivative[1][\[\Theta\]][t] -> dtheta,
Derivative[1][\[\Psi\]][t] -> dpsi, \[\Phi\][t] -> phi, \[\Theta\][t] ->
theta, \[\Psi\][t] -> psi};

f = {f1, f2, f3};

g = {{g11, g12, g13}, {g21, g22, g23}, {g31, g32, g33}};

```

ANEXO B – PROGRAMA PARA CÁLCULO DO CONTROLE NO EQUILÍBRIO

```
function Ue=equilibrio(xe)
global Xe

Xe=xe;
options=optimset('TolFun',1e-24,'MaxFunEvals',1e6,'MaxIter',1e5,'TolX',1e-24);
[Ue,fval, flag]=fsolve(@din_eq,[0;0;0],options);
%disp('Final objective function value:');disp(fval);disp('flag');disp(flag);

end

function Xp=din_eq(U)
global Xe
% O objetivo é encontrar o controle de equilíbrio para um dado estado de
% equilíbrio
X=Xe;
Xp=dinamica(0,X,U);
% Remove as primeiras derivadas
Xp=[Xp(4);Xp(5);Xp(6)];
end
```

ANEXO C – PROGRAMA DE LINEARIZAÇÃO

```
function [A,B]=linearizar(Xe,Ue)
eps=1e-5;
for i=1:6
    X=Xe; X(i)=Xe(i)+eps; f1=dinamica(0,X,Ue);
    X=Xe; X(i)=Xe(i)-eps; f2=dinamica(0,X,Ue);
    A(:,i)=(f1-f2)/(2*eps);
end

for i=1:3
    U=Ue; U(i)=Ue(i)+eps; f1=dinamica(0,Xe,U);
    U=Ue; U(i)=Ue(i)-eps; f2=dinamica(0,Xe,U);
    B(:,i)=(f1-f2)/(2*eps);
end
```

ANEXO D – PROGRAMA PARA O CÁLCULO DOS GANHOS

```
function [Ky,Kv]=ganhos(A,B,H)
global C F G D J

C=eye(size(A,1));
F=zeros(size(H,1));
G=eye(size(B,2));
D=eye(size(B,2));
J=zeros(size(B,2));

Aa=[A zeros(size(A,1),size(H,1)); -G*H F];
Ba=[B; zeros(size(H,1),size(B,2))];

Q=blkdiag(H'*H,eye(size(H,1))); n=size(A,1);

for i=1:size(B,2)
    R(i,i)=input(['Dê um valor para o termo R(' num2str(i) ',' num2str(i) ') da matriz']);
    Q(n+i,n+i)=input(['Dê um valor para o termo Q(' num2str(n+i) ',' num2str(n+i) ') da matriz']);
end

Ka=lqr(Aa,Ba,Q,R);
Ky=Ka(:,1:size(A,1)); Kv=Ka(:,size(A,1)+1:end);
```

ANEXO E – PROGRAMA PARA GERAR CONTROLE COM CME

```
function u = CME(y,yref,er,ur,B)

umin = ur(1); umax = ur(2);
% Cálculo do erro
et = yref - y; e = et / er;
if e < -1, e = -1; elseif e > 1, e = 1; end
% Cálculo de u
if abs(e) == 1, ut = e; else ss = e - sign(e); ut = sign(ss) * (abs(ss) ^ (2 ^ -B))
% Cálculo da saída
a = (umax - umin) / 2; u = a * (ut - 1) + umax;
```

ANEXO F – CÓDIGO DA S-FUNCTION SF_DINAMICA

```
function [sys,x0,str,ts]=sf_dinamica(t,x,u,flag,Xe)

switch flag
  case 0 % Inicialização
    sizes=simsizes; sizes.NumContStates=6;
    sizes.NumDiscStates=0; sizes.NumOutputs=6;
    sizes.NumInputs=3; sizes.DirFeedthrough=1;
    sizes.NumSampleTimes=1; sys=simsizes(sizes);
    x0 =Xe;
    str=[];ts =[0 0];
  case 1 % Dinâmica
    sys=dinamica(x,u);
  case 3 % Saídas
    sys=x;
end
```

ANEXO G – CÓDIGO DA S-FUNCTION SF_CME

```
function [sys,x0,str,tso]=sf_cme(t,x,u,flag)
global Ue eph eth eps Bph Bth Bps ts rco

switch flag
  case 0 % Inicialização
    sizes=simsizes; sizes.NumContStates=0;
    sizes.NumDiscStates=0; sizes.NumOutputs=3;
    sizes.NumInputs=6; sizes.DirFeedthrough=1;
    sizes.NumSampleTimes=1; sys=simsizes(sizes);
    x0 =[] ;
    str=[] ;tso =[0 0] ;
  case 3 % Saídas
    %gera uma referência cúbica para o controle não saltar
    x1f=rco(1); x2f=rco(2); x3f=rco(3);
    if t < ts(1), x1f=rco(1)*(3*t^2-2*t^3/ts(1))/ts(1)^2; end
    if t < ts(2), x2f=rco(2)*(3*t^2-2*t^3/ts(2))/ts(2)^2; end
    if t < ts(3), x3f=rco(3)*(3*t^2-2*t^3/ts(3))/ts(3)^2; end
    uph=CME(u(4),x1f,eph,[-3.75 3.75],Bph);
    uth=CME(u(5),x2f,eth,[-3.75 3.75],Bth);
    ups=CME(u(6),x3f,eps,[-3.75 3.75],Bps);
    sys=Ue+[uph; uth; ups];
  end
```