

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Joel Sebastian Riascos Hurtado

**Controle de Drones com Modelo Baseado em
Quaternios**

São Carlos

2024

Joel Sebastian Riascos Hurtado

Controle de Drones com Modelo Baseado em Quaternios

Monografia apresentada ao Curso de Engenharia Aeronáutica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Aeronáutico.

Orientador: Prof. Dr. Glauco Augusto de Paula Caurin

**São Carlos
2024**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

R481c Riascos Hurtado, Joel Sebastian
Controle de Drones com Modelo Baseado em
Quaternios / Joel Sebastian Riascos Hurtado;
orientador Glauco Augusto de Paula Caurin. São Carlos,
2024.

Monografia (Graduação em Engenharia Aeronáutica)
-- Escola de Engenharia de São Carlos da Universidade
de São Paulo, 2024.

1. Quaternios. 2. Drone. 3. Euler-Lagrange. 4.
MatLab. 5. LQR. 6. Lyapunov. I. Título.

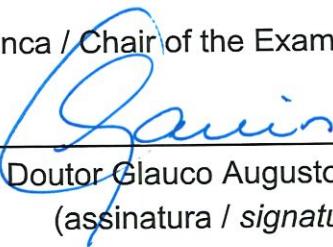
FOLHA DE APROVAÇÃO

Approval sheet

| |
|---|
| Candidato / Student: Joel Sebastian Riascos Hurtado |
| Título do TCC / Title : Controle de Drones com Modelo Baseado em Quaternios |
| Data de defesa / Date: 26/11/2024 |

| Comissão Julgadora / Examining committee | Resultado / result |
|---|---|
| Professor Doutor Glauco Augusto de Paula Caurin |  APROVADO |
| Instituição / Affiliation: EESC - SAA |  |
| Professor Doutor Jorge Henrique Bidinotto | Aprovado |
| Instituição / Affiliation: EESC - SAA | |
| Pesquisador Daniel Andres Rojas Perilla |  APROVADO |
| Instituição / Affiliation: EESC - SAA | |

Presidente da Banca / Chair of the Examining Committee:


Professor Doutor Glauco Augusto de Paula Caurin
(assinatura / signature)

*Esta tese é dedicada à minha família. Primeiro, à minha mãe, que sempre deu tudo pelo que considero mais valioso, minha **Eudaimonia**, e que me ensinou mais do que jamais alguém poderá ensinar; ao meu pai, por seu apoio incondicional quando preciso e por me ensinar o valor do esforço; ao meu irmão, por me ensinar lições de vida mesmo sendo mais novo que eu; aos meus avós, por me ensinarem a sonhar e a ver além do muro; à minha tia, por sempre acreditar em mim; e a toda minha família, que contribuiu de alguma forma para que eu esteja onde estou agora. À minha namorada, que me permite entender o mundo de um jeito que, para mim, seria impossível.*

Por fim, e não menos importante, dedico este trabalho aos meus professores desde a escola, aos das universidades, e àqueles que, sem saber, se tornaram fonte de aprendizado e inspiração para a realização de tantos projetos; escritores, filósofos, cientistas, artistas e amigos. A quem estiver lendo este trabalho, espero que ele ofereça um pouco mais de conhecimento em sua vida.

A todos, obrigado por permitirem que eu seja a pessoa que sou.

Ille,

Champion

AGRADECIMENTOS

Este trabalho deve muito à colaboração do Engenheiro Pedro Melo e do Engenheiro Glauco Caurin, que serviram como guias na realização desta tese. Agradeço aos meus pais por me darem a oportunidade de estudar no Brasil, o que me permitiu ter uma visão mais ampla do mundo. Agradeço também ao Físico Richard Avella, por sua dedicação em me fazer realmente compreender a física.

Agradeço aos meus amigos que me apoiaram ao longo do curso, especialmente a Kevin Castellanos e a Briam Castro, que sempre estiveram ao meu lado e apoiam minhas loucuras.

Por fim, agradeço tanto aos membros da Fundação Universitária Los Libertadores quanto aos da Universidade de São Paulo, que também tornaram isso possível.

*“Si quieres cambio verdadero,
pues camina distinto”*

René

RESUMO

RIASCOS, J. Controle de Drones com Modelo Baseado em Quatérnios. 2024. 67 p.
Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos,
Universidade de São Paulo, São Carlos, 2024.

Este trabalho apresenta uma investigação sobre a influência dos quatérnios no controle de drones, um tema de crescente interesse devido à expansão do uso dessa tecnologia em diversos campos. Para realizar as simulações, foi utilizado o software MATLAB, com o objetivo de estudar o comportamento de cada controlador e realizar sua comparação. O documento é dividido em 6 seções principais: a primeira contextualiza o problema e os objetivos buscados nesta investigação. A segunda seção contém a explicação de cada modelo do drone, tanto pelo método de Euler-Lagrange quanto por Quatérnios, além de uma análise sobre por que o segundo pode ser considerado superior ao primeiro. Esta seção inclui uma breve introdução aos quatérnios e suas propriedades algébricas. Na terceira seção, é apresentado o projeto dos controladores aplicados a cada modelo, sendo o LQR para Euler-Lagrange e um controlador simples por realimentação de estado para Quatérnios. A principal razão é testar a hipótese de que um controlador sem otimização para um modelo por Quatérnios pode ser equivalente ou até melhor que um controlador otimizado para um modelo de Euler-Lagrange. Sob a hipótese de que os quatérnios são a causa disso, a pesquisa pretende demonstrar que o controle do drone depende mais da natureza do seu modelo do que do projeto do controlador em si. Os controladores são simulados através da discretização da dinâmica no tempo, e cada controlador é testado sob os mesmos parâmetros, permitindo uma comparação equivalente entre ambos os enfoques. Nas últimas três seções, são apresentados os resultados e conclusões obtidos, assim como a comparação entre as simulações para cada caso, junto com as razões de por que ocorrem determinados fenômenos, como singularidades, erros e as características de cada controlador. Além disso, são sugeridas linhas de trabalho futuro derivadas dos resultados obtidos ao longo da investigação.

Palavras-chave: Quatérnios. Drone. Euler-Lagrange. MatLab. LQR. Lyapunov.

ABSTRACT

RIASCOS, J. Drone Control with Quaternion-Based Model. 2024. 67 p. Monograph (Conclusion Course Paper) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

This thesis presents a study on the influence of quaternions in drone control, a topic of increasing interest due to the growing use of this technology in various fields. MATLAB software was used to perform the simulations, enabling the study of each controller's behavior and their comparison. The document is divided into six main sections: the first contextualizes the problem and the objectives pursued in this research. The second section provides an explanation of each drone model, both Euler-Lagrange and Quaternion-based, and an analysis of why the latter can be considered superior to the former. It includes a brief introduction to quaternions and their algebraic properties. The third section presents the controller design applied to each model, with LQR for Euler-Lagrange and a simple state-feedback controller for Quaternions. The primary purpose is to test the hypothesis that an unoptimized controller for a Quaternion-based model can be equivalent to, or even better than, an optimized controller for an Euler-Lagrange model. Under the assumption that quaternions are the reason for this, the research aims to demonstrate that drone control relies more on the nature of the model than on the controller design itself. The controllers are simulated by discretizing the dynamics over time and applying each controller under the same parameters, which allows for an equivalent comparison between both approaches. The last three sections present the results and conclusions, along with a comparison between simulations for each case, explaining phenomena such as singularities, errors, and the characteristics of each controller. Additionally, future research directions are suggested based on the findings obtained during the study.

Keywords: Quaternions. Drone. Euler-Lagrange. MatLab. LQR. Luyapunov.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Arquitetura do sistema de controle utilizado para ambos modelos (Referência Escalonada) | 29 |
| Figura 2 – Ângulos de Euler. Imagem de <i>Lionel Brits</i> sob licença CC BY 3.0. ¹ | 30 |
| Figura 3 – Teoria do Elemento de Pá. Imagem de <i>Marino</i> sob licença CC BY-SA 4.0. ² . . | 30 |
| Figura 4 – Forças e Torques produzidos pelos rotores do Drone | 32 |
| Figura 5 – Modelo do Drone pelo seu Lagrangiano com o referencial inercial \mathcal{I} | 32 |
| Figura 6 – Rotação de um vetor $\vec{\lambda}$ | 34 |
| Figura 7 – Modelo do Drone por Quatérnios com um referencial inercial \mathcal{I} | 36 |
| Figura 8 – Trajetória projetada para LQR | 39 |
| Figura 9 – Trajetória projetada para Quatérnios | 40 |
| Figura 10 – Resposta de Força de (a) LQR y (b) Quatérnios | 49 |
| Figura 11 – Sobressinal de Força de (a) LQR y (b) Quatérnios | 50 |
| Figura 12 – Erro e Sobressinal de Torque LQR | 51 |
| Figura 13 – Erro e Sobressinal de Torque Quatérnios | 51 |
| Figura 14 – Acompanhamento Trajetória LQR | 52 |
| Figura 15 – Vistas da resposta do Controlador LQR | 52 |
| Figura 16 – Acompanhamento Trajetória Quatérnios | 53 |
| Figura 17 – Vistas da resposta do Controlador Quatérnios | 53 |
| Figura 18 – Resposta del quatérnio de rotação \mathbf{q} | 54 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Constantes Físicas | 48 |
|---|----|

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|-----------------------------------|
| USP | Universidade de São Paulo |
| USPSC | Campus USP de São Carlos |
| ANAC | Agência Nacional de Aviação Civil |
| ARE | Algebraic Riccati Equation |
| LQR | Linear Quadratic Regulator |
| eig | Eigenvalues |
| diag | Diagonal |
| Trs | Traslacional |
| Rot | Rotacional |
| arc cos | Arco-cosseno |

LISTA DE SÍMBOLOS

| | |
|---|----------------------|
| K | Ganho |
| S_α | Seno de α |
| C_α | Cosseno de α |
| \in | Pertence |
| e, \vec{n} | Vetor Unitário |
| T | Thrust |
| τ | Torque |
| ρ | Densidade |
| ω, Ω | Velocidade Angular |
| $\bar{\omega}$ | Frequência Angular |
| ϕ, θ, ψ | Roll, Pitch, Yaw |
| L | Lagrangiano |
| $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ | Efeito Coriolis |
| \mathbf{q} | Quaternião |
| \mathbf{q}_v | Versor |
| \mathcal{I} | Referencial Inercial |
| \mathcal{A} | Referencial Propio |

SUMÁRIO

| | | |
|----------|----------------------------------|-----------|
| 1 | INTRODUÇÃO | 25 |
| 1.1 | Aeronaves Não Tripuladas | 25 |
| 1.2 | Formulação do Problema | 26 |
| 1.3 | Objetivos | 26 |
| 1.3.1 | Objetivo Geral | 26 |
| 1.3.2 | Objetivos Específicos | 26 |
| 1.4 | Metodologia | 27 |
| 2 | MODELOS MATEMÁTICOS | 29 |
| 2.1 | Momentos e Forças em um Rotor | 29 |
| 2.2 | Modelo Dinâmico Euler-Lagrange | 31 |
| 2.3 | Modelo Dinâmico Quatérnios | 33 |
| 2.3.1 | Versor | 33 |
| 2.3.2 | Dinâmica por Quatérnios | 35 |
| 2.4 | Comparação de Modelos | 36 |
| 2.4.1 | Propriedade Intrínseca Principal | 36 |
| 2.4.2 | Rotações | 37 |
| 2.4.3 | Matemática e Computação | 37 |
| 3 | CONTROLADORES | 39 |
| 3.1 | Trajetória | 39 |
| 3.2 | Linear Quadratic Regulator | 39 |
| 3.2.1 | Equação Algébrica de Riccati | 40 |
| 3.2.2 | Controlador | 41 |
| 3.3 | Quatérnios | 42 |
| 3.3.1 | Translacional | 43 |
| 3.3.2 | Rotacional | 43 |
| 3.3.3 | Controlador | 43 |
| 3.4 | Comparação de Controladores | 45 |
| 3.4.1 | Principais Diferenças | 45 |
| 3.4.2 | Singularidades | 46 |
| 3.4.3 | Simplicidade | 46 |
| 4 | SIMULAÇÃO | 47 |
| 5 | RESULTADOS E COMPARAÇÃO | 49 |
| 5.0.1 | Resposta Força | 49 |

| | | |
|---|--|-----------|
| 5.0.2 | Resposta Torques | 50 |
| 5.1 | Acompanhamento de Trajetórias | 51 |
| 5.1.1 | Quatérnios | 52 |
| 5.2 | Comparação de Controladores | 53 |
| 6 | CONCLUSÕES | 57 |
| 6.1 | Trabalhos Futuros | 58 |
| REFERÊNCIAS | | 59 |
| ANEXOS | | 61 |
| ANEXO A – EFEITO CORIOLIS | | 63 |
| ANEXO B – ÁLGEBRA DE QUATERNIONS | | 65 |

1 INTRODUÇÃO

1.1 Aeronaves Não Tripuladas

Atualmente, os UAVs, sigla em inglês para Unmanned Aerial Vehicles (Veículos Aéreos Não Tripulados), tornaram-se uma tecnologia cada vez mais utilizada, graças à sua versatilidade e baixo custo, tanto em aplicações civis (Agricultura, Entretenimento, Recreação ou Transporte) quanto militares (Segurança, Armamentista). Como consequência, "a crescente demanda por voos autônomos com quadricópteros em diversas aplicações levou à introdução de novas estratégias de controle" [3, p. 1]. Daí a importância de estudar essa tecnologia, desde seu design até sua aplicação, para não apenas torná-los mais seguros, mas também mais precisos.

Os UAVs pertencem a uma classe específica de aeronaves multirotor, aquelas mais pesadas que o ar.⁴ No presente ano, são classificados de acordo com seu nível de autonomia como RPA (Remotely Piloted Aircraft), conforme o Regulamento Aeronáutico do Brasil (RBAC 91).⁵ De uma perspectiva mais rigorosa, eles se encontram em uma categoria distinta dos aeromodelos, daí a seriedade na regulamentação dessa tecnologia, sendo o cumprimento de procedimentos operacionais outro aspecto a ser considerado no momento do design dos controladores dos UAVs.

Ao longo deste documento, por motivos práticos e convenções de linguagem, os UAVs serão referidos simplesmente como “Drones”.

Esta pesquisa aprofunda-se nos parâmetros que compõem um controlador para drones, explorando tanto a influência do design adequado do controlador quanto a importância de contar com um modelo matemático preciso do sistema, que por si só apresenta alta complexidade. Para isso, foi realizada a simulação de dois tipos de controladores, cada um baseado em um modelo matemático diferente.

Esclarecendo que a natureza do sistema não depende do modelo matemático formulado, busca-se apresentar o fato de que as propriedades que para um modelo são *desacopladas*, para o outro já não o são, o que torna esse novo enfoque interessante.

O primeiro controlador é projetado mediante o método **LQR** (Controle Ótimo Linear-Quadrático), cujo objetivo principal é minimizar uma função de custo quadrática, por meio da solução da *Equação Algébrica de Riccati*, resultando em um (K_{LQR}).

O segundo controlador baseia-se em uma retroalimentação de estado por **Quaternios**. O ganho ($K_{Quaternios}$) é calculado a partir do critério de Lyapunov. É importante notar que ambos os sistemas possuem uma arquitetura “scaled reference”, a qual é ilustrada na 1.

Nos concentraremos então em como selecionamos (K), mas não na sua implementação.

1.2 Formulação do Problema

O design de algoritmos de controle e navegação em Drones tem sido amplamente estudado, elaborando diferentes abordagens que combinam algoritmos complexos e métodos de controle cada vez mais sofisticados. Isso se deve ao fato de que um Drone é um sistema dinâmico, instável, complexo, não linear e subatulado⁶; geralmente, a quantidade de entradas é menor que os graus de liberdade do sistema. Além disso, os drones estão sujeitos a efeitos aerodinâmicos que são complexos de modelar, entre eles "A dinâmica dos motores e suas interações com as forças de arrasto sobre as hélices",⁷ ou a influência de perturbações do ar (*i.e.* Blade-Flapping), e outros fatores que, por sua complexidade, costumam ser ignorados no controlador, afetando propriedades importantes do Drone.

Entre as abordagens propostas, estão controladores a partir da análise do sistema apenas em 2 dimensões ou desprezando os efeitos giroscópicos.⁴ Foram projetados controladores robustos buscando melhorar o desempenho otimizando a eficácia.⁸ Também foram utilizados enfoques onde são aplicados filtros de Kalman ou esquemas que aplicam várias teorias de controle.⁹ Outros autores optaram por incluir observadores ou algoritmos de controle adaptativo para obter melhores resultados.¹⁰ Uma das abordagens não convencionais foi utilizar redes neurais para melhorar a estabilidade "contra parâmetros importantes desconhecidos no modelo do sistema, bem como perturbações do vento".¹¹

Infere-se assim que reduzir a complexidade do controlador permitiria maiores e melhores intervalos de operação do Drone, conseguindo assim focar com maior detalhe naqueles aspectos complexos que também devem ser considerados no design dessa tecnologia.

1.3 Objetivos

1.3.1 Objetivo Geral

Comparar um controlador de Drone baseado em um modelo de quatérnios com um baseado na formulação de Euler-Lagrange, utilizando MATLAB.

1.3.2 Objetivos Específicos

- **Objetivo 1:** Compreender os fundamentos de cada modelo, bem como suas características intrínsecas.
- **Objetivo 2:** Projetar um controlador específico para cada modelo.
- **Objetivo 3:** Simular a resposta de cada um dos modelos para determinada trajetória de voo.
- **Objetivo 4:** Comparar os modelos e as respostas dos controladores.

1.4 Metodologia

A estrutura em que o texto está escrito corresponde a cada um dos passos realizados na pesquisa, na seguinte ordem:

- **Modelos Matemáticos:** Foram desenvolvidos modelos matemáticos para representar o comportamento dinâmico do Drone, utilizando tanto a formulação baseada em *Quaternios* quanto a formulação de *Euler-Lagrange*. Esses modelos foram fundamentais para o design dos controladores.
- **Controladores:** Foram projetados controladores para cada modelo, aproveitando as características específicas dos mesmos. Avaliando ambas as estratégias de controle, a partir dos parâmetros que devem ser considerados para cada modelo.
- **Simulação:** Foram realizadas simulações utilizando MATLAB, para avaliar a resposta de cada controlador diante de uma determinada trajetória de voo. Essas simulações permitiram analisar o comportamento dos modelos em condições controladas e obter dados para sua comparação.
- **Resultados e Comparações:** Foi realizado uma análise dos resultados obtidos das simulações, comparando o desempenho dos controladores projetados e sua relação com os modelos. Discutindo-se as vantagens e desvantagens de cada abordagem, destacando as características que fazem com que o controlador baseado em quatérnios seja superior às abordagens convencionais.

2 MODELOS MATEMÁTICOS

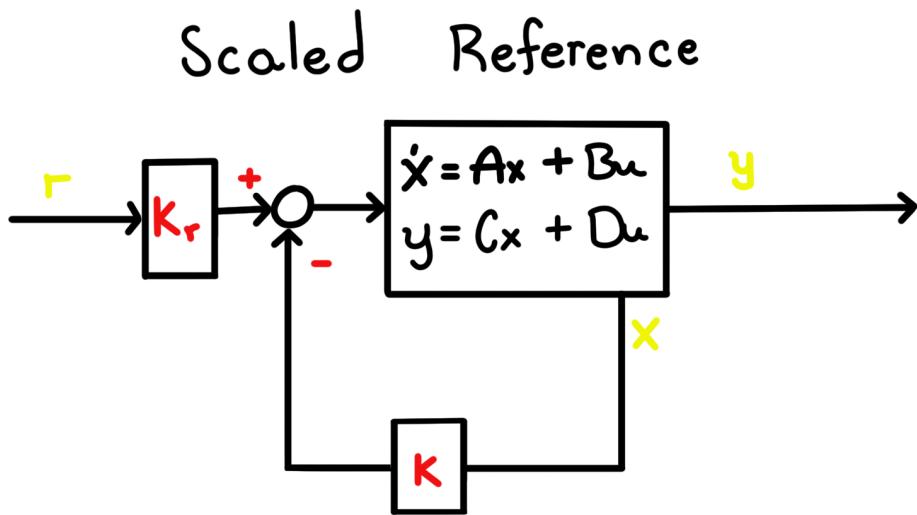


Figura 1 – Arquitetura do sistema de controle utilizado para ambos modelos (Referência Escalonada)

Os modelos que foram aplicados na maioria dos enfoques de controladores clássicos dependem de rotações sequenciais dos *Ângulos de Euler*^{12 7}. Consistem na rotação do corpo rígido em relação a um sistema inercial de coordenadas fixo (solidário à Terra), Figura 2.

A diferença entre um modelo por *Euler-Lagrange*¹³ e um por *Quaternios* está na sua capacidade de definir rotações de forma mais simples e compacta^{14 15}. O modelo dinâmico resultante para ambos os casos é um sistema subatuado.

Todos os sistemas de referência aqui apresentados seguem a regra da mão direita e os ângulos são representados em radianos. As funções trigonométricas $\text{Sen}(\alpha)$ e $\text{Cos}(\alpha)$ serão representadas como S_α e C_α , respectivamente.

2.1 Momentos e Forças em um Rotor

A partir da combinação da Teoria do Elemento de Pá Figura 3 e da Teoria do Disco Atuador, é possível modelar as forças e torques exercidos pelas hélices do Drone, considerando que a força de empuxo resultante é aplicada apenas no eixo e_z^a do referencial próprio ao Drone e sobre seu centro de massa, desconsiderando outros efeitos aerodinâmicos como o blade flapping ou o desalinhamento dos motores.

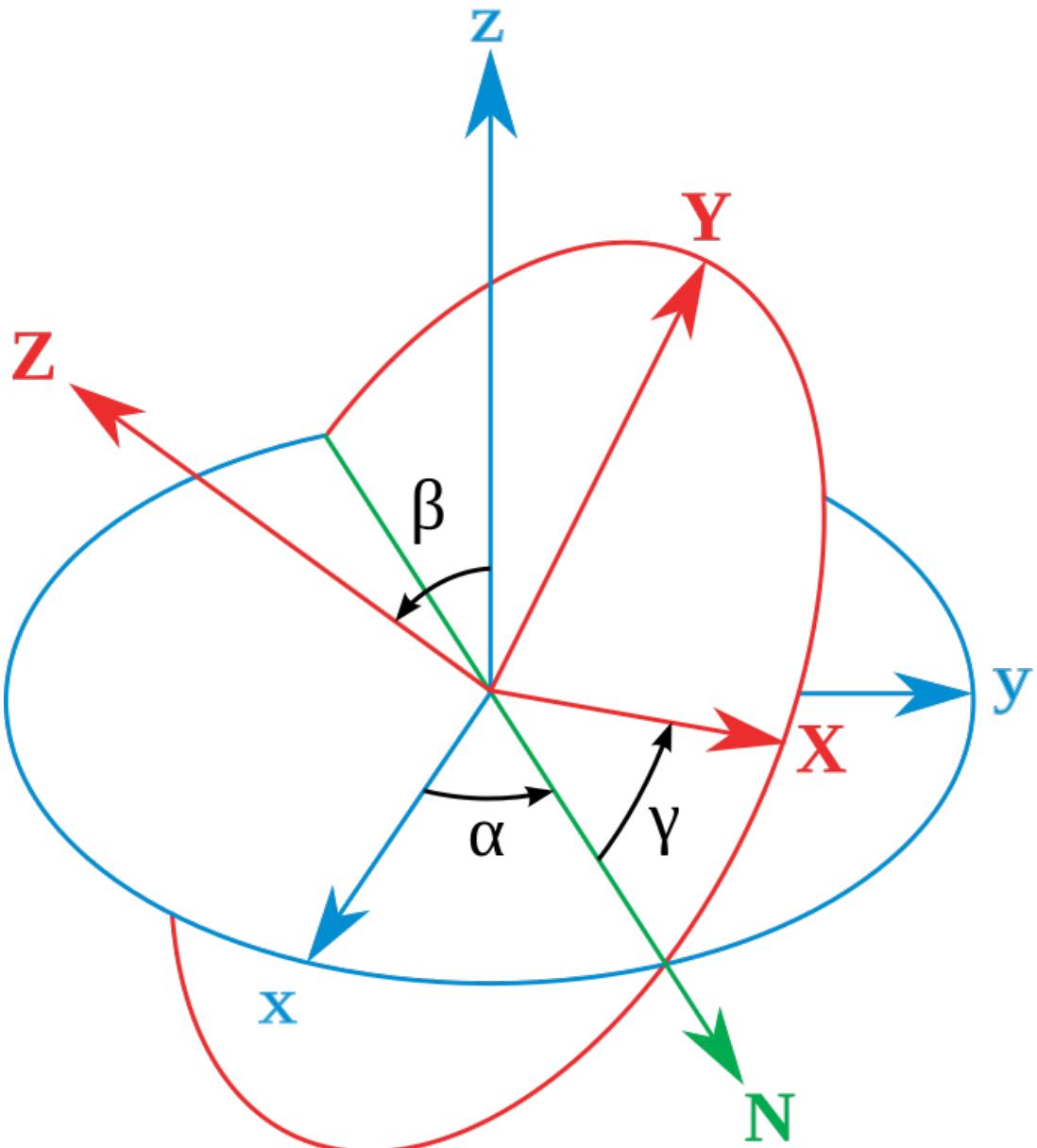


Figura 2 – Ângulos de Euler. Imagem de *Lionel Brits* sob licença CC BY 3.0.¹

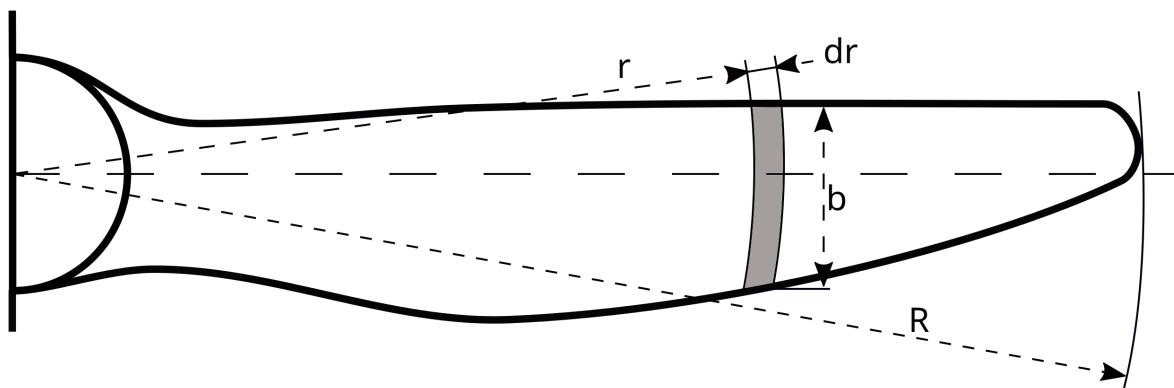


Figura 3 – Teoria do Elemento de Pá. Imagem de *Marino* sob licença CC BY-SA 4.0.²

O método decompõe a pá em pequenos elementos transversais (Perfis Aerodinâmicos), determina as forças sobre cada elemento e, em seguida, a força gerada pela pá é obtida integrando ao longo dela. O Drone estudo possui 4 rotores com 2 pás por hélice. Assim, obtém-se as equações^{16 17}:

$$T_i = C_T \rho A_h r_i^2 \omega_i^2 \quad (2.1)$$

$$\tau_i = C_Q \rho A_h r_i^3 \omega_i^2 \quad (2.2)$$

Assim, para um dado rotor i ; T_i representa o empuxo gerado, τ_i o torque produzido, r_i o raio da hélice, e ω_i sua velocidade angular. Os coeficientes de empuxo e de torque são C_T e C_Q , respectivamente. A densidade do ar é representada por ρ , e a área da hélice é simbolizada por A_h .

Considerando, primeiro, que algumas variáveis nas equações 2.1 e 2.2 são constantes, e segundo, que o Drone é simétrico, chegamos aos componentes das forças e torques totais produzidos pelos rotores do Drone:

$$\vec{T} = \begin{bmatrix} T_x \hat{\mathbf{e}}_x \\ T_y \hat{\mathbf{e}}_y \\ T_z \hat{\mathbf{e}}_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \bar{c}_T \sum_{i=1}^4 \omega_i^2 \end{bmatrix}, \quad \vec{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \bar{c}_T \sum_{i=1}^2 d_i S_{\Phi_i} \omega_i^2 (-\delta_{i,1} - \delta_{i,2} + \delta_{i,3} + \delta_{i,4}) \\ \bar{c}_T \sum_{i=1}^4 d_i C_{\Phi_i} \omega_i^2 (-\delta_{i,1} + \delta_{i,2} + \delta_{i,3} - \delta_{i,4}) \\ \bar{c}_Q \sum_{i=1}^4 \omega_i^2 \sigma_i \end{bmatrix} \quad (2.3)$$

Os torques dependem da distância em que são aplicados d_i (Braço do Drone), com sua direção $\sigma_i \in \{-1, +1\}$,⁷ sendo Φ_i o menor ângulo entre o braço de suporte do rotor e o eixo e_x^a do referencial próprio ao Drone, Figura 4.

O delta de Kronecker $\delta_{i,j}$ é igual a 1 quando $i = j$ e 0 caso contrário.

2.2 Modelo Dinâmico Euler-Lagrange

Este modelo é baseado na análise do Drone como um corpo sólido cujas forças $\in \mathbb{R}^3$. As coordenadas do veículo, $\mathbf{X}_{Drone} = (\vec{r}, \vec{\eta}) \in \mathbb{R}^6$, podem ser representadas considerando o vetor posição $\vec{r} = (x, y, z) \in \mathbb{R}^3$, do centro de massa do Drone, em relação ao referencial inercial \mathcal{I} , e o vetor $\vec{\eta} = (\phi, \theta, \psi) \in \mathbb{R}^3$, que representa a atitude do Drone (Roll, Pitch, Yaw). Definimos o Lagrangiano como:

$$L(\mathbf{X}_{Drone}, \dot{\mathbf{X}}_{Drone}) = T_{Trs} + T_{Rot} - U \quad (2.4)$$

Sabemos que a energia cinética translacional do Drone é $T_{Trs} = \frac{1}{2} m \dot{\vec{r}}^T \dot{\vec{r}}$, a energia cinética rotacional $T_{Rot} = \frac{1}{2} \vec{\Omega}^T I \vec{\Omega}$ e sua energia potencial é $U = -m \vec{g} \cdot \vec{r}$. O Drone possui uma massa m , um vetor de gravidade $\vec{g} = [0 \ 0 \ -g]^T$, uma matriz inercial I e um vetor de velocidade angular $\vec{\Omega} = [\omega_x, \omega_y, \omega_z]^T$.

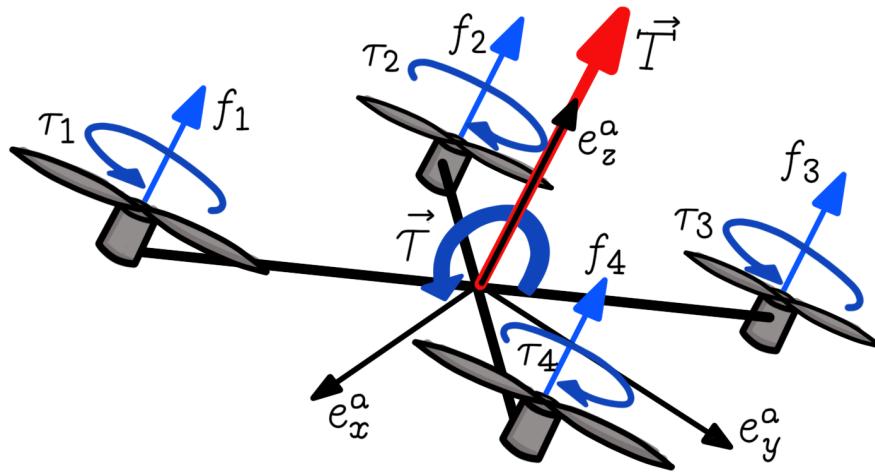


Figura 4 – Forças e Torques produzidos pelos rotores do Drone

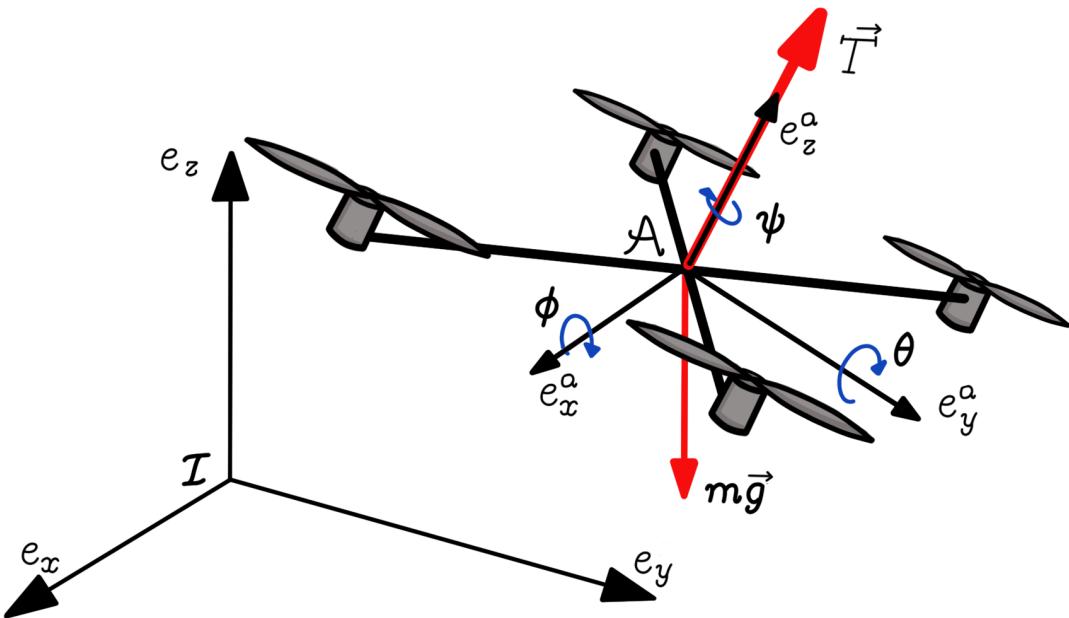


Figura 5 – Modelo do Drone pelo seu Lagrangiano com o referencial inercial \mathcal{I}

Substituindo $\vec{\Omega} = W_\eta \dot{\vec{\eta}}$, obtemos $T_{Rot} = \frac{1}{2} \dot{\vec{\eta}}^T \mathbb{I} \dot{\vec{\eta}}$. Podemos observar que a matriz inercial dependerá da rotação do Drone, com isso obtemos uma matriz inercial em função dos ângulos de Euler, calculada como $\mathbb{I} = W_\eta^T I W_\eta^{18}$, equivalente à forma como é feita por Walle,¹⁹ mas realizando a primeira rotação em ϕ .

$$W_\eta = \begin{pmatrix} -S_\theta & 0 & 1 \\ C_\theta S_\phi & C_\phi & 0 \\ C_\theta C_\phi & -S_\phi & 0 \end{pmatrix}, \quad I = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (2.5)$$

Considerando que a força de empuxo \vec{T} é aplicada apenas no eixo e_z^a , ela pode ser mapeada a partir do referencial inercial ${}^T\vec{T}_A = {}^T\mathbf{R}_A \vec{T}$, por meio de uma matriz de rotação pelos ângulos de Euler ${}^T\mathbf{R}_A \in SO(3)$:

$${}^T\mathbf{R}_A(\phi, \theta, \psi) = \begin{pmatrix} C_\psi C_\theta & -S_\psi C_\theta & S_\theta \\ S_\psi C_\phi + C_\psi S_\theta S_\phi & C_\psi C_\phi - S_\psi S_\theta S_\phi & -C_\theta S_\phi \\ S_\psi S_\phi - C_\psi S_\theta C_\phi & C_\psi S_\phi + S_\psi S_\theta C_\phi & C_\theta C_\phi \end{pmatrix} \quad (2.6)$$

Com isso, o modelo matemático da dinâmica do Drone pode ser obtido a partir da equação 2.7:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{X}}_{Drone}} - \frac{\partial L}{\partial \mathbf{X}_{Drone}} = \begin{bmatrix} {}^T\vec{T}_A \\ \vec{\tau} \end{bmatrix} \quad (2.7)$$

Obtendo,

$$m\ddot{\mathbf{r}} - m\vec{g} = {}^T\vec{T}_A, \quad \ddot{\mathbf{r}} = {}^T\vec{T}_A m^{-1} - \mathbf{g} \quad (2.8)$$

$$\mathbb{I}\ddot{\boldsymbol{\eta}} + \left(\dot{\mathbb{I}} - \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\eta}} (\dot{\boldsymbol{\eta}}^\top \mathbb{I}) \right) \dot{\boldsymbol{\eta}} = \vec{\tau}, \quad \mathbb{I}\ddot{\boldsymbol{\eta}} = \vec{\tau} - C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} \quad (2.9)$$

Sendo $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \left(\dot{\mathbb{I}} - \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\eta}} (\dot{\boldsymbol{\eta}}^\top \mathbb{I}) \right)$ o efeito Coriolis (em relação ao referencial inercial \mathcal{I}), produzido pela rotação do Drone enquanto ele está em movimento. A expansão da equação 2.9 está no Anexo A, utilizando o mesmo procedimento seguido por [6, p. 21]. Aqui se considera um dos efeitos giroscópicos que fazem parte da natureza do Drone.

2.3 Modelo Dinâmico Quaternios

2.3.1 Versor

Os quatérnios são uma extensão dos números reais, localizados em um espaço *Hiper-complexo* \mathbb{H} , introduzidos por William Rowan Hamilton em 1843. Diremos que são uma representação mais compacta de rotações, em comparação com os ângulos de Euler. Por estarem matematicamente relacionados ao grupo $SO(3)$, permitem representar rotações a partir de 4 números, ao invés de 9 como quando se utilizam matrizes por ângulos de Euler. Como o espaço tridimensional está incluído no espaço \mathbb{H} dos Quatérnios, os vetores podem ser representados como Quatérnios com sua parte escalar igual a zero.

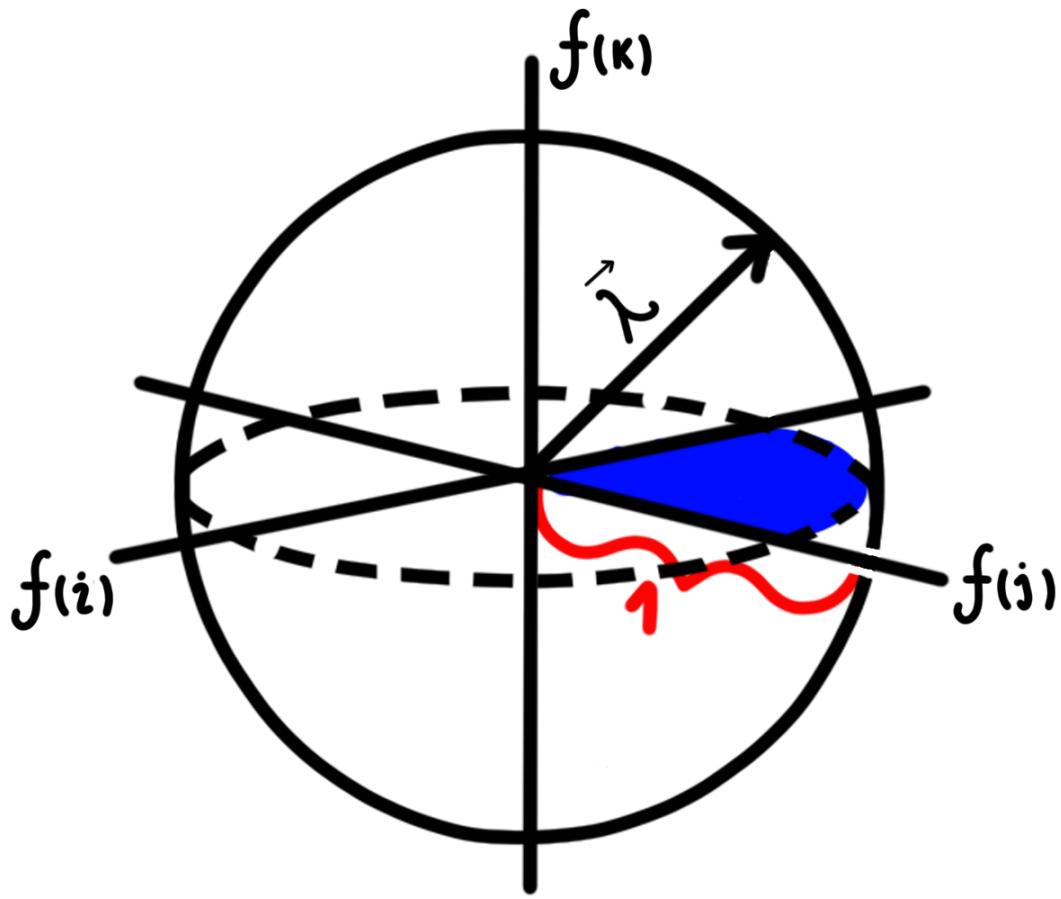


Figura 6 – Rotação de um vetor $\vec{\lambda}$

Os Quaternios são compostos por uma parte real escalar $(q_0, q_1, q_2, q_3) \in \mathbb{R}$ e uma parte imaginária $(\hat{i}, \hat{j}, \hat{k}) \in \mathbb{I}$,

$$\mathbf{q} = q_0 + q_1 \hat{i} + q_2 \hat{j} + q_3 \hat{k} = q_0 + \vec{q}, \quad \vec{q} = [q_1 \quad q_2 \quad q_3]^T \quad (2.10)$$

As regras que eles seguem são 2.11,

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1 \quad (2.11)$$

Tenhamos em mente que uma representação gráfica da rotação de quatérnios é difícil de exemplificar. Em um sentido abstrato, podemos dizer que uma rotação é denotada pelo vetor $\vec{\lambda}$. Este vetor, Figura 6, possui magnitude $\lambda = \|\vec{\lambda}\|$ em radianos, atuando sobre um eixo vetorial-unitário $\hat{h} = \vec{\lambda}/\|\vec{\lambda}\|$, cujo eixo de rotação é $\vec{\lambda} = \lambda \hat{h}$.

$$\vec{\lambda} = [\lambda_x \quad \lambda_y \quad \lambda_z]^T$$

Por meio da Identidade de Euler, sabe-se que uma rotação de λ radianos em um plano

é dada pela equação 2.12. Em 1840, Olinde Rodrigues realizou a expansão dessa identidade, parametrizando as rotações tridimensionais em função de quatérnios, obtendo a identidade 2.13.

$$e^{i\lambda} = \cos \lambda + i \sin \lambda \quad (2.12)$$

$$\mathbf{q}_v = e^{\lambda \hat{\mathbf{h}}/2} = \cos(\lambda/2) + \hat{\mathbf{h}} \sin(\lambda/2) \quad (2.13)$$

A magnitude de $|\mathbf{q}_v| = 1$, ou seja, o quatérnio unitário. Normalizar o quatérnio permite simplificar suas operações. Sabendo como o quatérnio realiza a rotação, podemos localizar seu eixo de rotação, despejando $\vec{\lambda}$.

$$\begin{aligned} \vec{\lambda} &= 2 \ln \mathbf{q} \\ \ln \mathbf{q} &= \begin{cases} [0 \ 0 \ 0]^T & \text{si } \|\vec{q}\| = 0 \\ \frac{\vec{q}}{\|\vec{q}\|} \arccos q_0 & \text{si } \|\vec{q}\| \neq 0 \end{cases} \end{aligned} \quad (2.14)$$

Com isso, podemos posicionar um vetor em qualquer localização do espaço $\in \mathbb{H}$.

Para realizar os cálculos, devemos ter em mente que a álgebra dos quatérnios segue regras específicas, as quais são apresentadas no Anexo B.

2.3.2 Dinâmica por Quatérnios

O modelo por Quatérnios^I é obtido a partir das considerações do modelo anterior (simétrico, sem blade-flapping, com empuxo apenas no eixo e_z^a), e considerando o drone como um corpo rígido em movimento, em seu sistema de referência próprio \mathcal{A} , em relação a um referencial inercial \mathcal{I} . Com isso, pelas equações de movimento de Newton, sua dinâmica é escrita por^{20 21 22}, representada na Figura 7:

$$x = \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \\ \mathbf{q} \\ \vec{\Omega} \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} \dot{\vec{r}} \\ \ddot{\vec{r}} \\ \dot{\mathbf{q}} \\ \dot{\vec{\Omega}} \end{bmatrix} = \begin{bmatrix} \dot{\vec{r}} \\ m^{-1} \mathbf{q} \circ \vec{T} \circ \mathbf{q}^* + \vec{g} \\ \frac{1}{2} \mathbf{q} \circ \vec{\Omega} \\ I^{-1} (\vec{\tau} - \vec{\Omega} \times I \vec{\Omega}) \end{bmatrix} \quad (2.15)$$

O vetor \vec{r} indica a posição do drone, $\dot{\vec{r}}$ sua velocidade e \mathbf{q} um versor denotando sua orientação, todas em relação a \mathcal{I} .

Por fim, $\vec{\Omega}$ é sua velocidade angular, I a matriz de inércia e $\vec{\tau}$ o torque total (2.3), no sistema de referência \mathcal{A} .

A força \vec{T} , neste caso, é aplicada perpendicular ao plano do drone, o qual está girando pelo quatérnio \mathbf{q} . Assim, podemos representá-la, como mostrado acima, a partir do que é apresentado em Anexo B.

^I A multiplicação de quatérnios será denotada por \circ

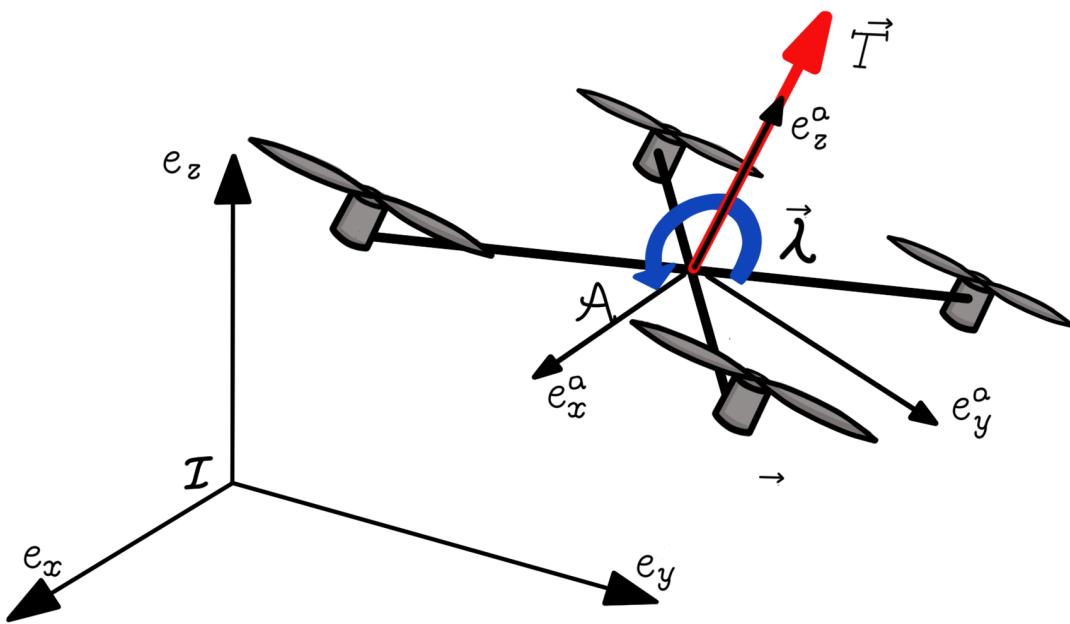


Figura 7 – Modelo do Drone por Quatérnios com um referencial inercial \mathcal{I}

O quatérnio $\mathbf{q} \rightarrow \mathbf{q}_v = e^{\lambda \hat{h}/2}$ implica a menor rotação possível de $\mathcal{I} \Rightarrow \mathcal{A}$.

2.4 Comparação de Modelos

Agora se sabe que a dinâmica do drone por *Euler-Lagrange* pode ser descrita pelas equações 2.8 e 2.9, as quais determinam sua translação e rotação, respectivamente. Neste caso, cada uma delas representa uma variável separada, ou seja, não há nada que as relate ou as conecte. Aqui, surge outra das vantagens ao utilizar o enfoque por quatérnios: ao atuar como conexões, eles se tornam uma excelente opção.

Para o caso do modelo dinâmico por *Quatérnios*, a translação e a rotação são modeladas pelo "mesmo conjunto de equações", equação 2.15. Aqui, as equações são compactadas porque a posição do drone depende apenas da força aplicada nele, e como esta depende do efeito de uma orientação, determinada por um quatérnio \mathbf{q} , então, precisamos controlar apenas a força gerada pelos motores, ou seja, a velocidade desses motores.

E então, nossa lei de controle será uma "simples" multiplicação por quatérnios, de modo que este quatérnio \mathbf{q} sirva como correção para se aproximar de \mathbf{q}_v .

2.4.1 Propriedade Intrínseca Principal

Pode parecer óbvio, mas a rotação é intrínseca ao drone, o que torna possível sua localização, e é o que define sua dinâmica. Daí derivam seus efeitos giroscópicos, e é por isso que seu controle é tão complexo. Para cada modelo, o principal objetivo é encontrar uma rotação da força no eixo z para levar o drone de um estado a outro. Faço essa anotação porque é isso que

torna o modelo por quatérnios mais eficiente. O quatérnio serve como uma representação mais elegante das propriedades intrínsecas do drone.

Aqui entram em consideração efeitos como os momentos de inércia e os efeitos giroscópicos, que são próprios do drone. A isso adicionamos, como propriedade extrínseca, a componente do efeito Coriolis, que surge da necessidade de um sistema de referência inercial, tornando o processo de determinar sua atitude complicado.

E se olharmos de uma forma abstrata, podemos pensar no drone como uma partícula tridimensional que se move e gira sobre um *plano tetradiimensional*^{II}, ou seja, estamos apenas nos reposicionando em um espaço vetorial, que segue regras especiais.

Observando as equações 2.8, 2.9 e 2.15, podemos ver que temos um total de 12 graus de liberdade: os vetores de posição, velocidade, ângulo e velocidade angular, cada um com 3 componentes. Para controlar o sistema, temos apenas 4 entradas: uma força e os torques exercidos pelos motores sobre o drone.

2.4.2 Rotações

A vantagem de usar quatérnios é que eles não são sequenciais. Se observarmos as equações 2.8 e 2.9, veremos que elas dependem de rotações produzidas por multiplicações de matrizes de cossenos direcionais, que podem ser decompostas por meio de um mapeamento inverso²¹. A escolha de ${}^T\mathbf{R}_A$ não é arbitrária; essa é uma rotação conhecida como *Ângulos de Cardano*, por Gerolamo Cardano, um matemático italiano do Renascimento; *Ângulos de Tait-Bryan*, por Peter Guthrie Tait, matemático escocês do século XIX; ou simplesmente ângulos de Euler, pelo matemático e físico suíço Leonhard Euler. Trata-se de uma sequência de rotações que possui certas vantagens frente ao **gimbal lock**, um dos maiores limitantes ao usar esse tipo de rotação. Sabe-se que nenhuma das rotações de Euler pode escapar desse fenômeno.

Ao usar quatérnios, temos a vantagem de nos desvincularmos do uso de ângulos. Incluí-los no modelo implica uma resposta contínua e natural do sistema, removendo restrições físicas e ampliando os limites do que o controlador pode fazer.

Por isso, melhora consideravelmente a adaptabilidade do sistema a diversas condições ou mudanças repentinhas, sem a necessidade de uma grande complexidade.

2.4.3 Matemática e Computação

Embora não pareça, usar quatérnios é mais simples, pois as rotações são representadas por uma única identidade, ao contrário das decomposições por ângulos de Euler, o que reduz o cálculo. Pode ser que sua representação não seja intuitiva, mas sabe-se que é a melhor forma para realizar rotações no espaço.

^{II} Os vetores são realmente equivalentes a um quatérnio com parte real zero

O custo computacional é mais baixo, pois, sendo matematicamente equivalentes aos ângulos de Euler, não alteramos as propriedades do sistema nem o comportamento do mesmo, e também não é necessário considerar correções. Apenas para comparação, para a multiplicação de duas matrizes 3×3 , são realizadas um total de 45 operações, enquanto os quatérnios exigem metade disso para se multiplicarem.

3 CONTROLADORES

3.1 Trajetória

Com $\mu \in [0, t]$, conseguimos discretizar o espaço, obtendo a trajetória desejada $\vec{r}_d = (x, y, z)$, com raio final ν e tempo final t , conforme mostrado na Figura 8.

Para aplicar essa trajetória usando quatérnios, diremos que uma força desejada aponta para o centro da espiral, Figura 9.

$$Tray(t) = \begin{cases} x = x_0 + \mu \cdot \nu \cdot \cos(\bar{\omega} \cdot \frac{2\pi\mu}{t})/t & x_0 = 0.1 \\ y = y_0 + \mu \cdot \nu \cdot \sin(\bar{\omega} \cdot \frac{2\pi\mu}{t})/t & y_0 = 0.1 \\ z = (z_0 + \frac{1}{2}g\nu^2) \cdot \frac{h}{1+g/2} & z_0 = 2, \nu = \mu/t \\ \bar{\omega} = 5 & \nu = 8 \\ g = 9.81 \text{ m/s}^2 & h = 15 \end{cases} \quad (3.1)$$

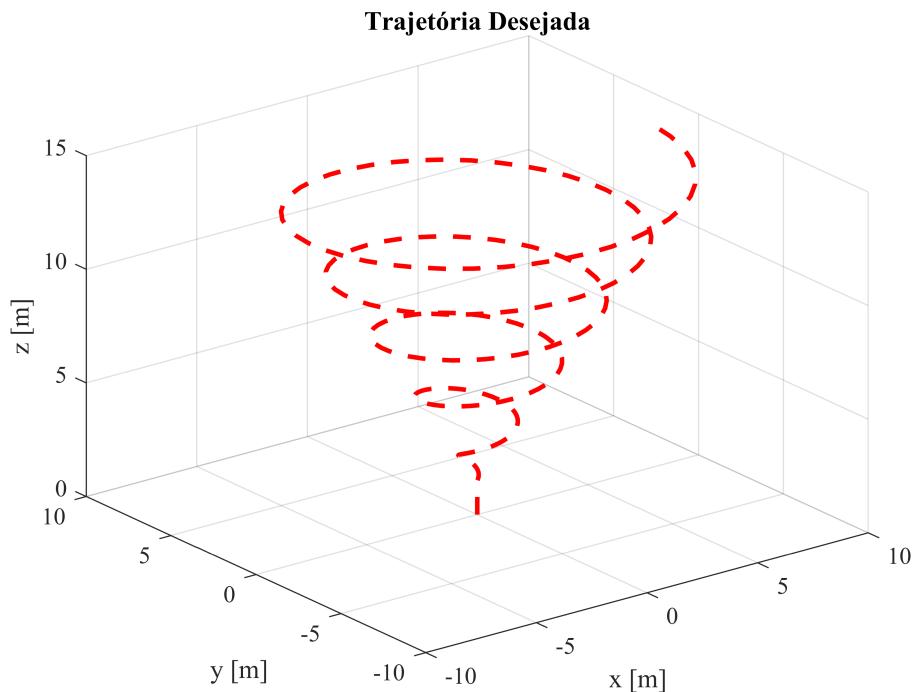


Figura 8 – Trajetória projetada para LQR

3.2 Linear Quadratic Regulator

Essa teoria baseia-se em operar a dinâmica de um sistema com o menor custo possível, utilizando uma função de custo que encontra o K_{LQR} ótimo para que os recursos «aplicados ao

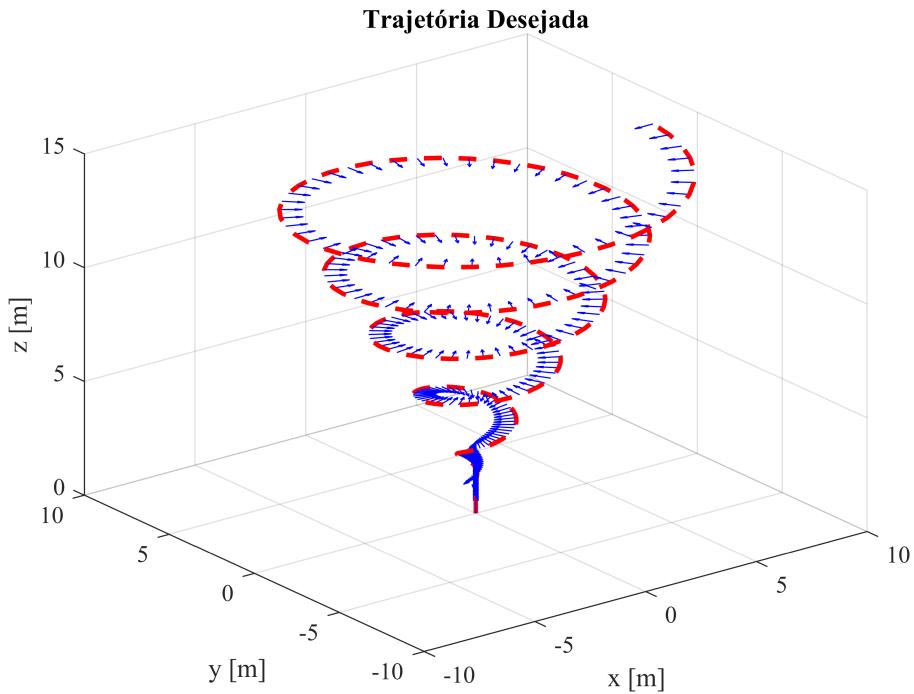


Figura 9 – Trajetória projetada para Quatérnios

sistema» sejam consumidos na menor quantidade possível. Em seguida, o resultado é elevado ao quadrado para evitar valores negativos no cálculo. O projeto de controladores por LQR é um método bastante eficiente, que permite reduzir o erro em regime permanente²³.

A função de custo é apresentada em 3.2, sendo \mathbf{Q} o componente que penaliza o desempenho e \mathbf{R} aquele que penaliza o esforço do atuador (energia).

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (3.2)$$

Integrar de 0 até ∞ significa que o controlador busca minimizar o erro durante toda a operação do sistema, forçando-o a convergir para o equilíbrio e a minimizar qualquer desvio ao longo do tempo. Com um horizonte infinito, a solução da ARE torna-se constante e estacionária, o que significa que não há necessidade de recalcular K_{LQR} a cada instante, como ocorre no controle ótimo de horizonte finito. Isso resulta em um controlador de realimentação de estado constante.

3.2.1 Equação Algébrica de Riccati

A equação apresentada em 3.3 é obtida ao incluir uma matriz simétrica $\mathbf{P} = \mathbf{P}^T$. Ao reorganizar e completar o quadrado em 3.2, obtemos, para um modelo linear, a respectiva ARE^I:

^I ARE, pela sigla em inglês

$$\begin{aligned}\dot{x} &= \mathbf{A}x + \mathbf{B}u & \mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} &= 0 \\ y &= \mathbf{C}x + \mathbf{D}u, & u &= -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} x\end{aligned}\tag{3.3}$$

3.2.2 Controlador

O controlador, então, utiliza uma realimentação com $K_{LQR} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$, onde \mathbf{P} é encontrada ao resolver a ARE.

Agora, observamos que é difícil linearizar a equação 2.9 para ser usada no método LQR, por isso considera-se \mathbb{I} como uma matriz diagonal constante:

$$\ddot{\boldsymbol{\eta}} = I^{-1}(\vec{\tau} - C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}) \tag{3.4}$$

Observamos que, ao resolver as equações de translação e rotação, temos 6 possíveis entradas \vec{u} para um vetor de estado $\ddot{\mathbf{X}}_{Drone}$:

$$\begin{aligned}\ddot{\mathbf{r}}_1 &= S_\theta m^{-1} U_1, \\ \ddot{\mathbf{r}}_2 &= -C_\theta S_\phi m^{-1} U_1, \\ \ddot{\mathbf{r}}_3 &= C_\theta C_\phi^{-1} U_1 - g, \\ \ddot{\boldsymbol{\eta}}_1 &= -I_{xx}^{-1} [(\dot{\psi} c_{11} + \dot{\theta} c_{12} + \dot{\phi} c_{13}) + U_2], \\ \ddot{\boldsymbol{\eta}}_2 &= -I_{yy}^{-1} [(\dot{\psi} c_{12} + \dot{\theta} c_{22} + \dot{\phi} c_{23}) + U_3], \\ \ddot{\boldsymbol{\eta}}_3 &= -I_{zz}^{-1} [(\dot{\psi} c_{13} + \dot{\theta} c_{32} + \dot{\phi} c_{33}) + U_4].\end{aligned}\tag{3.5}$$

Aqui surge um ponto importante: apenas $\ddot{\boldsymbol{\eta}}$ corresponde a 3 das 4 entradas disponíveis em um Drone, ou seja, com nossas entradas controlamos, em sua maior parte, a rotação do Drone.

Assim, temos uma matriz $[\mathbf{X}_{Drone}, \dot{\mathbf{X}}_{Drone}]$.

Escrevendo o sistema no espaço de estados, definimos que 0_3 é uma matriz quadrada de zeros e $I_{d_3}^{II}$ é a matriz identidade, ambas de dimensão 3×3 .

$$x = [\mathbf{r} \ \dot{\mathbf{r}} \ \boldsymbol{\eta} \ \dot{\boldsymbol{\eta}}]^T, \quad \vec{u} = [U_1, U_2 \ U_3 \ U_4]^T = [\vec{\mathbf{T}}_{\mathcal{A}} - \vec{g}, \ \vec{\tau}]^T$$

Para linearizar o modelo, considera-se que os ângulos são pequenos (entre iterações), de modo que $(S_\alpha = \alpha)$ e $(C_\alpha = 1)^{24}$. Com isso, é possível discretizar a equação.

Nossas equações no espaço de estados, com $\mathbf{G} = -I^{-1}C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$, são:

^{II} Não confundir com a matriz de inércia, I .

$$\mathbf{A} = \begin{bmatrix} 0_3 & \mathbf{I}_{d_3} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 & 0_3 & \mathbf{G} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0_3 & 0_3 \\ \mathbf{I}_{d_3} & 0_3 \\ 0_3 & 0_3 \\ 0_3 & I^{-1} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{I}_{d_3} & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & \mathbf{I}_{d_3} & 0_3 \end{bmatrix}, \quad \mathbf{D} = 0 \quad (3.6)$$

Resta agora a lei de controle, que é implementada criando um erro entre o estado atual e um estado de referência. Assim, nosso sistema em malha fechada é:

$$\dot{x} = \mathbf{Ax} + \mathbf{B}(-K_{LQR}(x - x_{\text{ref}})) \quad (3.7)$$

O erro, $(x - x_{\text{ref}})$, leva o sistema a um erro de estado estável por meio de uma compensação em relação à referência. Como já temos uma trajetória definida, encontraremos uma velocidade angular e os ângulos desejados por:

$$\dot{\vec{\eta}}_d = \frac{\vec{r}_d \times \dot{\vec{r}}_d}{|\vec{r}_d|} \quad (3.8)$$

$$\cos(\eta_d) = \frac{\vec{r}_d \cdot \dot{\vec{r}}_d}{|\vec{r}_d||\dot{\vec{r}}_d|} \quad (3.9)$$

$$\eta_d = \cos^{-1} \left(\frac{\vec{r}_d \cdot \dot{\vec{r}}_d}{|\vec{r}_d||\dot{\vec{r}}_d|} \right) \quad (3.10)$$

Calcula-se η_d para obter um ângulo de referência ao calcular os efeitos de Coriolis para cada iteração. Assim, definimos que nosso estado de referência $x_{\text{ref}} = x_d$, para a trajetória desejada \vec{r}_d , será:

$$x_d = [\dot{\vec{r}}_d, \dot{\vec{\eta}}_d], \quad u = -K_{LQR}[\vec{r} - \vec{r}_d, \vec{\eta} - \dot{\vec{\eta}}_d]$$

Com isso, definimos completamente nossas entradas, de modo que $\dot{\vec{r}} \rightarrow \dot{\vec{r}}_d$ e $\dot{\vec{\eta}} \rightarrow \dot{\vec{\eta}}_d$.

3.3 Quatérnios

Para explicar o design do controlador por quatérnios, o desacoplaremos em duas partes: translacional e rotacional. No entanto, cabe esclarecer, como mencionado anteriormente, que a dinâmica do Drone está compactada, ao contrário do modelo por Euler-Lagrange. O design do controlador segue um procedimento semelhante ao realizado em.⁶

Para simplificar, diremos que ${}^T\vec{T}_{\mathcal{A}} = \mathbf{q} \circ \vec{T} \circ \mathbf{q}^*$, e separaremos a equação 2.15 em sua componente translacional e rotacional, respectivamente:

$$\dot{x}_{Trs} = \begin{bmatrix} \dot{\vec{r}} \\ \ddot{\vec{r}} \end{bmatrix} = \begin{bmatrix} \dot{\vec{r}} \\ m^{-1} {}^T \vec{T}_{\mathcal{A}} + \vec{g} \end{bmatrix}, \quad \dot{x}_{Rot} = \begin{bmatrix} \dot{\vec{\Omega}} \\ \ddot{\vec{\Omega}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \mathbf{q} \circ \vec{\Omega} \\ I^{-1} (\vec{\tau} - \vec{\Omega} \times I \vec{\Omega}) \end{bmatrix} \quad (3.11)$$

3.3.1 Translacional

O design do controlador translacional baseia-se em fazer a posição do Drone (\vec{r}) e a posição desejada (\vec{r}_d) convergirem para zero. O objetivo é projetar uma força ${}^T \vec{T}_{\mathcal{A}}$, de modo que $x_{Trs}, \dot{x}_{Trs} \Rightarrow 0$, rastreando o erro da posição do Drone.

$$\vec{r}_e = \vec{r} - \vec{r}_d$$

$$\begin{bmatrix} \dot{\vec{r}}_e \\ \ddot{\vec{r}}_e \end{bmatrix} = \begin{bmatrix} \dot{\vec{r}}_e \\ m^{-1} {}^T \vec{T}_{\mathcal{A}} + \vec{g} \end{bmatrix}, \quad (3.12)$$

3.3.2 Rotacional

Para o caso do controlador rotacional, sabemos que seu controle dependerá de $\vec{\tau}$. Aqui, o objetivo é aplicar um quatérnio \mathbf{q} tal que a atitude do Drone converja para o versor $\mathbf{q}_v = 1 + [0 \ 0 \ 0]^T$, enquanto a orientação do eixo $\vec{\lambda}$ e sua velocidade angular convergem para 0.

Assim como antes, rastreamos a posição do Drone por um quatérnio de erro \mathbf{q}_e , em relação a um quatérnio desejado \mathbf{q}_d .

$$\mathbf{q}_e = \mathbf{q}_d^* \circ \mathbf{q} \quad (3.13)$$

Seguindo 3.13, a conclusão é então que $\mathbf{q} \rightarrow \mathbf{q}_d^*$, para que $\mathbf{q}_e \rightarrow \mathbf{q}_v$.

$$\begin{bmatrix} \dot{\vec{\Omega}}_e \\ \ddot{\vec{\Omega}}_e \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \mathbf{q}_e \circ \vec{\Omega}_e \\ I^{-1} (\vec{\tau} - \vec{\Omega}_e \times I \vec{\Omega}_e) \end{bmatrix} \quad (3.14)$$

3.3.3 Controlador

A trajetória desejada para o quatérnio é calculada a partir da consideração de uma força desejada \vec{T}_d , ao longo da trajetória, que aponta para o centro da mesma. Então, normalizando os vetores \vec{T}_d e \vec{T} , obtemos \vec{n}_d e \vec{n}_z , respectivamente, onde $\vec{n}_d \times \vec{n}_z$ determina o eixo $\vec{\Lambda}_d$ e a rotação com o menor ângulo possível λ_d entre \vec{T}_d e \vec{T} :

$$\vec{n}_d \times \vec{n}_z = \vec{\Lambda}_d \ S_{\lambda_d} \quad (3.15)$$

$$\vec{n}_d \circ \vec{n}_z^* = -\vec{n}_d \cdot \vec{n}_z^* + \vec{n}_d \times \vec{n}_z^* = \vec{n}_z \cdot \vec{n}_d + \vec{n}_z \times \vec{n}_d \quad (3.16)$$

$$\mathbf{q}_d = e^{\frac{\ln(\vec{n}_d \circ \vec{n}_z^*)}{2}} \circ e^{\frac{[0 \ 0 \ \psi_d]^T}{2}}, \quad (3.17)$$

O controlador por realimentação de estado, projetado para este caso, segue a lei de Lyapunov, que estabelece que, se existir uma função $V(x)^{III}$, contínua, diferenciável e positiva definida, cuja derivada em relação ao tempo $\dot{V}(x)^{IV}$ é negativa definida, então a origem do sistema é um ponto de equilíbrio estável pelo critério de Lyapunov.

$$\text{Criterio de Lyapunov: } V(x) = \frac{1}{2} x(t)^2 \quad \begin{cases} V(x) > 0, & \forall x \neq 0, \\ \dot{V}(x) < 0, & \forall x \neq 0. \end{cases}$$

Escrevendo os sub-sistemas pelas equações no espaço de estados:

$$\begin{bmatrix} \dot{\vec{r}}_e \\ \ddot{\vec{r}}_e \end{bmatrix} = \begin{bmatrix} 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} \vec{r}_e \\ \dot{\vec{r}}_e \end{bmatrix} + \begin{bmatrix} 0_3 \\ \mathbf{I}_{d_3} \end{bmatrix} (\vec{T}_d m^{-1} + \vec{g}) \quad (3.18)$$

$$\begin{bmatrix} \dot{\vec{\lambda}}_e \\ \ddot{\vec{\lambda}}_e \end{bmatrix} = \begin{bmatrix} 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} \vec{\lambda}_e \\ \dot{\vec{\lambda}}_e \end{bmatrix} + \begin{bmatrix} 0_3 \\ I^{-1} \end{bmatrix} (\vec{\tau}_d - \dot{\vec{\lambda}}_e \times I \dot{\vec{\lambda}}_e) \quad (3.19)$$

Assim, as equações positivas definidas são formuladas:

$$V_{Trs} = \frac{1}{2} \begin{bmatrix} \vec{r}_e \\ \dot{\vec{r}}_e \end{bmatrix} \cdot \begin{bmatrix} \vec{r}_e \\ \dot{\vec{r}}_e \end{bmatrix}, \quad V_{Rot} = \frac{1}{2} \begin{bmatrix} \vec{\lambda}_e \\ \dot{\vec{\lambda}}_e \end{bmatrix} \cdot \begin{bmatrix} \vec{\lambda}_e \\ \dot{\vec{\lambda}}_e \end{bmatrix}. \quad (3.20)$$

Suas derivadas são:

$$\begin{aligned} \dot{V}_{Trs} &= \begin{bmatrix} \dot{\vec{r}}_e \\ \ddot{\vec{r}}_e \end{bmatrix} \cdot \left(\begin{bmatrix} 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} \vec{r}_e \\ \dot{\vec{r}}_e \end{bmatrix} + \begin{bmatrix} 0_3 \\ \mathbf{I}_{d_3} m^{-1} \end{bmatrix} (\vec{T}_d + \vec{g}m) \right), \\ \dot{V}_{Rot} &= \begin{bmatrix} \dot{\vec{\lambda}}_e \\ \ddot{\vec{\lambda}}_e \end{bmatrix} \cdot \left(\begin{bmatrix} 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} \vec{\lambda}_e \\ \dot{\vec{\lambda}}_e \end{bmatrix} + \begin{bmatrix} 0_3 \\ I^{-1} \end{bmatrix} (\vec{\tau} - \dot{\vec{\lambda}}_e \times I \dot{\vec{\lambda}}_e) \right) \end{aligned} \quad (3.21)$$

As entradas serão propostas da mesma forma que antes, onde o erro é multiplicado por uma ganância K .

$$\vec{T}_d = - [K_{p,Trs} \ K_{d,Trs}] \begin{bmatrix} \vec{r}_e \\ \dot{\vec{r}}_e \end{bmatrix} - m\vec{g} \quad (3.22)$$

$$K_{p,Trs} = \text{diag}(\kappa_x, \kappa_y, \kappa_z), \quad K_{d,Trs} = \text{diag}(\bar{\kappa}_x, \bar{\kappa}_y, \bar{\kappa}_z)$$

^{III} Se $V(x) \rightarrow 0$ quando $x \rightarrow 0$, então o sistema é *assintoticamente estável*.

^{IV} Isso implica que o sistema tende a se aproximar do ponto de equilíbrio.

As entradas levam a que o critério de Lyapunov seja cumprido.

$$\vec{\tau} = -[K_{p,Rot} \quad K_{d,Rot}] \begin{bmatrix} \vec{\lambda}_e \\ \dot{\vec{\lambda}}_e \\ \ddot{\vec{\lambda}}_e \end{bmatrix} + \dot{\vec{\lambda}}_e \times I \dot{\vec{\lambda}}_e \quad (3.23)$$

$$K_{p,Rot} = \text{diag}(\epsilon_x, \epsilon_y, \epsilon_z), \quad K_{d,Rot} = \text{diag}(\bar{\epsilon}_x, \bar{\epsilon}_y, \bar{\epsilon}_z)$$

Substituindo 3.23 e 3.22 em 3.21, obtemos as matrizes que determinam os $K_{Quatérnios}$ do sistema:

$$\begin{aligned} & \text{eig} \left(\begin{bmatrix} 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 \end{bmatrix} + \begin{bmatrix} 0_3 \\ \mathbf{I}_{d_3} m^{-1} \end{bmatrix} [K_{p,Trs} \quad K_{d,Trs}] \right), \\ & \text{eig} \left(\begin{bmatrix} 0_3 & \mathbf{I}_{d_3} \\ 0_3 & 0_3 \end{bmatrix} + \begin{bmatrix} 0_3 \\ I^{-1} \end{bmatrix} [K_{p,Rot} \quad K_{d,Rot}] \right), \end{aligned} \quad (3.24)$$

Então, para estabilizar cada subsistema assintoticamente, as ganâncias K devem ser escolhidas de forma que as partes reais de 3.24 sejam negativas definidas.

Para realizar o acompanhamento do erro nos quatérnios, a fim de determinar qual quatérnio deve ser aplicado para corrigir esse erro, calculamos:

$$\vec{\lambda}_e = 2\ln(\mathbf{q}_e), \quad \mathbf{q}_e = \mathbf{q}_d^* \circ \mathbf{q}, \quad \dot{\vec{\lambda}}_e = \vec{\Omega} - 2\frac{d}{dt}\ln(\mathbf{q}_d) \quad (3.25)$$

3.4 Comparação de Controladores

3.4.1 Principais Diferenças

Note-se então que um controlador clássico requer uma grande quantidade de cálculos devido aos erros, que vêm da natureza sequencial das rotações. Por exemplo, para o cálculo de um estado para outro, operam-se, neste caso, matrizes $\in \mathbb{R}^{12}$, além de ser necessário calcular primeiro um eixo para então realizar a rotação.

Agora, sabendo que temos apenas 4 entradas de controle, a discussão é qual das posições \vec{r} usar. Isso dependerá do que se busca realizar. A dinâmica do Drone só permitirá um grau de liberdade em relação à posição. Por isso, são utilizados diferentes métodos de controle para levar o Drone de um estado a outro, sendo necessário sacrificar 2 das possíveis entradas. O problema reside na complexidade da dinâmica do Drone, induzida por operações entre ângulos, ao utilizar Matrizes de Rotação sequenciais.

3.4.2 Singularidades

Uma vantagem dos quatérnios é eliminar as singularidades devido aos ângulos de Euler. Para a rotação utilizada ${}^T\mathbf{R}_{\mathcal{A}}(\phi, \theta, \psi)$, se o pitch chega a $\theta \rightarrow \pi/2$, então $C_\theta = 0$, situação na qual o mapeamento inverso de ${}^T\mathbf{R}_{\mathcal{A}} \rightarrow \vec{\eta}$ não está definido. O processo de mapear os ângulos correspondentes a partir de uma matriz de rotação é importante porque requer saber a velocidade angular dos motores, para fazer girar o Drone.

Outra desvantagem dos ângulos de Euler é que eles perdem precisão ao integrar mudanças incrementais da atitude ao longo do tempo, mesmo sendo estas mudanças pequenas.²¹

3.4.3 Simplicidade

Observa-se que usar quatérnios remove as 3 atitudes , equação 3.5, e reduz os cálculos necessários para chegar de um estado a outro. A razão é que já não é necessário calcular um estado, revisar seu erro e depois mapear de volta o ângulo para aplicar uma entrada proporcional a esse erro. O que nos ajuda a simplificar os cálculos em quatérnios é que temos como referência um versor, assim nos preocupamos apenas em definir um eixo de rotação. A ideia, neste caso, é calcular um \mathbf{q} que nos aproxime o máximo possível de \mathbf{q}_v , quatérnio projetado paralelo a um desejado.

4 SIMULAÇÃO

Para a validação dos modelos, foram realizadas simulações utilizando MATLAB, com um tempo de simulação $t = 400$. Os cálculos são realizados de forma iterativa, discretizando a resposta no tempo. As posições e velocidades de referência são computadas antes do cálculo da resposta de cada controlador. Diremos que um valor presente é representado por \bullet , futuro com $^+$ e passado com $-$.

Esclarecemos que as derivadas foram calculadas utilizando diferenças finitas centrais, e aplicando diferenças progressivas e regressivas nas extremidades, com isso melhoramos a precisão do cálculo. A sequência que segue o algoritmo aplicado na simulação é:

1. Inicialização:

- Definimos as constantes físicas que nos permitirão calcular as condições iniciais, a partir dos dados da tabela 1.
- Definimos a matriz de inércia I com seus valores nos eixos I_{xx} , I_{yy} , I_{zz} .

2. Trajetória:

- Definimos a trajetória e calculamos os vetores associados.
- Calculamos a velocidade por diferenças finitas, $\vec{r}_d = (\vec{r}_d^+ - \vec{r}_d^-)/2dt$.
- Calculamos o ângulo entre o vetor posição e o vetor velocidade em cada ponto, 3.10, a diferença entre esses ângulos determina o nosso ângulo de roll desejado ψ_d .
- Calculamos a velocidade angular do Drone $\vec{\Omega}$, equação 3.8.

3. Definição das Matrizes do Sistema no Espaço de Estados:

- Colocamos as matrizes A , B , C , e D do sistema no espaço de estados^I, e suas respectivas ganâncias K .
- Criamos o sistema `sys` como um sistema discreto com tempo de amostragem dt ,^{II}.

4. Simulação do Sistema:

- Calculamos os erros, para saber as correções que devem ser aplicadas.
- Simulamos o estado seguinte y^+ , para t_\bullet , com a entrada $U_\bullet x_\bullet$ usando `lsim` no intervalo de tempo atual ($\Delta t = 0.1$ segundos).
- Armazenamos o estado x^+ e a saída y^+ atualizados para calcular o próximo estado.

^I Para os quatérnios devem ser acopladas 3.18 e 3.19, operando a dinâmica como um único sistema

^{II} Cuja função no MATLAB é `sys = ss(A, B, C, D, dt)`

- O processo é repetido até o fim da trajetória.

$$x(t + \Delta t) = \mathbf{A}x(t) - \mathbf{B} K u(t) \quad (4.1)$$

Para ambos controladores, o processo de simulação foi o mesmo, a diferença está no fato de que, para o controlador por quatérnios, a cada iteração deve-se mapear de volta o quatérnio \mathbf{q} que corrige nosso erro, equação 3.25. Então, tendo:

$$(\mathbf{q}_d^*)_\bullet^{-1} e^{\frac{\vec{\lambda}_e}{2}} = \mathbf{q}^+$$

Assim, podemos rotacionar $\mathbf{q}^+ \circ \vec{T}_\bullet \circ \mathbf{q}^{+\ast}$ ^{III}. A derivada do logaritmo do quatérnio desejado foi calculada da seguinte forma:

$$\frac{d}{dt} \ln(\mathbf{q}_d) = \ln(\mathbf{q}_d)^+ - \ln(\mathbf{q}_d)^-/2 dt$$

As constantes físicas utilizadas foram tomadas como proposição de um drone que possui essas características, e os valores de inércia pertencem ao projeto de um drone quadrado.

| Constante | Valor | Unidade | Ganho |
|------------------------|------------|----------------------------|-------------------------------|
| g | 9.81 | m/s^2 | $K_{p,Trs} = \text{diag}(10)$ |
| m | 1 | kg | $K_{d,Trs} = \text{diag}(2)$ |
| ω_{Init} | 1200 | rev/s | $K_{p,Rot} = \text{diag}(30)$ |
| C_T | $6.6e - 2$ | | $K_{d,Rot} = \text{diag}(5)$ |
| C_T/C_Q | 11.36 | | |
| I_{xx} | 0.2 | $\text{kg}\cdot\text{m}^2$ | Geométricos |
| I_{yy} | 0.2 | $\text{kg}\cdot\text{m}^2$ | |
| I_{zz} | 0.4 | $\text{kg}\cdot\text{m}^2$ | |
| ρ | 1.225 | kg/m^3 | |
| r^b | 2.2e-2 | m | $\Phi_i = \frac{\sqrt{2}}{2}$ |

^a Distância dos Braços

^b Raio da Hélice

Tabela 1 – Constantes Físicas

^{III} Ter cuidado, os quatérnios devem seguir uma ordem de pré-multiplicação, pois não comutam no produto

5 RESULTADOS E COMPARAÇÃO

Estes foram os resultados obtidos na simulação, e são apresentadas as respostas de cada controlador. Os gráficos mostrados exibem os erros de cada entrada.

Vale ressaltar que cores iguais denotam a mesma variável, independentemente do controlador ao qual se refere.

5.0.1 Resposta Força

A resposta à força é apresentada apenas como a exercida em z , uma vez que é a força de maior interesse neste estudo. Para os torques τ , consideramos suas componentes, assim definindo nossas 4 entradas.

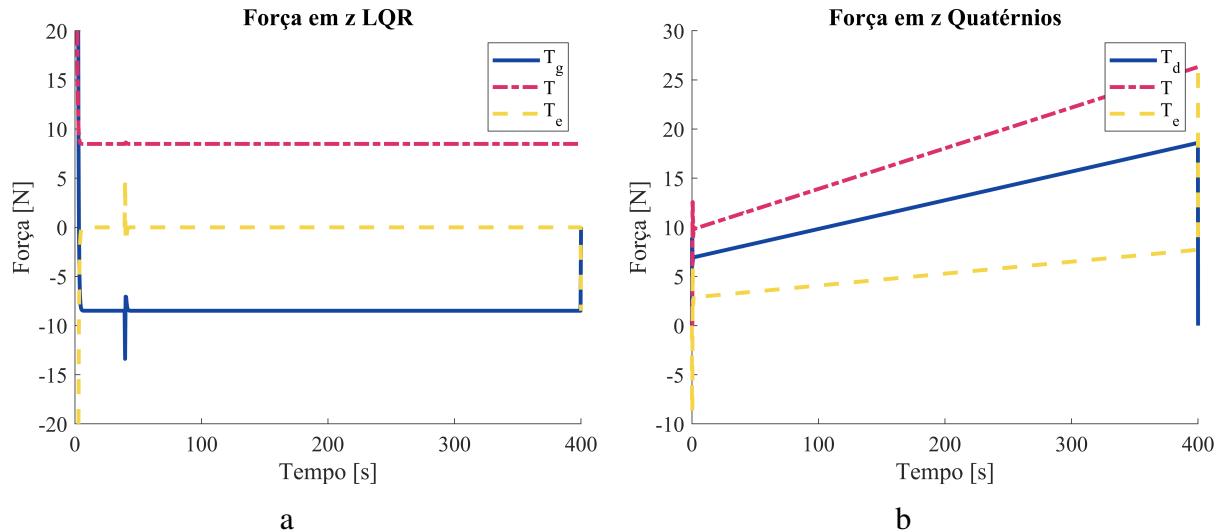


Figura 10 – Resposta de Força de (a) LQR y (b) Quatérnios

A Figura 10 permite evidenciar o comportamento da força em z de cada controlador. As forças na Figura (a) correspondem à força exercida pela gravidade (T_g), à força que o drone exerce durante a trajetória (T) e ao erro dessa força (T_e), respectivamente.

As forças na Figura (b) representam, respectivamente, a força desejada (T_d), a força aplicada pelo drone durante a trajetória (T) e o erro entre essas forças (T_e).

Para o caso da força em quatérnios, observamos que ela aumenta devido ao fato de que essa força está direcionada para o centro da trajetória. Ou seja, o drone exerce uma força cada vez maior para contrabalançar a força centrífuga, que depende do aumento do raio. O erro aumenta de 2.9 newtons, ao começo, para 7.71 newtons ao atingir 400 segundos.

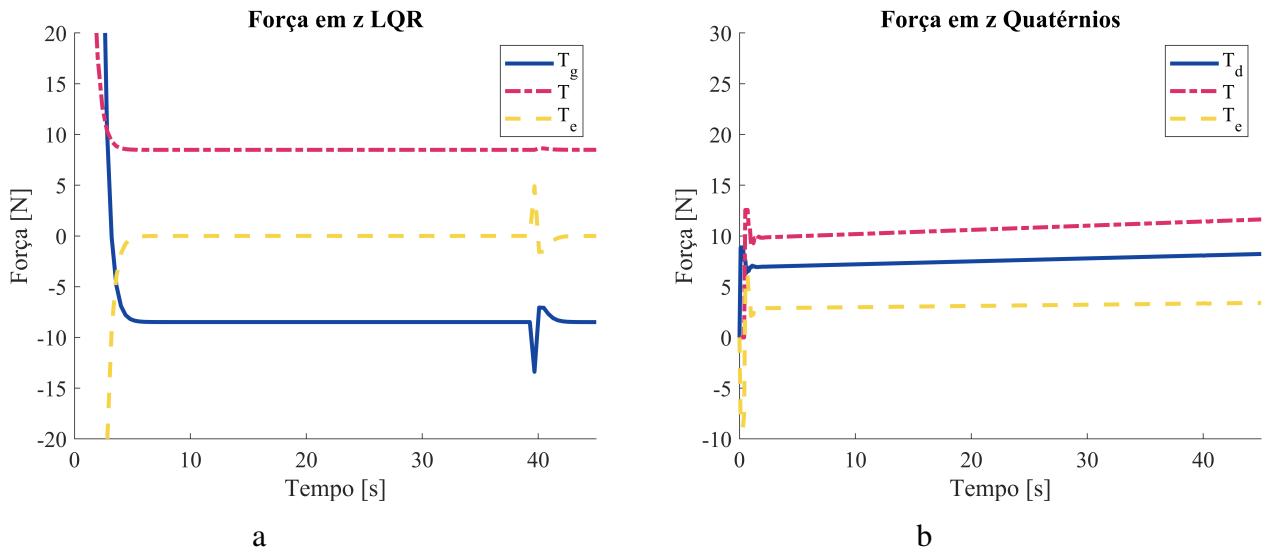


Figura 11 – Sobressinal de Força de (a) LQR y (b) Quatérnios

Na Figura 11 é apresentado um zoom das forças, para o LQR é possível observar que ocorre um sobressinal, de 5 newtons, aos 40 segundos devido à mudança de trajetória (de elevação para espiral). Para os quatérnios, o sobressinal ocorre no início da trajetória, e esse sobressinal também é de 5 newtons.

Além disso, evidencia-se que o erro para o controlador LQR é praticamente 0 durante quase toda a trajetória. Para ambas as figuras, a linha azul representa a força desejada, a linha vermelha a força alcançada pelo drone, e a linha amarela representa o erro dessa força.

5.0.2 Resposta Torques

Por outro lado, é apresentado apenas o erro dos torques, e não seus componentes, pois é suficiente para observar o quanto bem o controlador consegue realizar as rotações enquanto gira.

Vamos revisar aqui dois aspectos: primeiro, para quatérnios, o torque em τ_z tem um comportamento diferente (*diagonal*), e segundo, este possui sobressinal apenas no início do trajeto, inferindo que este controlador permite uma melhor resposta a mudanças repentinas de direção.

Nesta seção, são apresentados os erros dos torques de cada controlador à esquerda. À direita, é exibido um aumento na região inicial da curva, correspondente aos primeiros 50 segundos, para analisar os sobresinais em cada caso. Vale lembrar que cada torque τ_x, τ_y, τ_z corresponde aos torques aplicados aos movimentos de roll, pitch e yaw (ϕ, θ, ψ), respectivamente.

O sobre-sinal do torque no quatérnio aos 40 segundos é superior ao do controlador LQR. Vejamos que o erro da componente em z da Figura 12 é nulo ao longo da trajetória, por outro lado a Figura 13, tem uma componente diagonal, isto devido ao torque que está sendo produzido,

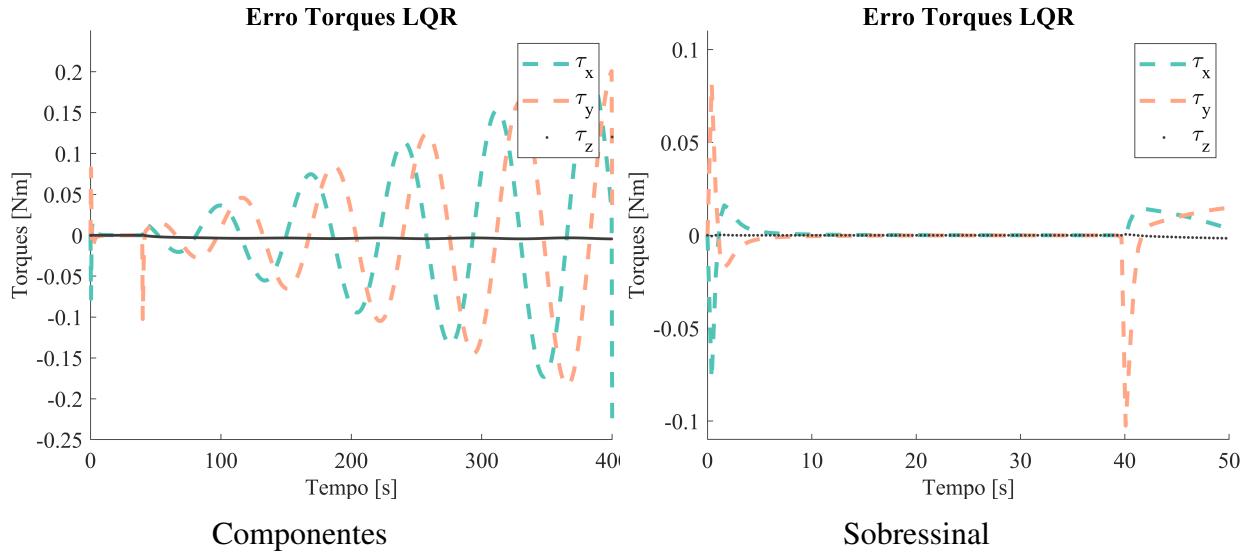


Figura 12 – Erro e Sobressinal de Torque LQR

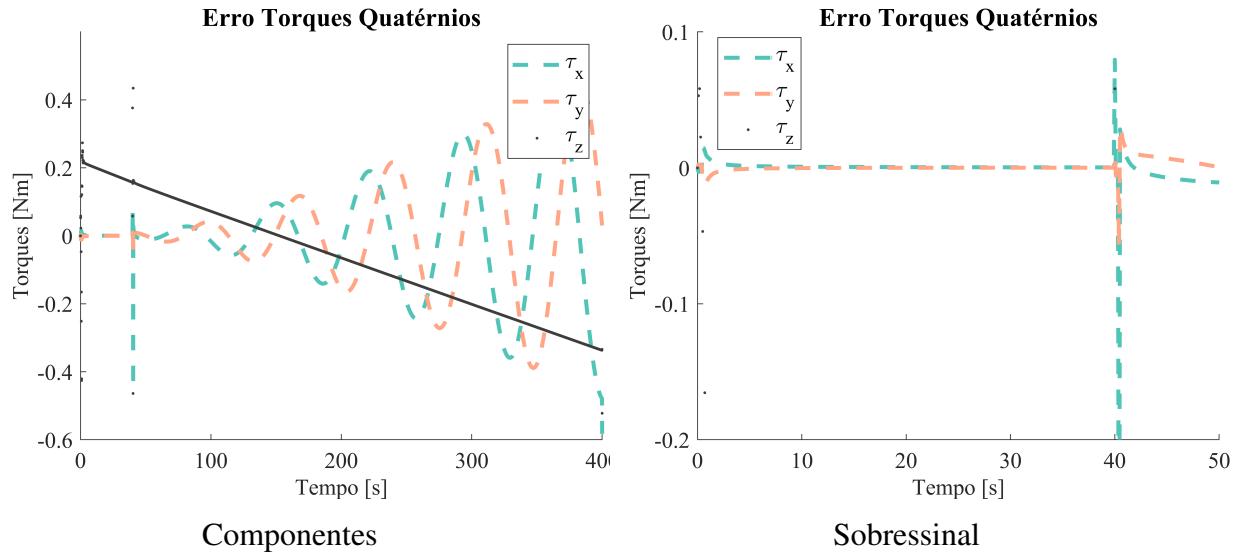


Figura 13 – Erro e Sobressinal de Torque Quatérnios

fazendo com que o drone tenha movimento de Yaw, obtemos que o seu erro é de 0.2 Nm no início da trajetória, nulo aos 200 segundos e 0.3 Nm aos 400 segundos.

5.1 Acompanhamento de Trajetórias

Agora apresentamos as trajetórias seguidas por cada controlador, com isso podemos evidenciar como é sua resposta às posições desejadas. Se os observarmos de uma vista lateral e superior, poderemos facilmente evidenciar seu comportamento em relação a cada eixo.

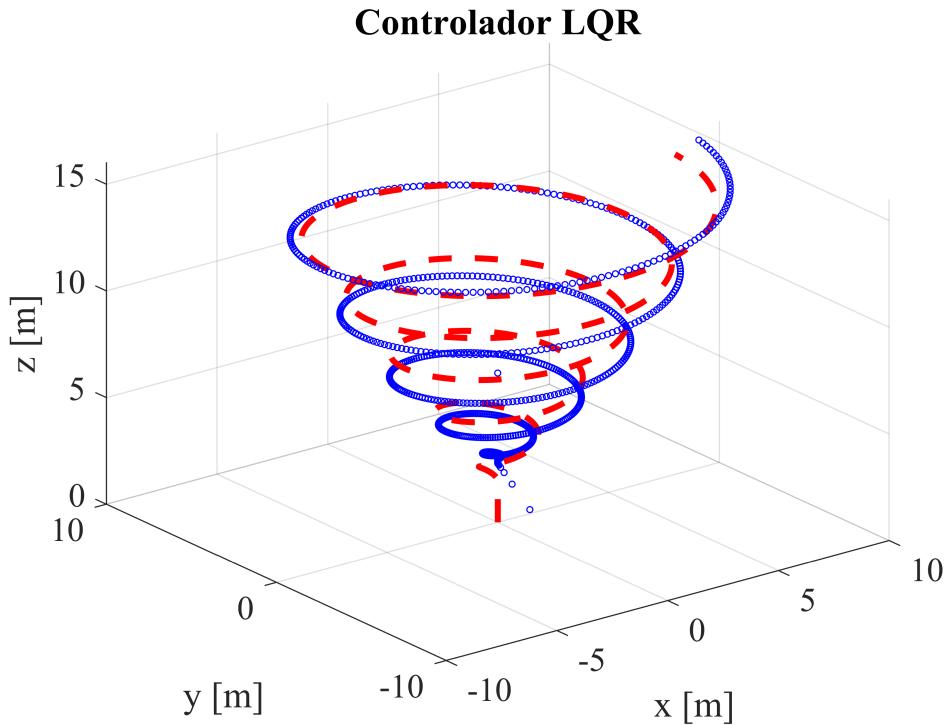


Figura 14 – Acompanhamento Trajetória LQR

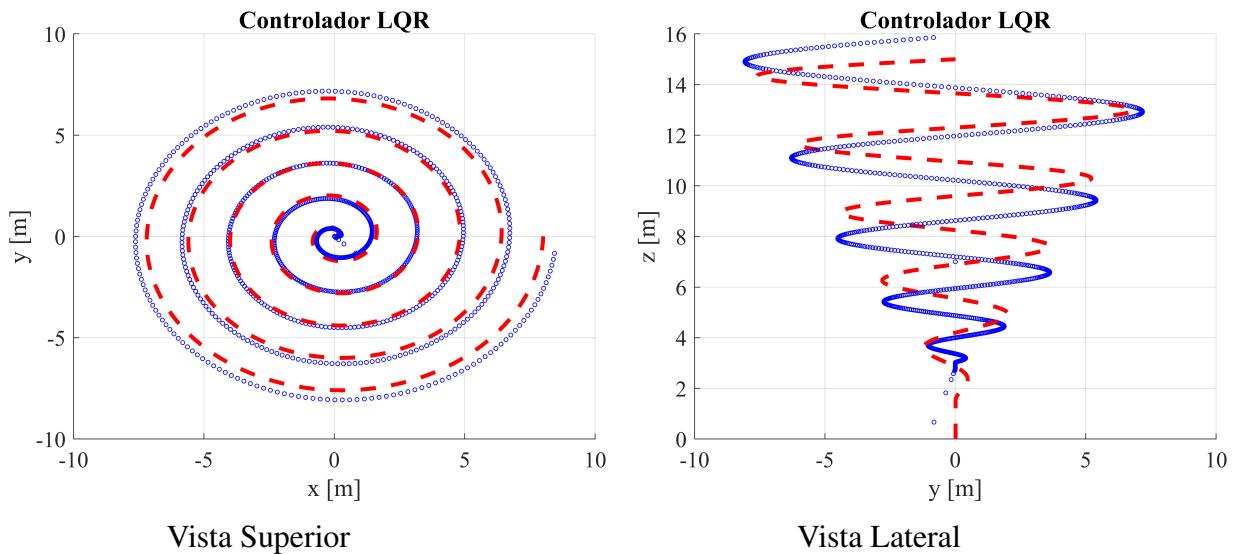


Figura 15 – Vistas da resposta do Controlador LQR

5.1.1 Quatérnios

Para os erros de quatérnios, além de visualizar o erro de cada componente do quatérnio desejado \mathbf{q}_d e do utilizado \mathbf{q} , apresentados na Figura 18, também é possível observar que a soma dos quatro componentes do quatérnio, em um mesmo instante de tempo, resulta em 1.

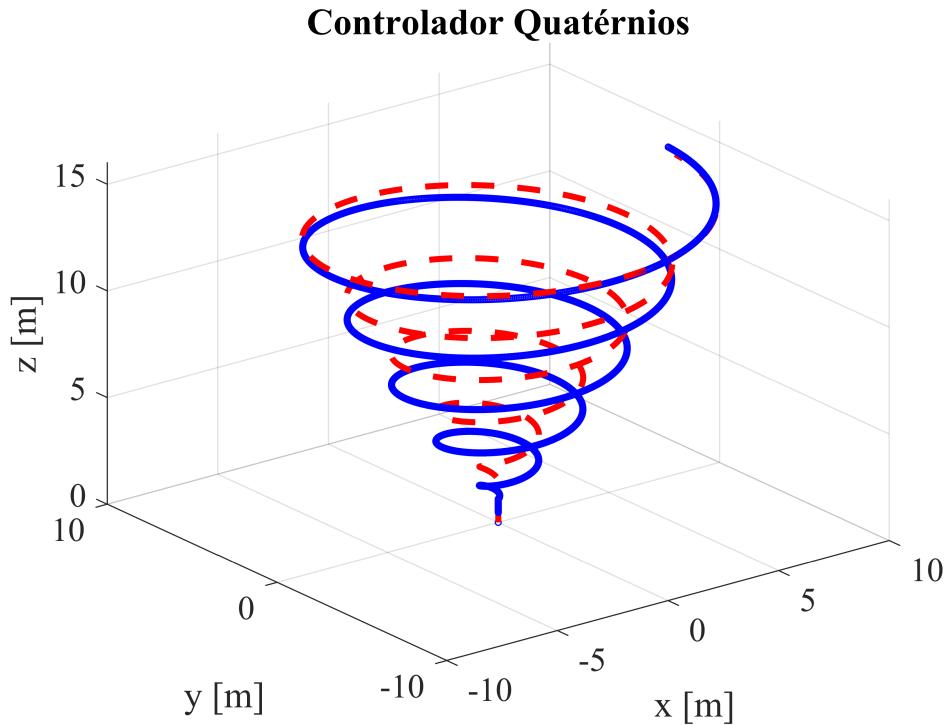


Figura 16 – Acompanhamento Trajetória Quatérnios

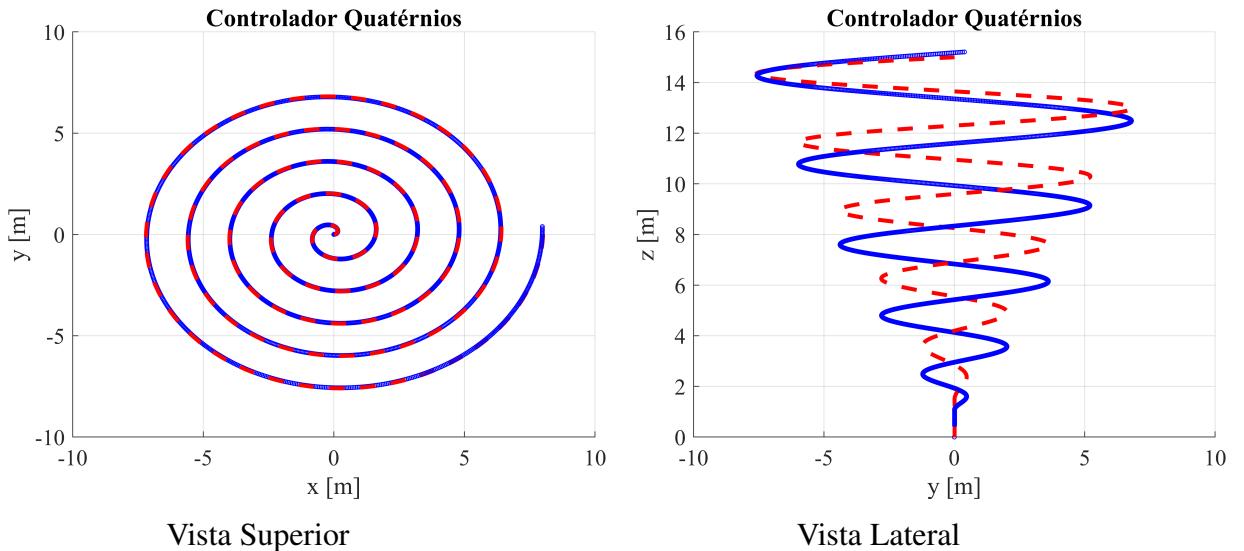


Figura 17 – Vistas da resposta do Controlador Quatérnios

5.2 Comparação de Controladores

Observemos que a força em z do controlador LQR e de Quatérnios, Figura 10 (a) e (b), respectivamente, são distintas. Isso ocorre porque a dinâmica do primeiro não permite mudanças de força em z , implicando que o drone se move de um ponto a outro em *hovering*, ou seja, o plano do drone está quase sempre paralelo ao solo, já que para este caso foram utilizados ângulos pequenos para as rotações.

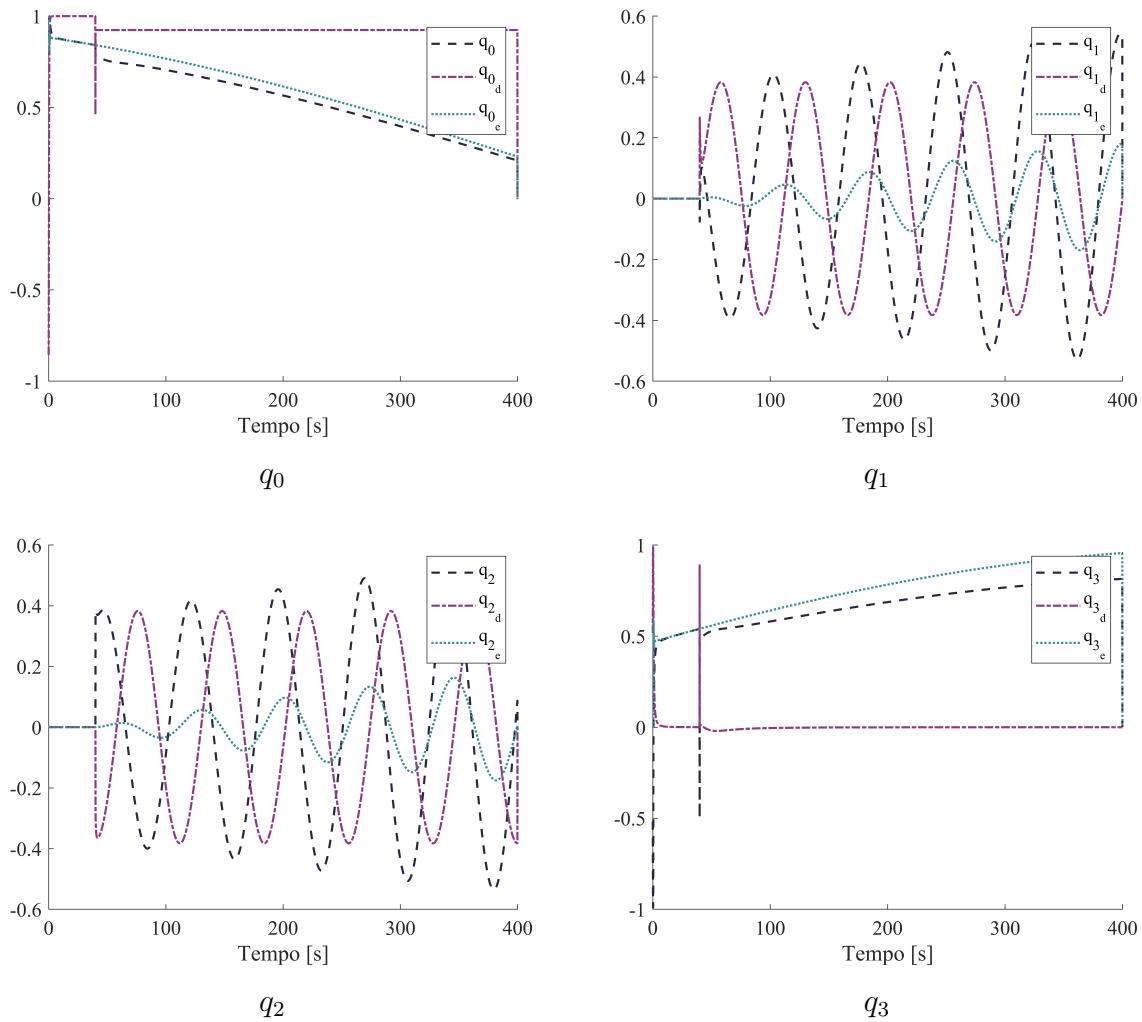


Figura 18 – Resposta del quatérnio de rotação \mathbf{q}

A dinâmica do segundo demonstra que o drone está acelerando, ao ter um ângulo de pitch e roll em cada ponto. Isso exige uma força que permita resistir à força centrífuga fictícia, provocada pela rotação. Tal força é proporcional ao raio, e como neste caso ela aumenta linearmente, equação 3.1, sua força também aumentará.

Além disso, observa-se que o torque em ψ tem o mesmo efeito, mas isso ocorre porque o drone também está girando sobre seu eixo.

O controlador por quatérnios possui uma precisão muito alta para seguir rotações, mas falha no momento de se deslocar no espaço. Vendo a Figura 17, vemos que sua posição vertical possui um erro maior no plano z , mas segue quase exatamente cada ponto enquanto gira. Por outro lado, o controlado por LQR tem uma vantagem na posição, mas suas rotações tendem a se afastar cada vez mais das desejadas.

Ainda assim, o controlador por Quatérnios parece se ajustar à trajetória após um tempo. Visto de uma perspectiva lateral, depois dos 12 metros, o erro da posição diminui drasticamente, podendo-se dizer que, aos 14 metros, o erro é zero.

Vejamos mais uma coisa, a densidade de pontos, *linha azul*, para o controlador LQR é menor do que para o controlador de Quaternios. Ambos os controladores foram simulados sob as mesmas condições, com deltas de tempo iguais e com o mesmo tempo final. Isso indica continuidade, pois os erros provenientes dos ângulos de Euler têm seu efeito sobre este parâmetro. É como se estivéssemos parados sobre um desses pontos, dando um passo sabendo que devemos estar em uma posição desejada, mas percebendo que, um instante depois, estamos um pouco mais à direita. Com a distância e os tempos constantes, só há uma razão para estarmos mais à direita: o nosso ângulo antes de dar o passo não foi o correto.

Controladores convencionais se esforçam para corrigir aquele *passo mal dado*, e é por isso que os quatérnios têm uma vantagem. A probabilidade de errar o passo usando quatérnios é menor.

Por fim, quero que percebamos que ambos os controladores apresentam um erro crescente em relação aos torques, devido ao fato de que, para essa trajetória específica, ela é uma sinal rampa e a arquitetura do controlador é de erro estático de posição. Se fosse desejado reduzir o erro dos torques, seria necessário projetar um controlador com uma arquitetura de retroalimentação que reduza erros estáticos de velocidade²⁵.

6 CONCLUSÕES

Os controladores aqui projetados são controladores do tipo 0, que apenas conseguem seguir sinais cujo erro estacionário é de posição, ou seja, sinais de passo. Entende-se então que a comparação dos controladores foi realizada sob as mesmas condições.

Conclui-se que um modelo por Quaternios é muito mais simples, e seu controlador associado permite respostas equivalentes ou, em alguns casos, melhores que um controlador otimizado com base em um modelo convencional.

Comparando as respostas de cada controlador, podemos dizer que o controlador por Quaternios possui uma maior precisão ao determinar rotações no drone, evitando descontinuidades, não caindo em singularidades e reduzindo a quantidade de cálculos exigidos. Assim, esse tipo de modelo e os controladores associados são melhores que os convencionais. Uma das principais razões para isso é o fato de separar as rotações de Euler.

A natureza sequencial dos ângulos de Euler não permite mudanças repentinas de rotação nem posições específicas; por exemplo, levar o drone perto de ângulos de roll perpendiculares ao solo provoca erros no seu cálculo.

O modelo por quatérnios possui uma melhor resposta a perturbações, o que é evidenciado pelos seus picos de sobressinal baixos. Outra vantagem é que os quatérnios exigem uma capacidade computacional até 40% menor²⁶, o que é fundamental em sistemas que requerem uma resposta rápida a mudanças no tempo, ou aqueles que estão realizando outros processos de forma paralela.

Os drones são completamente autônomos, e essa nova abordagem permite aproveitar recursos utilizados no cálculo de posições e velocidades, canalizando-os para outras questões. Sob uma perspectiva operacional, isso poderia permitir a inclusão de redundâncias ou a criação de algoritmos de segurança mais eficientes.

Conclui-se que a principal razão dessas vantagens é que os quatérnios possuem uma elegância matemática e uma simplicidade que permite modelar dinâmicas complexas de forma compacta. Além disso, são entidades cujo mapeamento é simples; ou seja, ir de um quatérnio a um vetor e vice-versa é mais fácil, ao contrário de decompor matrizes ou calcular ângulos. Devido à sua natureza, obtemos respostas contínuas e naturais do sistema, aumentamos os intervalos de operação e melhoramos a resposta do drone a rotações.

Os quatérnios possuem uma interpolação mais suave entre as orientações realizadas, o que faz com que a densidade de pontos seja mais uniforme ao longo da trajetória. Embora sua resposta à translação seja mais lenta em comparação com o controlador LQR.

Um problema de usar quatérnios é que, ao trabalhar sob a condição de um versor, é

difícil projetar métodos de otimização porque estamos restritos a uma magnitude unitária. Outra limitação é que, se quisermos realizar uma rotação completa¹, não é possível fazê-lo *de uma só vez*, como com os ângulos de Euler. Em videogames, por exemplo, esse problema é resolvido rotacionando o quatérnio de $0 \rightarrow \alpha$ e depois de $\alpha \rightarrow 2\pi$.

6.1 Trabalhos Futuros

Como trabalhos futuros, seria interessante considerar as forças dos motores e a gravidade de uma maneira mais realista, considerando suas componentes em x e y , e assim entender os efeitos que isso tem na dinâmica do drone.

Utilizar métodos de controle otimizados aplicados a um modelo por quatérnios, ou seja, combinar o modelo por quatérnios com arquiteturas mais complexas, permitiria observar sua resposta a tipos de entradas mais complexas. Outro ponto interessante seria utilizar o modelo por quatérnios em controladores que apliquem teorias de controle diferentes (LQR, PID, MC)^{27 28}, para encontrar o mais otimizado. O poder de generalizar o uso desses para ser aplicados em outros sistemas que utilizam rotações.

Modelar o drone por seu Lagrangiano, utilizando rotações por quatérnios, algo que poderia ser denominado *Quatérnios-Lagrange*, e assim determinar o drone por suas energias. O enfoque para linearizar novamente o modelo seria optar por usar séries de Taylor para expandir as equações dinâmicas.

Além disso, buscar separar o controlador por quatérnios de uma trajetória, para aplicar algum método de interpolação otimizado, garantindo que a rotação seja a menor *i.e.*, uma rotação pode ser de α ou de $2\pi - \alpha$, com os quatérnios ocorre o mesmo, e é vital garantir que o segundo caso não aconteça.

Realizar verificações experimentais em bancadas de teste, ou com modelos físicos, para verificar a resposta do modelo sob condições reais. Considerar efeitos aerodinâmicos como o *blade flapping*, o efeito de estela ou o arrasto induzido⁷, junto com os efeitos por erros de alinhamento dos motores, permitiria aumentar a precisão do drone. Também modelar efeitos giroscópicos por quatérnios, devido ao fato de que a ordem de grandeza destes é considerável.

Seria interessante desacoplar os motores, modelar cada rotor separadamente, como eixos que atuam sobre um objeto e acoplá-los novamente para obter uma dinâmica. Ou usar controladores separados, que controlem cada motor e depois um controlador maior que sirva como *chef d'orchestre*. Revisar uma teoria de controle, onde as condições se baseiem puramente na multiplicação de quatérnios, talvez usando mapas de Diagramas de Voronói.

^I Ou seja, uma rotação de 2π radianos

REFERÊNCIAS

- 1 BRITS, L. **CC BY 3.0.** <https://commons.wikimedia.org/wiki/File:Eulerangles.svg>. Vía Wikimedia Commons.
- 2 MARINO108LFS. **CC BY-SA 4.0.** https://commons.wikimedia.org/wiki/File:Propeller_blade_BET.svg. Vía Wikimedia Commons.
- 3 ASIGNACION A., J.; SATOSHI, S. Historical and current landscapes of autonomous quadrotor control: An early-career researchers' guide. **Drones**, v. 8, n. 3, p. 72, 2024.
- 4 BOUABDALLAH, S. Design and control of an indoor micro quadrotor. In: **Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2004. v. 5, p. 4393–4398.
- 5 REGULAMENTO Brasileiro de Aviação Civil (RBAC 91). 2017. Agência Nacional de Aviação Civil - ANAC. Disponível em: <https://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac>.
- 6 GONZALEZ, H. A. **Robust tracking of dynamic targets with aerial vehicles using quaternion-based techniques.** abr. 2019. Tese (Theses) — Université de Technologie de Compiègne, abr. 2019. Disponível em: <https://theses.hal.science/tel-02155857>.
- 7 MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. **IEEE Robotics and Automation Magazine**, v. 19, n. 3, p. 20–32, 2012.
- 8 ROJAS, J. S. Diseño de un controlador robusto basado en lmi aplicado en un quadrotor. **IngeCuC**, v. 18, n. 1, p. 38–50, mar 2022. Disponível em: <https://revistascientificas.cuc.edu.co/ingecuc/article/view/2911>.
- 9 FRANZETTI, I. D. D. M. de P. *et al.* Model-in-the-loop testing of control systems and path planner algorithms for quadrotor uavs. In: **2020 International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.: s.n.], 2020. p. 1809–1818.
- 10 ZULU, A.; JOHN, S. A review of control algorithms for autonomous quadrotors. **arXiv**, 2016. ArXiv:1602.02622.
- 11 NICOL, C.; MACNAB, C. J. B.; RAMIREZ-SERRANO, A. Robust neural network control of a quadrotor helicopter. In: **Proceedings of the Canadian Conference on Electrical and Computer Engineering**. Niagara Falls, Canada: [S.l.: s.n.], 2008. p. 1233–1237.
- 12 RUBIO, J. de J. *et al.* Comparison of two quadrotor dynamic models. **IEEE Latin America Transactions**, v. 12, n. 4, p. 531–537, 2014.
- 13 HAMEL, T. *et al.* Dynamic modelling and configuration stabilization for an x4-flyer. **IFAC Proceedings Volumes**, v. 35, n. 1, p. 217–222, 2002. ISSN 1474-6670. 15th IFAC World Congress. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1474667015392697>.
- 14 KUIPERS, J. B. **Quaternions and Rotation Sequences**. Princeton: Princeton University Press, 1999. v. 66.

- 15 SPRING, K. W. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review. **Mechanism and Machine Theory**, v. 21, n. 5, p. 365–373, 1986. ISSN 0094-114X. Disponível em: <https://www.sciencedirect.com/science/article/pii/0094114X86900844>.
- 16 GLAUERT, H. **The Elements of Aerofoil and Airscrew Theory**. 2nd. ed. Cambridge: Cambridge University Press, 1986.
- 17 PROUTY, R. W. **Helicopter Performance, Stability and Control**. [S.l.: s.n.]: Krieger Publishing Company, 2002.
- 18 GOLDSTEIN, H.; POOLE, C.; SAFKO, J. **Classical Mechanics**. 3rd. ed. [S.l.: s.n.]: Pearson, New International Edition, 2002.
- 19 WALLE, M.; MULATU, A. Dynamic modeling and trajectory tracking control of hexacopter using sliding mode controller a master's thesis by saba. *In: .* [S.l.: s.n.], 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:201021955>.
- 20 NEKOO, S. R.; ACOSTA, J. Á.; OLLERO, A. Quaternion-based state-dependent differential riccati equation for quadrotor drones: Regulation control problem in aerobatic flight. **Robotica**, v. 40, n. 9, p. 3120–3135, 2022.
- 21 DIEBEL, J. Representing attitude : Euler angles , unit quaternions , and rotation vectors. *In: .* [S.l.: s.n.], 2006. Disponível em: <https://api.semanticscholar.org/CorpusID:16450526>.
- 22 FRESK, E.; NIKOLAKOPOULOS, G. Full quaternion based attitude control for a quadrotor. *In: 2013 European Control Conference (ECC)*. [S.l.: s.n.], 2013. p. 3864–3869.
- 23 DHEWA, O. A. *et al.* Enhancement of stability on autonomous waypoint mission of quadrotor using lqr integrator control. **IIUM Engineering Journal**, v. 23, n. 1, p. 129–158, Jan. 2022. Disponível em: <https://journals.iium.edu.my/ejournal/index.php/iiumej/article/view/1803>.
- 24 SHAUQEE, P. R. M. N.; SUHADIS, N. M. An effective proportional-double derivative-linear quadratic regulator controller for quadcopter attitude and altitude control. **Automatika**, Taylor and Francis, v. 62, n. 3-4, p. 415–433, 2021. Disponível em: <https://doi.org/10.1080/00051144.2021.1981527>.
- 25 OGATA, K. **Modern Control Engineering**. 5. ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
- 26 ICKES, B. A new method for performing digital control system attitude computations using quaternions. *In: .* [S.l.: s.n.], 1968. Disponível em: <https://api.semanticscholar.org/CorpusID:62511008>.
- 27 OKASHA, M.; KRALEV, J.; ISLAM, M. Design and experimental comparison of pid, lqr and mpc stabilizing controllers for parrot mambo mini-drone. **Aerospace**, v. 9, n. 6, 2022. ISSN 2226-4310. Disponível em: <https://www.mdpi.com/2226-4310/9/6/298>.
- 28 ORTIZ, A.; GARCIA-NIETO, S.; SIMARRO, R. Comparative study of optimal multivariable lqr and mpc controllers for unmanned combat air systems in trajectory tracking. **Electronics**, v. 10, n. 3, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/3/331>.
- 29 OSWALD, N. **Hurwitz's Lectures on the Number Theory of Quaternions**. Berlin: EMS Press, 2023. (Heritage of European Mathematics).

ANEXOS

ANEXO A – EFEITO CORIOLIS

A expansão da equação 2.7 para obter o efeito Coriolis $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ é obtida reescrevendo a equação 2.9 levando em consideração a regra da cadeia para uma variável independente A.1. Se $z = f(x, y)$ onde $x = g(t)$ e $y = h(t)$, a regra da cadeia é expressa como:

$$\frac{\partial z}{\partial t} = \frac{\partial f}{\partial x} \frac{\partial g}{\partial t} + \frac{\partial f}{\partial y} \frac{\partial h}{\partial t} \quad (\text{A.1})$$

Obtendo que a dinâmica de atitude é:

$$\frac{d}{dt} \left(\vec{\Omega}^T I \frac{\partial \vec{\Omega}}{\partial \dot{\boldsymbol{\eta}}} \right) - \vec{\Omega}^T I \frac{\partial \vec{\Omega}}{\partial \boldsymbol{\eta}} = \tau, \quad \vec{\Omega}^T I \frac{\partial \vec{\Omega}}{\partial \dot{\boldsymbol{\eta}}} = [b_1 \ b_2 \ b_3]$$

$$\begin{aligned} b_1 &= I_{xx}(S_\theta^2 \dot{\psi} - S_\theta \dot{\phi}) + I_{yy}(C_\theta^2 S_\phi^2 \dot{\psi} + C_\theta C_\phi S_\phi \dot{\theta}) + I_{zz}(C_\theta^2 C_\phi^2 \dot{\psi} - C_\theta C_\phi S_\phi \dot{\theta}) \\ b_2 &= I_{yy}(C_\theta C_\phi S_\phi \dot{\psi} + C_\phi^2 \dot{\theta}) - I_{zz}(C_\theta C_\phi S_\phi \dot{\psi} + S_\phi^2 \dot{\theta}) \\ b_3 &= -I_{xx}(S_\theta \dot{\psi} - \dot{\phi}) \end{aligned}$$

$$\begin{aligned} \dot{b}_1 &= I_{xx}(2S_\theta \dot{S}_\theta \dot{\psi} + S_\theta^2 \ddot{\psi} - \dot{S}_\theta \dot{\phi} - S_\theta \ddot{\phi}) + \\ &\quad I_{yy}(2C_\theta \dot{C}_\theta S_\phi^2 \dot{\psi} + C_\theta^2 \dot{S}_\phi S_\phi \dot{\psi} + C_\theta^2 S_\phi^2 \ddot{\psi} + C_\phi S_\phi \dot{C}_\theta \dot{\theta} + C_\theta C_\phi \dot{S}_\phi \dot{\theta} + C_\theta C_\phi S_\phi \ddot{\theta}) + \\ &\quad I_{zz}(2C_\theta \dot{C}_\theta C_\phi^2 \dot{\psi} + C_\theta^2 \dot{C}_\phi C_\phi \dot{\psi} + C_\theta^2 C_\phi^2 \ddot{\psi} - \dot{C}_\theta C_\phi S_\phi \dot{\theta} - C_\theta \dot{C}_\phi S_\phi \dot{\theta} - C_\theta C_\phi \dot{S}_\phi \dot{\theta} - C_\theta C_\phi S_\phi \ddot{\theta}) \\ \dot{b}_2 &= I_{yy}(C_\phi S_\phi \dot{C}_\theta \dot{\psi} + C_\theta \dot{C}_\phi S_\phi \dot{\psi} + C_\theta C_\phi \dot{S}_\phi \dot{\psi} + C_\theta C_\phi S_\phi \dot{\psi} + 2C_\phi \dot{C}_\phi \dot{\theta} + C_\phi^2 \ddot{\theta}) - \\ &\quad I_{zz}(C_\phi S_\phi \dot{C}_\theta \dot{\psi} + C_\theta \dot{C}_\phi S_\phi \dot{\psi} + C_\theta C_\phi \dot{S}_\phi \dot{\psi} + C_\theta C_\phi S_\phi \ddot{\psi} + 2S_\phi \dot{S}_\phi \dot{\theta} + S_\phi^2 \ddot{\theta}) \\ \dot{b}_3 &= -I_{xx}(\dot{S}_\theta \dot{\psi} + S_\theta \ddot{\psi} - \ddot{\phi}) \end{aligned}$$

Usando,

$$\frac{\partial \Omega}{\partial \boldsymbol{\eta}} = \begin{bmatrix} 0 & -\psi C_\theta & 0 \\ 0 & -\psi C_\phi & -\dot{\psi} + \psi C_\theta C_\phi \\ 0 & -\psi C_\theta - C_\phi & -C_\theta \end{bmatrix}, \quad \Omega^T I \frac{\partial \Omega}{\partial \boldsymbol{\eta}} = [h_1 \ h_2 \ h_3]$$

$$h_1 = 0,$$

$$\begin{aligned} h_2 &= -\dot{\psi} \left[I_{xx}(C_\theta S_\theta \dot{\psi} - C_\theta \dot{\phi}) + I_{yy}(-C_\theta S_\theta S_\phi^2 \dot{\psi} - C_\phi S_\theta S_\phi \dot{\theta}) + I_{zz}(-C_\theta C_\phi^2 S_\theta \dot{\psi} + C_\phi S_\theta S_\phi \dot{\theta}) \right] \\ h_3 &= (I_{yy} - I_{zz})(C_\theta^2 C_\phi S_\phi \dot{\psi}^2 + C_\theta C_\phi^2 \dot{\psi} \dot{\theta} - C_\theta S_\phi^2 \dot{\psi} \dot{\theta} - C_\phi S_\phi \dot{\theta}^2). \end{aligned}$$

Obtendo assim as componentes do vetor torque, em função dos ângulos.

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \dot{b}_1 - h_1 \\ \dot{b}_2 - h_2 \\ \dot{b}_3 - h_3 \end{bmatrix}$$

Para a matriz inercial em função dos ângulos de Euler \mathbb{I} , obtemos:

$$\mathbb{I}(\eta) = \begin{bmatrix} I_{xx}S_\theta^2 + I_{yy}S_\phi^2C_\theta^2 + I_{zz}C_\phi^2C_\theta^2 & I_{yy}S_\phi C_\phi C_\theta - I_{zz}S_\phi C_\phi C_\theta & -I_{xx}S_\theta \\ I_{yy}S_\phi C_\phi C_\theta - I_{zz}S_\phi C_\phi C_\theta & I_{yy}C_\phi^2 + I_{zz}S_\phi^2 & 0 \\ -I_{xx}S_\theta & 0 & I_{xx} \end{bmatrix}$$

E para o efeito Coriolis o resultado é:

$$C(\dot{\eta}, \eta) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix},$$

$$\begin{aligned} c_{11} &= 2I_{xx}S_\theta C_\theta \dot{\theta} - 2I_{yy}S_\phi^2 S_\theta C_\theta \dot{\theta} + 2I_{yy}S_\phi C_\phi C_\theta^2 \dot{\phi} - 2I_{zz}S_\phi C_\phi C_\theta^2 \dot{\phi} - 2I_{zz}S_\theta C_\phi^2 C_\theta \dot{\theta} \\ c_{12} &= -I_{yy}S_\phi^2 C_\theta \dot{\phi} - I_{yy}S_\phi S_\theta C_\phi \dot{\theta} + I_{yy}C_\phi^2 C_\theta \dot{\phi} + I_{zz}S_\phi^2 C_\theta \dot{\phi} + I_{zz}S_\phi S_\theta C_\phi \dot{\theta} - I_{zz}C_\phi^2 C_\theta \dot{\phi} \\ c_{13} &= -I_{xx}C_\theta \dot{\theta} \\ c_{21} &= -(2S_\theta C_\theta \dot{\psi} - C_\theta \dot{\phi})I_{xx} - (S_\phi S_\theta C_\phi \dot{\theta} - 2S_\theta C_\phi^2 C_\theta \dot{\psi})I_{zz} - (-2S_\phi^2 S_\theta C_\theta \dot{\psi} - S_\phi S_\theta C_\phi \dot{\theta})I_{yy} - \\ &\quad I_{yy}S_\phi^2 C_\theta \dot{\phi} - I_{yy}S_\phi S_\theta C_\phi \dot{\theta} + I_{yy}C_\phi^2 C_\theta \dot{\phi} + I_{zz}S_\phi^2 C_\theta \dot{\phi} + I_{zz}S_\phi S_\theta C_\phi \dot{\theta} - I_{zz}C_\phi^2 C_\theta \dot{\phi} \\ c_{22} &= (I_{yy} - I_{zz})S_\phi S_\theta C_\phi \dot{\psi} - 2I_{yy}S_\phi C_\phi \dot{\phi} + 2I_{zz}S_\phi C_\phi \dot{\phi} \\ c_{23} &= I_{xx}C_\theta \dot{\psi} \\ c_{31} &= -((-S_\phi^2 \dot{\theta} + 2S_\phi C_\phi C_\theta \dot{\psi} + C_\phi^2 \dot{\theta})I_{yy} + (S_\phi^2 \dot{\theta} - 2S_\phi C_\phi C_\theta \dot{\psi} - C_\phi^2 \dot{\theta})I_{zz})C_\theta - I_{xx}C_\theta \dot{\theta} \\ c_{32} &= -(-S_\phi^2 C_\theta \dot{\psi} - 2S_\phi C_\phi \dot{\theta} + C_\phi^2 C_\theta \dot{\psi})I_{yy} - (S_\phi^2 C_\theta \dot{\psi} + 2S_\phi C_\phi \dot{\theta} - C_\phi^2 C_\theta \dot{\psi})I_{zz} \\ c_{33} &= 0 \end{aligned}$$

ANEXO B – ÁLGEBRA DE QUATERNIONS

Este sistema numérico possui propriedades específicas no momento de ser operado, cumprindo a condição 2.11, essas operações são fundamentais no momento do cálculo. Para uma revisão mais detalhada sobre o assunto, pode-se consultar.²⁹

Double Cover

Uma característica própria dos quaternions é que todo quaternion possui um oposto negativo, ou seja, cada rotação em $\in SO(3)$ possui dois quaternions que a representam. Portanto, sua rotação é equivalente, apenas com uma rotação de 2π em relação a \hat{h} . Para dois quaternions:

$$\mathbf{q}_1 := \cos(\lambda/2) + \vec{u} \sin(\lambda/2), \quad (\text{B.1})$$

$$\mathbf{q}_2 := \cos((\lambda + 2\pi)/2) + \vec{u} \sin((\lambda + 2\pi)/2) \quad (\text{B.2})$$

\mathbf{q}_1 e \mathbf{q}_2 representam a mesma rotação, mas a segunda está sendo aplicada em \hat{h} , como $\mathbf{q}_1 = -\mathbf{q}_2$.

Soma

Dado dois quaternions arbitrários, \mathbf{q} e \mathbf{a} , sua soma é definida como:

$$\mathbf{q} + \mathbf{a} := q_0 + a_0 + \vec{q} + \vec{a} \quad (\text{B.3})$$

Esta é comutativa, o que significa $\mathbf{q} + \mathbf{a} = \mathbf{a} + \mathbf{q}$. A subtração não possui as mesmas propriedades, mas seu cálculo é realizado componente a componente, como na soma.

Produto

Esta é a definição mais importante, pois a partir dela é possível demonstrar as outras operações. É uma operação não comutativa, ou seja, $\mathbf{q} \circ \mathbf{a} \neq \mathbf{a} \circ \mathbf{q}$, sendo definida por:

$$\mathbf{q} \circ \mathbf{a} := (q_0 a_0 - \vec{q} \cdot \vec{a}) + (q_0 \vec{a} + a_0 \vec{q} + \vec{q} \times \vec{a}) \quad (\text{B.4})$$

É importante ressaltar que ela não é comutativa, ou seja, $\mathbf{q} \circ \mathbf{a} \neq \mathbf{a} \circ \mathbf{q}$.

Norma

A norma em quaterniones é a mais estudada em matemática, pois determina várias propriedades no cálculo de quaterniones. Para um quaternión arbitrário, define-se sua norma como:

$$\begin{aligned}\|\mathbf{q}\|^2 &= \mathbf{q} \circ \mathbf{q}^* = q_0^2 + q_1^2 + q_2^2 + q_3^2, \\ \mathbf{q}^2 &= 2q_0\mathbf{q} - \|\mathbf{q}\|^2\end{aligned}\tag{B.5}$$

Conjugado

A definição é determinada por:

$$\mathbf{q}^* := q_0 - \vec{q},\tag{B.6}$$

$$(\mathbf{q} \circ \mathbf{r})^* = \mathbf{r}^* \circ \mathbf{q}^*\tag{B.7}$$

O conjugado de um *versor* representa uma rotação inversa em relação ao mesmo eixo \hat{h} .

Inversa

Para um quaternón não nulo, verifica-se que sua inversa será seu conjugado multiplicado pela inversa de sua norma.

$$\mathbf{q}^{-1} := \frac{\mathbf{q}^*}{\|\mathbf{q}\|},\tag{B.8}$$

$$\mathbf{q} \circ \mathbf{q}^{-1} = \mathbf{q}^{-1} \circ \mathbf{q} = 1 + [0 \ 0 \ 0]^T\tag{B.9}$$

Rotação de Vetor

Dado um vetor $\vec{v} \in \mathbb{R}^3$, em um referencial \mathcal{A} e sendo $\vec{v}_b \in \mathbb{R}^3$ o mesmo vetor em relação a um referencial \mathcal{B} , a transformação $\vec{v} \Rightarrow \vec{v}_b$ pode ser realizada da seguinte maneira:

$$\vec{v}_b = \mathbf{q}^{-1} \circ \vec{v} \circ \mathbf{q} = \mathbf{q}^* \circ \vec{v} \circ \mathbf{q}\tag{B.10}$$

Aqui \mathbf{q} é o quaternón que indica a rotação de \vec{v}_b em relação a \vec{v} , propriedades do quaternón.

Isso é feito dessa forma, pois assim é possível rotacionar o vetor em relação a um plano. Para mais informações, consulte.¹⁴

Derivada

Por regra da cadeia, podemos definir a derivada de um vetor posição $\dot{\vec{r}}_b$ em relação a \mathcal{A} , isolando obtemos:

$$\dot{\vec{r}}_b = \dot{\mathbf{q}}^{-1} \circ \vec{r} \circ \mathbf{q} + \mathbf{q}^{-1} \circ \dot{\vec{r}} \circ \mathbf{q} + \mathbf{q}^{-1} \circ \vec{r} \circ \dot{\mathbf{q}} = \dot{\mathbf{q}}^{-1} \circ \mathbf{q} \circ \vec{r}_b + \vec{r}_b \circ \dot{\mathbf{q}}^{-1} \circ \dot{\mathbf{q}} \quad (\text{B.11})$$

Definimos \mathbf{q} como um versor, então

$$\mathbf{q} \circ \mathbf{q}^{-1} = 1$$

Cuja derivada

$$\dot{\mathbf{q}}^{-1} \circ \mathbf{q} = -\mathbf{q}^{-1} \circ \dot{\mathbf{q}}$$

Substituindo em B.11 obtemos:

$$\dot{\vec{r}}_b = \vec{r}_b \circ \mathbf{q}^{-1} \circ \dot{\mathbf{q}} - \mathbf{q}^{-1} \circ \dot{\mathbf{q}} \circ \vec{r}_b = 2(\mathbf{q}^{-1} \circ \dot{\mathbf{q}}) \times \vec{r}_b \quad (\text{B.12})$$

Sabendo que a velocidade transacional pode ser escrita como, $\dot{\vec{r}}_b = \vec{\Omega} \times \vec{r}_b$, onde $\vec{\Omega}$ é a velocidade angular de \vec{r}_b .

$$\vec{\Omega} \times \vec{r}_b = 2(\mathbf{q}^{-1} \circ \dot{\mathbf{q}}) \times \vec{r}_b \quad (\text{B.13})$$

A derivada de um quaternión unitário em função de $\vec{\Omega}$, é então:

$$\begin{aligned} \vec{\Omega} &= 2(\mathbf{q}^{-1} \circ \dot{\mathbf{q}}) \\ \dot{\mathbf{q}} &= \frac{1}{2}\mathbf{q} \circ \vec{\Omega} \end{aligned} \quad (\text{B.14})$$