



UNIVERSIDADE FEDERAL DO PIAUÍ

CENTRO DE TECNOLOGIA

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PEDRO IAGO CARVALHO MARTINS

**ATITUDE EM UM DRONE QUADROTOR: IDENTIFICAÇÃO, SIMULAÇÃO E
CONTROLE**

TERESINA

2018

PEDRO IAGO CARVALHO MARTINS

ATITUDE EM UM DRONE QUADROTOR: IDENTIFICAÇÃO, SIMULAÇÃO E CONTROLE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica do
Centro de Tecnologia da Universidade Federal
do Piauí, como requisito parcial à obtenção do
grau de bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Otacílio da Mota
Almeida

TERESINA

2018

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Biblioteca Comunitária Jornalista Carlos Castello Branco
Divisão de Processos Técnicos

M386a Martins, Pedro Iago Carvalho.
Atitude em um drone quadrotor: identificação, simulação e
controle. / Pedro Iago Carvalho Martins. -- 2018.
77 f.

Trabalho de Conclusão de Curso (Graduação) –
Universidade Federal do Piauí, Centro de Tecnologia,
Bacharelado em Engenharia Elétrica, Teresina, 2018.

“Orientação: Prof. Dr. Otacílio da Mota Almeida”.

1. Controle - Drone quadrotor. 2. Drone quadrotor -
Identificação. 3. Drone quadrotor - Simulação. I. Título.

CDD: 621.317

PEDRO IAGO CARVALHO MARTINS

ATITUDE EM UM DRONE QUADROTOR: IDENTIFICAÇÃO, SIMULAÇÃO E CONTROLE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica do
Centro de Tecnologia da Universidade Federal
do Piauí, como requisito parcial à obtenção do
grau de bacharel em Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Otacílio da Mota Almeida (Orientador)
Universidade Federal do Piauí (UFPI)

Prof. Dr. José Maria Pires de Menezes Júnior
Universidade Federal do Piauí (UFPI)

Prof. Esp. Rômullo Randell Macedo Carvalho
Instituto Federal do Piauí (IFPI)

AGRADECIMENTOS

Ao Prof. Dr. Otacílio da Mota Almeida, pela confiança e os ensinamentos passados no PET, na monitoria, e no TCC. Ao Prof. Dr. José Maria Menezes, por me apresentar à pesquisa e me incentivar o pensamento lateral. Ao Engenheiro e Professor Vinicius, pela liderança, o trabalho em equipe, e a oportunidade de conhecer tecnologias tão importantes no ambiente hospitalar. Ao meu co-orientador e amigo Francisco Victor Esteves Lemos, por me ajudar sempre que precisei, e contribuir com a criação do meme que marcou a minha apresentação. À Alexandra Elbakyan, pela curiosidade e a perspectiva de um futuro melhor. A sua virtude também é compartilhada por todos aqueles que disponibilizam conhecimento, livre de barreiras. À Toby Fox, pela trilha sonora que me conectou à um mundo acolhedor, e facilitou a finalização deste trabalho. À minha família, por identificar as dificuldades, e me apoiar incondicionalmente; em alguns momentos quase não o terminei e somente com o apoio de vocês consegui continuar.

“Não existe trabalho ruim.
O ruim é ter que trabalhar.”

(Don Ramón)

RESUMO

Neste trabalho foram identificados componentes de hardware e de software essenciais ao controle de atitude de um drone quadrotor. O drone foi fixado por cordas nas extremidades de um eixo horizontal, de forma que a atitude ao seu redor fosse o único grau de liberdade. Uma aproximação do sistema de atitude, incluindo a atuação não-linear dos dois pares motor e hélice, foi identificada. Os parâmetros obtidos foram utilizados para construir uma simulação sem a presença das cordas. O simulador foi validado por comparação ao modelo de função de transferência, identificado por uma análise do teste do relé em malha fechada. Três controladores PID de diferentes estruturas foram projetados, e analisados através da simulação. A estrutura PDF-D apresentou a melhor performance, no critério do índice ITAE para o rastreamento de referência degrau. Por último, o simulador e o trabalho foram publicados na plataforma de colaboração GitHub.

Palavras-chave: Drone quadrotor. Identificação. Simulação. Controle.

ABSTRACT

In this work hardware and software components essential to the attitude control of a quadrotor drone were identified. The drone was tied by strings on the extremities of a horizontal axis, such that the attitude around it represented the single degree of freedom. An approximation of the attitude system, including the non-linear actuation of the two pairs of motor and propeller, was identified. The obtained parameters were utilized to construct a simulation without the presence of the strings. The simulator was validated by comparison to the transfer function model, identified by an analysis of the relay feedback test. Three PID controllers of different structures were designed, and analysed through the simulation. The PDF-D structure presented the best performance, in the criterium of the ITAE index for step reference tracking. By the end, the simulator and the computer code were published in the GitHub collaboration platform.

Keywords: Quadrotor drone. System identification. Simulation. Controller.

LISTA DE ILUSTRAÇÕES

Figura 1 – Quadrotor utilizado no trabalho.	14
Figura 2 – Quadrotor simulado no trabalho.	15
Figura 3 – Pinagem do MPU6050.	17
Figura 4 – Bateria LiPo 2200mAh 3s.	19
Figura 5 – Modelo de ESC utilizado.	20
Figura 6 – Motor, base e conector da hélice.	21
Figura 7 – Formas de tensão e corrente em um motor BLDC.	22
Figura 8 – Modelos de hélice 1045 utilizados.	23
Figura 9 – Fluxo de ar em uma hélice simulada.	24
Figura 10 – Modelo de microcontrolador utilizado.	25
Figura 11 – Console serial de calibração do MPU.	26
Figura 12 – Console serial de calibração do ESC.	27
Figura 13 – Resposta ao controle on-off.	28
Figura 14 – Simulador em pause.	31
Figura 15 – Sistema de testes, com a sustentação por cordas.	32
Figura 16 – Exemplos de resposta ao degrau.	36
Figura 17 – Exemplos de curvas de Nyquist.	37
Figura 18 – Diagrama de blocos do sistema motor-hélice	39
Figura 19 – Simplificação do diagrama de blocos do sistema motor-hélice	40
Figura 20 – Diagrama de blocos do sistema de balanço horizontal em roll	41
Figura 21 – Análise da função descritiva para um relé com histerese.	45
Figura 22 – Diagrama de conexão do relé por feedback de erro.	45
Figura 23 – Resultado do relé sem histerese no sistema teste.	46
Figura 24 – Resultado do relé sem histerese, com um filtro pente na referência.	47
Figura 25 – Resultado do relé com histerese, com um filtro pente na referência.	48
Figura 26 – Identificação de pontos na curva de Nyquist no sistema teste.	52
Figura 27 – Identificação de pontos na curva de Nyquist no sistema simulado.	52
Figura 28 – Validação do simulador pela resposta ao relé.	53
Figura 29 – Diagrama do controlador PID por feedback de erro.	55
Figura 30 – Resposta ao degrau com os controladores da Tabela 3.	56
Figura 31 – Diagrama do controle com dois graus de liberdade.	57

Figura 32 – Diagrama do controlador I-PD.	61
Figura 33 – Resposta ao degrau com os controladores da Tabela 5.	62
Figura 34 – Diagrama do controlador PDF-D.	63
Figura 35 – Propriedades previstas aos controladores da Tabela 6.	65
Figura 36 – Configuração do trem de pulsos em <i>pitch</i> e <i>roll</i>	66
Figura 37 – Comparaçāo da malha fechada pelos controladores propostos.	67
Figura 38 – Atuação e resposta para o controlador PD ZN.	68
Figura 39 – Atuação e resposta para o controlador I-PD.	69
Figura 40 – Atuação e resposta para o controlador PDF-D.	70
Figura 41 – Trem de degraus para o controlador PD ZN.	71
Figura 42 – Trem de degraus para o controlador I-PD.	71
Figura 43 – Trem de degraus para o controlador PDF-D.	72

LISTA DE TABELAS

Tabela 1 – Valores médios para quatro medições de $G(j\omega)$.	49
Tabela 2 – Método de Ziegler-Nichols.	55
Tabela 3 – Método de Ziegler-Nichols aplicado.	56
Tabela 4 – Otimização do ITAE para referência degrau.	60
Tabela 5 – Controladores I-PD por alocação de pólos dominantes.	62
Tabela 6 – Controladores PDF-D por alocação de pólos dominantes.	65

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Referencial do quadrotor	14
1.2	Objetivos	16
1.3	Estrutura do trabalho	16
2	COMPONENTES ESSENCIAIS	17
2.1	Unidade de processamento inercial MPU6050	17
2.1.1	<i>Algoritmo DMP e a frequência de amostragem</i>	18
2.2	Bateria LiPo 3s	19
2.3	Controlador de velocidade eletrônico ESC	20
2.4	Motor de corrente contínua sem escovas BLDC	21
2.5	Hélice	23
2.6	Microcontrolador Arduíno Mega 2560	25
2.6.1	<i>Calibração do MPU6050</i>	26
2.6.2	<i>Calibração do ESC</i>	27
2.7	Controle on-off	28
2.8	O simulador	29
2.8.1	<i>Unity3D</i>	29
2.8.2	<i>Arcadia e Clojure</i>	30
2.9	Balanço horizontal	32
3	FUNÇÕES DE TRANSFERÊNCIA	34
3.1	Exemplo no MATLAB	35
3.2	Realização dos controladores	38
3.3	Transformada de Laplace para o sistema motor-hélice	38
3.4	Transformada de Laplace para o sistema de atitude	41
3.5	Considerações de simulação	42
4	IDENTICAÇÃO DO SISTEMA	43
4.1	Método do relé em malha fechada	43
4.2	Considerações práticas	45
4.3	Medidas da curva de Nyquist	49
4.4	Aproximação estatística do modelo	50

4.5	Considerações de simulação	53
5	DESIGN DOS CONTROLADORES	54
5.1	O controlador PID	55
5.2	Método de sintonia de Ziegler Nichols	55
5.3	Método de sintonia por alocação de pólos	57
5.4	Pólos notáveis	59
5.4.1	<i>Pólos complexos dominantes</i>	59
5.4.2	<i>Pólos otimizados à redução do índice ITAE</i>	60
5.5	O controlador I-PD	61
5.6	O controlador PDF-D	63
5.7	Considerações de simulação	66
6	RESULTADOS DE SIMULAÇÃO	67
7	CONCLUSÕES E TRABALHOS FUTUROS	73
	REFERÊNCIAS	74

1 INTRODUÇÃO

Com o avanço da tecnologia, computadores tornam-se cada vez mais autônomos para atender as mais diversas atividades. Os drones, ou VANTs (Veículos Aéreos Não Tripulados), são veículos utilizados em processos como a captura de imagens, entrega de produtos, manutenção de linhas de transmissão, mapeamento do solo, e vários outros.

ESQUIVEL (2015) desenvolveu um kit de monitoramento por drone para áreas de preservação permanente na Floresta Atlântica do Rio de Janeiro, analisando imagens geográficas capturadas. Conclui que a tecnologia representa uma solução de bom custo-benefício para mapear e monitorar a floresta, especialmente contra a expansão agrícola clandestina.

Agricultura é outro área em que o uso de drones já se difundiu, e as tecnologias de agricultura de precisão já são uma realidade no campo para os técnicos e produtores rurais. JORGE e INAMASU (2014), apresentaram várias técnicas de processamento de imagens capturadas por drones, utilizadas na identificação de alterações na plantação, como o estresse hídrico e o seu vigor. Concluem que falhas no sistema, em geral, não tem sido a principal causa de quedas, mas sim a falta de manutenção, e falhas de operação por parte do operador.

Em contraste à difusão da tecnologia no ambiente rural, no ambiente urbano ainda existe o desafio legal referente a operação completamente autônoma, o que vem sendo contestado pela Amazon, que propõe e experimenta o sistema aéreo de entregas de mercadoria Amazon Prime Air. Nesse sistema os protótipos são construídos com múltiplas redundâncias e algoritmos sofisticados de evasão de obstáculos, porém permanecem em fase de testes e não possuem um design e mecanismo final (AMAZON, 2018). WELCH (2015), analisou o custo-benefício do sistema, e projetou o potencial de trazer uma significativa vantagem contra os competidores de mercado; como obstáculo, identificou as regulações da FAA, Federal Aviation Administration, que vem impedindo o estabelecimento do serviço de entrega autônoma nos Estados Unidos, como por exemplo ao restringir que em áreas urbanas o veículo só possa ser operado enquanto na linha de visão do seu piloto (ROMM, 2017).

O presente trabalho iniciou-se de uma curiosidade sobre o limite da confiança nesses veículos, e por uma apreciação das técnicas reunidas na Teoria de Controle. Três alternativas de controlador PID são projetadas para a atitude de um drone quadrotor, através da identificação do seu modelo matemático, por análise da resposta ao relé em malha fechada. Ao mesmo tempo, um simulador do drone quadrotor é construído na plataforma Unity, na tentativa de isolar a dinâmica dominante e antecipar possíveis riscos ao tornar essa tecnologia autônoma, e suas soluções.

1.1 Referencial do quadrotor

O drone utilizado na pesquisa classifica-se como quadrotor, porque possui quatro pares motor-hélice utilizados para criar uma força de sustentação no ar. O drone em particular era utilizado em pesquisas no curso de graduação de Engenharia Elétrica da UFPI, e nas condições encontradas já possuia os elementos básicos de hardware para a aquisição de dados e a atuação; entretanto, enfrentava-se dificuldades para implementar o software do controle de equilíbrio.

O drone é apresentado na Figura 1, e foi montado por um conjunto de peças adquiridas anteriormente ao trabalho. Sua estrutura consiste em quatro braços de plástico, conectados por uma peça superior central sobre a qual fixa-se o microcontrolador Arduino do autor. Abaixo dessa peça metálica, existe um vâo e então uma base plástica onde fixa-se a bateria Turnigy LiPo 3S 2200 mAh, de alimentação dos circuitos elétricos. Na parte inferior dessa base, parafusa-se uma placa de circuito impresso com as conexões de uma unidade de processamento inercial MPU6050, e de um sensor de distância não utilizado HCSR04. Em cada uma das quatro extremidades parafusa-se um motor Turnigy D2830-11 1000 KV, acoplado à uma hélice Gemfan 1045, e acionado por um controlador de velocidade eletrônico Hobbyking UBEC 30 A.

Figura 1 – Quadrotor utilizado no trabalho.



Fonte – O Autor.

Matematicamente, representam-se os três eixos referenciais de um corpo físico por três simples letras: x, y, e z. Entretanto, no aeromodelismo, também convém nomear os eixos referenciais em uma alusão a geometria de um avião, priorizando a rápida e comum identificação. Assim, nomeia-se como *frente* o sentido que o avião percorre, *abaixo* o sentido que o avião pousa, e *direita* o sentido da referente asa vista por trás. Naturalmente, nomeam-se os sentidos contrários respectivamente como: *atrás*, *acima*, e *esquerda*. Em um drone quadrotor, é arbitrário a diferenciação entre frente e direita, devido a sua simetria em torno do eixo vertical. Assim, a atitude do drone quadrotor define-se como qualquer uma das duas rotações horizontais possíveis.

Para a rotação ao redor dos eixos frente, direita, e abaixo, aqui nomea-se respectivamente os ângulos *roll*, *pitch*, e *yaw*, ou ainda x, y, e z. Os ângulos são medidos em graus, têm sentido positivo ou negativo, e se equivalem a cada $\pm 360^\circ$. Dessa forma, é comum que softwares de programação armazenem os seus valores dentro do intervalo de $\pm 180^\circ$, com uma transição descontínua entre um limite e o outro. Para o vôo de um quadrotor, é suficiente que *roll*, *pitch* e *yaw* permaneçam entre $\pm 90^\circ$, tal que a priori a descontinuidade não deva ser alcançada.

Para situar os motores enumera-se cada braço do quadrotor de 0 a 3, a partir daquele alinhado à frente, na ordem de rotação vertical positiva. No simulador construído, discutido na Seção 2.8, convém ainda colorir cada motor de acordo com o seu referente número, além de mostrar uma seta circular indicando o sentido da rotação vertical positiva. Apresenta-se na Figura 2 o drone simulado e as cores referentes a cada motor; no referencial apresentado deve-se interpretar cada semi-eixo colorido como uma seta direcionada ao sentido na respectiva cor.

Figura 2 – Quadrotor simulado no trabalho.



Fonte – O Autor.

Observa-se que existem múltiplas maneiras de se realizar uma manobra, e o plano de vôo que referencia os atuadores deve ser um controle à parte. Neste trabalho limitou-se ao objetivo de atingir o referencial de cada ângulo separadamente, por meio de uma combinação linear dos valores de atuação. Tal combinação é geralmente nomeada nos controladores comerciais como *mixing* (BIDDEN *et al.*, 2015). Em condições ideais e simétricas, para se manter o drone fixo no ar os quatro motores devem atuar com a mesma velocidade de rotação, expelindo o ar de cima para baixo, e balanceando a força gravitacional no drone. Além disso, para se equilibrar o torque proveniente do arrasto do ar e manter o ângulo *yaw*, o sentido de rotação de cada motor deve ser alternado entre positivo e negativo (BRESCIANI, 2008).

Visto a complexidade associada ao controle autônomo de quatro atuadores, optou-se por focar o estudo no comportamento da atitude *roll*, fixando o drone nas extremidades 0 e 2, de tal forma que apenas os atuadores 1 e 3 fossem necessários ao equilíbrio. Por analogia, o estudo pode então ser adaptado para os ângulos *pitch* e *yaw*, e analisado através da simulação.

1.2 Objetivos

Como objetivo geral deseja-se projetar um controle de atitude um drone quadrotor, capaz de alcançar um degrau de referência de até 10° , após 1 segundo da transição. O ângulo de atitude não deve ultrapassar 5% do valor de referência. Para tal, são objetivos específicos:

- Identificar um modelo matemático de função de transferência, para o sistema de atitude do drone quadrotor preso por cordas, e extrapolar o modelo para a situação de vôo livre.
- Construir um simulador fiel ao comportamento do drone, e disponibilizar o mesmo online em Martins (2018), para que seja útil a outros pesquisadores ou proprietários de drone.
- Avaliar diferentes estruturas de controladores e escolher a que melhor se adequa aos critérios de performance estabelecidos.

1.3 Estrutura do trabalho

No Capítulo 2 são apresentados os componentes essenciais de hardware e software utilizados no trabalho, bem como o método utilizado para os testes e a sua fundamentação teórica e prática. O simulador desenvolvido e publicado em Martins (2018) também é apresentado.

No Capítulo 3 apresenta-se a teoria de funções de transferência, utilizadas como modelo matemático para o sistema de atitude discutido nos capítulos seguintes.

No Capítulo 4 desenvolve-se um método de identificação de modelo de função de transferência, a partir da resposta ao relé em malha fechada (ASTROM; HAGGLUND, 1995) e da aquisição de pontos da curva de Nyquist, para o sistema de atitude fixado por cordas; repete-se o processo no simulador e se realiza a sua validação por comparação ao modelo identificado.

Depois, com o modelo identificado, no Capítulo 5 três estruturas de controladores PID são analizadas: o controlador PD sintonizado pelo método de Ziegler-Nichols, o controlador I-PD sintonizado por alocação de pólos dominantes, e o controlador PDF-D sintonizado pela otimização do critério ITAE para uma referência do tipo degrau.

Por fim, no Capítulo 6 mostra-se os resultados obtidos por simulação dos três controladores propostos, observando a resposta para um trem de degraus executado no simulador desenvolvido; e no Capítulo 7 conclui-se sintetizando o conhecimento adquirido e enumerando propostas de continuação do trabalho, para viabilizar um controle autônomo.

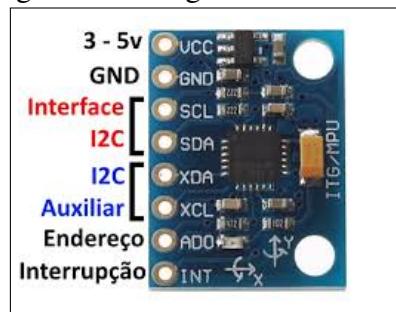
2 COMPONENTES ESSENCIAIS

2.1 Unidade de processamento inercial MPU6050

Unidades de processamento inercial consistem em dispositivos aplicados na identificação do movimento, geralmente a partir de dois ou mais sensores de diferentes grandezas físicas. O MPU6050, unidade fabricada pela empresa InvenSense Inc., utiliza dois sensores internos de tecnologia MEMS (*Micro Electro Mechanical System*) fundamentada na pressão e sinal elétrico correspondente em piezoelétricos. Como mostra Wiki (2018), os dois sensores são: o sensor accelerômetro, que quantifica a influência do campo gravitacional terrestre através da inércia de uma massa microscópica interna; e o sensor giroscópio, que quantifica o deslocamento angular através do efeito coriolis observado em uma massa microscópica vibrante.

Os dados axiais dos sensores são armazenados no total de seis registradores, cada um referente a um eixo identificado pela marcação X e Y, na Figura 3, onde mostra-se o MPU6050 e a sua pinagem. O sentido positivo de rotação em cada eixo pode ser configurado por software (INVENSENSE, 2013). Tais sensores são gerenciados por um processador interno, com registradores acessíveis externamente pela interface de comunicação serial I2C. A necessidade de se utilizar mais de um sensor para o processamento dos ângulos de rotação, vêm da precisão adicional criada pela fusão dos dados sensoriais. No MPU6050, os dados adquiridos pelo accelerômetro possuem boa resposta em baixa frequência, porém podem apresentar picos ruídosos de curta duração. De maneira complementar, o sensor giroscópio possui boa resposta imediata, porém pode acumular um desvio considerável após longa duração.

Figura 3 – Pinagem do MPU6050.



Fonte – filipeflop.com

O referencial nulo do MPU, no qual o drone deve se encontrar em equilíbrio e os ângulos devem medir 0° , precisa ser ajustado nos registradores de *offset* (INVENSENSE, 2014). O programa que identifica os valores de *offset* finais a se configurar é discutido na Seção 2.6.

O MPU6050 é uma unidade de processamento prática, devido a sua compatibilidade com o algoritmo DMP (*Digital Motion Processing*), patenteado pela IvenSense Inc., que calcula internamente os ângulos de rotação a partir dos dados brutos sensoriais, e assim preserva tempo de processamento necessário ao microcontrolador externo. Para corrigir o erro acumulado, o algoritmo DMP possibilita ainda a adição de um sensor magnetômetro pela comunicação I2C (INVENSENSE, 2013), porém esse sensor não foi adquirido. Sempre que o MPU é reiniciado, o algoritmo precisa ser reprogramado pelo microcontrolador externo através da interface I2C. O software DMP não é aberto, e é disponível apenas a interface que envia o seu código-fonte em hexadecimal para o MPU. A biblioteca i2cdevlib (ROWBERG, 2018), reune o esforço da comunidade open-source, isto é, que compartilha código livre, ao interpretar o significado de algumas linhas hexadecimais do código-fonte, que possibilitam a configuração do DMP.

2.1.1 Algoritmo DMP e a frequência de amostragem

A taxa de amostragem do algoritmo DMP escolhida foi a de 100 Hz, porque encontrou-se instabilidades na leitura ao se utilizar a opção mais rápida, de 200 Hz. Como o DMP é um software de código fechado, não se sabe ao certo a técnica utilizada no cálculo dos ângulos; apesar disso, a taxa de amostragem de 100 Hz e a praticidade na aquisição direta dos ângulos foram os diferenciais que incentivaram a escolha do DMP, ao invés de um processamento *ad-hoc* dos valores brutos registrados pelos sensores.

Para a interface entre o microcontrolador e o MPU6050, este possui uma pilha de registradores FIFO (*First In First Out*), utilizada para habilitar a concorrência entre os processamentos interno e externo. No algoritmo DMP, armazena-se na pilha FIFO as leituras de ângulos sem a necessidade de requisições externas, deixando à cargo do microcontrolador a opção de obter os dados e esvaziar a pilha em um momento oportuno. Quando a pilha não é esvaziada a tempo perde-se a primeira leitura; para evitar essa situação, no MPU existe a saída digital de interrupção, denominada INT na Figura 3. Dentre as possíveis interrupções a ser configuradas, existe a DATA_RDY_INT, habilitada pelo bit número 0 do registrador INT_ENABLE. Essa interrupção indica ao microcontrolador que uma nova leitura foi recém-armazenada.

No software escrito pelo autor, a ser apresentado na Seção 2.6, optou-se por não realizar o processamento de toda a pilha durante a interrupção DATA_RDY_INT, mas apenas sinalizar que esta ocorreu, e realizar o seu processamento durante o próximo loop de programa, facilitando assim uma maior consistência do seu tempo de execução.

2.2 Bateria LiPo 3s

A bateria utilizada foi a LiPo 2200mAh 3s 20C da Turnigy, apresentada na Figura 4. Através da sua nomenclatura identifica-se o reagente químico polímero de Lítio, com capacidade de energia elétrica para a descarga de 2200 mA durante uma hora, através de 3 células em série de tensão nominal 3,7 V cada, somando 11,1 V, e corrente mantida por 30 segundos de até 20 vezes a sua capacidade por hora, $20 \times 2,2 = 44$ A.

Figura 4 – Bateria LiPo 2200mAh 3s.



Fonte – [mercadolivre.com](https://www.mercadolivre.com)

As baterias representam um grande desafio para aplicações autônomas, nas quais deseja-se uma operação ininterrupta por mais que alguns minutos. Para o drone e a bateria em especial, equilibrar o seu 1,0 kg consome uma corrente aproximada de 5 A em cada motor. Nessa corrente total de 20 A, o drone teria uma autonomia de por volta de 7 minutos sem precisar recarregar a bateria. Maiores autonomias necessitariam de uma bateria com maior capacidade, e portanto maior peso e consumo associado. Como o drone em si já possui um peso considerável, acredita-se que ainda possa se beneficiar do aumento de capacidade da bateria.

Do ponto de vista de controle, a tensão da bateria representa uma variável importante, porque altera o ganho de velocidade nos motores e a produção de torque mecânico. Em todos os experimentos práticos, procurou-se carregar a bateria com uma tensão inicial um pouco acima da nominal, de 11,1 V, e utilizou-se o atuador configurado para cortar a alimentação com tensão menor que 9,0 V. A opção utilizada para carga foi sempre a de balanceamento simultâneo das três células, evitando assim a má distribuição energética e a potencial sobrecarga de uma célula.

2.3 Controlador de velocidade eletrônico ESC

O ESC (*Electronic Speed Controller*), em português controlador de velocidade eletrônico, é a interface de atuação que assegura a comutação trifásica de um motor. Na Figura 5 mostra-se o modelo de ESC utilizado: o Hobbyking UBEC 30 A; nota-se à esquerda a entrada de comando de velocidade, conectada ao microcontrolador, e à direita a saída trifásica de potência, conectada à um motor. O controle de velocidade em cada motor é independente e possui o seu ESC, num total de 4. Hobbyking (2018) apresenta o manual do usuário do UBEC 30 A, utilizado para configurar os níveis de comando máximo e mínimo, como discutido na Seção 2.6.

Figura 5 – Modelo de ESC utilizado.



Fonte – [mercadolivre.com](https://www.mercadolivre.com)

O sinal enviado pelo microcontrolador ao ESC, é um pulso PWM (*Pulse Width Modulation*) de 5 V, que codifica na sua largura a referência de torque do motor. O pulso deve ter largura entre 1 e 2 ms, com repetição à cada 20 ms. Essa modulação é a mesma utilizada por servo motores e foi adotada por fabricantes de ESC, principalmente por diferenciar o sinal mínimo do sinal nulo. Quando o pulso enviado tem largura menor que 1 ms, interpreta-se que houve uma falha de comunicação e o ESC inicia a tentativa de pouso, reduzindo a velocidade.

Semelhante ao MPU6050, o ESC corresponde à um processador paralelo ao programa principal. Não por acaso, a frequência do loop de controle no programa principal coincide com o período de atualização dos ESCs, de 20 ms. Dessa maneira, assume-se que exista algo próximo ao sincronismo, porém com a presença de um jitter ou oscilação de frequência não controlada. No pior caso, o programa principal envia o sinal de controle logo após o tempo de atualização do ESC, e o mesmo só será efetivado após um período de amostragem. Além disso, nada impede que um ESC modifique o seu valor antes ou depois dos outros.

2.4 Motor de corrente contínua sem escovas BLDC

Os quatro motores Turnigy D2830/11 utilizados são classificados como PM BLDC (*Permanent Magnet Brushless Direct Current*), isto é, são motores de corrente contínua sem escovas, de ímãs permanentes. Dentre as suas especificações, as mais relevantes para este trabalho foram o peso, de 52 g, o número de pólos, 14, a resistência 0.09Ω , e a sua constante Kv , de valor 1000 rpm V^{-1} , que é a razão entre a velocidade angular do motor e a tensão induzida correspondente. A Figura 6 apresenta um exemplo de D2830/11, onde 28 mm é o seu diâmetro, 30 mm é o comprimento da base até o início do eixo, e 11 é o número de voltas em cada bobina.

Figura 6 – Motor, base e conector da hélice.



Fonte – amazon.co.uk

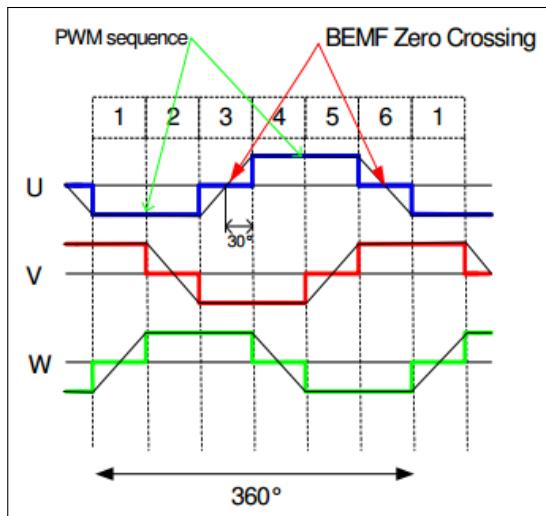
De maneira geral, o funcionamento de um motor elétrico tem origem na força de Lorentz, experenciada por cargas elétricas em movimento relativo à um campo magnético. No motor elétrico, essa força atua sobre as espirais condutoras de corrente, as bobinas, que produzem um campo elétrico-magnético, como postulado pelas Leis de Ampére e Faraday (SADIKU, 2012).

Os motores PM BLDC classificam-se como motores síncronos, nos quais os sinais elétricos de corrente e tensão oscilam em uma frequência f_e múltipla a frequência de rotação mecânica w , de acordo com (2.1), onde n_p é o número de pólos. Apesar deste motor classificar-se como de corrente contínua, na realidade sobre os seus três terminais a corrente assume um formato alternado, com o detalhe de que apenas dois terminais conduzem corrente por vez.

$$f_e = \frac{wn_p}{2}. \quad (2.1)$$

As ondas de tensão e corrente em cada uma das fases são representadas na Figura 7, que mostra um possível esquema de atuação utilizado pelo ESC, o responsável por inverter a corrente de alimentação da bateria. Nota-se que em cada momento apenas duas fases conduzem corrente, a forma de onda quadrada e colorida. Durante dois intervalos de 30° no período de repetição, cada fase deixa de ser alimentada, e a tensão deixa de ser positiva ou negativa, ao transitar de um extremo ao outro. O momento em que essa tensão contra-eletromotriz passa por zero, é utilizado pelo ESC para temporizar a próxima comutação de corrente, de forma a sincronizar o ímã do motor BLDC ao campo magnético da corrente, e transferir a maior parte de potência elétrica como potência de giro mecânica (GAMAZO-REAL *et al.*, 2010).

Figura 7 – Formas de tensão e corrente em um motor BLDC.



Fonte – (ATMEL, 2006).

Em (2.2) relaciona-se a tensão de entrada V e a velocidade w de um motor BLDC, como discutido em (KRYKOWSKI; HETMANCZYK, 2013). Assim, o motor BLDC equivale a um motor de corrente contínua monofásico, de resistência R e indutância L o dobro das individuais por fase. Ambos podem ser caracterizados por sua constante $K_e = \frac{30}{\pi K_v}$, que relaciona a tensão induzida $E = K_e w$ com a velocidade w de rotação em rad s^{-1} . Como é discutido na Seção 3.3, E atua como um *feedback* negativo, assim como o torque de arrasto do ar na hélice.

$$V = RI + L \frac{dI}{dt} + E. \quad (2.2)$$

A indutância L e a resistência R relacionam a diferença de potencial entre V e E com o consumo de corrente I . Assim como a tensão e a velocidade, o torque e a corrente são diretamente proporcionais, também através da constante K_e .

2.5 Hélice

A hélice, junto ao motor e o ESC, compõe a interface de atuação do drone quadrotor. Em resumo, o papel do atuador no sistema de atitude é rotacionar o drone através da alteração da velocidade das hélices, e do deslocamento de ar ao seu redor. Desse deslocamento origina-se as forças e os torques responsáveis pelo movimento angular do drone. De maneira geral, as hélices tem formato helicoidal que expele o ar na direção do eixo da sua rotação, e em sentido (acima ou abaixo) particular à sua geometria. Para o quadrotor em questão, utilizou-se quatro hélices de dimensão 1045, duas apropriadas à rotação horária, e duas apropriadas à rotação anti-horária. Mostra-se na Figura 8 os dois modelos de hélice utilizados, bem como os adaptadores que ajustam a largura do encaixe ao eixo do motor. O sentido adequado de rotação de uma hélice pode ser identificado a partir da face com a superfície superior mais visível, quando a hélice é observada na sua posição horizontal.

Figura 8 – Modelos de hélice 1045 utilizados.

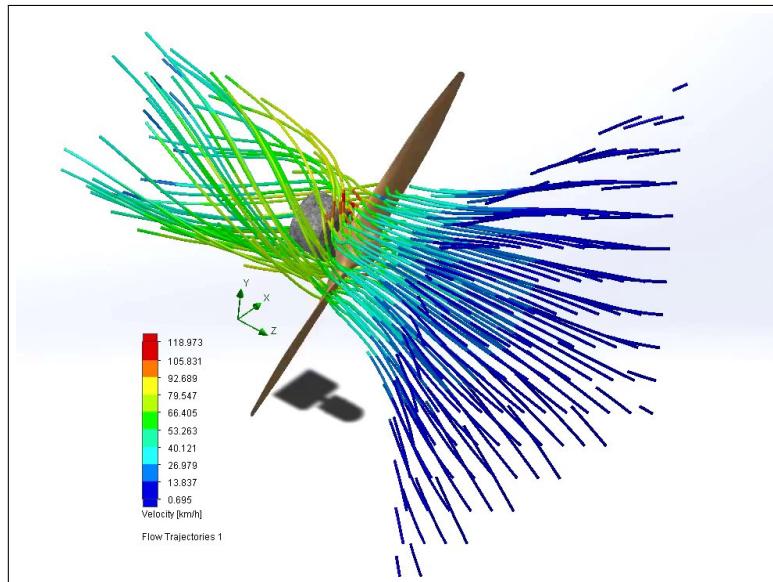


Fonte – mercadolivre.com

A dimensão da hélice é dada em polegadas, e pode ser conferida encravada na sua superfície. O primeiro número corresponde ao diâmetro da hélice, de 10 polegadas, e o segundo número corresponde ao *pitch* da hélice, de 4,5 polegadas. Apesar da nomenclatura *pitch*, este não tem relação direta à um só ângulo presente na hélice. Na realidade, o ângulo de ataque da hélice, isto é, o ângulo em que a sua superfície encontra as partículas de ar, aumenta gradualmente das suas extremidades até o seu centro, e o seu valor *pitch* resume o impacto dessa geometria em um único número. Essa grandeza é importante à especificação da hélice, porque junto ao diâmetro e à velocidade de rotação, quantifica o deslocamento de ar (STAPLES, 2014).

A Figura 9 ilustra o campo de velocidade provocado nas partículas de ar nas proximidades de uma hélice em rotação. Iniciado pelas partículas mais imediatas à superfície, o ar desloca-se rotacionando de cima para baixo, formando um disco de pressão diferencial no sentido contrário, que atua como uma força de sustentação sobre aquela área. Ao mesmo tempo, o arrasto das partículas de ar provoca um torque de carga ao motor BLDC, aumentando o consumo da corrente elétrica necessária para estabelecer a velocidade angular da hélice.

Figura 9 – Fluxo de ar em uma hélice simulada.



Fonte – SOLIDWORKS.

A velocidade angular w da hélice tem um papel fundamental no cálculo da força e do torque que influenciam cada motor. A força de propulsão F_p de cada hélice, ao atuar no seu eixo vertical distanciado l do centro de massa do quadrotor, cria o torque de atitude T_p em um eixo perpendicular. Decorre da aproximação estática de Staples (2014) que o torque T_p é então proporcional ao quadrado da velocidade w pelo coeficiente P , de acordo com (2.3). Além disso, o torque de arrasto do ar pode ser calculado por (2.4), proporcional à w^2 pelo coeficiente A .

$$T_p = F_p l = P w^2, \quad (2.3)$$

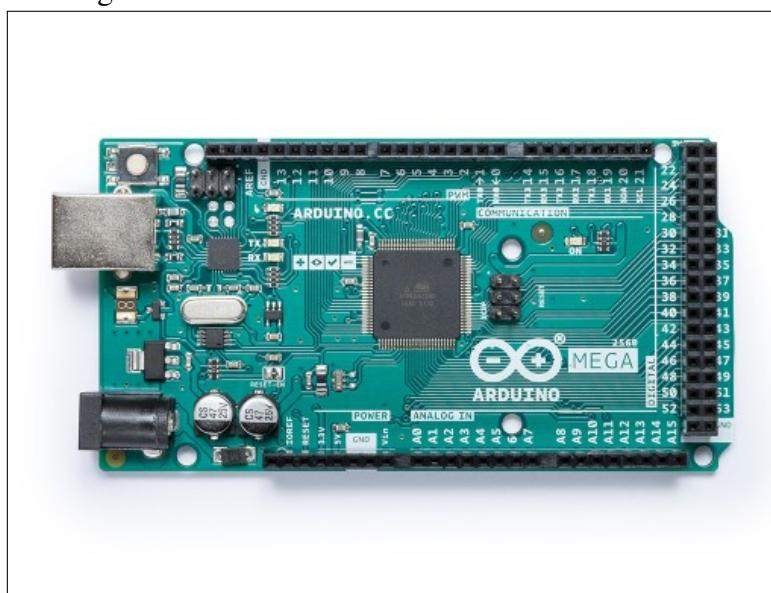
$$T_a = A w^2. \quad (2.4)$$

Os coeficientes P e A são quase constantes para o mesmo valor de diâmetro e *pitch*. Embora não seja necessário os identificar para determinar um modelo de atuação, para a simular a partir da velocidade w nominal foi útil realizar a medição de P por uma balança digital, e a inferência de A por comparação à hélice APC SL 1047 no banco de dados de J.B. *et al.* (2018).

2.6 Microcontrolador Arduíno Mega 2560

Um microcontrolador é um processador embarcado com periféricos apropriados para a leitura e o acionamento por sinais elétricos de baixa tensão. As tarefas realizadas por um microcontrolador são escritas em linguagens de programação como Assembler, C, ou C++, e descarregadas em sua memória. Tarefas comuns podem ser encontradas nas chamadas bibliotecas, um conjunto de arquivos com o papel de definir uma interface, geralmente não específica ao microcontrolador, que facilitam a prototipagem rápida de um software funcional. A interface de bibliotecas é importante devido a grande diversidade de microcontroladores disponíveis no mercado, cada um com características de hardware e de software específicas. O microcontrolador escolhido, Arduino Mega 2560, é apresentado na Figura 10.

Figura 10 – Modelo de microcontrolador utilizado.



Fonte – arduino.cc

O Arduino Mega 2560 não foi o microcontrolador original utilizado em pesquisas anteriores, mas sim uma adição particular do autor. Os fatores da sua escolha foram a familiaridade pessoal e o amplo conhecimento *open source* difundido na internet, por diversos usuários de Arduino. Especificamente adaptou-se dois programas, um utilizado para a calibração do MPU6050, e outro para a calibração do ESC HobbyKing UBEC: o primeiro desenvolvido por Rodenas (2015), e o segundo desenvolvido por Dû (2018). Os dois programas adaptados, e o programa utilizado para a identificação do sistema de atitude no Capítulo 4, podem ser acessados no repositório do GitHub pjago/tcc (MARTINS, 2018), do próprio autor. O GitHub é uma plataforma de contribuição de software, na qual é possível hospedar um controle de versão Git.

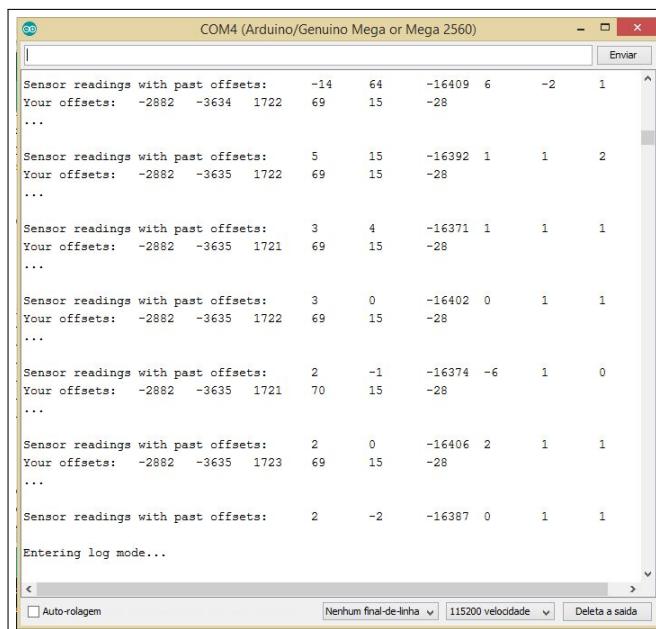
2.6.1 Calibração do MPU6050

A calibração do MPU6050 se resume a identificar o *offset* dos dois sensores internos, o acelerômetro e o giroscópio, em relação à distribuição da força gravitacional em cada eixo. No MPU6050, o *offset* consiste em seis valores, um para cada eixo e sensor, subtraídos internamente de cada valor sensoriado. Para que o drone equilibre a força peso na marcação de atitude 0°, o programa de calibração deve ser executado com o MPU6050 nivelado na horizontal.

O programa *MPU6050_calibration.ino* (RODENAS, 2015) baseou-se na biblioteca i2cdev de ROWBERG (2018), que implementa uma interface ao protocolo de comunicação serial I2C, entre o MPU e o Arduino. Nesse programa, um *offset* inicial é pré-configurado no MPU e então subtraído de um valor proporcional à média de 100 leituras, repetindo-se o processo até que a leitura obtida se encontre próxima ao valor neutro. Nesta condição, a leitura do eixo vertical do sensor acelerômetro deve marcar algo próximo de ±16384, correspondente a aceleração gravitacional de ±1G, e as outras cinco leituras devem se aproximar de zero.

Na Figura 11 apresenta-se o monitor serial do Arduino IDE, que deve ser configurado com baud rate de 115200 bps, utilizado para imprimir a atualização do *offset* na tela. Valores exatos dificilmente serão obtidos, em decorrência de ruídos ou pequenas oscilações do sensor durante a calibração. Os valores finais de *offset* devem ser anotados, e copiados para a variável *MPUOffsets* do programa de identificação, a ser configurada no MPU cada vez que for energizado.

Figura 11 – Console serial de calibração do MPU.



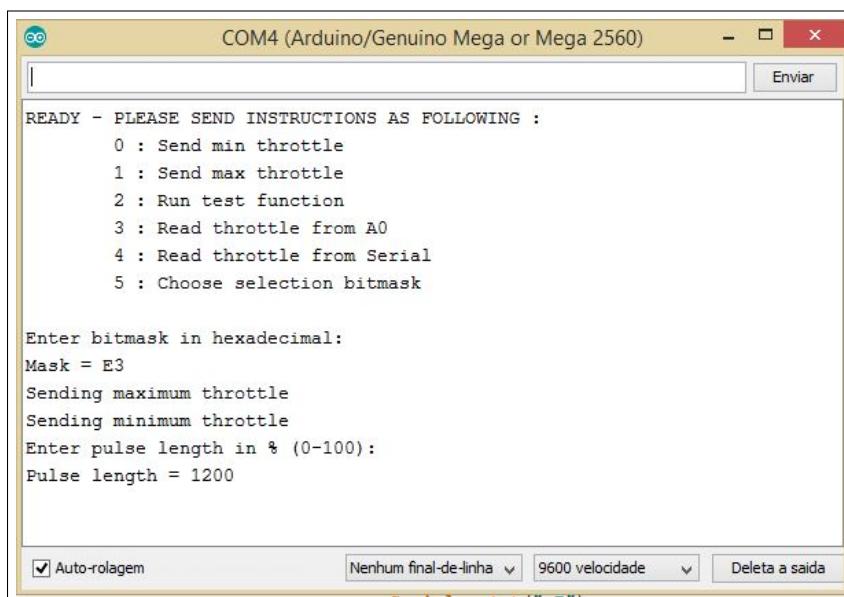
Fonte – O Autor.

2.6.2 Calibração do ESC

A calibração de cada ESC pode ser realizada simultaneamente a partir do programa *ESC-calibration.ino* (Dû, 2018). O programa original pode ser encontrado no seu repositório no GitHub lobodol/ESC-calibration, onde também é disponível uma breve explicação das três opções que esse programa espera do usuário pelo monitor serial. Na opção 0 cada ESC recebe a menor largura de pulso, configurada na variável *MIN_PULSE_LENGTH* como 1000 us. Na opção 1 cada ESC recebe a maior largura de pulso, configurada na variável *MAX_PULSE_LENGTH* como 2000 us. E por fim na opção 2, realiza-se uma varredura entre esses dois limites, do menor para o maior, com incrementos de 5 us na largura à cada 200 ms.

Na versão adaptada pelo autor, adiciona-se mais três opções visíveis na Figura 12, com o monitor serial do Arduino IDE configurado em um baud rate de 9600 bps. Na opção 3 atualiza-se o comando enviado de maneira proporcional à um sinal analógico obtido na porta A0 do Arduino. Essa opção pode ser utilizada para facilitar a mudança de largura de pulso através de um potenciômetro. Outra opção semelhante é a 4, porém nela o valor do comando é proporcional a um decimal de 0 à 100 requisitado no monitor serial, logo após o envio da opção 4. A última opção, 5, restringe a escolha do ESC o qual o sinal correspondente as opções anteriores será direcionado; o grupo escolhido corresponde aos bits de valor 1 na representação binária do hexadecimal, de 0 à F, requisitado após o envio da opção 5.

Figura 12 – Console serial de calibração do ESC.



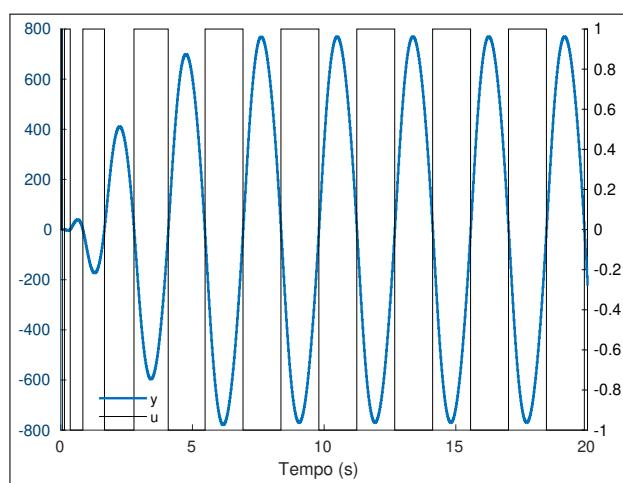
Fonte – O Autor.

2.7 Controle on-off

Nesta monografia, o termo *loop* de controle é utilizado para se referir ao constante esforço de se realizar a leitura do sensor, cálculo do erro, cálculo do próximo valor de atuação, e a atuação em si, repetidos especificamente nessa ordem. Tais instruções, quando realizadas em frequência adequada ao processo, posicionam uma variável representativa, de saída, em um valor de referência. Por outro lado, na entrada, a variável manipulada é aquela proporcional à potência de atuação controlada, capaz de modificar a variável de saída do processo de acordo com a sua dinâmica, como o ganho e as constantes de tempo.

Talvez a estratégia mais simples que exemplifique um *loop* de controle seja o controlador *on-off*. Nesse controle, a variável manipulada varia entre dois valores, *on* e *off*, com transições imediatas e em oposição à variável de saída. Em cada *loop*, a saída y é sensoriada e o seu erro $e = r - y$ com a referência r é calculado. Quando o erro é positivo a atuação u toma o valor *on*, caso contrário toma o valor *off*. Limitada por atrasos do processo, a saída y segue gradualmente o chaveamento em u , resultando na resposta característica da Figura 13.

Figura 13 – Resposta ao controle on-off.



Fonte – O Autor.

O controle on-off geralmente não possui a performance desejada, devido à ondulação provocada na saída, em azul na Figura 13. Ainda assim, para realizar esse controle na prática, é necessária boa parte da instrumentação presente nos demais controladores. Não obstante, o controlador *on-off* é uma ferramenta importante para o método de identificação aplicado no Capítulo 4. Portanto, optou-se por desenvolver o software embarcado o suficiente para realizar o controle *on-off*, e através dele identificar um modelo matemático para a atitude y .

2.8 O simulador

Um dos objetivos deste trabalho é que os alunos de graduação da UFPI, ao pagar as disciplinas de controle, tenham a opção de treinar o conteúdo teórico em um ambiente de simulação, sem a necessidade de realizar muita programação. O problema é que, nem todos chegam às cadeiras de Controle Analógico e Controle Digital com proeficiência em programação, e acabam esbarrando em dificuldades relacionadas não somente ao conteúdo das disciplinas, mas também à programação em si. Ainda assim, a criação de um sistema simulado é uma boa ferramenta para a familiarização dos conceitos básicos de loop de controle, funções de transferência, métodos de identificação, e outros. Com esse objetivo, imaginou-se um simulador de fácil parametrização do controle pré-estabelecido, ainda com a autonomia de se criar novos controladores, e novos sistemas, à partir de uma interface de programação opcional.

Um aspecto vantajoso da simulação é a acessibilidade que proporciona para com o sistema real. O drone em estudo não é acessível à todos os alunos, e antes de que alguns criem interesse para investigar o sistema, outras atividades da graduação os impede de se dedicar à tarefa. Inicialmente, o simulador criado é direto por ser específico ao drone da UFPI, e assim espera-se que alguns alunos iniciem a sua pesquisa com o drone, à partir da simulação.

Além do aspecto educativo, a simulação é uma ferramenta útil na previsão e mitigação dos riscos associados ao sistema, antes que estes se tornem custosos no aspecto monetário ou da saúde. No desenvolvimento de sistemas é importante errar, reconhecer, e corrigir os erros, mas muitas vezes não é seguro o fazer no sistema real, especialmente no início, quando esse ainda é desconhecido. Para o drone quadrotor, existe o risco de queda e destruição das hélices, ou ainda de cortes profundos, provocados por sua agudez e alta velocidade.

2.8.1 Unity3D

O ambiente de simulação criado utilizou a plataforma livre Unity3D da Unity Technologies, um conjunto de bibliotecas e editor 3D que reunem várias ferramentas auxiliares à criação de jogos. Unity é nesse sentido uma *engine*, porque possui camadas de software comuns à múltiplas aplicações, como exemplo implementações de *shaders*, física, e interface ao usuário. O principal motivo da escolha da Unity como plataforma foi a sua *engine* de física, a Nvidia Physx, que possibilitou a parametrização de um drone virtual, a partir de medidas do sistema real. O critério de semelhança entre simulação e prática foi a própria identificação do sistema.

A interface de programação na UnityEngine gira em torno do objeto de jogo, uma classe, no paradigma da orientação objeto, que participa de uma hierarquia com outros objetos de jogo, cada um com um componente de posição, rotação, e escala, acumulados ao longo da hierarquia. A *engine* Unity utiliza o conceito de componentes que podem ser adicionados aos objetos de jogo, para adicionar comportamentos pré-programados.

O componente *Rigidbody* é o qual habilita o objeto à participar da *engine* física, e ser sujeito as forças de gravidade, de arrasto, e de colisão com outros objetos. Para o drone em específico, foi necessário ainda calcular e adicionar as forças de propulsão e o torque de arrasto das hélices. Além do objeto de jogo principal, também adicionou-se componentes *Rigidbody* aos objetos filhos bateria e motores, porém os configurou como *isKinematic*, que na nomenclatura da *engine* significa que os mesmos não estão sujeitos às forças físicas. Entre dois componentes *Rigidbody* pode-se adicionar um componente *HingeJoint*, para simular uma conexão do tipo pêndulo, ou ainda uma *FixedJoint*, para simular uma conexão que fixa os dois objetos juntos. A utilização de *Joints* foi importante para a replicação do sistema teste no simulador.

2.8.2 Arcadia e Clojure

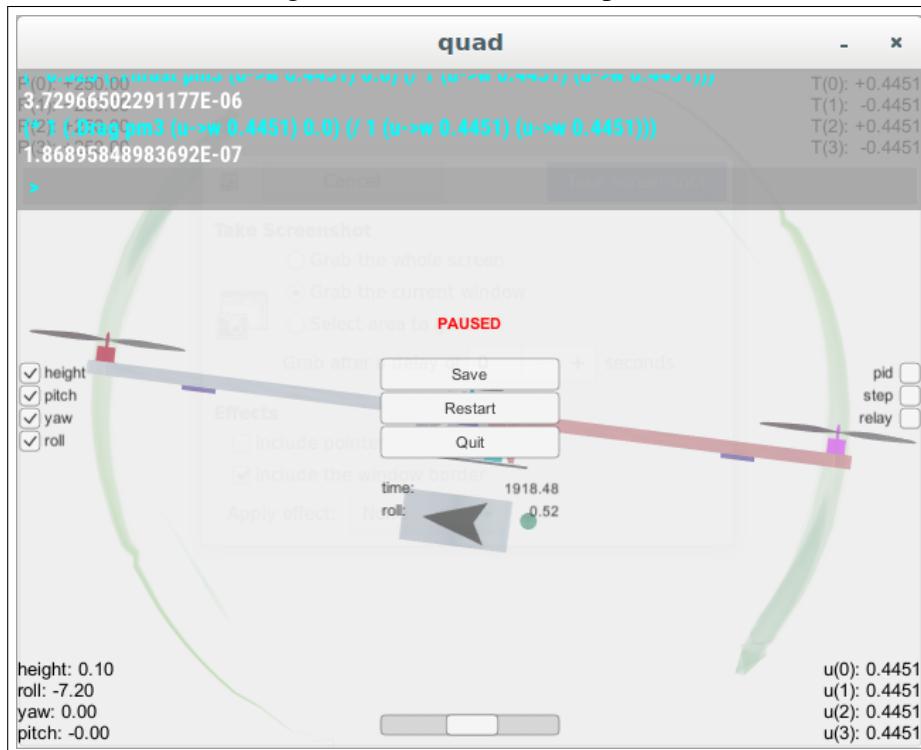
A programação do simulador em Unity3D foi realizado majoritariamente em Clojure-CLR, apesar de algumas utilidades terem sido programadas na linguagem oficial C#. O C#, é uma linguagem de programação criada pela Microsoft como uma alternativa ao Java, que para a Unity possibilita a criação de *scripts* como componentes que podem ser adicionados à objetos, assim como o é o *Rigidbody*. Por outro lado, Clojure-CLR é a implementação da linguagem Clojure, originalmente executada na JVM (*Java Virtual Machine*), adaptada para o CLR (*Common Language Runtime*) que é o mesmo ambiente que executa o C#. Dessa maneira, programas em Clojure-CLR e C# compartilham o mesmo ambiente e podem interagir entre si.

Essa aplicação de Clojure só é possível pelos esforços de Gardner e Nasser (2017), assim como toda a comunidade open-source do repositório Arcadia-Unity no Github. Arcadia é uma série de bibliotecas que funcionam como um plugin ao editor Unity, e o habilita a compilar *scripts* programados em Clojure-CLR, que se torna então uma alternativa à linguagem principal C#. A motivação desta monografia em utilizar Clojure, para desenvolver a maior parte do simulador, veio do desenvolvimento interativo proporcionado por seu REPL (*read-eval-print loop*) que funciona similar ao *workspace* da linguagem preferida para a análise, o MATLAB, estabelecendo de certa forma um diálogo entre o programador e o ambiente desenvolvido.

Em Clojure, o REPL funciona como uma linha de comando que recebe uma expressão nessa linguagem (geralmente contida entre parênteses), lê essa expressão com um compilador, a executa, e então imprime o resultado na tela. Essa interface, simples porém capaz, possibilitou o desenvolvimento interativo do simulador, ao habilitar a consulta do estado das variáveis programadas enquanto a simulação ocorre, alterando funções e observando o impacto imediato da mudança. Devido esse aspecto, que facilitou o desenvolvimento, o autor achou apropriado adicionar um REPL no executável do simulador, o que significa que o usuário pode modificar variáveis e funções lidando diretamente com o código fonte e a linguagem Clojure, sem que houvesse a necessidade do autor implementar uma interface gráfica em específico.

O simulador final, disponível em Martins (2018), ainda é habilitado com alguma interface ao usuário além do REPL, como os marcadores que limitam ou liberam a influência da *engine* física, e ainda a barra de arrasto na qual é possível alterar diretamente o valor do ângulo de rotação escolhido. Ou seja, adicionou-se interface ao usuário apenas para as aplicações mais simples, como a alteração de alguns parâmetros como a escala de tempo, ou um valor de distúrbio enviado aos motores. A Figura 14 mostra o simulador em *pause* e o REPL na extremidade superior, o REPL pode ser acessado apertando a tecla 1, e fechado apertando ESC; a interface será discutida mais a fundo nos próximos capítulos, a medida que for utilizada.

Figura 14 – Simulador em pause.



Fonte – O Autor.

2.9 Balanço horizontal

O sistema quadrotor é por sua natureza um sistema com quatro entradas, a tensão elétrica em cada motor, e pelo menos três saídas, os três ângulos *roll*, *pitch*, e *yaw*. Entretanto, nessa monografia limitou-se ao controle de apenas uma saída de atitude, a partir de uma entrada: o torque total produzido no eixo atuante. Devido a sua simetria, o controle do ângulo *roll*, e o controle do ângulo *pitch*, são equivalentes e podem utilizar a mesma análise a ser discutida.

Para realizar o controle *on-off* de atitude, o quadrotor precisa ser limitado a rotacionar em apenas um eixo, fixado na mesma altura, e de forma que o conector USB não deva ser atingido. A estrutura montada para realizar esse experimento pode ser observada na Figura 15. Nota-se que o eixo de rotação precisa ser fixado de ponta à ponta, e isto pode ser realizado com dois cordões e duas cadeiras, além de uma mesa sobre a qual se localiza um notebook. Uma terceira cadeira com rodinhas foi utilizada para facilitar a manutenção do drone e do notebook, além de um osciloscópio digital para observar os sinais elétricos de comando e de tensão nos motores.

Figura 15 – Sistema de testes, com a sustentação por cordas.



Fonte – O Autor.

Como se vê na Figura 15, o eixo deve estar nivelado com a horizontal, e a altura do eixo até o nó que o sustenta deve ser o suficiente para que o drone possa realizar um giro completo, sem que as hélices colidam com o chão. Essa situação pode ocorrer por exemplo quando o sentido de rotação de um dos motores está trocado, e os torques de cada propulsão se adicionam positivamente. Desde que esse limite de altura seja preservado, quanto maior a distância entre o drone e o nó, melhor, porque maior será a força de restauração para o mesmo ângulo, o que limita a amplitude do giro e facilita a medição.

Desconsiderando o efeito giroscópio, da interação entre um eixo de rotação e outro, um modelo simplificado para a rotação pode ser obtido aplicando-se a terceira Lei de Newton. A equação (2.5) especifica a influência do torque T_x da propulsão das hélices no seu respectivo ângulo x . De acordo com a distribuição dos quatro atuadores sobre os eixos, controla-se o ângulo *roll* pelo torque das hélices 3 e 1, e controla-se o ângulo *pitch* pelo torque das hélices 2 e 0. O momento de inércia I_x limita a aceleração angular no eixo x , e o arrasto do ar devido a velocidade de rotação dx/dt produz um torque contrário ao movimento através da constante de arrasto C_x .

$$T_x = I_x \frac{d^2x}{dt^2} + C \frac{dx}{dt} + K_x. \quad (2.5)$$

Especificamente durante a realização do controle *on-off* no sistema teste, um terceiro termo é introduzido, proveniente do torque de restauração característico em pêndulos limitados a pequenos ângulos. De fato, substituindo o quadrotor por seu centro de massa, quando este é pendurado em um ponto fixo e $\text{sen}(x) \approx x$, um torque restaurador de constante elástica K_x se opõe ao seu deslocamento angular, em relação à origem. A esse sistema é utilizada a nomenclatura balanço horizontal. A equação (2.5) com $K_x \neq 0$, apresenta a dinâmica de balanço para o ângulo *roll*, durante o controle *on-off*. A constante K_x é um efeito colateral da restrição do sistema, porém facilita a identificação de I_x e C_x , o que é objetivo do sistema teste.

3 FUNÇÕES DE TRANSFERÊNCIA

A análise do sistema de rotação do par motor e hélice, e do sistema de atitude do quadrotor, utiliza conceitos como a derivada de um sinal contínuo no tempo, e representa a evolução desse sinal por meio de uma equação diferencial. A transformação do sinal no domínio do tempo t , para um sinal equivalente no domínio da frequência s , foi desenvolvida por Laplace, inicialmente como uma ferramenta auxiliar a solução de equações diferenciais (DEAKIN, 1981).

Um sinal $x(t)$ é classificado como LTI (linear e invariante no tempo) quando segue a equação diferencial (3.1), que o relaciona a uma sequência de n derivadas da sua função $x(t)$, ou seja, as taxas de crescimento desse sinal. As constantes a_i , com i de 0 à n , devem ser invariantes, isto é, devem ter sempre o mesmo valor, independente do instante de tempo t . Na prática, admite-se a elas uma pequena variação, associada ao ponto de operação ao qual a variável $x(t)$ se aproxima. Denomina-se a evolução de x através de (3.1) de resposta natural.

$$a_n \frac{d^n x}{dt^n} + \dots + a_3 \frac{d^3 x}{dt^3} + a_2 \frac{d^2 x}{dt^2} + a_1 \frac{dx}{dt} + a_0 x = 0. \quad (3.1)$$

O controle LTI do sinal x é representado por um agente externo u que o influencia, com a adição de m derivadas, ponderado por constantes b_j alterando a equação (3.1) de resposta natural, para a equação (3.2) de resposta forçada. No grupo de sistemas aqui discutidos, apenas o sinal de atuação $u(t)$, sem a presença das suas derivadas, influencia diretamente na evolução de $x(t)$. Como exemplo da duplidade entre o tempo e a frequência, apresenta-se em (3.3) a relação equivalente nas transformadas de Laplace $X(s)$ e $U(s)$, respectivas aos sinais $x(t)$ e $u(t)$.

$$a_n \frac{d^n x}{dt^n} + \dots + a_2 \frac{d^2 x}{dt^2} + a_1 \frac{dx}{dt} + a_0 x = b_m \frac{d^m u}{dt^m} + \dots + b_2 \frac{d^2 u}{dt^2} + b_1 \frac{du}{dt} + b_0 u, \quad (3.2)$$

$$a_n s^n X + \dots + a_2 s^2 X + a_1 s X + a_0 X = b_m s^m U + \dots + b_2 s^2 U + b_1 s U + b_0 U. \quad (3.3)$$

Para os propósitos desta monografia, o operador s multiplicando uma transformada de Laplace $X(s)$ pode ser interpretado como uma derivada do sinal $x(t)$ no tempo, e vice e versa. Dessa forma, a multiplicação por s^i corresponde à aplicação de i derivadas ao sinal correspondente no tempo. Decorre dessa interpretação que o operador s compartilha das propriedades lineares assim como um derivador, e dessa forma a análise de sistemas LTI pode ser feita algébricamente, colocando-se as transformadas $U(s)$ e $X(s)$, de entrada e de saída, em evidência.

$$G(s) = \frac{X(s)}{U(s)} = \frac{B(s)}{A(s)} = \frac{b_m s^m + \dots + b_2 s^2 + b_1 s + b_0}{a_n s^n + \dots + a_2 s^2 + a_1 s + a_0}. \quad (3.4)$$

O resultado é a chamada função de transferência $G(s)$, em (3.4), um modelo matemático do sistema recíproco a equação diferencial (3.2) que o governa. Os polinômios $B(s)$ e $A(s)$, correspondentes ao numerador e ao denominador da função de transferência $G(s)$, têm papel fundamental nas características de tempo e frequência do sistema modelado. Em particular, as raízes complexas dos polinômios, podem ser consideradas os “átomos” da análise de sistemas lineares, assim como os números primos são os “átomos” da teoria dos números. Para diferenciar as raízes, denominam-se de zeros as raízes do numerador $B(s)$, e de pólos as raízes do denominador $A(s)$. É comum a representação de zeros e pólos em um plano cartesiano complexo, onde os zeros são marcados com “o” e os pólos são marcados com “x”. Nesta monografia, o sistema a ser controlado não possui zeros, ou seja, o polinômio B se resume à uma constante.

3.1 Exemplo no MATLAB

Na toolbox *ControlSystems* do MATLAB, uma representação da função de transferência G pode ser instanciada pelo comando $G = tf(B, A)$, onde B e A são representados pelo vetor de seus coeficientes, da maior para a menor ordem em s . O comportamento da função G no tempo pode ser observado pelo comando *step(G)*, que simula a sua resposta na saída $x(t)$ para o sinal de entrada denominado degrau unitário, $u(t) = 1$. Já a análise da função G na frequência, pode ser feita pelo comando *nyquist(G)*, que apresenta o chamado diagrama de Nyquist; um diagrama polar com o caminho traçado por $G(j\omega)$, que compreende a informação de ganho e de fase da resposta senoidal $x(t)$ para uma entrada senoidal $u(t)$, parametrizada pela frequência ω . O código a seguir foi executado e é utilizado aqui para exemplificar os comandos *step* e *nyquist*.

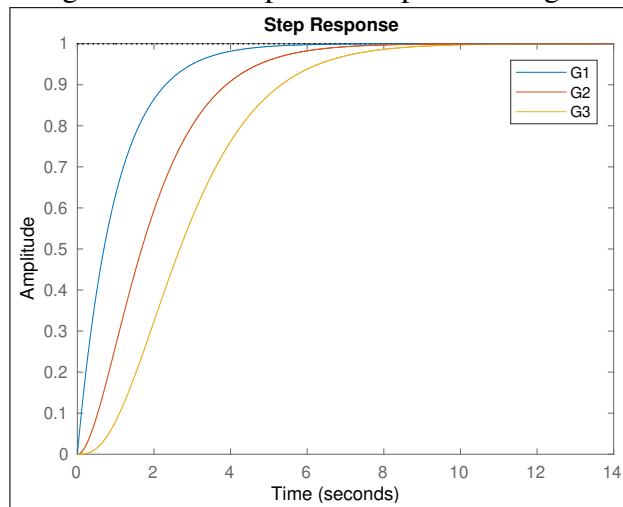
```

G1 = tf([1], [1 1]);      % u(t) = \dot{x}(t) + x(t)
G2 = tf([1], [1 2 1]);    % u(t) = \ddot{x}(t) + 2\dot{x}(t) + x(t)
G3 = tf([1], [1 3 3 1]);  % u(t) = \dddot{x}(t) + 3\ddot{x}(t) + 3\dot{x}(t) + x(t)
% Sobrepõe o plot do step (e nyquist), de três funções exemplo
figure; hold on;
step(G1); step(G2); step(G3);
legend('G1', 'G2', 'G3', 'Location', 'NorthEast')
figure; hold on;
nyquist(G1); nyquist(G2); nyquist(G3);
legend('G1', 'G2', 'G3', 'Location', 'NorthWest')

```

Como exemplo escolheu-se as funções G_i , onde i coincide a sua ordem, isto é, ao grau do seu denominador. As três funções possuem os pólos em -1 , repetidos i vezes. O resultado do comando *step* nessas funções pode ser observado na Figura 16. Primeiramente chama-se a atenção de que as três respostas são estáveis pelo critério BIBO, ou seja, uma entrada limitada, produz uma saída limitada (LATHI, 2007). Isso ocorre porque todos os pólos de cada função têm parte real negativa; bastaria um pôlo com parte real positiva para que a resposta fosse ilimitada. Outro fator notável é que as três respostas possuem valor final igual à 1; esse valor, para uma entrada degrau unitário, pode ser previsto aplicando a função G no ponto $s = 0$. Por último, chama-se a atenção de que o aumento da ordem da função aumenta o atraso de resposta, no caso de pólos iguais; caso os pólos sejam diferentes, aqueles com o maior módulo da parte real têm a resposta individual mais rápida, porém os atrasos de cada pôlo ainda se acumulam.

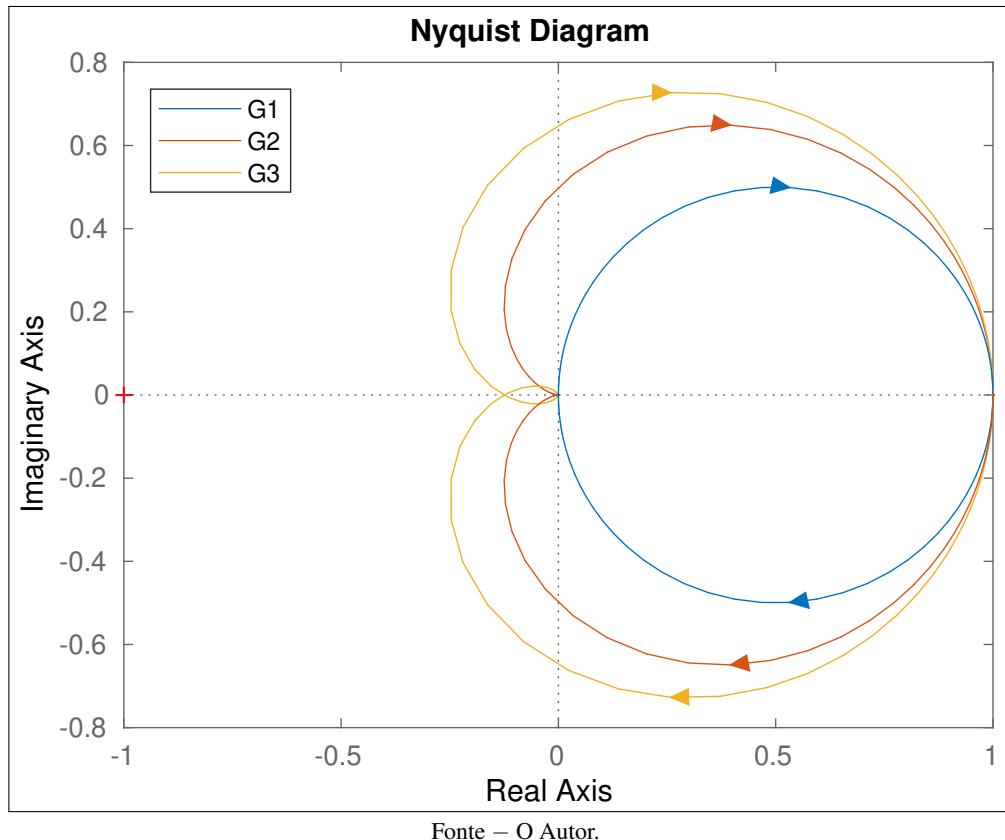
Figura 16 – Exemplos de resposta ao degrau.



Fonte – O Autor.

O resultado do comando *nyquist* pode ser observado na Figura 17. O diagrama de Nyquist consiste em um mapa conforme do domínio $s = j\omega$, através da função de transferência $G(s)$ resultando no valor complexo $G(j\omega)$, que percorre o caminho traçado na Figura 17 pelo sentido da seta, a medida que o parâmetro ω vai de 0 à infinito. Especificamente em funções com maior número de pólos do que zeros, no limite em que ω vai ao infinito o número complexo $G(j\omega)$ se aproxima de zero. O diagrama de Nyquist de funções reais é necessariamente refletido pelo eixo real, porém uma de suas metades, nesse caso a superior, corresponde a valores negativos de frequência, o que não possui interpretação física. Na curva inferior, nota-se que a defasagem da saída aumenta a medida que a frequência de entrada aumenta. Como será apresentado no Capítulo 4, o diagrama de Nyquist foi fundamental para o método de identificação utilizado.

Figura 17 – Exemplos de curvas de Nyquist.



Fonte – O Autor.

Para o nyquist das três funções exemplo, chama-se atenção de que apenas a curva da função de terceira ordem G_3 atravessa o eixo real negativo; denomina-se frequência crítica aquela na qual isso ocorre. Uma conclusão notável de Nyquist é que, a proximidade dessa curva ao ponto $(-1, 0)$, o + na extremidade esquerda da Figura 17, reduz a estabilidade da sua malha fechada. O método do relé proposto por Astrom e Hagglund (1995), permite identificar esse ponto de frequência crítica de maneira experimental, por meio da análise da entrada e saída de um controlador on-off, sem a necessidade de um modelo matemático para o sistema.

Através do critério de estabilidade de Nyquist (1932), se a função de malha aberta G não possuir pólos de parte real positiva conclui-se que, a função resultante de malha fechada será estável se e somente se a curva G não circular o ponto $(-1, 0)$. Assim, uma medida de estabilidade é a margem de ganho, a mudança de escala da curva G que a faz alcançar o ponto $(-1, 0)$. Outra medida de estabilidade é a margem de fase, a menor rotação necessária à curva G para que compreenda o ponto $(-1, 0)$. Existe ainda uma segunda ferramenta aqui utilizada para analizar a resposta em frequência, o diagrama de Bode, que apresenta os valores de ganho e de fase separadamente, em função da frequência, e será analisado no Capítulo 4.

3.2 Realização dos controladores

A transformada de Laplace aplica-se a um sinal $x(t)$ contínuo no tempo, isto é, sobre o qual o tempo t pode assumir qualquer valor real. Entretanto, na prática, apesar das variáveis de controle e controladas existirem a todo momento, sistemas microcontrolados as têm acesso somente em intervalos de tempo $t = kT$ específicos, onde k é um número inteiro e T é o período de amostragem. Uma notação comum para representar as amostragens $x(kT)$, é a do sinal discreto $x[k]$, sobre o qual aplica-se a transformada Z, que se assemelha em vários aspectos à transformada de Laplace (LATHI, 2007). No contexto discreto, pode-se considerar a multiplicação do operador z^{-i} à transformada $X(z)$, equivalente à um atraso do sinal $x[k]$ para o sinal $x[k - i]$, com i inteiro e positivo. Em paralelo às equações (3.2) e (3.3) estão as equações (3.5) e (3.6), onde a equação diferencial é substituída por uma equação de diferenças, e a transformada de Laplace é substituída pela transformada Z.

$$a_n x_{[k-n]} + \dots + a_1 x_{[k-1]} + a_0 x_{[k]} = b_m u_{[k-m]} + \dots + b_1 u_{[k-1]} + b_0 u_{[k]}, \quad (3.5)$$

$$a_n X z^{-n} + \dots + a_1 X z^{-1} + a_0 X = b_m U z^{-m} + \dots + b_1 U z^{-1} + b_0 U. \quad (3.6)$$

Neste trabalho, a transformada Z foi utilizada apenas para implementar os controladores projetados, em um processador discreto. Os controladores $C(s)$ foram todos projetados em Laplace, pelos métodos discutidos no Capítulo 5, através do modelo $G(s)$ identificado para o sistema. Depois, os zeros e os pólos aproximados da sua versão discreta $C(z)$ foram identificados pela aproximação $z = e^{sT}$, onde s é o zero ou pôlo correspondente em $C(s)$, e T é o período de amostragem, de 0,02 s. Tal aproximação pôde ser automatizada pelo comando $Cz = c2d(Cs, T, 'matched')$, no MATLAB. Por fim, a equação de diferença correspondente à $C(z)$ foi identificada, e utilizada para calcular a atuação discreta $u[k]$, no loop de controle.

3.3 Transformada de Laplace para o sistema motor-hélice

No sistema motor-hélice, a transformada de Laplace não pode ser aplicada diretamente, devido a não-linearidade entre a velocidade angular w e os torques T_a e T_p , de arrasto e de propulsão. Todavia, ainda é possível utilizar Laplace após a aproximação de pequenos sinais alternados (NILSSON; RIEDEL, 2007), de comportamento quase linear no intervalo de operação, por volta de 50% do torque máximo. Nessa aproximação, demonstrada em (3.7) para um k qualquer, o sinal $w(t)$ é separado entre o seu termo fixo \bar{w} e o seu termo oscilatório $\tilde{w}(t)$.

$$w(t) = \bar{w} + \tilde{w}(t), \quad (3.7)$$

$$T(t) = \bar{T} + \tilde{T}(t) = k\bar{w}^2 + \tilde{T}(t),$$

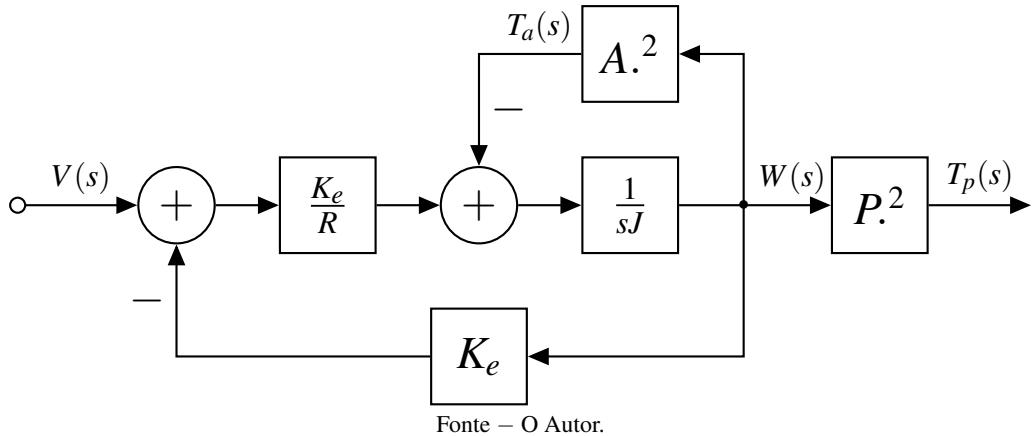
$$\frac{\tilde{T}(t)}{\tilde{w}(t)} \approx \frac{dT}{dw} = 2kw(t) = 2k(\bar{w} + \tilde{w}(t)),$$

$$\tilde{T}(t) \approx 2k(\bar{w}\tilde{w}(t) + \tilde{w}^2(t)), \text{ onde } \bar{w} \gg |\tilde{w}(t)| \implies \bar{w}|\tilde{w}(t)| \gg \tilde{w}^2(t),$$

$$\tilde{T}(t) \approx 2k\bar{w}\tilde{w}(t).$$

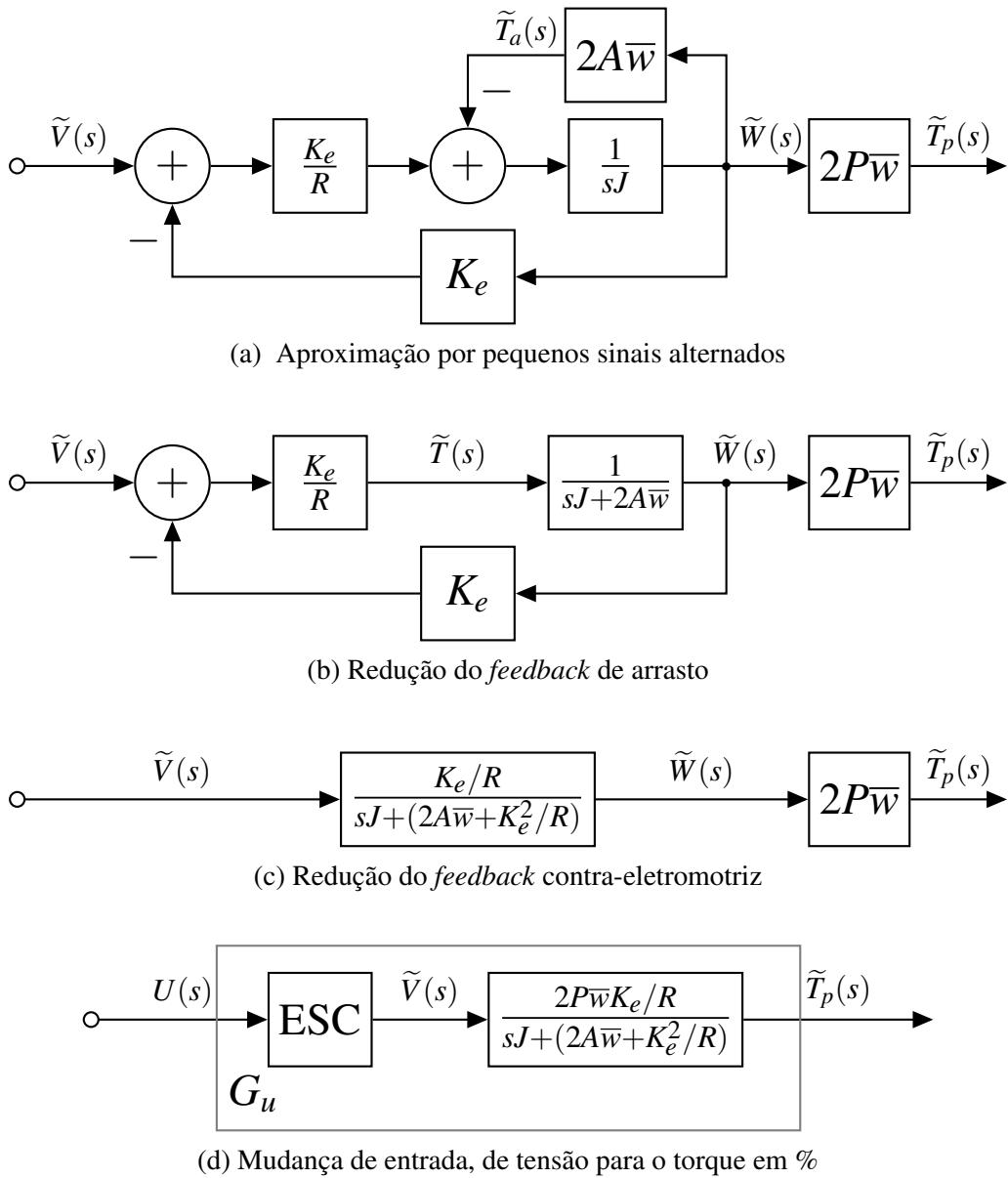
Uma maneira de visualizar o sistema motor-hélice é através do diagrama de blocos na Figura 18. O diagrama de blocos é uma ferramenta na qual as operações de adição, multiplicação, e até mesmo operações não lineares, são representadas a partir de blocos conectados, nos quais cada conexão representa um sinal, evidenciando as transformações no sistema desde a entrada até a saída. No referente a análise da atitude do quadrotor, o sistema motor-hélice pode ser resumido à sua entrada de tensão V , e à sua saída de torque T_p . O efeito da indutância do motor, e o efeito do torque de atrito mecânico foram desconsiderados.

Figura 18 – Diagrama de blocos do sistema motor-hélice



Na Figura 18, os blocos não lineares A e P , referentes às funções quadráticas da hélice, inicialmente impossibilitam modelar o sistema por uma única função de transferência linear, ou seja, de se reduzir o diagrama do sistema à um único bloco. Todavia, o desequilíbrio entre torques necessário ao controle de atitude, pode ser considerado um pequeno sinal alternado, correspondente à uma pequena diferença de velocidade entre os motores. Dessa forma, o sistema motor-hélice pode ser linearizado com base em (3.7), substituindo os sinais da Figura 18 por suas componentes de oscilação, e tornando possível a simplificação do diagrama de blocos.

Figura 19 – Simplificação do diagrama de blocos do sistema motor-hélice



Fonte – O Autor.

A partir das Figuras 19a e 19b, as retroalimentações de torque e tensão são simplificadas uma de cada vez, através da fórmula $\frac{G}{(1+GC)}$, onde G é o produto dos blocos no caminho direto, e C é o produto dos blocos no caminho de retorno (LATHI, 2007). Tratam-se de retroalimentações por diferença, que no sistema em questão, são mecanismos de malha fechada por construção, que limitam a aceleração instantânea do motor, e preservam suas partes mecânicas. Uma estrutura com dupla retroalimentação similar à essa será utilizada no Capítulo 5, no design de controladores por alocação de pólos. Da Figura 19c para a 19d evidencia-se a mudança de variável a partir da hipótese de que o ESC, como parte da linearização “super” suave descrita no seu manual Hobbyking (2018), atua modificando o ganho de entrada.

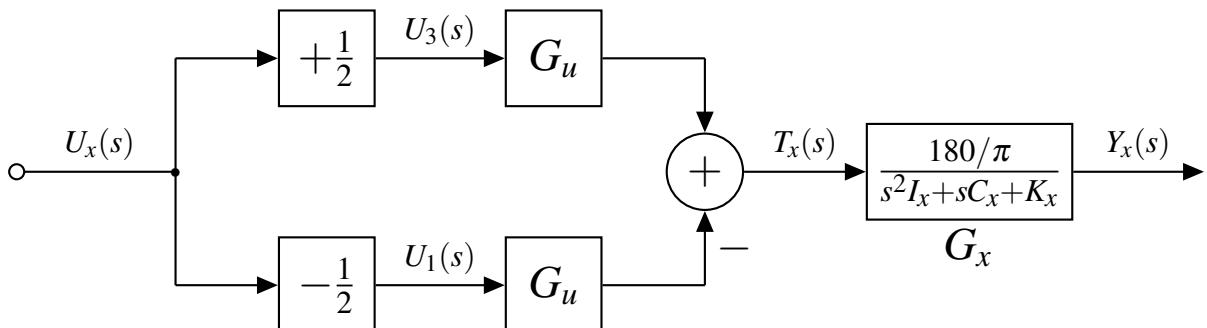
Para os fins desta monografia, considera-se o papel do ESC equivalente à constante necessária para produzir o torque final, em cada ponto de operação, correspondente ao que foi medido na Seção 2.5. Dessa forma, o ganho da função de transferência G_u torna-se o torque máximo $T_{p_{max}}$, visto que a entrada u é dada em porcentagem. Aceitando a hipótese de que o ESC, por chavear a tensão em altas frequências, não altera a dinâmica de tempo do sistema, o pólo p_{gu} da função final será o mesmo pólo do bloco de transição entre \tilde{V} e \tilde{T}_p . A tarefa de identificar a inércia J do par motor-hélice, pode então ser substituída pela identificação do pólo p_{gu} , e a aplicação de (3.8). O valor, a ser calculado, é utilizado como parâmetro na simulação.

$$J = \frac{2A\bar{w} + K_e^2/R}{p_{gu}}. \quad (3.8)$$

3.4 Transformada de Laplace para o sistema de atitude

As saídas T_p 3 e 1, correspondentes ao *mixing* de atuação, são somadas e alimentadas no sistema de atitude do quadrotor. Para simplificar a análise, considera-se que os atuadores tem a função de transferência G_u idêntica, com o mesmo pólo e o mesmo ganho, diferindo apenas no sentido de contribuição. A função de transferência resultante, desde a variável manipulada u referente à um eixo, até a saída correspondente y , segue o exemplo da Figura 20, para o eixo roll. Nota-se que o *mixing* escolhido para a variável u_x a torna uma porcentagem do torque final T_x produzido, o que também ocorre para u_y e u_z , nos seus respectivos eixos.

Figura 20 – Diagrama de blocos do sistema de balanço horizontal em roll



Fonte – O Autor.

Denomina-se G a função de malha aberta do sistema, à ser identificada no teste do balanço horizontal, além de calculada a partir de (3.9). Os parâmetros específicos à dinâmica de rotação em roll, presentes no denominador da função G_x , podem ser identificados fatorando G_u de G , assumindo que p_{gu} corresponde ao pólo real mais rápido de G . Essa hipótese é razoável

visto que o torque deve se estabelecer mais rápido do que a resposta natural G_x , de forma a facilitar o controle. Após a identificação dos parâmetros de G , será possível realizar a sua simulação, com e sem a presença da constante elástica K_x . Além disso, um controlador poderá ser projetado, de forma a proporcionar a performance desejada.

$$G = G_u G_x = \frac{T_{p_{max}}}{1 - s/p_{gu}} \times \frac{180/\pi}{s^2 I_x + s C_x + K_x}. \quad (3.9)$$

Uma característica notável de G é o fato dessa função possuir três pólos, devido ao polinômio de ordem 3 em s , no seu denominador. A localização dos pólos no plano complexo s pode ser utilizada para caracterizar a resposta em malha aberta, em respeito à precisão e velocidade, dois fatores essenciais ao sistema. O controlador C a ser proposto, têm então o objetivo de adaptar o sistema para uma outra função de transferência G_f , de malha fechada, com pólos e zeros adequados aos requisitos, de precisão e velocidade.

3.5 Considerações de simulação

Apesar de ser utilizada com sucesso para a simulação mecânica do drone em si, optou-se por não utilizar a engine física de Unity para o sistema motor-hélice. Um dos motivos foi a alta velocidade, da ordem de 5000 rpm, que a hélice precisaria ser simulada. Por servir de referência para o sentido de aplicação da força propulsora, o eixo vertical de cada hélice precisou permanecer fixo na mesma orientação, em relação ao frame do quadroto. A introdução da hélice na engine física de Unity, mesmo após a adição de uma *HingeJoint*, produziu pequenos erros nos eixos horizontais, e portanto foi descartada.

Além disso, como o tempo de amostragem da engine foi de 20 ms, seria muito custoso, entre um frame e outro de simulação, sobreamostrar e simular o comportamento elétrico do motor. Essa opção poderia ser considerada caso não fosse objetivo imprimir na tela o quadroto em tempo real. Se cada motor fizesse parte da engine, seriam necessários mais quatro *FixedJoint*, e ainda quatro *HingeJoint*, uma para cada hélice. A complexidade nesse caso seria muito grande para a engine simular de maneira fiel, e portanto nem o motor e nem a hélice fizeram parte da engine física; ambos foram marcados como *isKinematic*.

Todavia, ainda foi possível computar o diagrama não linear de blocos do conjunto motor-hélice. O mesmo pôde ser simulado tratando as funções não lineares como novos sinais. O integrador $\frac{1}{sJ}$ foi implementado pela acumulação do sinal de entrada, em cada frame de 20 ms.

4 IDENTICAÇÃO DO SISTEMA

A estratégia de identificação de sistemas utilizada, viabilizada a partir dos dados adquiridos no teste de balanço horizontal, consistiu em uma aproximação dos parâmetros da função de terceira ordem G , em formato representado em (4.1). Primeiramente, mensurou-se pontos da curva de Nyquist do sistema através do método do relé, como proposto por Astrom e Hagglund (1995); depois, identificou-se os parâmetros a e b que melhor se adequam aos pontos mensurados, pelo critério do menor erro quadrático; por fim, calculou-se os parâmetros c e d necessários para que a curva do modelo G , interceptasse o eixo real negativo na mesma frequência e ganho que os mensurados na prática.

$$G(s) = \frac{1}{as^3 + bs^2 + cs + d}. \quad (4.1)$$

4.1 Método do relé em malha fechada

O relé retroalimentado, como proposto por Astrom e Hagglund (1995), é um método de identificação de sistemas muito utilizado na indústria, geralmente aplicado nos chamados *autotuners*, responsáveis pela adaptação de controladores à variações no sistema. O método, na sua forma mais simples, é equivalente à um controlador *on-off*, e portanto pode ser aplicado na maioria dos processos industriais, sem que ocorra a perca de controlabilidade. Os dados adquiridos são a frequência e a amplitude de oscilação da saída, denominadas críticas no caso especial do controle *on-off*. Além do controle *on-off*, existem diversas variações do método do relé, que possibilitam a identificação da resposta em outras frequências do processo; pontos da sua curva de Nyquist. A variação do método do relé utilizada foi a do relé com histerese.

O relé com histerese baseia-se na defasagem, ou atraso, do sinal de atuação em relação ao erro de saída. O pseudocódigo a seguir mostra uma implementação do relé com histerese em um ambiente microcontrolado, similar a utilizada no programa real *euler_ctrl_dmp.ino*, disponível em Martins (2018). As variáveis r , y , e , u referem-se aos arrays de referência, saída, erro, e atuação, atualizados em cada período do loop. Com esse controlador, a saída y oscila em amplitude A e frequência ω específicas ao parâmetro histerese ϵ , e ao parâmetro amplitude de atuação D . O controle *on-off* ocorre no caso especial em que $\epsilon = 0$. As funções *read* e *write* são utilizadas apenas para ilustrar as operações de entrada e saída.

```

loop {
    k = k + 1
    y[k] = read()
    r[k] = 0
    e[k] = r[k] - y[k]
    if (e[k] > ε)
        u[k] = D
    else if (e[k] < -ε)
        u[k] = -D
    else
        u[k] = u[k-1]
    write(u[k])
}

```

A histerese ε é utilizada para atrasar o chaveamento u em relação ao momento de cruzamento de zero em e . Por se tratar de uma não-linearidade, o relé com histerese normalmente é aproximado por sua primeira componente fundamental da expansão de Fourier, como explica Astrom e Hagglund (1995). Dessa forma, uma componente senoidal de frequência ω na entrada do relé, sofre uma mudança de ganho e de fase através da chamada função descritiva. O ponto correspondente na curva de Nyquist em $G(j\omega)$, pode então ser identificado através de (4.2). Primeiramente, $|G(j\omega)|$ pode parecer independente de ε , entretanto ω é uma função de ε .

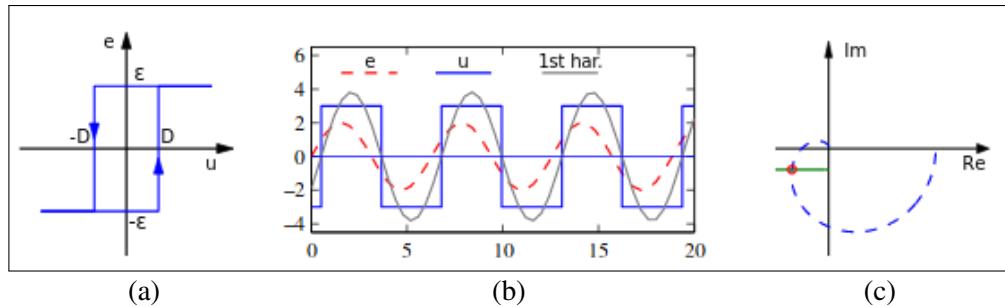
$$G(j\omega) = \frac{\pi}{4D} \sqrt{A^2 - \varepsilon^2} - i \frac{\pi\varepsilon}{4D}. \quad (4.2)$$

A amplitude D atua como um fator de escala para a amplitude A correspondente; a sua escolha é um compromisso entre a resolução de controle e os limites aceitáveis de oscilação. Quando $\varepsilon = 0$, a frequência ω_c de oscilação denomina-se frequência crítica, e o ganho $|G(j\omega_c)|$ denomina-se ganho crítico. A medida que ε é aumentado, menores frequências são identificadas, e a defasagem entre e e u aumenta. O ganho $|G(j\omega)|$ também aumenta. Um requisito importante para a validade de (4.2) é que o relé deve ser aproximadamente simétrico, isto é, permanecer no estado *on*, um intervalo de tempo próximo ao que permanece no estado *off*.

A Figura 21 ilustra o relé com histerese e a sua interpretação através do conceito de função descritiva. Em 21a a relação entre a entrada e a saída do relé com histerese é ilustrada.

Em 21b nota-se a defasagem e a mudança de amplitude entre a primeira harmônica do sinal do relé, u , e o sinal retroalimentado, e . Por fim, em 21c nota-se que o ponto identificado $G(j\omega)$ é a interseção entre a curva de Nyquist do sistema, e a reta horizontal definida por ϵ e D , os dois parâmetros de controle. Nota-se que a relação entre esses e a resposta, é específica ao sistema.

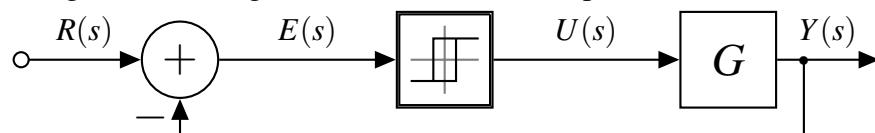
Figura 21 – Análise da função descritiva para um relé com histerese.



Fonte – Adaptado de Astrom e Murray (2009)

Ilustra-se na Figura 22 o diagrama de blocos da conexão do relé ao sistema G . Esta configuração é a mesma utilizada em diversos controladores. Como o relé é implementado em um processador discreto, a aquisição do sinal y em intervalos definidos, através de um conversor analógico-digital, está implícita. Em contraste, o atuador pode ser digital, já que necessita de somente dois valores. O programa *euler_ctrl_dmp.ino* foi adaptado para implementar o relé.

Figura 22 – Diagrama de conexão do relé por feedback de erro.



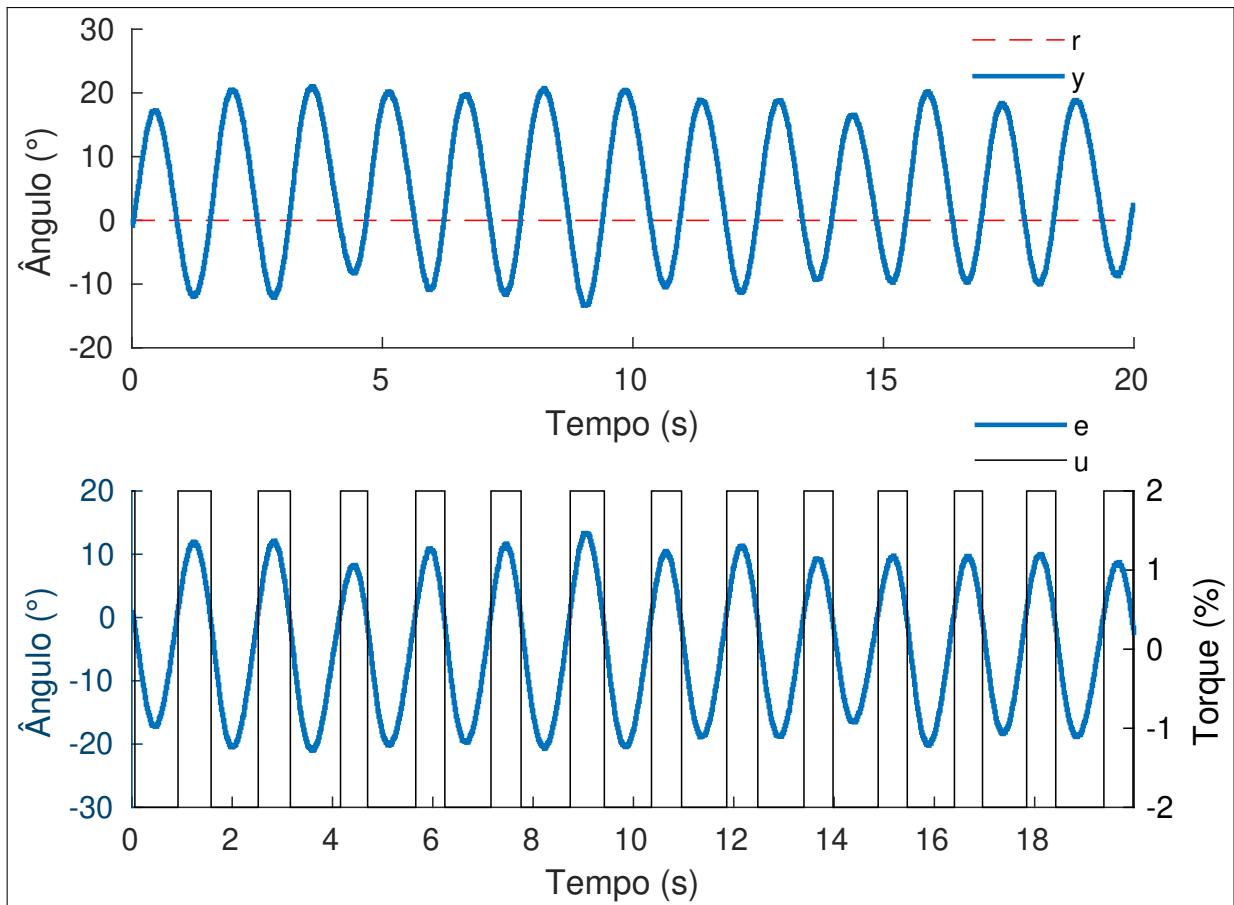
Fonte – Adaptado de Astrom e Hagglund (1995).

4.2 Considerações práticas

A maior adversidade prática encontrada, à manutenção de ciclos limitados no sinal y , foi a presença de frequências menores do que a frequência fundamental ω , algo não previsto por uma função de transferência linear. Em especial, um pequeno desequilíbrio entre os atuadores já foi suficiente para introduzir um *offset* constante no sinal de saída, e dificultar o tratamento simétrico idealizado para o relé, com o sinal $r = 0$. Para alguns valores de ϵ , não era possível alcançar ambos os pontos de chaveamento, e os sinais u e y permaneciam em um valor fixo.

Na Figura 23 apresenta-se os sinais durante a aplicação do relé sem histerese, com $D = 0,02$. Nota-se que o sinal y demora mais tempo em um ângulo positivo, resultando em período maior no estado *off*, do que no estado *on*. Além da oscilação característica do relé, nota-se o distúrbio constante, provocado pelo desequilíbrio dos atuadores. Uma possível explicação é a existência de assimetrias no circuito de alimentação, entre a bateria e cada ESC. Para reverter esse distúrbio, é necessário a redução do *offset* em y através de u , ou ainda, o aumento do *offset* em e através de r . Além disso, nota-se uma oscilação de baixa frequência na envoltória do sinal y , que torna ambígua a determinação de A . Atribui-se essa oscilação à imprecisão introduzida pelo protocolo analógico PWM do ESC, e talvez à uma baixa resolução do próprio ESC.

Figura 23 – Resultado do relé sem histerese no sistema teste.

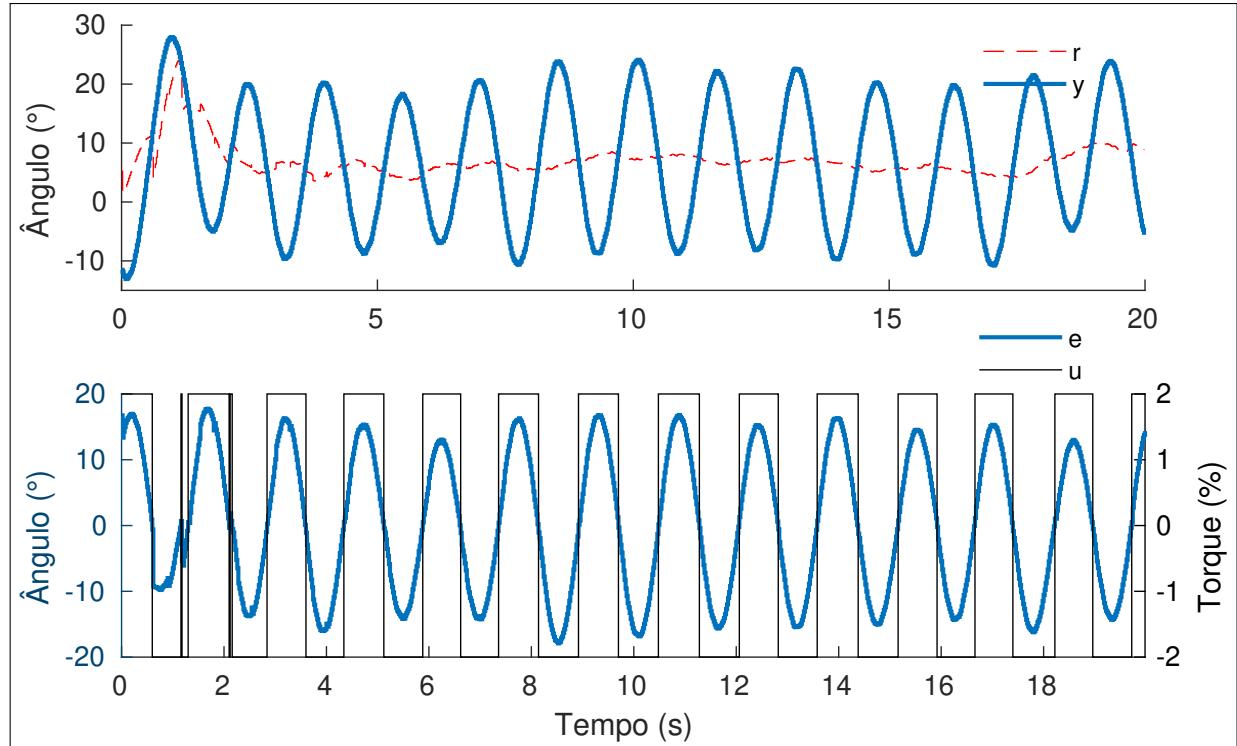


Fonte – O Autor.

A estratégia adotada para mitigar a assimetria dos atuadores foi a de atualizar o sinal r de forma a seguir y aproximadamente, com exceção na frequência de oscilação específica ao relé. Optou-se por calcular r a partir de um filtro pente aplicado em y , de frequência adaptativa correspondente ao período mais recente do sinal u .

Implementando-se o filtro pente de y para r , como definido por Smith (2006), viabilizou-se a manutenção de ciclos limitados e simétricos no sinal e , e por consequência em u , como pode ser observado na Figura 24. Ainda assim, o valor de A foi ambíguo, entre 10° e 20° , e teve que ser determinado estatisticamente. A estratégia utilizada será discutida na Seção 4.3.

Figura 24 – Resultado do relé sem histerese, com um filtro pente na referência.



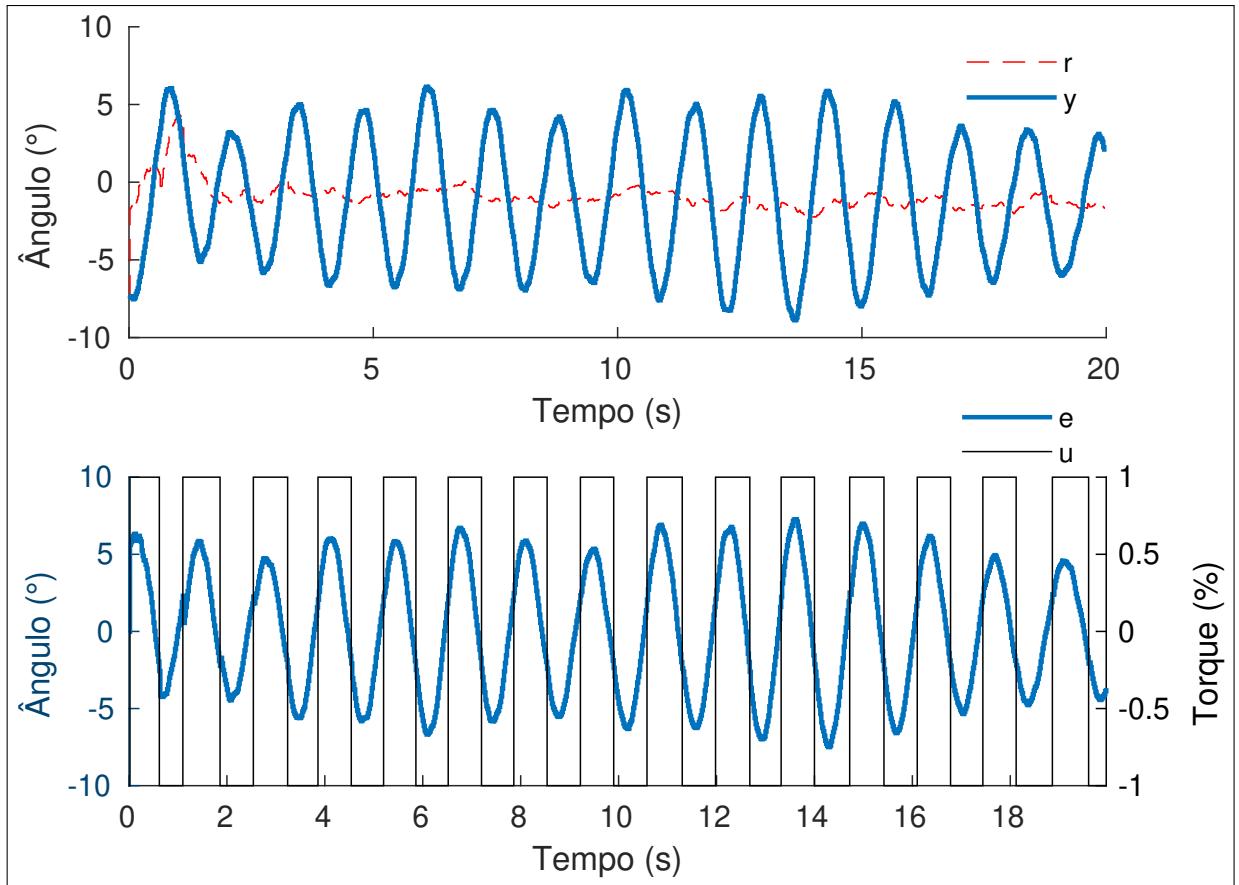
Fonte – O Autor.

A Figura 24 mostra o auxílio de um filtro pente de qualidade 8 dB aplicado em y . O valor de r em cada período é a média entre a saída atual do filtro e a saída do semi-ciclo anterior. Nota-se que praticamente eliminou-se o *offset* constante de e , porém a modulação em baixa frequência, da envoltória, foi apenas parcialmente removida. Em teoria, até mesmo ruídos poderiam ser removidos, porém o sinal y não possuiu ruídos notáveis. A única dificuldade dessa estratégia ocorre quando o sinal r alcança y antes que o período de oscilação seja estabelecido. Tal dificuldade pode ser observada em parte pela resposta transitória entre 0 e 2 segundos.

O resultado do relé com histerese $\epsilon = 2^\circ$, e amplitude $D = 0,01$, é apresentado na Figura 25. A adição da histerese provoca um atraso do sinal u em relação ao sinal e . Dessa vez os ruídos podem ser notados, especialmente quando o torque elástico é máximo, nas extremidades. Os ruídos periódicos são removidos com a subtração $e = r - y$, porém impulsos ruidosos, como o do pico anterior à 12 segundos, são reintroduzidos 90° à frente. Como o valor de ϵ é pequeno

em relação à amplitude A , é suficiente uma precisão na medida em instantes próximos a $e = 0$. Como essa é a região em que o ângulo y se move com maior velocidade, e portanto tem maior variação entre duas amostras, o valor real de ϵ pode possuir uma pequena oscilação, e portanto, assim como A , também teve que ser identificado estatísticamente.

Figura 25 – Resultado do relé com histerese, com um filtro pente na referência.



Fonte – O Autor.

Considerando que a amplitude A do sinal y precisa ser pequena de forma a preservar a aproximação $\sin(y) \approx y$, optou-se por realizar os testes com um valor de $D = 0,01$. Caso a amplitude fosse maior que 10° , como acontece para $D = 0,02$, a aproximação linear para o torque elástico devido as cordas auxiliares introduziria um erro de modelagem. Como o aumento de ϵ também provoca um aumento em A , restringiu-se a programar três valores de ϵ : 0° , 1° , e 2° ; nos quais constatou-se que A permanecia no intervalo entre 5° e 10° . Para reduzir as incertezas de medição e do modelo, repetiu-se o processo quatro vezes para cada valor de histerese programado, e utilizou-se a média dos valores de $G(j\omega)$ e ω identificados.

4.3 Medidas da curva de Nyquist

O método estatístico utilizado para identificar os valores medidos de A , ε e ω , foi programado na função *relay_info.m* e no script *tune.m*, disponíveis em Martins (2018). Cada medição foi composta de um ponto $G(j\omega)$ da curva de Nyquist, e a sua correspondente frequência. O intervalo completo de uma medição foi de 10 segundos, sobre o qual analisou-se cada semi-ciclo delimitado pelo sinal u , separadamente. Amplitude, histerese e frequência, em cada medição, foram aproximados como a sua mediana em todos os semi-ciclos. A mediana foi preferida como medida estatística devido a sua robustez em relação a média, que seria influenciada pela oscilação em ganho atribuída aos ESCs.

A Tabela 1 lista os valores médios encontrados para os três grupos de quatro medições. Nota-se que os valores médios para ε são aproximadamente $0,25^\circ$ maiores do que aqueles programados. O aumento em ε é natural visto o ambiente discreto em que é feito o processamento de controle; dificilmente os chaveamentos ocorrem exatamente nos valores programados. Ainda assim, as medidas de ε não deixam de ser precisas, com uma variação de no máximo 1,91% em relação ao valor médio de A . No ponto de frequência crítica, notou-se uma grande variação do valor de A , cerca de 13,7% do valor médio correspondente. Uma explicação é que o ângulo no qual a curva de Nyquist intercepta o eixo imaginário não é reto, e portanto a pequena variação em ε foi amplificada em A ; além disso, pela hipótese da baixa resolução do ESC, o mesmo contribui introduzindo uma variação desconhecida em D , e por conseguinte em A .

Tabela 1 – Valores médios para quatro medições de $G(j\omega)$.

D (%)	A ($^\circ$)	ε ($^\circ$)	ω (rad/s)	$\text{Re}\{G(j\omega)\}$ (rad)	$\text{Im}\{G(j\omega)\}$ (rad)
1.00	5.2578 ± 0.7191	0.2457 ± 0.1002	5.0704 ± 0.1494	-7.1987 ± 0.9824	-0.3368 ± 0.1374
1.00	5.7001 ± 0.1325	1.2687 ± 0.0834	4.8722 ± 0.0975	-7.6170 ± 0.1714	-1.7392 ± 0.1143
1.00	6.7490 ± 0.1536	2.2298 ± 0.0408	4.7504 ± 0.2301	-8.7317 ± 0.2172	-3.0566 ± 0.0560

Fonte – O Autor.

Os valores reais e imaginários de $G(j\omega)$, calculados a partir de (4.2) também estão presentes na Tabela 1. A unidade apresentada foi a de radianos, porém no design dos controladores é importante utilizar G na unidade $^\circ$, utilizada em y no algoritmo de controle. Observa-se que a variação de A propagou-se na parte real $\text{Re}\{G(j\omega)\}$, cuja variação também chega próxima à 13,7% na frequência crítica; nas demais frequências não ultrapassa 2,5%. O valor imaginário $\text{Im}\{G(j\omega)\}$, como em um paralelo ao ε , mostrou-se bem preciso em relação ao módulo de $G(j\omega)$, não ultrapassando 1,5%. Tal precisão é um sinal de que o período de

amostragem de 50 Hz é bem apropriado; além disso a frequência crítica é de aproximadamente 0,8 Hz, 62 vezes menor do que a amostragem. Como mencionado por Astrom e Hagglund (1995), além de útil na identificação de sistemas, o método do relé também é apropriado para a determinação do período de amostragem, em controladores digitais.

4.4 Aproximação estatística do modelo

Partindo do princípio de que o modelo G é de terceira ordem como apresentado na Seção 3.4, o ganho e a fase de $G(j\omega)$ podem ser calculados em função de ω , e dos coeficientes no denominador de G , a se determinar. A resolução não é trivial, devido a não linearidade entre os coeficientes e os pontos de Nyquist identificados. Inicialmente, é definido os vetores $\omega[\varepsilon]$ e $G[\varepsilon]$, correspondentes às medições da Tabela 1, enumeradas sequencialmente pela histerese. Para fins de cálculo, é útil ainda definir os vetores $\alpha[\varepsilon]$ e $\beta[\varepsilon]$, a partir de (4.3a) e (4.3b).

$$\alpha[\varepsilon] = \frac{+ \operatorname{Re}\{G[\varepsilon]\}^2}{(\operatorname{Re}\{G[\varepsilon]\}^2 + \operatorname{Im}\{G[\varepsilon]\}^2)^2} \times \frac{1}{\omega[\varepsilon]}, \quad (4.3a)$$

$$\beta[\varepsilon] = \frac{- \operatorname{Im}\{G[\varepsilon]\}^2}{(\operatorname{Re}\{G[\varepsilon]\}^2 + \operatorname{Im}\{G[\varepsilon]\}^2)^2}. \quad (4.3b)$$

Dessa forma, os coeficientes a e b , respectivos aos termos de terceira e segunda ordem em G , podem ser aproximados como as soluções de mínimo erro quadrático das equações lineares (4.4a) e (4.4b). Define-se aqui o vetor $\omega^2[\varepsilon]$ como aquele no qual os elementos de $\omega[\varepsilon]$ são elevados individualmente ao quadrado; o operador de diferenças Δ é utilizado para representar o vetor $\Delta x[k] = x[k+1] - x[k]$. E, assim como em Matlab (2018), utiliza-se o operador *pinv* ou \backslash para computar a pseudoinversa de Moore-Penrose, a solução de mínimo erro quadrático.

$$(\Delta\alpha)a = \Delta\omega^2 \implies a = \Delta\omega^2 \backslash \Delta\alpha, \quad (4.4a)$$

$$(\Delta\beta)b = \Delta\omega^2 \implies b = \Delta\omega^2 \backslash \Delta\beta. \quad (4.4b)$$

Os coeficientes c e d são utilizados para garantir que o modelo G possua frequência crítica ω_c e ganho crítico K_c iguais aos identificados na Tabela 1. A exatidão especificamente nesse ponto é importante porque é utilizado em diversos métodos de sintonia de controladores. Nesse propósito, (4.5a) e (4.5b) são aplicadas, garantindo a condição $G(j\omega_c) = -K_c$.

$$c = a\omega_c^2, \quad (4.5a)$$

$$d = b\omega_c^2 - K_c. \quad (4.5b)$$

Finalmente, o modelo G de atitude nas condições de teste, pode ser computado. Apresenta-se em (4.6) a função de transferência G , e a sua fatoração em G_u e G_x , para o exemplo específico do ângulo *roll*. Como discutiu-se no Capítulo 2.9, as cordas de sustentação serão eliminadas no sistema final; isso corresponde a fazer $K_x = 0$ em G_x , e recalcular $G = G_u G_x$.

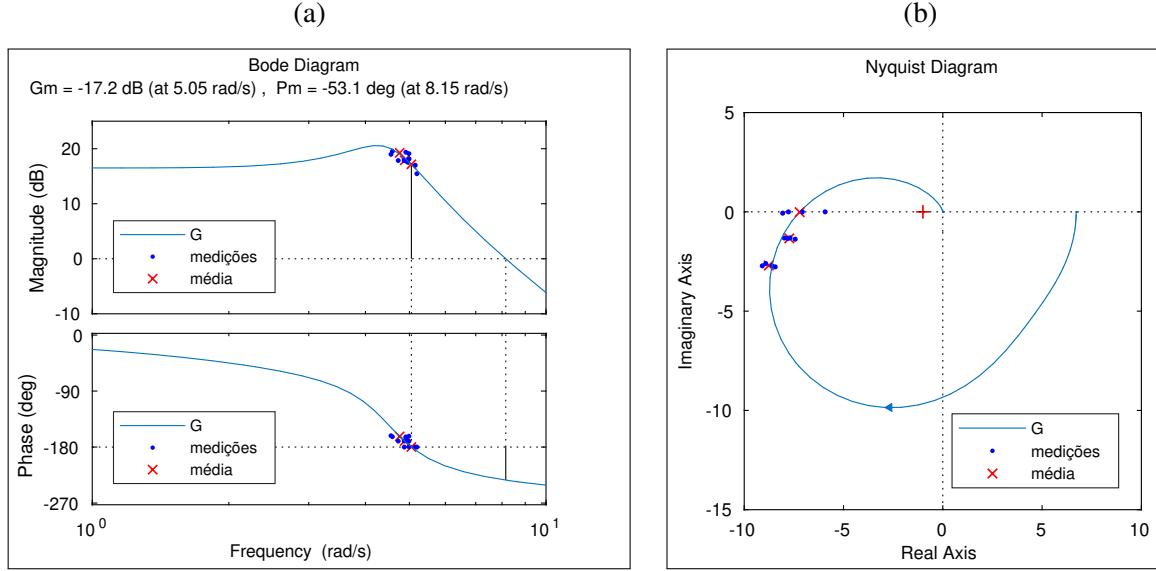
$$\begin{aligned} G(s) &= \frac{1}{as^3 + bs^2 + cs + d} = \frac{23932}{s^3 + 4.71s^2 + 25.5s + 62.18}, \\ G_u(s) &= \frac{T_{p_{max}}}{1 - s/p_{gu}} = \frac{1.791}{1 + s/3.044}, \\ G_x(s) &= \frac{180/\pi}{s^2 I_x + s C_x + K_x} = \frac{180/\pi}{0.01305s^2 + 0.02175s + 0.2666}. \end{aligned} \quad (4.6)$$

O modelo para o ângulo *pitch* pode ser identificado movendo-se as cordas de sustentação para a posição dos atuadores 1 e 3, e repetindo o processo de identificação, dessa vez com o relé nos atuadores 0 e 2. Outra opção é assumir que o modelo em *pitch* deve ser igual a G , por razão de simetria. No eixo *yaw*, o modelo pode ser aproximado a partir dos parâmetros obtidos. Tomando como exemplo o trabalho de Bresciani (2008), em um quadrotor similar, a inércia de rotação I_y deve ser próxima ao dobro de I_x . Outra possibilidade é desenvolver os controladores de atitude a partir da função G , e depois realizar o processo de identificação em *yaw* com o relé, enquanto os demais controladores estabilizam o drone no ar.

A congruência de G com os pontos medidos pode ser analisada a partir da Figura 26, que apresenta o seu diagrama de Bode em 26a, e a sua curva de Nyquist em 26b. O diagrama de Bode consiste em dispor a curva de ganho sobre a curva de fase, em função da frequência em escala logarítmica, aplicada em uma função de transferência. Nesse sentido, o diagrama de Bode e a curva de Nyquist apresentam a mesma informação sobre a resposta em frequência, porém no primeiro é possível analisar a escala de frequência, enquanto o segundo prioriza o formato da curva, cujo o parâmetro é a frequência.

Analizando a Figura 26 nota-se a proximidade entre os pontos medidos e a curva de Nyquist do modelo G identificado. As medições com histereses programadas maior que zero foram bem mais precisas do que a medição do ponto crítico, como pode-se notar pelo espaçamento em um grupo de medição e a sua média marcada por um x . Assim como foi projetado, a curva do modelo atravessa o eixo real exatamente no ponto crítico médio, também na frequência média de 5,05 rad/s. Através do modelo, outras informações como o valor final, a margem de ganho, e a margem de fase podem ser identificadas.

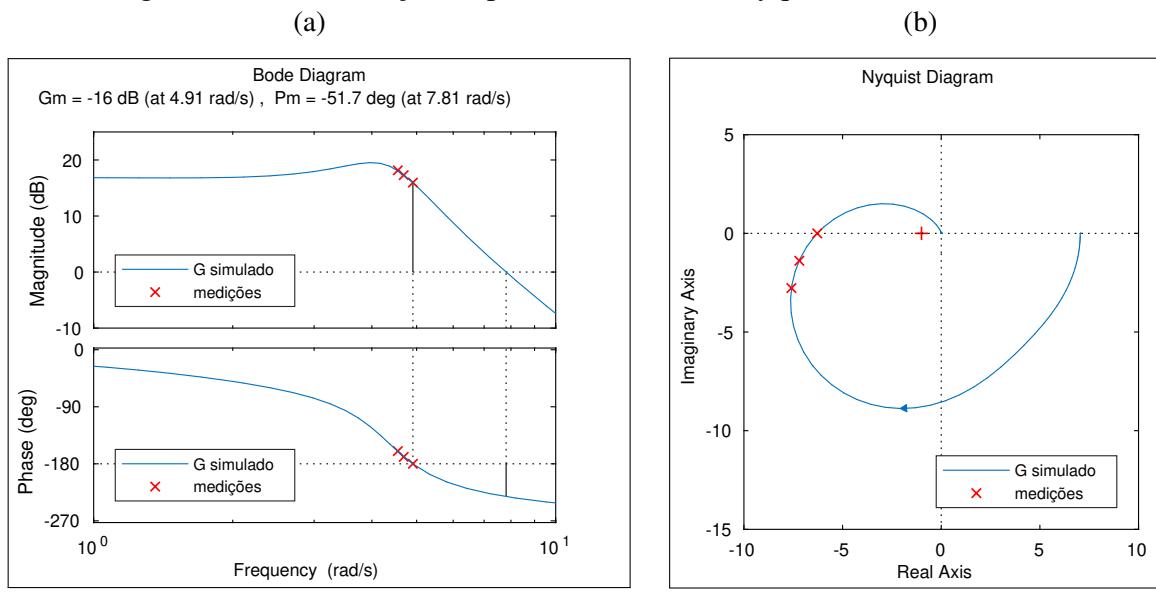
Figura 26 – Identificação de pontos na curva de Nyquist no sistema teste.



Fonte – O Autor.

Na Figura 27 apresenta-se o equivalente a Figura 26, porém para o modelo simulado, na situação de teste, através da engine física simulada, parametrizada com as propriedades do quadrotor. O modelo correspondente a simulação foi bem semelhante ao modelo G , apesar de uma pequena redução de ganho, e das frequências em geral. Ainda assim, os valores de margem de ganho e de margem de fase, ilustrados no diagrama de bode como G_m e P_m , ocorrem em frequências com desvio de apenas 2,7% e 4,2%, em respeito ao modelo G . Em ambos os modelos, para uma entrada contínua existirá uma saída limitada; esta é justamente a ação das cordas de segurança, desde que a entrada não seja o suficiente para provocar uma rotação completa.

Figura 27 – Identificação de pontos na curva de Nyquist no sistema simulado.

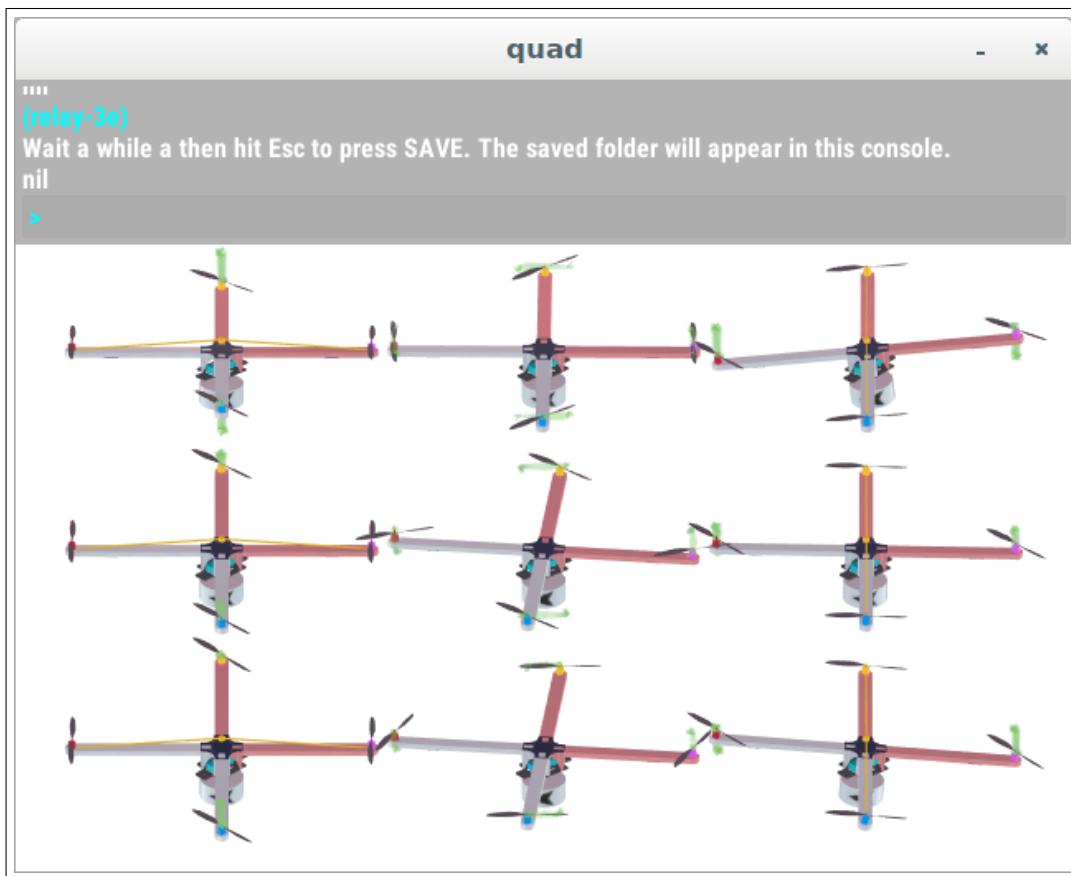


Fonte – O Autor.

4.5 Considerações de simulação

Observa-se na Figura 28 a repetição do esquema de teste no ambiente simulado. Cada ponto amarelo representa um suporte de sustentação, simulado através de uma *HingeJoint*. O comando “(relay-3e)” deve ser enviado ao console, e então três relés, cada com um valor de ϵ , serão instanciados para cada eixo. Após 10 segundos, o usuário deve apertar ESC e salvar os resultados obtidos. Os dados são salvos no formato .csv, na pasta *data*, no mesmo local do .exe do simulador. Como explicitado, o método de identificação serviu para validar o simulador.

Figura 28 – Validação do simulador pela resposta ao relé.



Fonte – O Autor.

Utilizando os parâmetros do drone na plataforma Unity, além de ser possível reproduzir a configuração de teste, também é possível extrapolar o vôo livre, quedas, e colisões. Dessa forma, o ambiente simulado pode ser utilizado para representar o modelo G , e realizar previsões sobre o seu comportamento, identificando e mitigando riscos previamente. Uma prática razoável pode ser, antes de se testar um controle no sistema real, verificar na simulação se o mesmo atinge as especificações, principalmente no que for em respeito a segurança contra quedas e acidentes.

5 DESIGN DOS CONTROLADORES

De acordo com Ogata (2010), sistemas de controle são projetados para realizar tarefas específicas, rotineiramente associadas à precisão, estabilidade relativa, e velocidade de resposta. Objetivos como o rastreamento de referência ou a rejeição de distúrbio, podem ter requisitos conflitantes, dependendo do sistema, e a escolha do controlador e os seus parâmetros deve realizar um compromisso entre tais tarefas.

Nesta monografia, concentrou-se os esforços no rastreamento de referência, com o objetivo de que a atitude do drone quadrotor obtivesse um tempo de resposta próximo à 1 segundo, com poucas oscilações. Também se objetivou um baixo sobressinal, a porcentagem ultrapassada do ângulo em relação à referência, na região de 2 à 5%. Quando não se conseguiu atingir os dois compromissos, priorizou-se a manutenção do tempo de resposta, a duração do início de um degrau de referência até o instante de alcance na saída, de 98% do seu valor final.

O controle de atitude do drone é a primeira peça no desafio de controle da sua posição. Na situação ideal, o controle de posição considera que a referência é seguida em um tempo muito curto, de forma a simplificar as equações de movimento do drone quadrotor. Nesse sentido, é feita a hipótese de que seja suficiente um baixo tempo de estabilização em 5% da referência. Ainda assim, o sobressinal relaciona-se a sensibilidade e a robustez do processo em malha fechada, e não deixa de ser um parâmetro importante.

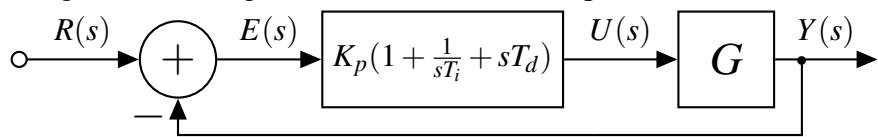
O modelo G utilizado a partir desse Capítulo já não considera mais a constante elástica identificada no teste do relé com as cordas. Dessa forma, o modelo fica definido a partir de (5.1), uma função de transferência em que se nota um pólo em zero, chamado pólo integrador, e outros dois pólos bem próximos, em -3,044 e -1,667. Como será apresentado, esse sistema de terceira ordem representa um maior desafio na parametrização dos controladores convencionais PID, geralmente aplicados em sistemas de primeira e segunda ordem. Os termos b_0 , a_1 , e a_2 serão utilizados para representar os coeficientes do modelo G .

$$\begin{aligned}
 G(s) &= G_u(s)G_x(s) \\
 &= \frac{1.791}{1+s/3.044} \times \frac{180/\pi}{0.01305s^2 + 0.02175s} \\
 &= \frac{23932}{s(s+3.044)(s+1.667)} \\
 &= \frac{23932}{s^3 + 4.7103s^2 + 5.0726s} = \frac{b_0}{s^3 + a_2s^2 + a_1s}
 \end{aligned} \tag{5.1}$$

5.1 O controlador PID

Em Ogata (2010), nota-se que mais da metade dos controladores industriais em uso empregavam esquemas de controle PID ou PID modificado. Naturalmente, uma primeira expectativa é se algum controlador PID será capaz de atingir a performance desejada, no rastreamento de referência, para a atitude do quadroto. Existem diversas estruturas e métodos de parametrização de controladores PID. O controlador PID na sua forma mais simples consiste no bloco da Figura 29, conectado ao sistema na configuração feedback de erro. A função de malha fechada é calculada por $G_f(s) = \frac{G(s)C(s)}{1+G(s)C(s)}$, em que $C(s) = \frac{U(s)}{E(s)}$ é o controlador.

Figura 29 – Diagrama do controlador PID por feedback de erro.



Fonte – Adaptado de (OGATA, 2010).

5.2 Método de sintonia de Ziegler Nichols

Em Haugen (2004), nota-se que um controlador razoável deve produzir uma resposta rápida e com estabilidade satisfatória, dois requisitos geralmente contraditórios. Apresenta-se o critério de Ziegler e Nichols como uma quantificação desse compromisso, ao classificar como satisfatório o decaimento de 1/4, entre sobressinais de oscilações consecutivas. Na tentativa de atingir tal decaimento, Ziegler e Nichols sugeriram regras para a sintonia de controladores PID, úteis quando o modelo matemático do sistema é desconhecido (OGATA, 2010).

No método de Ziegler e Nichols frequencial, o período de oscilação crítico $T_u = 2\pi/\omega_c$, e o inverso do ganho crítico $K_u = -1/G(j\omega_c)$, identificáveis com um relé sem histerese, são utilizados para calcular os parâmetros K_p , T_i , e T_d , através do esquema da Tabela 2, onde apresenta-se a correspondência entre o tipo de controlador e a sua sintonia.

Tabela 2 – Método de Ziegler-Nichols.

Controlador	K_p	T_i	T_d
P	$0.5K_u$	–	–
PI	$0.45K_u$	$T_u/1.2$	–
PD	$0.8K_u$	–	$T_u/8$
PID	$0.6K_u$	$T_u/2$	$T_u/8$

Fonte – Ziegler e Nichols (1993)

É importante notar que o ponto crítico identificado no sistema teste, no qual $K_u = 2.421 \times 10^{-3}$ e $T_u = 1.244$ s, não deve ser utilizado para o método de Ziegler Nichols, por que sofreu a influência da sustentação utilizada em teste. Os valores corretos correspondem ao modelo G modificado, em (5.1), no qual $K_u = 9.984 \times 10^{-4}$ e $T_u = 2.790$ s. A diferença dos parâmetros entre o teste e o vôo livre é tão significativa, que produziria controladores instáveis se o sistema G não fosse identificado e modificado para a situação prática. Os parâmetros calculados para o vôo livre, pelo método de Ziegler Nichols, são apresentados na Tabela 3.

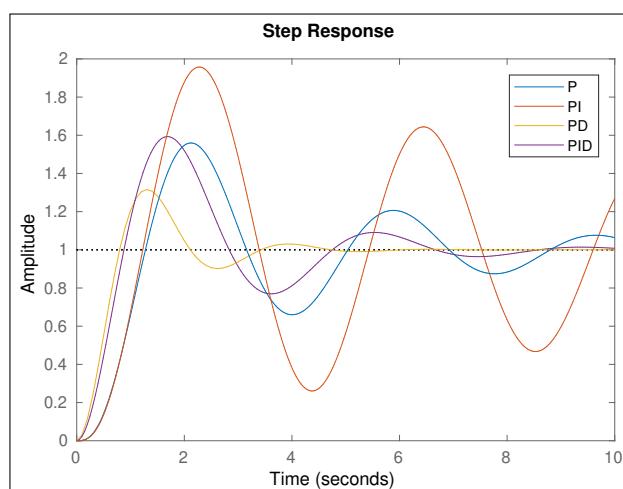
Tabela 3 – Método de Ziegler-Nichols aplicado.

<i>Controlador</i>	K_p	T_i	T_d
<i>P</i>	4.992×10^{-4}	–	–
<i>PI</i>	4.493×10^{-4}	2.323 s	–
<i>PD</i>	7.987×10^{-4}	–	0.3487 s
<i>PID</i>	5.990×10^{-4}	1.395 s	0.3487 s

Fonte – O Autor.

A performance dos controladores sintonizados, como na Tabela 3, pode ser prevista com a simulação do comando *step* no MATLAB, apresentada na Figura 30. Nota-se que apenas o controlador PD possui uma perfomance razoável, ainda que o sobressinal ultrapasse 30%. Observa-se que termo integrador I torna a resposta mais oscilatória. Além disso, como o modelo G já possuí um pólo em zero, a malha aberta não necessita do termo I para eliminar o erro de regime. O controlador P por si só não tem amortecimento satisfatório, o que é corrigido pelo termo D. No restante desta monografia, este controlador PD será referenciado como PD ZN.

Figura 30 – Resposta ao degrau com os controladores da Tabela 3.



Fonte – O Autor.

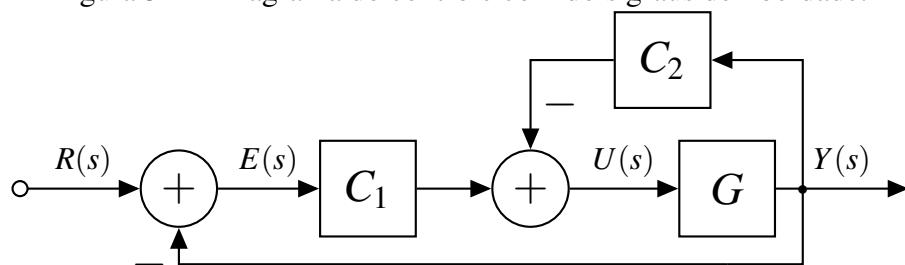
5.3 Método de sintonia por alocação de pólos

Como se observou na Seção 5.2, uma desvantagem do método de Ziegler-Nichols é que provoca uma resposta agressiva, com alto sobressinal. Ogata (2010) observa que as regras de sintonia de Ziegler-Nichols proporcionam apenas um ponto de partida para a sintonia fina. Todavia, esse processo pode ser demorado e nem sempre é possível alcançar a performance desejada. Um método de sintonia mais preciso, por utilizar o modelo da função de transferência, é o método de alocação dos pólos em malha fechada.

Como discutido no Capítulo 3, os pólos de uma função de transferência representam os seus “átomos” característicos, porque a resposta de uma função sem zeros é uma combinação linear da resposta de frações parciais com seus pólos (OGATA, 2010). Assim, no projeto por alocação de pólos, os controladores são escolhidos para que a função de malha fechada possua pólos desejáveis, que atendam aos requisitos de performance. Todavia, dependendo da estrutura do sistema e dos controladores, nem todos os parâmetros da função resultante podem ser escolhidos, ou ainda, a função resultante pode possuir zeros indesejáveis.

A estrutura utilizada para a alocação de pólos, distinguiu o feedback por dois caminhos, um pelo sinal de erro e outro pelo sinal de saída, de forma à parametrizar um controlador para cada sinal, respectivamente C_1 e C_2 . O diagrama de blocos com a conexão desses controladores e o processo G pode ser observado na Figura 31. A função de transferência resultante entre a referência e a saída, $G_f = \frac{Y(s)}{R(s)}$, é apresentada em (5.2). Tal estrutura denomina-se controle com dois graus de liberdade (ASTROM, 2002). Uma vantagem que possui sobre o feedback de somente o erro, é que o termo derivador pode ser separado do sinal r , que pode apresentar degraus associados a mudança de referência; a derivada do degrau é um impulso, que poderia provocar desgaste nos atuadores. Para o método de alocação de pólos, a vantagem principal é a maior independência entre o ajuste dos zeros e o ajuste dos pólos, em malha fechada.

Figura 31 – Diagrama do controle com dois graus de liberdade.



Fonte – O Autor.

$$G_f(s) = \frac{Y(s)}{R(s)} = \frac{GC_1}{1 + G(C_1 + C_2)} = \frac{b'_m s^m + b'_{m-1} s^{m-1} + \dots + b'_0}{s^n + a'_{n-1} s^{n-1} + \dots + a'_0}. \quad (5.2)$$

Em (5.2), também ilustra-se a representação de G_f como uma razão de polinômios, que pode ser determinada substituindo-se o valor de G , C_1 , e C_2 , em função dos seus parâmetros. Os coeficientes a'_i , determinados pela posição desejada dos pólos, resultam em um sistema de equações que relacionam os parâmetros de C_1 e C_2 , aos coeficientes do modelo G . Para que esse sistema de equações tenha uma solução possível e única, a ordem de cada controlador deve ser apropriada a ordem do modelo G .

Astrom (2002) apresenta a solução de (5.2) para controladores PI, PD e PID, atuando em processos de primeira e segunda ordem. Astrom observa que a complexidade necessária no controlador, é um reflexo da complexidade do processo; um controlador PI é suficiente à um sistema de primeira ordem, porém para um sistema de segunda ordem, um controlador PID é mais apropriado. Sistemas de ordem maior que dois são classificados como de alta ordem, nos quais para um controlador PID, existe uma restrição sobre a região alcançável dos pólos de malha fechada.

Quando a ordem do modelo G é alta, convém utilizar na alocação de pólos, uma modelo aproximado G_a , de menor ordem. Para o modelo G em questão, uma aproximação G_a pode ser obtida substituindo-se os dois pólos reais e negativos, p_{gu} e p_{gx} , por um único polo $p_{ga} = \frac{1}{1/p_{gu} + 1/p_{gx}}$. Esse tipo de aproximação é apresentada por Astrom e Murray (2009) como a soma das constantes de tempo, ou seja, o recíproco do polo. Por acumular os atrasos associados à cada polo, mostra-se uma melhor alternativa do que simplesmente desconsiderar o polo menos dominante; já que a relação $\frac{p_{gu}}{p_{gx}}$ é de apenas $\frac{-3.044}{-1.667} = 1.826$. Ajustando-se o ganho K_{ga} para compensar a redução dos pólos, obtém-se a aproximação G_a a partir de (5.3). Os termos \tilde{a}_1 e \tilde{b}_0 serão utilizados para representar os coeficientes do modelo G_a .

$$\begin{aligned} G_a(s) &= \frac{K_{ga}}{s(1 - s/p_{ga})}, \\ &= \frac{\left| \frac{p_{ga}}{p_{gu}p_{gx}} \right| K_g}{s(1 - s/p_{ga})} = \frac{\left| \frac{-1.077}{5.074} \right| 23932}{s(1 + s/1.077)}, \\ &= \frac{5080.8}{s(s + 1.077)} = \frac{\tilde{b}_0}{s^2 + \tilde{a}_1 s}. \end{aligned} \quad (5.3)$$

5.4 Pólos notáveis

Nem todos os pólos são criados iguais. Em termos da localização no plano complexo s frequencial, e em respeito a resposta no tempo, pode-se afirmar que pólos distantes do eixo imaginário são rápidos, e pólos distantes do eixo real são oscilatórios. Além disso, pólos no semi-plano direito são instáveis, porque a sua resposta temporal não estabiliza em um valor final. Na aplicação, projetada para o rastreamento de referência, apenas certos grupos de pólos produzem a performance desejada, com baixo tempo de resposta e pouca oscilação.

Primeiro, considerou-se a alocação em pólos complexos dominantes, projetados por um controlador PID com dois graus de liberdade. Nessa estratégia, foi possível reduzir o sobressinal em relação ao controlador PD ZN, embora ao custo de um pequeno aumento no tempo de resposta. Em alternativa, utilizou-se um compensador avanço de fase no bloco C_1 , e um derivador no bloco C_2 , projetando-se os pólos para a otimização do índice ITAE para a entrada degrau (DORF; BISHOP, 2000), de forma a se obter o tempo de resposta próximo à 1 s, e o sobressinal menor do que 5%, que foram estipulados como satisfatórios.

5.4.1 Pólos complexos dominantes

De posse do modelo G ou da aproximação G_a , os controladores C_1 e C_2 podem ser escolhidos de forma que a equação $G(C_1 + C_2) = -1$ possua pólos desejáveis. Um exemplo pode ser os dois pólos complexos de (5.4), especificados por um coeficiente de amortecimento ζ , e uma frequência natural ω_n . O valor ζ normalmente deve situar-se entre 0.4 e 0.8, pois $\zeta < 0.4$ resulta em um sobressinal excessivo, e $\zeta > 0.8$ resulta em um demorado tempo de resposta (OGATA, 2010). A frequência natural ω_n atua como um fator de escala dos pólos e do tempo de resposta, e deve ser escolhida observando-se as frequências de operação do sistema original.

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \implies s = (-\zeta \pm \sqrt{1 - \zeta^2})\omega_n. \quad (5.4)$$

Nessa estratégia de alocação, posiciona-se a parte real dos dois pólos complexos, $-\zeta\omega_n$, em uma frequência próxima a dos pólos dominantes no sistema em malha aberta; quando o polinômio de malha fechada possuir ordem maior que 2, o restante dos pólos deve ser afastado para uma região de frequências pelo menos 5 vezes maior. Ogata (2010) cita que a relação entre a parte real dos pólos, quando maior que 5, pode ser utilizada como critério para desprezar os pólos mais rápidos, porque nessa situação os pólos mais lentos dominam o tempo de resposta.

5.4.2 Pólos otimizados à redução do índice ITAE

O índice ITAE, integral do tempo multiplicado pelo erro absoluto, pode ser calculado através de (5.5). Apesar da sua definição ser através de uma integral de duração infinita, na prática o valor $e(t)$ se aproxima de zero mais rápido do que o valor de t aumenta, e a integral pode ser aproximada até o instante de estabilização em 2% do valor final.

$$ITAE = \int_0^{\infty} t |e(t)| dt. \quad (5.5)$$

O ITAE consiste em uma métrica da acumulação de erro, correspondente ao rastreamento de referência, ponderado pelo tempo em que o erro ocorre. Dessa forma, oscilações após o tempo de resposta possuem um peso maior do que o erro de atraso inicial. Os polinômios característicos que possuem o menor ITAE são tabelados de acordo com o tipo do sinal de referência para os quais foram calculados (DORF; BISHOP, 2000). Para uma referência do tipo degrau, os polinômios de menor ITAE de ordem 1 à 5, estão apresentados na Tabela 4.

Tabela 4 – Otimização do ITAE para referência degrau.

$$\begin{aligned} & s + \omega_n \\ & s^2 + 1.4\omega_n s + \omega_n^2 \\ & s^3 + 1.75\omega_n s^2 + 2.15\omega_n s + \omega_n^3 \\ & s^4 + 2.1\omega_n s^3 + 3.4\omega_n^2 s^2 + 2.7\omega_n^3 s + \omega_n^4 \\ & s^5 + 2.8\omega_n s^4 + 5.0\omega_n^2 s^3 + 5.5\omega_n^3 s^2 + 3.4\omega_n^4 s + \omega_n^5 \end{aligned}$$

Fonte – Adaptado de Dorf e Bishop (2000).

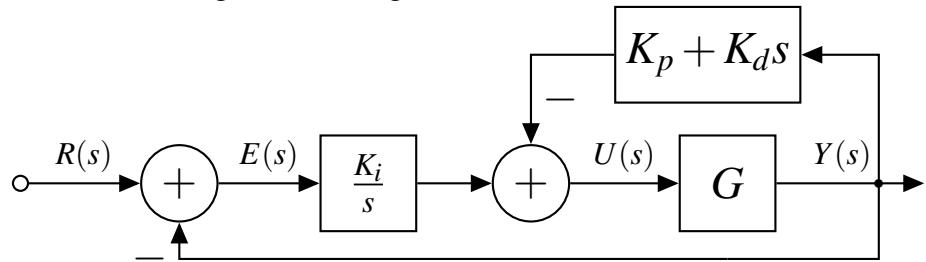
Em polinômios de ordem 2 nota-se que, observando a equação (5.4), o coeficiente de amortecimento $\zeta = 0.7$ corresponde ao menor ITAE. De fato, observa-se que o índice ITAE é influenciado pelo equilíbrio entre a oscilação e o tempo de resposta, assim como ζ . A frequência ω_n continua apenas como um fator de escala de tempo, de forma que, os coeficientes de um polinômio na Tabela 4 podem ser comparados entre si normalizando-os para $\omega_n = 1$. À medida que ω aumenta, para uma mesma relação de coeficientes, menor será o ITAE calculado, porque o formato de resposta permanece o mesmo, a não ser pela menor escala de tempo.

Como já foi observado no Capítulo 3, quanto maior é a ordem do denominador maior é o atraso associado, e assim maior será o ITAE. Embora seja possível utilizar C_1 e C_2 para o cancelamento de pólos em G , e a consequente redução da ordem do polinômio, tal estratégia introduz, na função resultante G_f , zeros que não são considerados pela otimização referente a Tabela 4. Por esse motivo, o polinômio de ordem 4, uma ordem maior que G , será utilizado.

5.5 O controlador I-PD

A primeira alternativa considerada, ao controlador PD por *feedback* de erro, foi o controlador PID com dois graus de liberdade. Como se vê na Figura 39, optou-se por utilizar o termo I no bloco C_1 , e o termo PD no bloco C_2 . O motivo da introdução do termo I no *feedback* de erro, apesar do sistema G já possuir um integrador, foi devido uma observação feita no simulador, de que um distúrbio do tipo degrau no sistema, como por exemplo uma corrente de vento, pode alterar a dinâmica da malha aberta e provocar um erro permanente em malha fechada. O controlador PD então, foi utilizado no bloco C_2 para balancear o efeito do integrador, a partir do sinal relativamente suave da saída, com o objetivo de reduzir a oscilação em relação aos controladores de Ziegler-Nichols considerados na Seção 5.2.

Figura 32 – Diagrama do controlador I-PD.



Fonte – O Autor.

Em um primeiro momento, calculou-se a função de malha fechada G_f utilizando o controlador I-PD, a partir do modelo G de 3ºordem; o resultado pode ser visto em (5.6).

$$G_f(s) = \frac{\frac{b_0}{s^3+a_2s^2+a_1s} \frac{K_i}{s}}{1 + \frac{b_0}{s^3+a_2s^2+a_1s} \frac{K_d s^2 + K_p s + K_i}{s}} = \frac{b_0 K_i}{s^4 + a_2 s^3 + (a_1 + b_0 K_d) s^2 + b_0 K_p s + b_0 K_i}. \quad (5.6)$$

Como pode-se observar nos coeficientes do denominador de G_f , apenas os termos de ordem 0, 1 e 2 podem ser ajustados. O termo de ordem 3 é fixado apenas à constante a_2 do sistema, o que limita as possibilidades de alocação dos pólos. Por esse motivo, optou-se por utilizar a aproximação G_a de segunda ordem, e projetar pela aproximação da malha fechada G_{fa} , em (5.7). Feita a aproximação, a alocação por pólos dominantes pode então ser utilizada.

$$G_{fa}(s) = \frac{\frac{b_0}{s^2+\tilde{a}_1s} \frac{K_i}{s}}{1 + \frac{b_0}{s^2+\tilde{a}_1s} \frac{K_d s^2 + K_p s + K_i}{s}} = \frac{b_0 K_i}{s^3 + (\tilde{a}_1 + b_0 K_d) s^2 + b_0 K_p s + b_0 K_i}. \quad (5.7)$$

Escolhendo-se um par de pólos de dominantes e um terceiro pôlo afastado 10 vezes no eixo real, é possível definir os parâmetros necessários ao controlador I-PD através de (5.7), e se obter o polinômio de malha fechada projetado. Os parâmetros de três controladores, para diferentes escolhas de ω_n e ζ , podem ser observados na Tabela 5. O valor $\zeta = 0,7$ foi inicialmente testado por representar o polinômio de menor ITAE, de segunda ordem. Depois, considerou-se $\zeta = 0,4$, que está no limite dos valores recomendados por Ogata (2010). Os valores de ω_n , com exceção de 3,08, foram escolhidos para que a parte real dos pólos dominantes de malha fechada $-\zeta\omega_n$, fosse próxima ao pôlo de malha aberta da aproximação G_a , em -1,077.

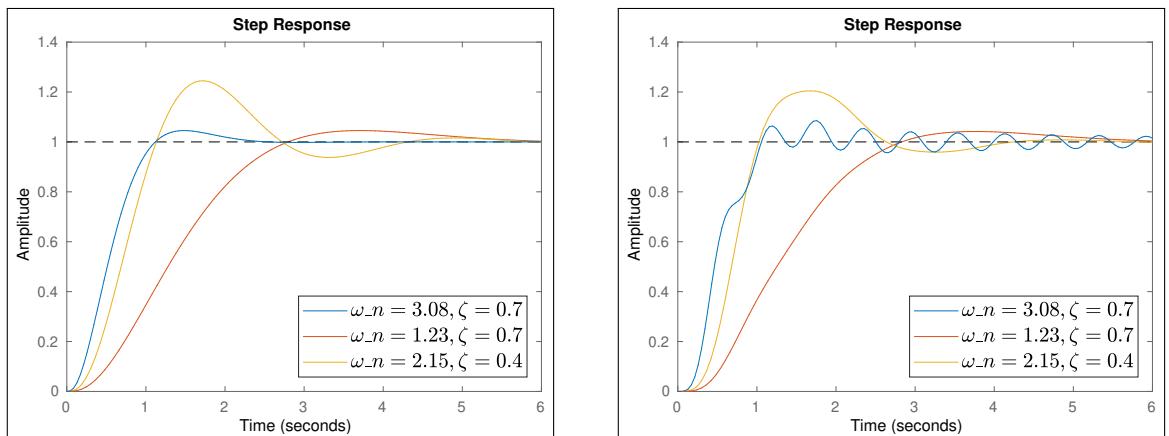
Tabela 5 – Controladores I-PD por alocação de pólos dominantes.

Controlador	K_p	T_i	T_d
$\omega_n = 3.08, \zeta = 0.7$	4.992×10^{-3}	0.5014 s	0.2422 s
$\omega_n = 1.23, \zeta = 0.7$	3.220×10^{-3}	1.253 s	0.5661 s
$\omega_n = 2.15, \zeta = 0.4$	3.834×10^{-3}	0.4875 s	0.4753 s

Fonte – O Autor.

O resultado dos controladores da Tabela 5 foi simulado pelo *step* no MATLAB como na Figura 33a, para a aproximação G_a , e na Figura 33b, para o modelo original G . É interessante notar que em média os dois resultados são bem parecidos, especialmente quando $-\zeta\omega_n$ foi 80% do pôlo em G_a . Somente no caso em que ω_n foi 3,08, e $-\zeta\omega_n$ foi o dobro do pôlo, é que uma sensibilidade não capturada em G_a provocou oscilações na malha fechada, esperada ao sistema real, com o modelo G . O controlador preferido foi aquele em que $\zeta = 0,4$, porque apresentou um tempo de resposta próximo a 1 segundo, com curta oscilação, apesar do sobressinal de 20%.

Figura 33 – Resposta ao degrau com os controladores da Tabela 5.

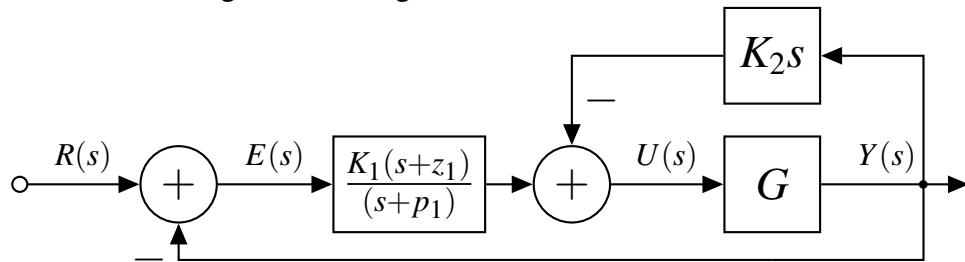


Fonte – O Autor.

5.6 O controlador PDF-D

Os dois controladores projetados apresentaram características semelhantes, ambos cumprindo o requisito de tempo, porém o controlador PD ZN com um sobressinal de 30% e o controlador I-PD com um sobressinal de 20%. Para o controle de atitude, as restrições de tempo são bem maiores do que as de sobressinal, e nesse aspecto ambos controladores são satisfatórios. Ainda assim, em uma tentativa final de aumentar a precisão sem perder em velocidade, a estrutura PD inicial foi expandida a dois graus de liberdade, atuando com dois derivadores distintos para os sinais de erro e de saída. O diagrama de blocos da estrutura é apresentado na Figura 34.

Figura 34 – Diagrama do controlador PDF-D.



Fonte – O Autor.

Como se observa na Figura 34, o bloco C_1 consiste em um ganho, um pólo, e um zero, definidos por três parâmetros, K_1 , p_1 , e z_1 ; já o bloco C_2 consiste em apenas um derivador puro, de ganho K_2 . A idéia fundamental dessa estrutura é tornar possível o aumento de K_1 sem perca de estabilidade, através da ação dos dois zeros, um em $-z_1$, e outro em 0, um derivador. Além disso, o impacto do degrau de referência é dosado através do pólo em $-p_1$, que atua como um filtro de primeira ordem, suavizando a ação derivativa. Por esse motivo, a estrutura de controle C_1 é conhecida como PDF, o controlador PD com um filtro no derivativo. Na literatura, tal controlador ainda é encontrado como um compensador avanço de fase (OGATA, 2010). Uma maneira mais convencional de definir o controlador PDF é através de (5.8).

$$C_1(s) = K_p \frac{sT_d + 1}{sT_f + 1}. \quad (5.8)$$

Para fins de comparação, também é útil definir C_2 à partir de (5.9). Dessa forma, as constantes de tempo e o ganho K_p podem ser relacionadas aos controladores prévios.

$$C_2(s) = sK_p T_{dy}. \quad (5.9)$$

Resolvendo a malha fechada para o controlador PDF-D, obtém-se a equação (5.10). Uma vantagem inicial dessa estrutura em relação ao I-PD, é que todos os coeficientes no denominador podem ser ajustados, habilitando a alocação por pólos otimizados ao critério ITAE, para a referência degrau. A única ressalva é que a malha fechada G_f terá um zero adicionado por C_1 , e portanto pode não ter a otimização esperada. Astrom (2002) avalia a influência de um zero em um sistema de segunda ordem e conclui que, para um zero de módulo 5 vezes maior que a frequência natural do sistema, o efeito do zero é pequeno; caso contrário, o zero adiciona sobressinal se for negativo, ou até mesmo um pico contrário a referência, se for positivo.

$$\begin{aligned} G_f(s) &= \frac{\frac{b_0}{s^3+a_2s^2+a_1s} \frac{K_1(s+z_1)}{s+p_1}}{1 + \frac{b_0}{s^3+a_2s^2+a_1s} \left(\frac{K_1(s+z_1)}{s+p_1} + K_2 s \right)}, \\ &= \frac{b_0 K_1 (s+z_1)}{s^4 + (a_2 + p_1)s^3 + (a_2 p_1 + a_1 + b_0 K_2)s^2 + (a_1 p_1 + b_0 K_1 + b_0 p_1 K_2)s + b_0 K_1 z_1} \end{aligned} \quad (5.10)$$

Igualando-se os coeficientes do denominador de (5.10) aos do polinômio de 4º ordem na Tabela 4, de otimização do ITAE, chega-se à um sistema linear de equações, cuja solução, especificada uma frequência natural ω_n , é apresentada no sistema de Equações .

$$p_1 = 2.1\omega_n - a_2, \quad (5.11a)$$

$$K_2 = (3.4\omega_n^2 - a_1 - a_2 p_1)/b_0, \quad (5.11b)$$

$$K_1 = (2.7\omega_n^3 - a_1 p_1 - b_0 p_1 K_2)/b_0, \quad (5.11c)$$

$$z_1 = \omega_n^4/(b_0 K_1). \quad (5.11d)$$

Substituindo-se (5.11a) em (5.11b) e assim sucessivamente, pode-se calcular o valor de cada parâmetro do controlador, escolhido um valor de ω_n . Por outro lado, fazendo o valor $z_1 = 5\omega_n$, no qual espera-se uma baixa influência do zero, é possível calcular a frequência exata sobre a qual este é 5 vezes maior, através da solução das raízes do polinômio de terceira ordem correspondente, onde são descartadas as soluções complexas e utilizada a raiz real.

Apesar da análise de Astrom (2002) se referir à uma função de 2º ordem, e o polinômio alocado ter ordem 4, os pólos de malha fechada otimizados ao ITAE são próximos uns aos outros, de maneira que o sistema assim projetado comporta-se semelhante à uma função de ordem 2, após um curto tempo de atraso. Resolvendo-se a equação de terceira ordem com a ajuda do MATLAB, identificou-se a frequência $\omega_n = 4.652 \text{ rad s}^{-1}$, como a qual em que a influência do zero foi quase imperceptível, por se localizar em $z_1 = 5\omega_n$.

O resultado da alocação para esse e outros três valores de ω_n pode ser analisado a partir da Tabela 6. Nota-se que até então os valores de K_p não ultrapassavam a ordem de 10^{-3} . Dessa vez, apenas para $\omega_n = 2.78 \text{ rad s}^{-1}$, encontra-se K_p nessa ordem. Como referência, a margem de ganho do modelo G é $K_u = 9.984 \times 10^{-4}$, indicando que se a ação derivativa fosse removida, nenhum desses controladores seria estável. O valor T_d talvez seja mais peculiar ainda, visto que chega a ser negativo para $\omega_n = 8.35 \text{ rad s}^{-1}$, e $\omega_n = 6.39 \text{ rad s}^{-1}$. Nesse caso o termo derivativo em C_1 atua contra a variação de referência, e tende a acelerar a saída. Em contrapartida, T_{dy} é sempre positivo e suficiente, para retardar a ação de C_1 e garantir a estabilidade em G_f .

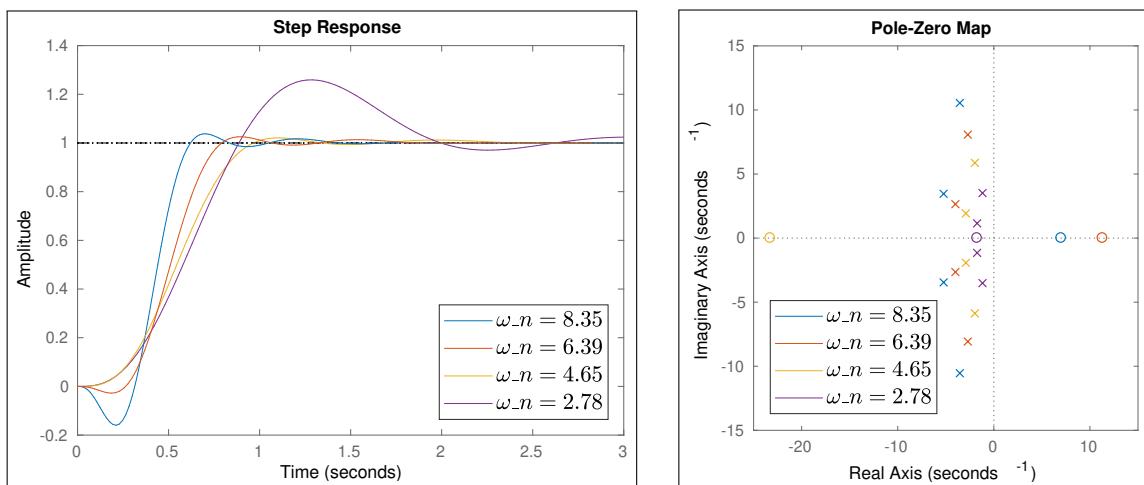
Tabela 6 – Controladores PDF-D por alocação de pólos dominantes.

Controlador	K_p	T_f	T_d	T_{dy}
$\omega_n=8.35 \text{ rad s}^{-1}$	2.031×10^{-1}	0.0780s	-0.1427s	0.0353s
$\omega_n=6.39 \text{ rad s}^{-1}$	6.975×10^{-2}	0.1148s	-0.0884s	0.0556s
$\omega_n=4.65 \text{ rad s}^{-1}$	1.957×10^{-2}	0.1977s	+0.0430s	0.0954s
$\omega_n=2.78 \text{ rad s}^{-1}$	2.501×10^{-3}	0.8840s	+0.5742s	0.2657s

Fonte – O Autor.

As respostas ao *step*, da malha fechada pelos controladores PDF-D, podem ser analisadas na Figura 35a. Nota-se que para o valor de $\omega_n = 2.78 \text{ rad s}^{-1}$, z_1 é apenas $0.626\omega_n$, e portanto age no transitório aumentando o sobressinal; em uma performance similar aos controladores prévios, PD ZN e I-PD. Na Figura 35b ilustra-se o mapa de pólos e zeros no plano complexo s . Observa-se que a posição relativa do zero está diretamente relacionada ao tipo de resposta, assim como previsto para sistemas de 2º ordem por Astrom (2002).

Figura 35 – Propriedades previstas aos controladores da Tabela 6.

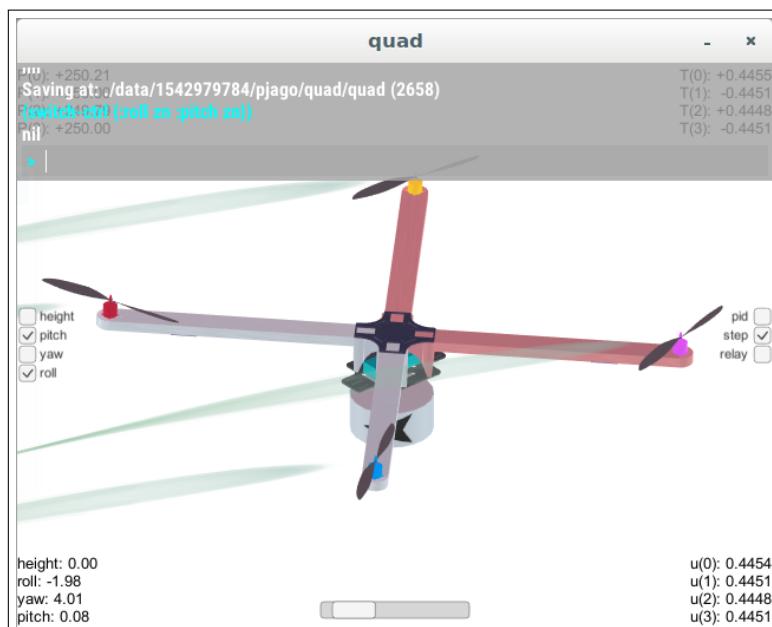


Fonte – O Autor.

5.7 Considerações de simulação

Desenvolvidos os três controladores, o próximo passo foi analisar o seu sinal de atuação u através do ambiente simulado. Na simulação, é possível habilitar somente os eixos de atitude, e deixar os valores de yaw e altura inalterados. Foi nessa configuração que os resultados finais nesta monografia foram obtidos. Na extrema esquerda da Figura 36, pode-se observar os marcadores utilizados para especificar quais eixos são habilitados. Na extrema direita, existem outros marcadores que dizem respeito ao tipo de referência gerada automaticamente, e ao controle em atuação. Caso o marcador 'pid' não esteja marcado, o controlador PDF-D será utilizado. O controlador PD ZN só pode ser escolhido pelo comando *switch-ctrl* na Figura 36.

Figura 36 – Configuração do trem de pulsos em *pitch* e *roll*.



Fonte – O Autor.

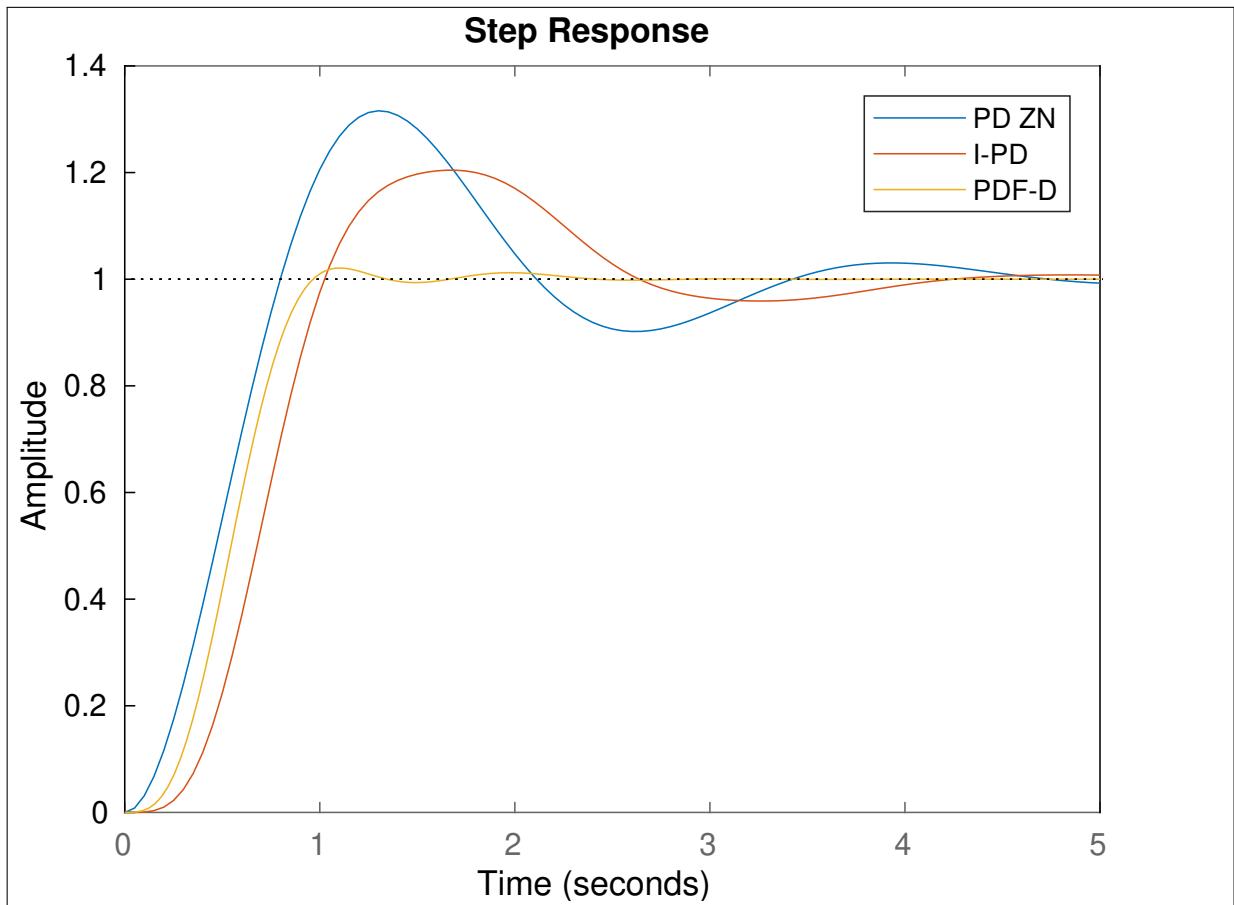
O marcador *step* inicia um trem de pulsos do tipo degrau, espaçados de 5 segundos, nos eixos habilitados. Cada degrau é associado à uma rotação na seta de orientação abaixo do drone, a qual o mesmo precisa nivelar. O traçado verde representa o caminho percorrido por cada motor. Por default, a câmera que apresenta a visão do simulador acompanha a translação do drone, de forma a parecer que esse permanece estático no centro da tela, a não ser por sua atitude. Caso o usuário desejar observar o movimento do drone por um referencial estático, pode segurar o botão esquerdo do mouse, em qualquer lugar vazio de tela branca, e ali o referencial será fixado. Para aumentar o campo de visão, a barra inferior pode ser movida toda para direita.

6 RESULTADOS DE SIMULAÇÃO

As três estruturas de controle, PD, I-PD, e PDF-D, podem ser comparadas por sua resposta projetada em malha fechada, na Figura 37. Apenas a melhor sintonia encontrada de cada estrutura é considerada. Nota-se que a resposta do controlador PD se atencia em relação aos outros, porém estabiliza na referência somente após 4 segundos. O controlador I-PD também teve oscilação considerável, e apesar de possuir um menor sobressinal, foi 0,2 s mais lento. Apenas o controlador PDF-D foi capaz de cumprir ambos os requisitos de velocidade e precisão. Seu tempo de resposta foi um pouco menor que 1 s e o sobressinal não ultrapassou 5%.

É importante notar que esse resultado não leva em conta o modelo não-linear dos motores, pois utiliza apenas a aproximação do modelo G por pequenos sinais. Outra limitação, dessa vez associada ao comando *step* do MATLAB, é que não apresenta o sinal de atuação u que produziu a referente saída y . O resultado no simulador, que utiliza o modelo não-linear, e também simula o sinal de atuação u , será analisado a seguir.

Figura 37 – Comparação da malha fechada pelos controladores propostos.

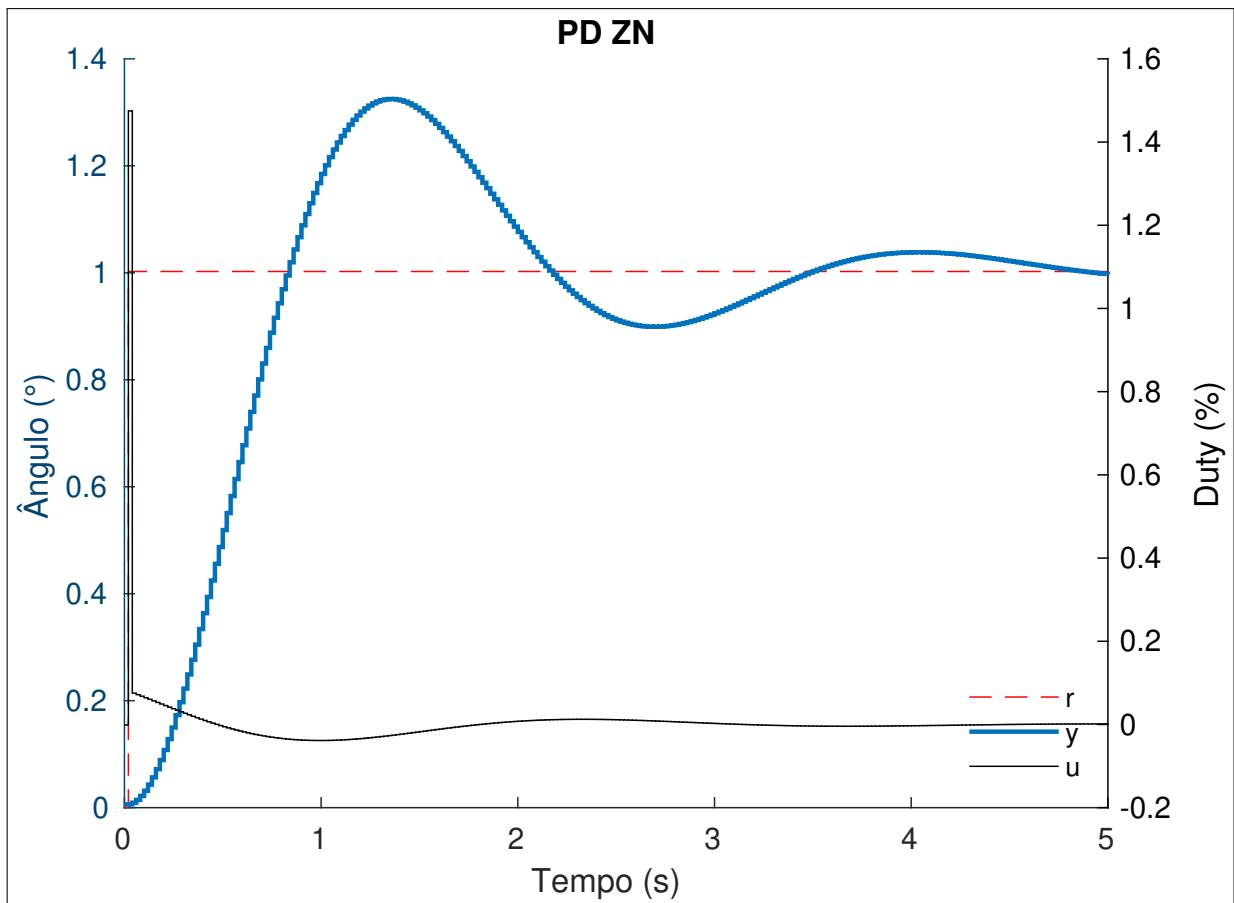


Fonte – O Autor.

Na Figura 38, mostra-se os gráficos da atuação u e da resposta y para o degrau r . Primeiramente, nota-se a semelhança entre y e o que foi projetado na Figura 37, no que diz respeito ao formato das curvas. O sobressinal, assim como no projeto também foi próximo a 30%. A temporização da resposta também foi semelhante. Ou seja, mesmo dadas as diferenças entre o modelo simulado e modelo da função de transferência, os resultados de ambos foram bem semelhantes, reafirmando a validação realizada no Capítulo 4 pelo método do relé.

Em preto, pode-se finalmente analizar o sinal de atuação u , e imediatamente nota-se o seu pico exagerado no instante do degrau em $t = 0$, correspondente a ação derivativa na estrutura feedback de erro. Apesar do salto relativo de u ser bem discrepante, a sua porcentagem $< 1,5\%$, é muito pequena em relação a capacidade de torque, visto que o ponto de operação dos motores situa-se em torno de 50%. Em fato, o sinal u é tão pequeno, que desconfia-se de que na realidade o ESC Hobbyking UBEC disponível nem tenha a resolução suficiente para a sutil ação de controle, após $t = 0$. Na prática, o sinal u seria interpretado apenas como dois valores, praticamente um empurrão inicial, que dificilmente seria preciso assim como na simulação.

Figura 38 – Atuação e resposta para o controlador PD ZN.

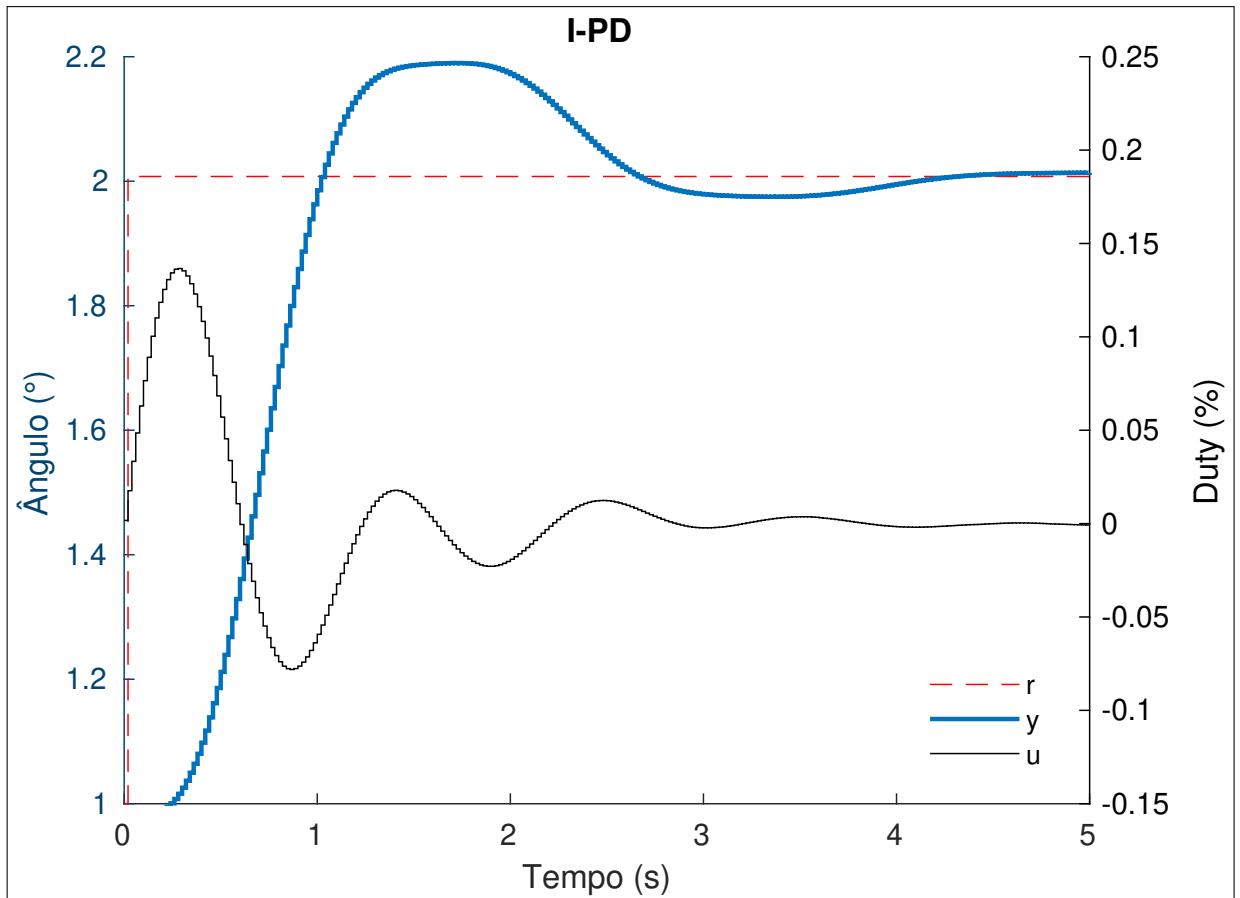


Fonte – O Autor.

Analizando agora o controlador I-PD, na Figura 39, observa-se que o pico inicial em $t = 0$ foi removido, como era de se esperar para a sua estrutura com dois graus de liberdade. Todavia, a ação de controle u continua sendo bem sutil, permanecendo entre os limites de $\pm 0,15\%$, para um degrau de referência de 1° . Uma possível solução poderia ser aumentar a amplitude do degrau, na tentativa de amplificar as variações do sinal u . Porém, isso também amplificaria a variação de atitude y , o que pode não ser desejado. No simulador, previu-se que uma inclinação entre 0° e 5° já é suficiente para manter o drone em movimento. A solução apropriada seria utilizar um ESC de maior resolução, capaz de capturar as mudanças apresentadas entre um período e outro de controle, uma transição de pequenos degraus em u .

Como se pode notar, mesmo com a relação não-linear presente no modelo simulado, os resultados têm se mostrado bastante fieis ao modelo da função de transferência. Com o controlador I-PD, a resposta y manteve o seu formato de sobressinal reto característico, que pode-se considerar até mesmo uma vantagem à locomoção do drone, por compensar o tempo de atraso inicial; isso desde que o sinal r consista em degraus de duração maior que 5 segundos.

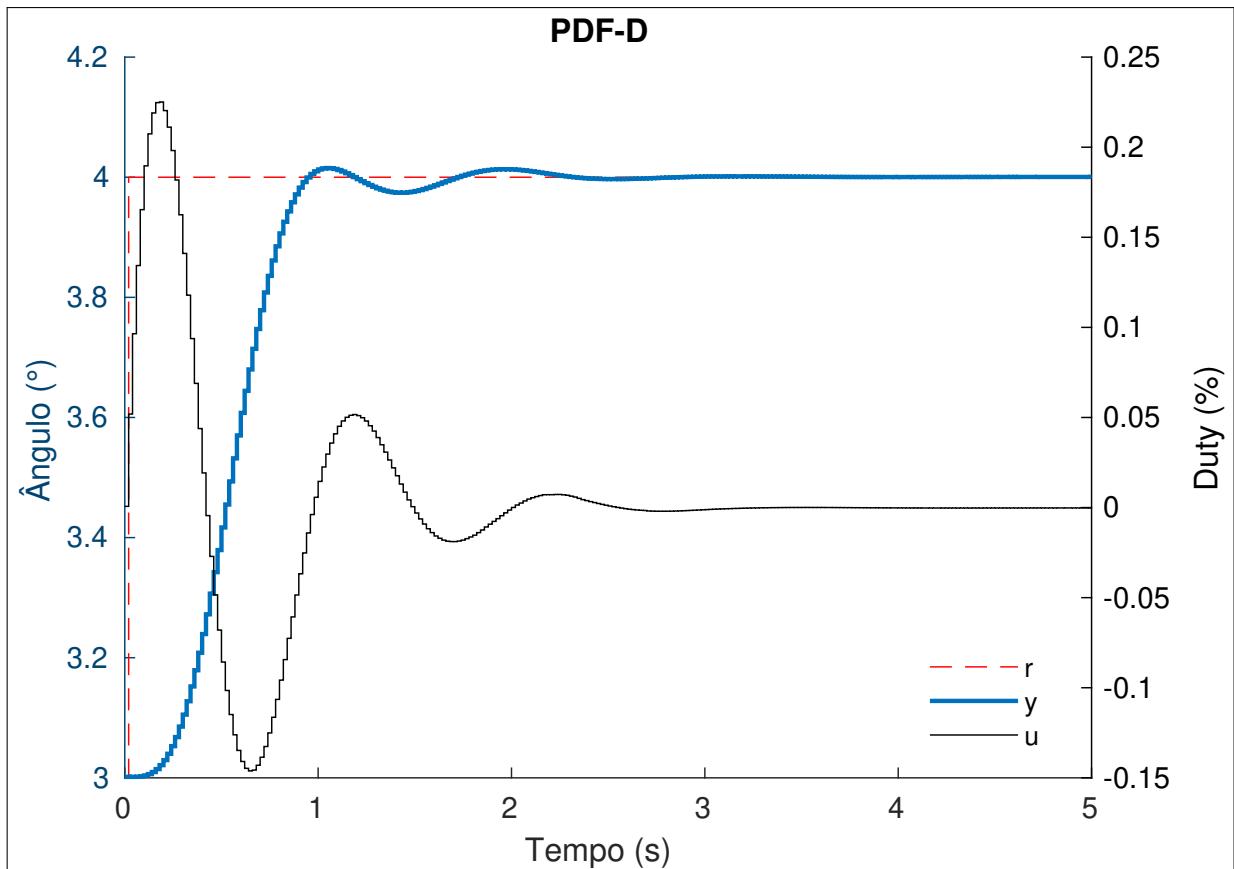
Figura 39 – Atuação e resposta para o controlador I-PD.



Fonte – O Autor.

Por último, apresenta-se na Figura 40 aquele que foi considerado o controlador mais eficaz, o controlador PDF-D. Realmente, comparando a sua atuação u em relação a do controlador I-PD, nota-se que o número de suas oscilações visíveis caiu de 7 para 5, e isso foi refletido/realimentado na precisão do sinal y . Analisando o inicio imediato à primeira oscilação, observa-se que a ação P seguida do filtro F predomina. Nessa região, a saída y quase não se locomoveu, e assim permanece por volta de 8 períodos de controle, de maneira bem semelhante à um atraso de transporte, até o momento em que a atuação u passa a diminuir. Nesse momento, a ação dos derivadores antecipa o sobressinal eminente, e começa a reduzir a atuação u , à medida que y acelera. Essa dinâmica se repete a medida que y começa a perder velocidade.

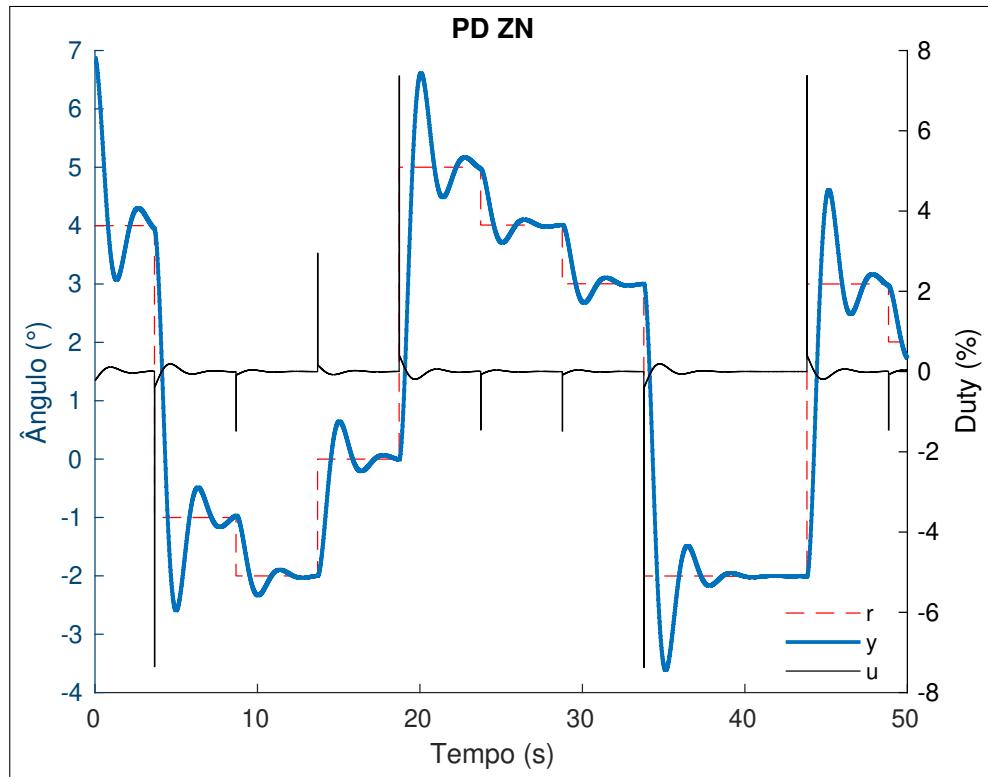
Figura 40 – Atuação e resposta para o controlador PDF-D.



Fonte – O Autor.

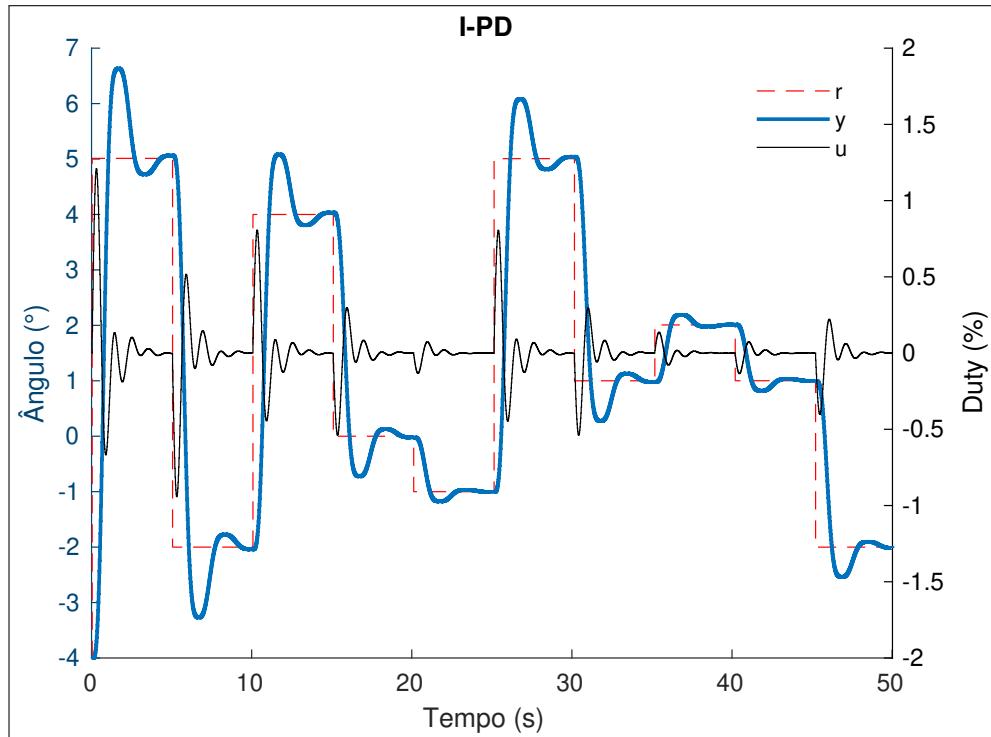
Um mérito na redução do sobressinal, alcançada pelo PDF-D, é visível observando-se o trem de degraus produzidos no simulador em diferentes tamanhos, e apresentados nas Figuras 41, 42, e 43. Como o sobressinal é uma porcentagem do valor final de referência, quando o tamanho do degrau aumenta, o valor absoluto de erro também aumenta, criando a sensação de que a ação de controle agiu de forma menos eficaz, quando na realidade apenas mudou de escala.

Figura 41 – Trem de degraus para o controlador PD ZN.



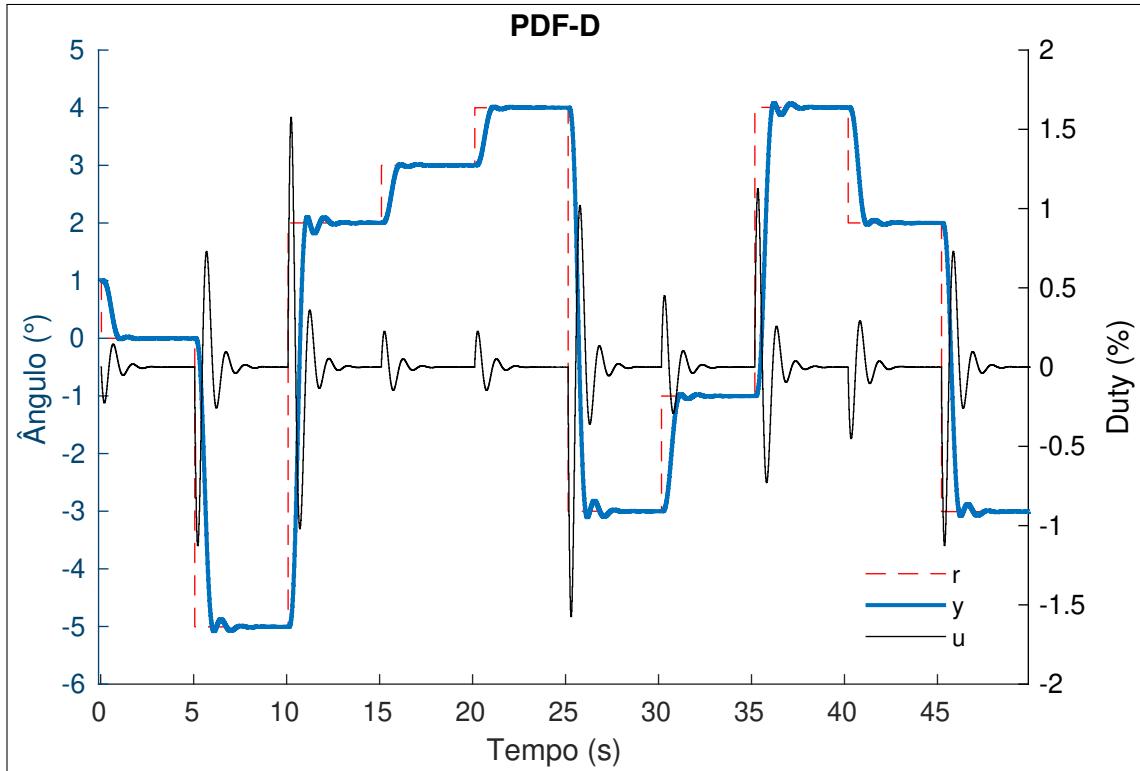
Fonte – O Autor.

Figura 42 – Trem de degraus para o controlador I-PD.



Fonte – O Autor.

Figura 43 – Trem de degraus para o controlador PDF-D.



Fonte – O Autor.

Como pode-se notar na Figura 43, a amplitude das oscilações nesse caso é bem pequena, e são pouco perceptíveis no drone simulado; o maior valor de atuação não ultrapassou 2%. Apesar do controle PDF-D ser um pouco mais agressivo do que os outros, visto o aumento do ganho proporcional, ainda não utiliza boa parte do potencial de atuação disponível. Em situações que deseja-se uma resposta suave e precisa, o controlador PDF-D já é uma boa alternativa.

7 CONCLUSÕES E TRABALHOS FUTUROS

O método de identificação desenvolvido foi utilizado para validar o simulador do sistema criado em Unity3D. A ferramenta Arcadia-Unity foi importante para habilitar o desenvolvimento do simulador, apesar de requisitar muita manutenção por parte do autor. A comunidade *open source* que desenvolveu Arcadia ajudou indiretamente na conclusão do simulador. Uma característica importante é que o mesmo pode ser reprogramado durante a sua execução, dentro do mesmo ambiente que o criou, reduzindo a necessidade da criação de interfaces.

O desenvolvimento *open source* foi prioridade nesse trabalho. A opção de escolha do Arduino pode ser subestimada por diversos motivos, porém é inegável o alcance que essa plataforma tem na internet. O alcance é importante para que o projeto ganhe tração, e mais pessoas possam o desenvolver. Assim, todo software foi publicado no GitHub em Martins (2018), na esperança de que outros se inspirem a continuar o esforço aqui iniciado.

Analisou-se as estruturas PD, I-PD, e PDF-D de controladores para a atitude, e concluiu-se que todas são capazes de alcançar o tempo de resposta de 1 segundo que era desejado, porém apenas com a estrutura PDF-D se conseguiu obter um sobressinal menor que 5%. A estrutura PD, e I-PD obtiveram sobressinal de 30% e 20%, respectivamente. Em todos os controladores, observou-se que a porcentagem de atuação utilizada foi muito baixa, e não ultrapassou 2% para degraus de referência de 1°.

Como trabalho futuro planeja-se realizar a comunicação do software embarcado a partir de um controle remoto, e habilitar o teste dos controladores desenvolvidos na prática, sem o suporte das cordas. Para tal, uma comunicação por radiofrequênciia ou equivalente terá que ser estudada. Além disso, planeja-se substituir cada ESC por outro de maior resolução, e re-confeccionar o circuito impresso, para que a sua alimentação seja melhor distribuída.

Por fim, deseja-se investigar novos tipos de controladores e novos métodos de identificação, dessa vez aplicados a movimentação do drone em um plano horizontal; e então projetar um controlador autônomo de posição com a adição de um sensor GPS.

REFERÊNCIAS

- AMAZON. Frequently Asked Questions.** 2018. Acessado em: 22-11-2018. Disponível em: <<https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>>.
- ASTROM, K. J. Control System Design Lecture Notes for ME 155A.** 2002. Disponível em: <<https://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom.html>>.
- ASTROM, K. J.; HAGGLUND, T. PID Controllers: Theory, Design, and Tuning.** 2. ed. [S.l.]: Instrument Society of America, 1995.
- ASTROM, K. J.; MURRAY, R. M. Feedback Systems.** [S.l.]: Princeton University Press, 2009.
- ATMEL. AVR493: Sensorless Commutation of BLDC Motor control using AT90PWM3/3B and ATAVRMC100.** 2006. Acessado em: 22-11-2018. Disponível em: <<http://ww1.microchip.com/downloads/en/AppNotes/doc7658.pdf>>.
- BRESCIANI, T. Modelling, Identification and Control of a Quadrotor Helicopter.** 2008. (MSc Theses). Student Paper.
- BUDDEN, M. et al. Mixer.** 2015. Disponível em: <<https://github.com/martinbudden/betaflight/blob/master/docs/Mixer.md>>.
- DEAKIN, M.** The development of the laplace transform, 1737–1937. **Archive for History of Exact Sciences**, v. 25, p. 343–390, 12 1981.
- DORF, R. C.; BISHOP, R. H. Modern Control Systems.** 9th. ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2000. ISBN 0130306606.
- DÙ, G. L. Arduino sketch to calibrate ESC (Electrical Speed Controllers).** 2018. Disponível em: <<https://github.com/lobodol/ESC-calibration>>.
- ESQUIVEL, L. B. Forest Monitoring with Drones: Application Strategies for Protected Riverine Forest Ecosystems in the Atlantic Forest of Rio de Janeiro, Brazil.** Dissertação (Mestrado) — Technical University of Cologne, 2015.
- GAMAZO-REAL, J. C. et al.** Position and speed control of brushless dc motors using sensorless techniques and application trends. **Sensors**, v. 10, n. 7, p. 6901–6947, 2010. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/10/7/6901>>.
- GARDNER, T.; NASSER, R. Arcadia - The integration of the Clojure Programming Language with the Unity 3D game engine.** 2017. Disponível em: <<https://github.com/arcadia-unity/Arcadia>>.
- HAUGEN, F. PID Control.** Tapir Academic Press, 2004. ISBN 9788251919456. Disponível em: <<https://books.google.com.br/books?id=8CIIAAAACAAJ>>.
- HOBBYKING. Hobbyking 30A UBEC, User Manual.** 2018. Acessado em: 27-11-2018. Disponível em: <<http://rosa-hp.sakura.ne.jp/811103388X7478X20.pdf>>.
- INVENSENSE. MPU-6000/MPU-6050 Product Specification.** 2013. Acessado em: 22-11-2018. Disponível em: <<https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>>.

- INVENSENSE. **MPU Hardware Offset Registers App Note**. 2014. Acessado em: 22-11-2018. Disponível em: <<https://www.digikey.com/en/pdf/i/invensense/mpu-hardware-offset-registers>>.
- J.B., B. *et al.* **UIUC Propeller Database**. 2018. Disponível em: <<http://m-selig.ae.illinois.edu/props/propDB.html>>.
- JORGE, L. C.; INAMASU, R. **Uso de veículos aéreos não tripulados (VANT) em Agricultura de Precisão**. [S.l.]: Embrapa, 2014.
- KRYKOWSKI, K.; HETMANCZYK, J. Constant current models of brushless dc motor. **Electrical, Control and Communication Engineering**, v. 3, n. 1, p. 19–24, 2013. Disponível em: <<https://content.sciendo.com/view/journals/ecce/3/1/article-p19.xml>>.
- LATHI, B. **Sinais e Sistemas Lineares - 2.ed.** BOOKMAN COMPANHIA ED, 2007. ISBN 9788560031139. Disponível em: <<https://books.google.com.br/books?id=ySxoo2TVeeYC>>.
- MARTINS, P. **Repositório Github com o código fonte da monografia**. 2018. <<https://github.com/pjago/tcc>>. Acessado em: 22-11-2018.
- MATLAB. **Moore-Penrose pseudoinverse - MATLAB pinv - MathWorks**. MathWorks, 2018. Disponível em: <<https://www.mathworks.com/help/matlab/ref/pinv.html>>.
- NILSSON, J. W.; RIEDEL, S. **Electric Circuits (8th Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007. ISBN 0131989251.
- NYQUIST, H. Regeneration theory. **The Bell System Technical Journal**, v. 11, n. 1, p. 126–147, Jan 1932. ISSN 0005-8580.
- OGATA, K. **Modern control engineering**. 5. ed. [S.l.]: Pearson Education do Brasil Ltda, 2010.
- RODENAS, L. **Arduino script for MPU-6050 auto-calibration**. 2015. Disponível em: <<https://42bots.com/tutorials/arduino-script-for-mpu-6050-auto-calibration/>>.
- ROMM, T. **A new Trump policy could let Amazon and Google test more drones in U.S. cities**. 2017. Disponível em: <<https://www.recode.net/2017/10/25/16542940/amazon-google-drones-us-government-trump>>.
- ROWBERG, J. **I2C device library collection for AVR/Arduino or other C+-based MCUs**. 2018. <<https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>>. Acessado em: 22-11-2018.
- SADIKU, M. **Elementos de Eletromagnetismo**. 5. ed. [S.l.]: Saraiva, 2012.
- SMITH, I. J. O. **Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects**. [s.n.], 2006. Disponível em: <<https://www.dsprelated.com/freebooks/pasp/Comb\ Filters.html>>.
- STAPLES, G. **Propeller Static & Dynamic Thrust Calculation**. 2014. Disponível em: <<https://www.electricrcaircraftguy.com/2014/04/propeller-static-dynamic-thrust-equation-background.html>>.
- WELCH, A. **A cost-benefit analysis of Amazon Prime Air**. 2015. Disponível em: <<https://scholar.utc.edu/honors-theses/47>>.

WIKI, E. **Inertial Measurement Unit IMU Use — ESE205 Wiki**, 2018. Acessado em: 22-11-2018. Disponível em: <[https://classes.engineering.wustl.edu/ese205/core/index.php?title=Inertial_Measurement_Unit_\(IMU\)_Use&oldid=12354](https://classes.engineering.wustl.edu/ese205/core/index.php?title=Inertial_Measurement_Unit_(IMU)_Use&oldid=12354)>.

ZIEGLER, J.; NICHOLS, N. Optimum setting for automatic controllers. **Transactions of ASME**, p. 759–768, 06 1993.