

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA ELÉTRICA**

BRUNO DE AGUIAR SILVA MIRANDA

**MONTAGEM DE UMA PLATAFORMA PARA TESTES DE CONTROLE DE UM
VEÍCULO AÉREO NÃO TRIPULADO QUADRIMOTOR**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

Julho 2019

BRUNO DE AGUIAR SILVA MIRANDA

**MONTAGEM DE UMA PLATAFORMA PARA TESTES DE CONTROLE DE UM
VEÍCULO AÉREO NÃO TRIPULADO QUADRIMOTOR**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso 2, do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Cristiano Marcos Agulhari.

CORNÉLIO PROCÓPIO

Julho 2019



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento Acadêmico de Elétrica
Curso de Engenharia Elétrica



FOLHA DE APROVAÇÃO

Bruno de Aguiar Silva Miranda

MONTAGEM DE UMA PLATAFORMA PARA TESTES DE CONTROLE DE UM VEÍCULO AÉREO NÃO TRIPULADO QUADRIMOTOR

Trabalho de conclusão de curso apresentado às 14:00hs do dia 01/07/2019 como requisito parcial para a obtenção do título de Engenheiro Eletricista no programa de Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). Cristiano Marcos Agulhari - Presidente (Orientador)

Prof(a). Dr(a). Paulo Rogério Scalassara - (Membro)

Prof(a). Dr(a). Kleber Romero Felizardo - (Membro)

Prof(a). Dr(a). Marcos Banheti Rabello Vallim - (Membro)

DEDICATÓRIA

Dedico este trabalho à minha família,
Especialmente à minha avó Elenilda.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Cristiano Marcos Agulhari por todo auxílio, paciência e disponibilidade nos momentos que precisei, mesmo quando não fiz por merecer.

Agradeço ao Prof. Dr. Alessandro Do Nascimento Vargas por todo suporte no projeto, pois além do empréstimo de peças e componentes disponibilizou o laboratório para a realização dos testes.

À Deus por todas as bênçãos e por sempre ter cuidado de mim.

Aos os meus colegas e professores da UTFPR que me ajudaram nesta trajetória.

Aos meus pais por todo apoio e por sempre me ajudarem nesta longa caminhada. Vocês são meus melhores amigos, razão que me faz seguir em busca de ser uma pessoa melhor a cada dia e que me dão forças para continuar, mesmo nos momentos mais difíceis.

Ao meu irmão Rodrigo por toda ajuda durante os momentos que mais precisei, desde emocional e até mesmo financeira para que eu pudesse morar em outro estado. Sempre foi mais que um irmão, o considero um segundo pai pra mim.

A minha amiga, namorada e futura esposa Thayane Dias dos Santos por toda ajuda e incentivo para finalização deste projeto.

“Não temas, porque eu sou contigo; não te assombres, porque eu sou teu Deus; eu te fortaleço, e te ajudo, e te sustento com a destra da minha justiça.”

(Isaías 41:10)

RESUMO

MIRANDA, Bruno de A. S. **Montagem de uma plataforma para testes de controle de um veículo aéreo não tripulado quadrimotor.** 2019. 71 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Veículos Aéreos não Tripulados (VANTs) tem sido amplamente desenvolvidos e utilizados tanto no meio militar como no meio civil, sendo observado um crescimento exponencial de sua aplicação nesta última área nos últimos anos. Diante das vantagens e contribuições do VANT muitas pesquisas têm sido realizadas, entretanto grande parte destes estudos trazem apenas soluções considerando condições ideais para simular o funcionamento do mesmo. Assim, o presente estudo tem como objetivo realizar a montagem de uma plataforma para testes de controle de VANTs, o que possibilita a validação experimental de métodos de controle levando em consideração situações não previstas em simulações, além de incentivar novas pesquisas e trabalhos após ser doado para o Laboratório de Automação e Controle de Processos (Lab Control) da Universidade Tecnológica Federal do Paraná (UTFPR), campus Cornélio Procópio. Utilizou-se um controle PID para testes na plataforma, a fim de verificar a viabilidade da mesma na utilização de métodos de controle, com estrutura que suporte as características de funcionamento dos VANTs. Os resultados obtidos nos testes da plataforma foram satisfatórios e possibilitam, em trabalhos futuros, a implementação de métodos de controle mais completos, visto que poucos ajustes podem se fazer necessários, como a troca da placa microcontroladora por uma com maior capacidade de processamento, em caso de perda de desempenho.

Palavras-chave: Testes de Controle em VANTs. Veículos Aéreos Não Tripulados. Controle de Estabilidade. Quadrimotor.

ABSTRACT

MIRANDA, Bruno de A. S. **Assembly of a test platform for the control of an unmanned aerial vehicle.** 2019. 71 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Unmanned Aerial Vehicles (UAVs) have been extensively developed and used in both military and civilian environments, with an exponential growth in their application in the civil area during the last years. As a result of the advantages and contributions of the UAV, many researches have been carried out, however many of these studies bring only solutions considering ideal conditions to simulate the operation of the UAV. Thus, the present study has the objective of assembling a platform for control tests of UAVs, which allows the experimental validation of control methods taking into account situations not considered in simulations, besides encouraging new researches and works after being donated for the Laboratory of Automation and Control of Processes (Lab Control) of the Universidade Tecnológica Federal do Paraná (UTFPR), Cornélio Procópio campus. A PID control was used to test the platform, in order to verify its feasibility in the use of control methods, with structure that supports the operating characteristics of the UAVs. The results obtained in the tests of the platform were satisfactory and allow future implementations of more complete control methods, since few adjustments may be necessary, such as the exchange of the microcontroller board for one with greater processing capacity, in case of performance loss.

Keywords: Control Tests in UAVs. Unmanned Aerial Vehicles. Stability Control. Four-engine.

LISTA DE SIGLAS

RNA	Rede Neural Artificial
PID	Proporcional-Integrativo-Derivativo
PWM	Pulse Width Modulation
ABS	Acrylonitrile Butadiene Styrene
UTFPR	Universidade Tecnológica Federal do Paraná
GWS	Grand Wing Servo
IMU	Inertial Measurement Unit
I2C	Inter-Integrated Circuit
SDA	Serial Data
SCL	Serial Clock
PCI	Placa de Circuito Impresso

LISTA DE ACRÔNIMOS

VANT	Veículo Aéreo Não Tripulado
ESC	Electronic Speed Control
Lab Control	Laboratório de Automação e Controle de Processos

LISTA DE ILUSTRAÇÕES

Figura 1 – VANTs de asas fixas	16
Figura 2 – VANTs de asas móveis	16
Figura 3 – Manobrabilidade dos VANTs de asas rotativas e de asas fixas	17
Figura 4 – Diagrama de blocos e comunicação do VANT	21
Figura 5 – Pinos do módulo GY-80	22
Figura 6 – Esquema elétrico do módulo GY-80.....	23
Figura 7 – PWM gerado para diferentes valores de entrada	24
Figura 8 – Ligação entre motor e ESC	25
Figura 9 – Correntes dos motores geradas pelos ESCs	25
Figura 10 – Controle em malha fechada do sistema	27
Figura 11 – Eixo X e Y dos sensores do VANT	27
Figura 12 – Barras Retangulares de Alumínio.....	31
Figura 13 – Estrutura DJI Flame Wheel 450 F450	31
Figura 14 – Placa Arduino UNO	32
Figura 15 – Bateria Turnigy Li-Po 3S 2200mAh	33
Figura 16 – Disposição interna das células de baterias Li-Po 3S.....	34
Figura 17 – Bateria Zippy Flightmax Li-Po 3S 4000mAh	34
Figura 18 – Carregador de baterias Li-Po 2S/3S	35
Figura 19 – Carregador de baterias Li-Po 1-6S <i>AccuCel 6</i> Turnigy	36
Figura 20 – Motor brushless CF2822 da EMAX	37
Figura 21 – Hélice 9x5 e adaptador spinner.....	38
Figura 22 – ESC Simon Series 20A da EMAX	39
Figura 23 – Módulo multissensor GY-80	40
Figura 24 – Montagem da estrutura Física F450.....	42
Figura 25 – Possibilidade de Energização dos ESCs via soldagem na PCB	43
Figura 26 – Plataforma para testes de controle de VANTs	43
Figura 27 – Aceleração obtida em testes variando a posição do VANT.....	45
Figura 28 – Velocidade obtida em testes variando a posição do VANT	45
Figura 29 – Testes realizados com o módulo GY-80.....	46
Figura 30 – Teste do controle de estabilidade com 2 motores	48
Figura 31 – Teste utilizando Controle PID com 2 motores	49
Figura 32 – Sinal PWM dos motores em função do Controle PID	50

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	17
1.2	Problemas e premissas	18
1.3	Objetivos	19
1.2.1	Objetivo geral	19
1.2.2	Objetivos específicos	19
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Comunicação	21
2.3.1	Transmissão de dados do módulo sensor.....	21
2.3.2	Transmissão do sinal PWM para os ESCs.....	23
2.3.3	Transmissão de energia para os motores	24
2.2	Controle de estabilidade	25
2.2.1	Controle PID e princípios de funcionamento do VANT	26
2.2.2	Calibração do controle PID.....	28
3	MATERIAIS E MÉTODOS	30
3.1	Hardware	30
3.3.1	Estrutura física	30
3.3.2	Placa microcontroladora.....	32
3.3.3	Alimentação do sistema	32
3.3.4	Carregador de bateria	35
3.3.5	Motor de corrente contínua sem escovas	36
3.3.6	Hélices propulsoras.....	37
3.3.7	Controlador eletrônico de velocidade	38
3.3.8	Módulo sensor.....	39
3.2	Resultado esperado e avaliação	40
4	RESULTADOS	42
4.1	Montagem da plataforma	42
4.2	Aferição dos sensores	43
4.3.1	Acelerômetro de 3 eixos.....	44
4.3.2	Giroscópio de 3 eixos.....	45
4.3	Controle de velocidade dos motores	46
4.4	Implementação do código de controle da plataforma	47

4.5	Testes do controle PID na plataforma	48
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	51
	REFERÊNCIAS.....	53
	APÊNDICE A – Código do Controlador PID de Estabilidade do VANT	57

1 INTRODUÇÃO

Os Veículos Aéreos Não Tripulados (VANTs) são definidos como máquinas voadoras que dispensam a presença de tripulação a bordo, sem caráter recreativo e que possui carga útil embarcada. Podem ser controladas à distância por meios eletrônicos e computacionais, sob a supervisão e governo humano, ou por meio de Controladores Lógicos Programáveis, sem intervenção humana simultânea ao seu funcionamento (BRASIL, 2018).

Essas aeronaves têm seu uso registrado primeiramente em um ataque do Exército Austríaco à cidade italiana de Veneza em 12 de julho de 1849, onde balões foram carregados com explosivos e lançados do navio austríaco Vulcano, visando se precipitarem sobre a cidade e explodir sua carga. Entretanto, alguns retornaram às linhas austríacas devido a uma repentina mudança de vento (HARDGRAVE, 2005). Entre os anos de 60 e 70, os EUA durante as Guerras da Coreia e do Vietnã utilizaram o modelo Firebee, uma aeronave com propulsão a jato e de dimensão de um pequeno jato executivo (WIDMAIER, 2005).

No Brasil, os primeiros experimentos com essas aeronaves são da década de 80, com o desenvolvimento do Projeto Acauã pelo Centro Tecnológico Aeroespacial (CTA), em que foram projetados cinco protótipos, que realizaram o primeiro voo em 1985 (OLIVEIRA, 2005).

Nesse sentido, verifica-se que inicialmente os VANTs eram projetados com fins bélicos, sendo utilizados apenas para espionagem militar, reconhecimento aéreo e bombardeios em alvos estratégicos. Este tipo de utilização se deve ao fato de não colocar a vida do piloto em risco, facilitando voos em ambientes hostis (SÁ, 2012 p. 1). Devido às tecnologias da época, o uso desses veículos era pouco confiável e impreciso.

O avanço da tecnologia tem proporcionado diminuição nos custos de produção e manufatura, sendo a automatização de processos de produção um dos fatores que mais tem influenciado nesta redução. Tecnologias que outrora eram de uso restrito de instituições militares e centros de pesquisa, principalmente devido aos altos custos dos componentes e dimensões elevadas, estão sendo cada vez mais utilizadas pela sociedade civil (MARINESCU, 2013). Apesar da predominância de sua aplicação no meio militar, o autor Jaxa-Malachowski (1996) revela a tendência de crescimento exponencial em ambas as áreas.

Para THOMPSON et al (1994) o fator que mais colaborou recentemente para o desenvolvimento dos VANTs foi o aperfeiçoamento dos sistemas de navegação, como o Sistema de Posicionamento Global (GPS). Ademais, a criação da tecnologia MEMS (sistemas microeletromecânicos em inglês) proporcionou a diminuição das dimensões dos componentes eletromecânicos e o aumento de sua produção em larga escala, tornando-os mais acessíveis, possibilitando o desenvolvimento de diversos bens de consumo com o uso dessa tecnologia e potencializando a demanda por essas aeronaves. (MELO, 2010 p. 14).

Cada vez mais presente em meio civil, suas aplicações são inúmeras, tais como localização de pessoas em locais de difícil acesso, fotos e filmagens panorâmicas, entrega de botes salva vidas e até mesmo inspeção de linhas de transmissão de energia elétrica.

De acordo com o tipo de voo, os VANTs podem ser divididos entre os de asas rotativas e os de asas fixas, como pode ser visto nas Figuras 1 e 2.

Figura 1 – VANTs de asas fixas



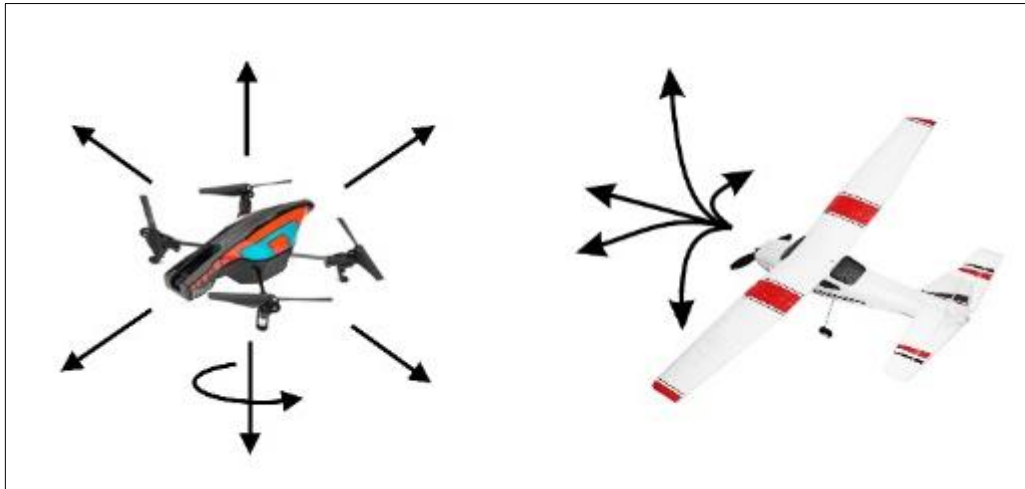
Figura 2 – VANTs de asas móveis



Fonte: Adaptado de Santana (2016, p. 18).

Uma relevante diferença entre essas categorias é a sua manobrabilidade, ilustrada na Figura 3. Os veículos de pás rotativas apresentam como vantagem a decolagem e o pouso vertical, mantêm-se parados em uma posição e movimentam-se em diversas direções, em contrapartida, os veículos de asas fixas realizam movimentos frontais contínuos para permanecerem voando. (SANTANA, 2016 p. 18).

Figura 3 – Manobrabilidade dos VANTs de asas rotativas e de asas fixas



Fonte: Adaptado de Santana (2016, p. 19).

Os VANTs de asas rotativas podem ter diversas configurações quanto ao número de motores, porém, o modelo quadrimotor, que é o objeto de estudo do presente trabalho, prevalece frente aos outros (MARINESCU, 2013 p. 11), sendo o mais utilizado, devido à sua arquitetura que proporciona o equilíbrio estático no ar, se dotado de sistema de controle, que pode ser automático ou manual. (PFEIFER, 2013 p.3).

Atualmente o controle de VANTs pode ser feito através de lógica Fuzzy, RNA (Redes Neurais Artificiais) e PID (Proporcional, Integral e Derivativo), utilizado neste trabalho para testar a plataforma elaborada, por sua simplicidade de aplicação em comparação aos demais, assim como utilizado por Benigno (2015).

1.1 Justificativa

O quadrimotor é um VANT que possui muitas vantagens na sua construção e manutenção, o que justifica seu constante uso na literatura e progresso no desenvolvimento de pesquisas. Com o aumento de pesquisas e publicações, as possibilidades de aplicação também crescem notavelmente (BENIGNO, 2015 p.18).

Em diversos projetos de VANTs publicados, tais como o desenvolvido por Melo (2010), Sá (2012) e Marinescu (2013), percebe-se que um dos problemas na manipulação dos VANTs é a complexidade do controle para atingir a estabilidade e navegação com resultados aceitáveis. Tais trabalhos tratam apenas de garantir a estabilidade do VANT a partir da implementação de leis de controle propostas.

No contexto apresentado, este trabalho encontra seu propósito ao propor a montagem de uma plataforma para testes de controle de um quadrimotor, realizando a aplicação de um controle PID em dois motores a fim de comprovar a eficácia da plataforma.

1.2 Problemas e premissas

A sociedade aumentou consideravelmente o uso de mecanismos eletroeletrônicos no decorrer dos últimos anos, não somente de VANTs, mas de sistemas embarcados em geral. Estes sistemas se tornam cada vez mais complexos, de acordo com sua capacidade de processamento e sensoriamento. (BOTELHO, 2018 p. 23).

Com tantas aplicações noticiadas em mídias de todo o mundo, nem sempre as informações sobre VANTs foram motivo de entusiasmo como atualmente. Em sua grande maioria, os comunicados de sua utilização eram sobre bombardeios e espionagens realizadas, pois os VANTs “possibilitam a aplicação letal da força de forma discreta e precisa” como destacado por Vicente (2013).

Dispondo de tamanha precisão para executar tamanhas atrocidades, em contrapartida os VANTs estão sendo utilizados cada vez mais para salvar vidas, seja em busca por pessoas perdidas em locais de difícil acesso, lançamento de botes salva vidas em casos de situações de afogamento e até mesmo em transporte de órgãos entre hospitais.

O desenvolvimento de controladores e estratégias de controles visando estabilização de um VANT é um grande desafio, principalmente por seu grande número de variáveis. A quantidade de variáveis pode aumentar mais ainda se for necessário sistema de direção e navegação, além do controle de estabilidade, pois em resumo são definidos como:

- Navegação: abrange o conjunto de ferramentas que captam os movimentos do VANT e determinam seus estados (posição, velocidade e aceleração por exemplo);
- Controle: responsável por utilizar os dados dos sensores e, sem intervenção manual durante a operação, sintetizar as leis de controle dos movimentos a fim de concluir o objetivo programado, como por exemplo, manter a estabilidade do VANT;

- Direção: encarregado de determinar a missão desejada, no qual a uma trajetória ou posição é previamente informada, ou definidas ao mesmo tempo que recebe informações do sistema de navegação. (SANTANA apud KENDOUL, 2012).

Este trabalho tem por finalidade propiciar diversas validações experimentais de VANTs, através da montagem de uma plataforma de testes que possibilitará a aplicação das metodologias de controle por meio de códigos de programação que, após a avaliação do mesmo, será doado para o Laboratório de Automação e Controle de Processos (Lab Control) da Universidade Tecnológica Federal do Paraná (UTFPR), campus Cornélio Procópio, com a finalidade de auxiliar e incentivar futuras pesquisas quanto ao assunto abordado neste trabalho.

1.3 Objetivos

1.2.1 Objetivo geral

Realizar a montagem de uma plataforma para testes de controle de um VANT quadrimotor, com pás rotativas e sistema de controle via microcontrolador, com sensores tipo acelerômetro e giroscópio para navegação e controle de estabilidade, quatro controladores eletrônicos de velocidade (*ESCs*, do inglês *Electronic Speed Control*), quatro motores de corrente contínua sem escovas (do inglês *brushless*) e alimentação através de bateria recarregável. Por conseguinte, realizar testes de um controle de estabilidade do VANT mediante programação do microcontrolador.

1.2.2 Objetivos específicos

Os objetivos específicos deste projeto seguem abaixo:

- estudar conceitos sobre VANTs e materiais que o compõem, a fim de especificar os componentes necessários para a montagem da plataforma;
- definir qual tipo de microcontrolador será utilizado;
- realizar a calibração dos sensores que serão utilizados;

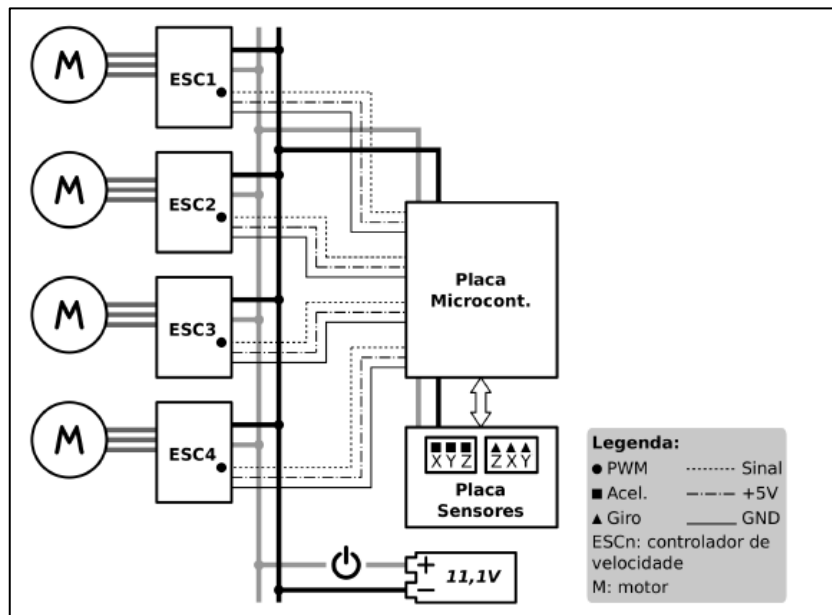
- viabilizar a leitura dos dados do acelerômetro e do giroscópio utilizando o microcontrolador;
- desenvolver um código específico para modulação de largura de pulso (PWM), a fim de efetuar o controle de velocidade dos motores;
- desenvolver e implementar um código para testes da plataforma, utilizando controle PID em dois motores opostos;
- obter os resultados dos testes do controle PID utilizados na plataforma, tais como amostras dos sensores, variáveis calculadas no controle PID e velocidades dos motores.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Comunicação

O sistema deve se comunicar com precisão para que os dados dos sensores possam ser transmitidos para o Arduino, os valores de velocidade sejam enviados em forma de sinal PWM para os ESCs e estes, conectados à bateria de Li-Po, liberem mais (ou menos) energia para os motores. No diagrama de blocos, apresentado na Figura 4, é possível visualizar a estrutura do sistema de comunicação a ser utilizado.

Figura 4 – Diagrama de blocos e comunicação do VANT



Fonte: Adaptado de Melo (2010 p. 21)

2.3.1 Transmissão de dados do módulo sensor

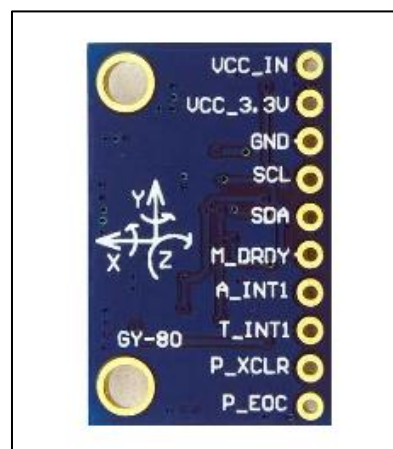
A interface I2C (*Inter-Integrated Circuit*) é um protocolo de comunicação *serial* que utiliza apenas dois fios, no qual a entrada SDA (do inglês *Serial Data*) pode ser utilizada para envio e recebimento de dados utilizando uma taxa binária para transmissão, ao receber um sinal de *clock* em outra linha comumente chamada de SCL (do inglês *Serial Clock*). (MELO, 2010 p. 32).

Alguns pinos do Arduino têm funções específicas, como as entradas analógicas A4 e A5 do UNO que são respectivamente utilizados como SDA e SCL

para comunicação a dois fios (ARDUINO SERIAL, 2016). Os pinos do módulo GY-80 podem ser observados na Figura 5, em que a função dos pinos mais relevantes são:

- VCC_IN caso queira alimentar com 5V;
- VCC_3.3V caso queira alimentar com 3,3V;
- GND deve ser conectado ao GND do UNO;
- SCL deve ser conectado ao pino A5 do UNO;
- SDA deve ser conectado ao pino A4 do UNO;

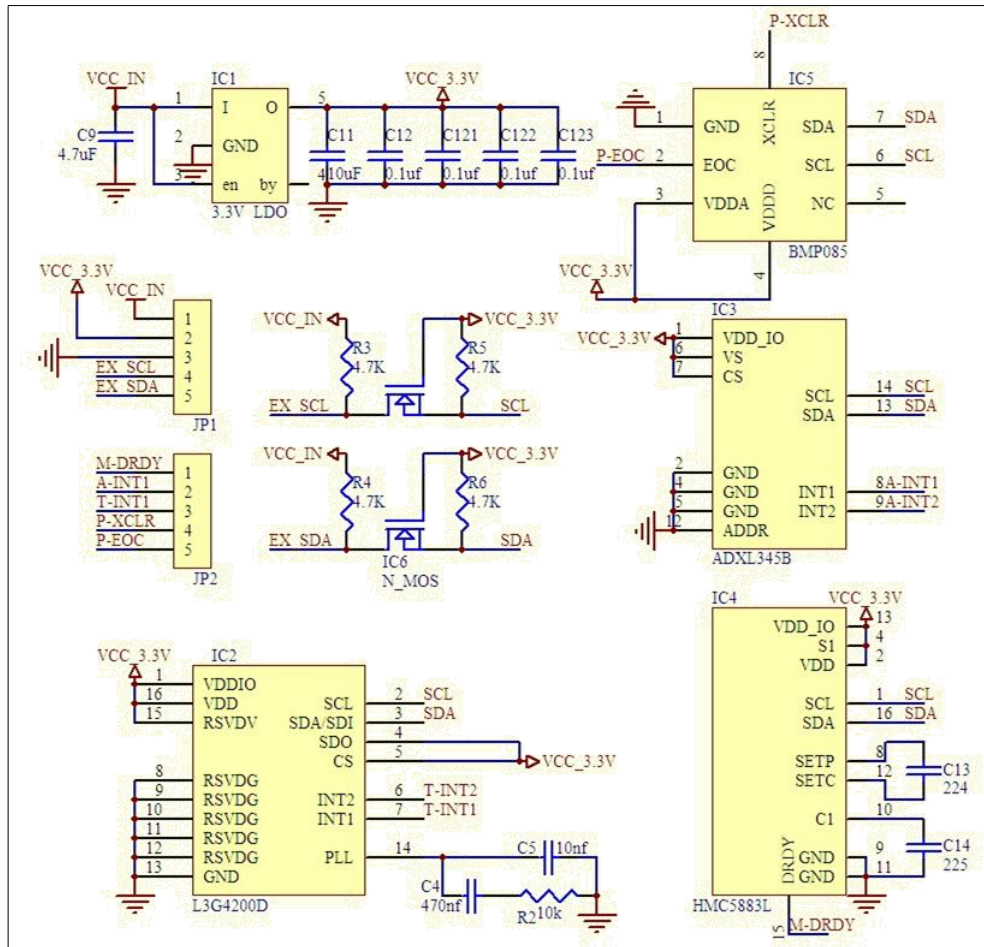
Figura 5 – Pinos do módulo GY-80



Fonte: Arduino e Cia (2014).

Como o módulo GY-80 é multissensor, cada um dos sensores contidos no mesmo tem seu próprio endereço para comunicação I2C dado em números hexadecimais. Utilizando a biblioteca *Wire* no código de programação do Arduino é possível realizar a seleção e inicialização de funcionamento de um sensor específico, sendo necessário informar o endereço hexadecimal para iniciar a transmissão de dados do SDA do respectivo sensor para o SDA do Arduino. O esquema elétrico, que contém as configurações internas do GY-80 e dos sensores contidos, pode ser visto na Figura 6.

Figura 6 – Esquema elétrico do módulo GY-80



Fonte: Arduino e Cia (2014).

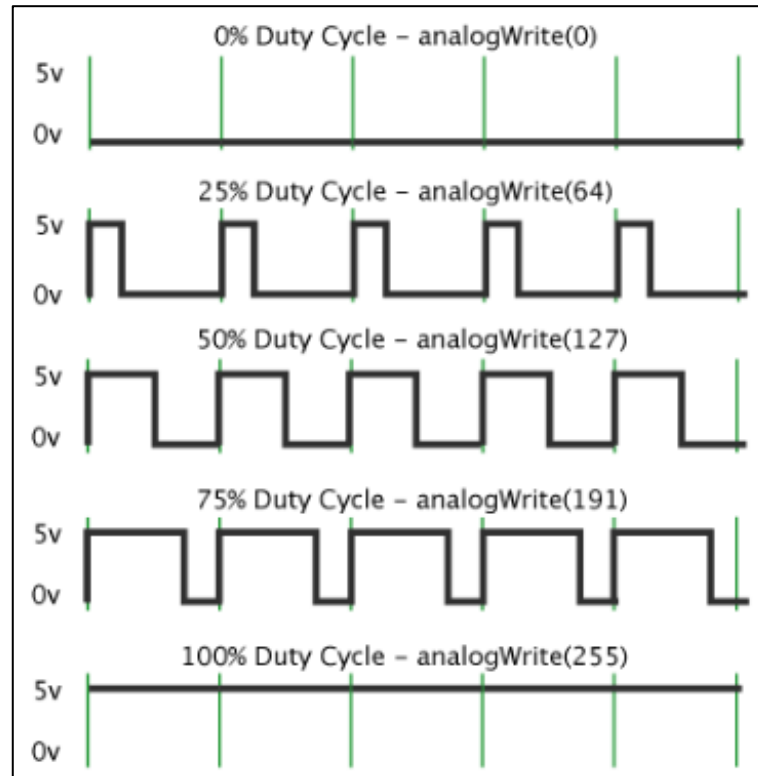
Iniciada a transmissão da comunicação I2C, faz-se necessário realizar a requisição dos dados do sensor específico e posteriormente a leitura dos mesmos. Além disso, utilizando a biblioteca *Serial* do Arduino os dados podem ser disponibilizados na porta *serial* do computador via conexão USB, tornando possível visualiza-los instantaneamente pela ferramenta *Monitor Serial do Arduino*.

2.3.2 Transmissão do sinal PWM para os ESCs

Das 6 saídas digitais contidas no Arduino UNO, são necessárias apenas 4 para ligar os ESCs. Elas são responsáveis pelo comando realizado pelo UNO nos ESCs, enviando sinais PWM com diversos valores de razão cíclica. A razão cíclica é a porcentagem que um sinal digital permanece em nível alto (5V no caso do Arduino) e possibilita variar a tensão média na saída de 0 a 5V, diretamente

proporcional ao seu valor. O Arduino contém originalmente somente uma função para gerar sinal **PWM** em suas saídas digitais, chamada *analogWrite*(entrada, saída). Esta função pode receber um valor de entrada de 0 a 255 (8 bits) e varia a razão cíclica da saída informada de 0 a 100% respectivamente (ARDUINO PWM, 2016). A Figura 7 ilustra o sinal PWM gerado para diferentes valores de entrada.

Figura 7 – PWM gerado para diferentes valores de entrada



Fonte: ARDUINO PWM (2016).

2.3.3 Transmissão de energia para os motores

Os *ESCs* realizam eletronicamente a comutação da energização das bobinas (fases), ou seja, os *ESCs* geram 3 sinais (normalmente trapezoidais) defasados em 120° entre si para as 3 fases dos motores (SÁ, 2012, p. 26). A conexão entre o motor CF2822 e o *ESC* pode ser observada na Figura 8.

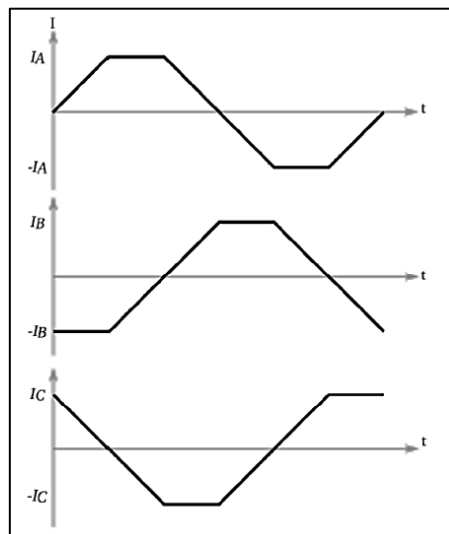
Figura 8 – Ligação entre motor e ESC



Fonte: Autoria própria.

A quantidade de energia que os *ESCs* fornecem para os motores é diretamente proporcional à razão cíclica do sinal PWM que recebem da placa microcontroladora. Na Figura 9 é apresentado um exemplo de corrente da fase A, B e C de um motor, fornecida pela saída do *ESC* ao receber um sinal PWM em sua entrada.

Figura 9 – Correntes dos motores geradas pelos *ESCs*



Fonte: Nascimento (2011, p. 28).

Assim como em motores de corrente alternada, para inverter o sentido da rotação dos motores brushless basta inverter uma das conexões com o *ESC*, mudando as fases de ABC para ACB por exemplo. A plataforma necessita ter dois motores que realizam giro no sentido horário (ABC) e dois no sentido anti-horário (ACB).

2.2 Controle de estabilidade

2.2.1 Controle PID e princípios de funcionamento do VANT

Um dos fatores mais importantes para a estabilidade dos VANTs é o controle de velocidade dos motores. Sem uma resposta dos motores precisa, que possa gerar pequenas variações de velocidade quando necessário, a estabilidade se torna impossível. O controle PID é uma técnica bastante utilizada em sistemas de controle realimentados, devido a sua estrutura simplificada, por estar relacionado a sistemas instáveis e sensíveis a distúrbios, susceptíveis a variações em parâmetros do sistema e a possibilidade de implementação sem modelo específico do sistema de controle. (BOTELHO, 2018, p. 54; OGATA, 2011, p. 521).

Este tipo de controle combina ações de controle proporcional, integral e derivativo, com as vantagens individuais de cada um deles (OGATA, 2011 p. 21), e foi utilizado para desenvolvimento do controle de estabilidade do VANT. O sinal de controle $u(t)$, dado pela Equação (1), representa ação combinada do controle PID:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

No qual tem-se:

- $e(t)$ é o sinal de erro;
- K_p , K_i e K_d são respectivamente o ganho proporcional, integrativo e derivativo;
- $de(t)/dt$ é a variação do sinal de erro;
- $\int_0^t e(t) dt$ é a integral do erro (soma dos erros).

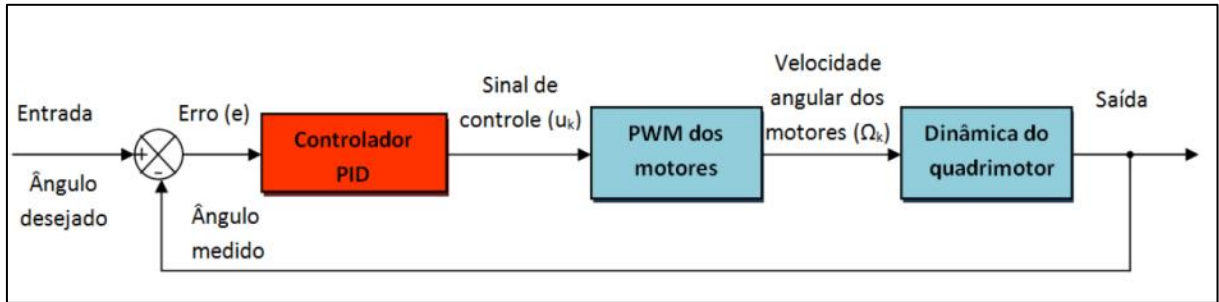
O sinal de erro é calculado pela diferença entre o ângulo medido pelo sensor e o ângulo desejado, que é zero caso não haja *offset*. O controle PID atua diferentemente em cada par de motores, pois é necessário calcular o erro em cada eixo horizontal (x e y). A função de transferência do controle PID é dada pela Equação (2):

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2)$$

Em que T_i é igual a K_p/K_i e T_d é igual a K_d/K_p .

O esquema de controle em malha fechada demonstra o comportamento do sistema frente ao erro calculado, no qual a saída é definida pela ação do controlador PID no sinal PWM dos motores e sofre influência da dinâmica do quadrimotor devido às forças de empuxo e propulsão, está contido na Figura 10.

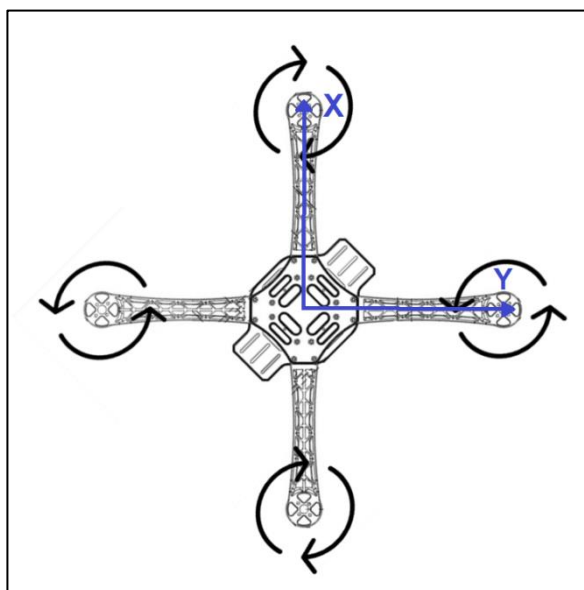
Figura 10 – Controle em malha fechada do sistema



Autoria: Adaptado de Sá (2012 p. 35).

No controle de estabilidade horizontal do VANT usam-se apenas 2 dos 3 eixos dos sensores, no qual cada eixo é responsável por um par de motores. Devido às reações de momento angular gerado pelas hélices, faz-se necessário que os motores de um eixo girem em sentido contrário aos do outro eixo. Esse é o princípio básico para que um VANT quadrimotor não gire em torno do próprio eixo, tal como na Figura 11.


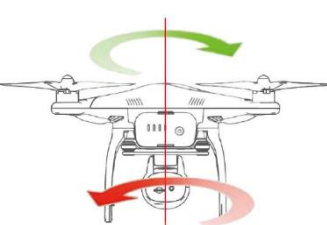
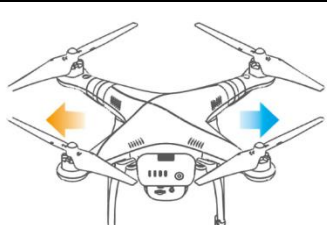
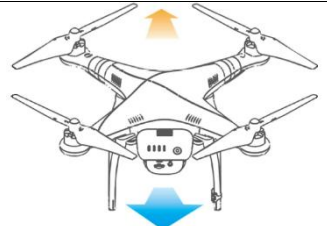
Figura 11 – Eixo X e Y dos sensores do VANT



Fonte: Autoria própria.

Os VANTs necessitam de certas combinações de velocidades entre seus motores para realizar deslocamentos verticais, horizontais ou em torno de seu próprio eixo, tais como descrito no Quadro 1, sendo estes os princípios utilizados em modelos matemáticos para desenvolvimento de controladores.

Quadro 1 – Funções e movimentos executados por um VANT

VANT	Função utilizada
	<p>Propulsão Vertical (do inglês Throttle): conforme indicado na imagem, é utilizada para realizar mudanças de altitude. Nesta função ambos os motores tem sua velocidade aumentada igualmente.</p>
	<p>Guinada (do inglês Yaw): utilizada para mudança de orientação, seja pro sentido horário ou anti-horário, ao realizar giro em torno do próprio eixo. Nesta função os motores que giram no sentido desejado devem ter sua velocidade aumentada.</p>
	<p>Rolagem (do inglês Roll): utilizada para realizar deslocamentos laterais, seja no sentido horário ou anti-horário. Nesta função o par de motores (esquerda ou direita) opostos ao sentido desejado devem ter sua velocidade aumentada.</p>
	<p>Arfagem (do inglês Pitch): utilizada para realizar deslocamentos para frente ou para trás. Nesta função o par de motores (frontal ou traseiro) opostos ao sentido desejado devem ter sua velocidade aumentada.</p>

Fonte: Adaptado de Santana (2016, p. 37).

2.2.2 Calibração do controle PID

Visto que um dos objetivos do presente trabalho é a aplicação de um controle PID em apenas dois motores, a fim de testar a plataforma montada e levando-se em consideração os demais componentes do VANT, influência do vento e outras características em seu funcionamento, a calibração e definição dos ganhos

do controle PID serão realizadas baseando-se nos resultados obtidos perante seus valores definidos previamente, de forma empírica. Sendo assim, será possível gerar gráficos com os valores dos erros, da saída do controle PID e individuais que o compõem (ação proporcional, ação derivativa e ação integrativa), no qual servirá de base para melhorar os valores das constantes do controle PID (K_p , K_d e K_i).

A análise do comportamento das ações individuais do controle PID possibilita aumentar ou diminuir os valores de K_p , K_d e K_i de acordo com o necessário para garantir a estabilidade. Aumentar o valor de K_i , por exemplo, reduz o erro em regime permanente. Por outro lado, alterar K_d corretamente irá suavizar a oscilação do VANT, diminuindo a força de possíveis variações (efeito gangorra). A alteração de K_p corrige o erro atual, porém, utilizar um valor elevado poderá causar instabilidade ou efeito gangorra. Existem inúmeras técnicas de sintonia de controle PID, tal como os métodos de Ziegler e Nichols, porém, demandam complexidade maior que o necessário para validação da plataforma.

3 MATERIAIS E MÉTODOS

As soluções e métodos de controle de alguns trabalhos não envolvem uma aplicação experimental, e apenas utilizam programas considerando condições ideais para simular o funcionamento do VANT, tal como desenvolvido por Botelho (2018) e Benigno (2015), nos quais não são consideradas as intempéries que podem causar instabilidade, ruídos na estrutura ou até mesmo leituras incorretas dos sensores.

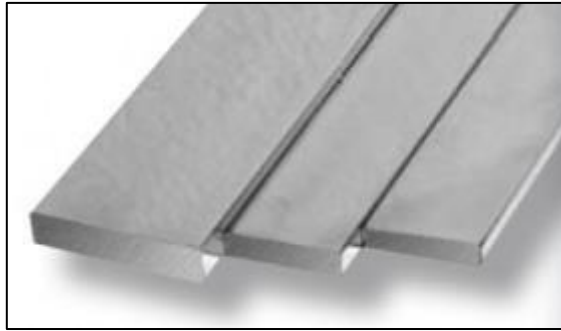
Sendo assim, ao realizar a aplicação experimental de um método de controle, o resultado poderá ser diferente da simulação e, além de se fazerem necessárias correções na programação, ajustes físicos também poderão ser pertinentes, tais como utilização de borrachas na fixação dos sensores para absorver vibrações ou troca de hélices empenadas por exemplo. A utilização de uma plataforma de testes proporciona, além validação de métodos de controle, o fomento da utilização de VANTs em novas pesquisas, não apenas em trabalhos de conclusão de curso. Para tanto, neste capítulo serão apresentados os materiais utilizados para a construção da plataforma, bem como a metodologia a ser empregada para testes e validação dos resultados

3.1 Hardware

3.3.1 Estrutura física

Necessária para a fixação dos componentes eletrônicos e motores, a estrutura precisa suportar as forças de impulso vertical e reduzir vibrações exercidas pelo giro das hélices. Caso seja feita de material demasiadamente maleável o VANT poderá sofrer interferência nos sensores, devido a vibrações que poderão ser interpretadas como movimento, e instabilidade causada pelas pequenas mudanças de angulação do eixo dos motores.

Com a finalidade de realizar testes dos motores e sensores, um protótipo inicial foi construído com barras retangulares de alumínio, por apresentarem peso reduzido e pouca flexibilidade, tais como as da Figura 12, com estrutura em forma de cruz (ângulos retos entre eixos).

Figura 12 – Barras Retangulares de Alumínio

Fonte: Adaptado de Incometal (2016).

A estrutura final da plataforma é do tipo *DJI Flame Wheel 450 (F450)*, feita do termoplástico ABS (do inglês *Acrylonitrile Butadiene Styrene*) em impressão 3D, com rigidez suficiente para reduzir vibrações e suportando hélices com tamanho de 8 a 10 polegadas. Além de conter placas de circuito impresso (PCI) que possibilitam realizar a alimentação dos componentes, a F450 tem massa de apenas 282g e suporta peso máximo de decolagem de até 1,6kg (DJI, 2015), sendo este o limite para alçar voo com segurança no qual a estrutura suporta. Projetada para diminuir as forças aerodinâmicas e facilitar a distribuição dos componentes, tem sido “escolhida não só por causa da resistência e leveza do material, mas também por causa do *layout*” (MARINESCU, 2013 p. 27), como pode ser observado na Figura 13.

Figura 13 – Estrutura *DJI Flame Wheel 450 F450*

Fonte: Adaptado de Marinescu (2013, p. 28).

3.3.2 Placa microcontroladora

O Arduino UNO é uma plataforma de baixo custo, código aberto e *software* de fácil utilização, que será utilizado como placa principal na plataforma de testes, que “é uma placa de microcontrolador baseado no ATmega328. Contém 14 entradas/saídas digitais, e seis delas podem ser usadas como saídas PWM” (D’AUSILIO, 2011, p. 3). Suas entradas analógicas serão responsáveis por receber os dados dos sensores e, através das saídas digitais, efetuará o acionamento dos motores em forma de sinal PWM. Com massa de apenas 25g e 6,86cm de comprimento por 5,34cm de largura, o Arduino UNO pode ser visualizado na Figura 14.

Figura 14 – Placa Arduino UNO



Fonte: Arduino UNO (2016).

O Arduino UNO utilizado na plataforma foi cedido temporariamente pelo Laboratório de Automação e Controle de Processos (Lab Control) da Universidade Tecnológica Federal do Paraná (UTFPR), campus Cornélio Procopio.

3.3.3 Alimentação do sistema

Alguns dos fatores determinantes para a definição do tipo de bateria são a necessidade de ter peso e tamanho reduzidos, capacidade de carga suficiente principalmente para alimentação dos motores, da placa microcontroladora e dos sensores que serão utilizados. O tipo de bateria mais utilizado em VANTs é a de Polímero de Lítio (Li-Po, do inglês *Lithium Polymer*), pois contém peso menor que as baterias de Chumbo-Ácido.

Com intuito de testar os motores e sensores, inicialmente foi adquirida uma bateria da marca *Turnigy*, ilustrada na Figura 15. A escolha por tal marca foi feita em razão da facilidade de aquisição no mercado nacional e seu baixo custo, no qual o modelo adquirido contém as seguintes características:

- 3 células (3S) em série com 3,7V cada, totalizando 11,1V de tensão nominal;
- capacidade de carga de 2200mAh;
- pode fornecer até 55A constantemente (25 vezes sua capacidade de carga);
- suporta até 77A por no máximo 10 segundos (35 vezes sua capacidade de carga);
- 10,6cm de comprimento por 2,4cm de largura e 3,4cm de altura;
- massa de apenas 188g.

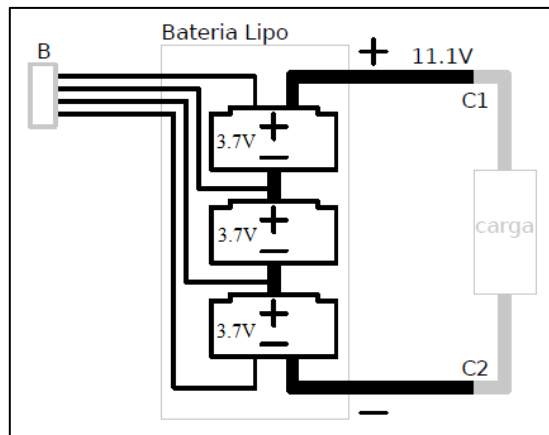
Figura 15 – Bateria *Turnigy* Li-Po 3S 2200mAh



Fonte: Adaptado de *Turnigy Blue* (2016).

Por ser uma bateria composta de três células conectadas em série, os valores de tensão de cada célula raramente são iguais. Rodrigues (2010, p. 13) ressalta que as baterias de Li-Po conseguem operar normalmente enquanto o valor de tensão de cada célula estiver entre 4,2 e 3, com risco de serem danificadas ou terem sua vida útil reduzida para valores fora do intervalo considerado. A disposição das células que compõem uma bateria Li-Po com 3 células pode ser observada na Figura 16, bem como o conector “B” que serve para o carregamento das células de forma individual.

Figura 16 – Disposição interna das células de baterias Li-Po 3S



Fonte: Adaptado de Melo (2011, p. 38).

Visando assegurar testes mais longos e utilizações constantes, optou-se pela aquisição de uma bateria com melhor qualidade e capacidade de carga, da marca *Zippy Flightmax*, ilustrada na Figura 17 e que também foi cedida temporariamente pelo Laboratório de Automação e Controle de Processos (Lab Control) da Universidade Tecnológica Federal do Paraná (UTFPR), campus Cornélio Procópio. As características mais relevantes desta bateria são:

- 3 células (3S) em série com 3,7V cada, totalizando 11,1V de tensão nominal;
- capacidade de carga de 4000mAh;
- pode fornecer até 160A de forma contínua (40 vezes sua capacidade de carga);
- suporta até 200A por no máximo 10 segundos (50 vezes sua capacidade de carga);
- 14,5cm de comprimento por 2,1cm de largura e 5,0cm de altura;
- massa de apenas 351g.

Figura 17 – Bateria *Zippy Flightmax* Li-Po 3S 4000mAh



Fonte: Autoria Própria.

3.3.4 Carregador de bateria

Em virtude das características intrínsecas das baterias de Li-Po, seu carregamento demanda cuidados especiais. É necessário conectar o terminal B da Figura 16) em carregadores próprios para baterias de Li-Po, que alimentam cada célula separadamente sem exceder seus limites e realizam o balanceamento das mesmas. Caso o carregamento seja realizado pelos terminais de saída (C1 e C2), a tensão de uma das células poderá aumentar além de sua capacidade (maior que 4,2V) e, como qualquer outra bateria, poderá superaquecer e até mesmo explodir.

Para minimizar os riscos de aquecimento, o carregamento não deve ser feito com corrente superior à capacidade de carga da bateria. O carregador que foi inicialmente utilizado é da marca *Turnigy*, com tensão de entrada de 12V e corrente máxima de saída de 800mA, abaixo dos 2200mAh da bateria *Turnigy*. Este modelo foi adquirido em conjunto da bateria, para auxílio nos testes pois possibilita o carregamento de baterias com 2 ou 3 células, podendo ser observado na Figura 18.

Figura 18 – Carregador de baterias Li-Po 2S/3S



Fonte: Autoria própria.

Para o carregamento da bateria *Zippy Flightmax* de 4000mAh foi utilizado o carregador *Accucel 6* da marca *Turnigy*, ilustrado na Figura 19, também cedido temporariamente pelo Laboratório de Automação e Controle de Processos (Lab Control) da Universidade Tecnológica Federal do Paraná (UTFPR), campus Cornélio Procópio. Dentre os principais benefícios deste carregador, vale destacar a possibilidade de selecionar a corrente de carregamento com variação de 100mA, podendo chegar até 6A no máximo. Além disso, existe a função *Auto Cut-Off*, que para automaticamente o carregamento quando a bateria se encontra com a corrente

máxima informada, a fim de evitar que a mesma seja carregada além de sua capacidade de carga.

Durante o carregamento, no *display* do carregador *Accucel 6* é possível verificar a tensão atual da bateria, a corrente atual de carregamento e o total de corrente injetada na bateria.

Figura 19 – Carregador de baterias Li-Po 1-6S *Accucel 6* Turnigy



Fonte: Autoria própria.

3.3.5 Motor de corrente contínua sem escovas

Diferente dos motores de corrente contínua com escovas, no qual a energização de suas bobinas tem comutação mecânica, os motores sem escovas (*brushless*) realizam comutação eletrônica. Em virtude desta diferença, Melo (2010, p. 112) afirma:

“[...] torna-se mais adequado o uso do motor DC *brushless* num quadrotor microcontrolado quando comparado com um motor DC com escovas, pois:

- por se controlar a energização das bobinas eletronicamente, o motor brushless gera menos ruído que o motor DC com escovas. Isto é importante pois os motores estão próximos dos circuitos de instrumentação do quadrotor;
- possuem rendimento superior aos motores DC com escova (bom para aplicações cuja fonte de energia é uma bateria).”

Outra particularidade do motor *brushless* é o fato do rotor ser magnetizado por ímãs permanentes, enquanto o estator é magnetizado pela corrente elétrica (NASCIMENTO, 2011, p. 26). São utilizados 4 motores *brushless* CF2822 da fabricante EMAX, como o ilustrado na Figura 20, por seu ótimo custo benefício e

facilidade de aquisição, além de ser “indicado por muitos que trabalham com aeromodelos” (SÁ, 2012, p.25). contendo as seguintes características:

- massa de apenas 39g;
- Eficiência máxima de 82%
- pode ser alimentado com tensão de 7,4V (2S) ou 11,1V (3S);
- potência de 166W;
- velocidade de 1200rpm/V;
- suporta 16A por até 60s;
- carrega uma massa de 840g drenando 15A com hélices 10x5.

Figura 20 – Motor *brushless* CF2822 da EMAX



Fonte: EMAX brushless motors (2016).

3.3.6 Hélices propulsoras

A escolha das hélices foi realizada levando-se em consideração as características da estrutura física F450, que suporta hélices com diâmetro de 8 a 10 polegadas, e dos motores CF2822 que suportam hélices de até 10 polegadas. Para não colocar em risco a F450 e não trabalhar utilizando o tamanho máximo suportado pelo CF2822, optou-se pelas hélices 9x5 do fabricante *Grand Wing Servo* (GWS). Este modelo de hélice tem diâmetro de giro de 9 polegadas e, considerando-se que ao rotacionar não haja deslizamentos, seriam necessárias 5 voltas completas para causar um deslocamento de 1 polegada (passo). Em virtude das características necessárias para que o VANT não fique girando em torno de si, são necessárias 2 hélices (9x5) que irão operar girando no sentido horário e 2 hélices (9x5r) no sentido anti-horário, comumente chamadas de contrarrotativas.

O adaptador necessário para fixação da hélice no eixo do motor é chamado de *spinner*, tal qual ilustrado à direita da Figura 21, que também ilustra um exemplar de hélice 9x5.

Figura 21 – Hélice 9x5 e adaptador spinner



Fonte: Autoria própria.

3.3.7 Controlador eletrônico de velocidade

Os *ESCs* são responsáveis pela alimentação dos motores e, devido à corrente máxima de operação do motor CF2822 ser 15A (com hélices 10x5), o *ESC* da EMAX com corrente mais próxima do necessário é o *Simon Series 20A*, que foi utilizado neste trabalho e pode ser visto na Figura 22. Algumas de suas características principais encontram-se a seguir:

- massa de apenas 28g;
- corrente de operação de 20A;
- suporta uma corrente de 25A por até 10s;
- opera com 7,4V (2S) ou 11,1V (3S).

Figura 22 – ESC Simon Series 20A da EMAX



Fonte: EMAX (2016).

Portanto, ainda que o motor CF2822 esteja operando com velocidade máxima e drenando uma corrente de 16A, o ESC irá manter o funcionamento sem riscos de operar em sobrecarga ou sofrer picos de corrente, nos casos de variações bruscas de velocidades.

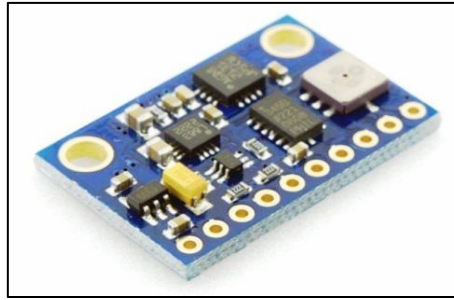
3.3.8 Módulo sensor

Para o VANT se manter estável o sistema precisa ter dados precisos como referência, sendo necessário que seus sensores tenham alta velocidade de leitura e resposta, bem como exatidão. Os sensores mais utilizados para garantir a estabilidade de VANTs são o acelerômetro e o giroscópio, dos quais seus valores são utilizados em cálculos matemáticos responsáveis por aumentar ou diminuir a velocidade dos motores.

Visando diminuir a quantidade de placas utilizadas na plataforma, optou-se pela aquisição de um módulo contendo ambos os tipos de sensores. O módulo GY-80 que será utilizado é uma unidade de medição inercial (IMU, do inglês *Inertial Measurement Unit*) multissensor que pode ser alimentado tanto com 5V quanto com 3,3V (disponíveis no Arduino UNO) e seus valores devem ser enviados através de comunicação I2C. Este módulo contém quatro sensores, que estão listados a seguir:

- Acelerômetro digital de 3 eixos no chip ADXL345;
- giroscópio digital de 3 eixos no chip L3G4200D;
- magnetômetro (bússola) de 3 eixos no chip HMC5883L;
- temperatura e pressão no chip BMP085;

A placa do módulo GY-80 que será utilizada neste trabalho se encontra na Figura 23.

Figura 23 – Módulo multissensor GY-80

Fonte: Arduino e Cia (2014).

Deste módulo foram utilizados os sensores acelerômetro e giroscópio, com o intuito de possibilitar a utilização das informações nos cálculos na programação. O acelerômetro digital de 3 eixos ADXL345 tem um consumo de energia de apenas 140 μ A (23 μ A no modo espera), resolução variável de 10 bits (valores entre $\pm 2g$) até 13 bits (valores próximos a $\pm 16g$), velocidade de saída dos dados de 800Hz e possibilita a leitura de angulações com erro de apenas $\pm 0,1^\circ$. (ANALOG, 2016).

O giroscópio digital de 3 eixos L3G4200D contém três escalas, sendo possível obter seus valores em 250, 500 ou 2000 graus por segundo. A corrente consumida em operação é de apenas 6,1mA e de 1,5mA no modo espera. Contém uma resolução de 16 bits e velocidade de saída dos dados ajustável de 100 a 800 ciclos por segundo (ST, 2010).

3.2 Resultado esperado e avaliação

Fisicamente, um VANT pode ser considerado estável se for capaz de restaurar seu equilíbrio frente a uma pequena perturbação. A qualidade do controle de estabilidade de um VANT pode ser definida pela sua capacidade de estabilização, que pode estar sujeito a pequenas variações constantes (desde que não se torne instável) ou, em um sistema perfeito, imobilidade total durante seu funcionamento. Com a construção da plataforma de testes, espera-se tornar possível realizar aplicações experimentais de métodos de controle. Após a implementação do código que será responsável pelo cálculo de velocidade dos motores, o qual utiliza os dados dos sensores como referência, o controle PID será implementado para testes com dois motores opostos. O resultado esperado neste

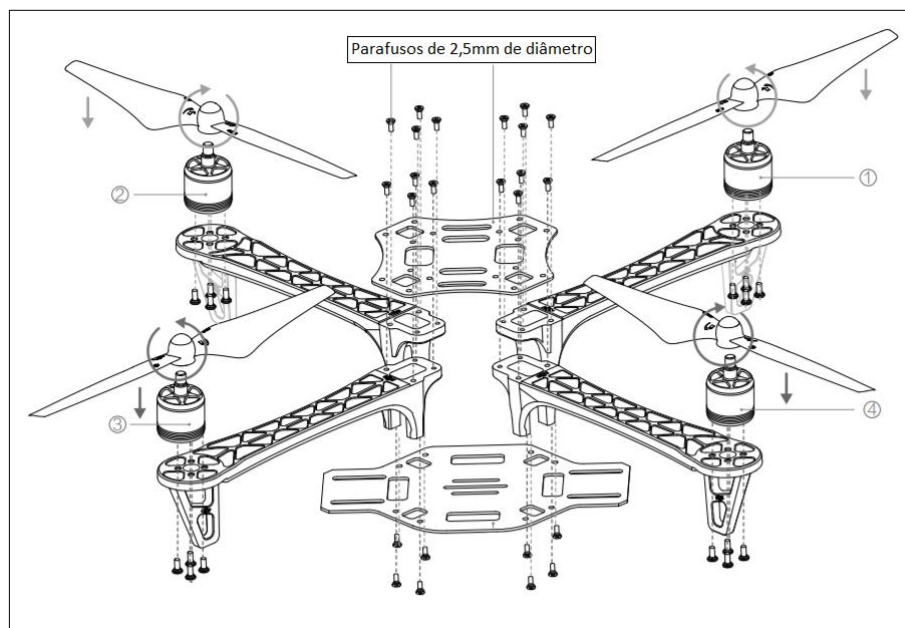
teste é que se consiga deixar o VANT ligado e estável, ainda que com apenas 2 motores, independentemente de pequenas variações.

4 RESULTADOS

4.1 Montagem da plataforma

A estrutura física F450 foi montada conforme orientação do manual, utilizando os parafusos de 2,5mm de diâmetro por 6mm de comprimento disponibilizados em conjunto, conforme visualizado na Figura 24.

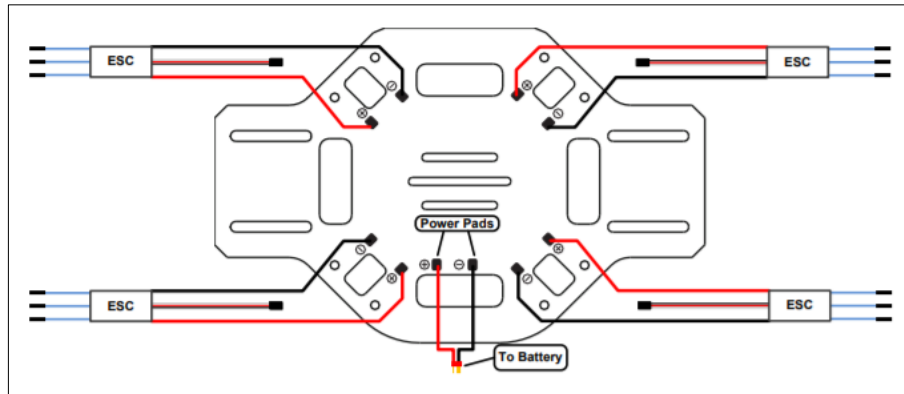
Figura 24 – Montagem da estrutura Física F450



Fonte: Adaptado de DJI (2015).

A base do motor CF2822 não tem furação igual ao do motor demonstrado na Figura 24, fazendo-se necessário fixar os motores utilizando parafusos com porcas. Além disso, ainda que as PCIs da F450 possibilitem soldar os terminais positivo e negativo da bateria e dos ESCs, conforme orientação da Figura 25 do manual da F450, evitou-se sua utilização visando facilitar uma possível troca de componentes

Figura 25 – Possibilidade de Energização dos ESCs via soldagem na PCB



Fonte: Adaptado de DJI (2013).

A conexão da bateria com os ESCs foi realizada utilizando-se conectores comumente chamados de *bullet* do tipo macho e fêmea, que são semelhantes aos bornes e pinos bananas encontrados em fontes. A plataforma final montada com todos os componentes, conexões de comunicação e de energia pode ser observada na Figura 26 .

Figura 26 – Plataforma para testes de controle de VANTS



Fonte: Autoria Própria.

4.2 Aferição dos sensores

Os testes com o módulo GY-80 (ARDUINO E CIA, 2014) foram realizados utilizando a biblioteca *Wire* do Arduino, onde a seleção do sensor a ser utilizado

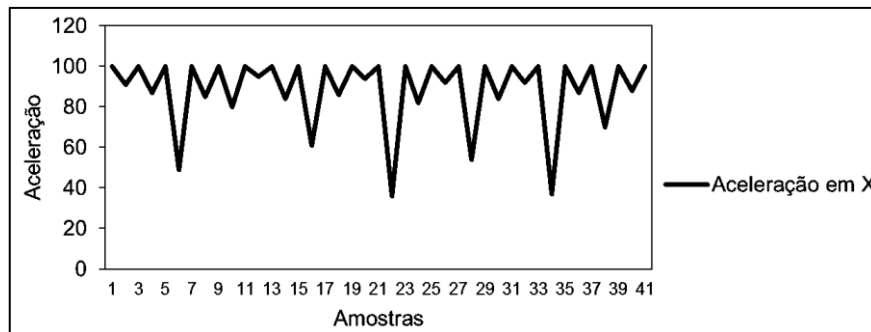
deve ser feita via código devido a comunicação I2C. Devido ao fato de o módulo GY-80 ser um multissensor, o endereço hexadecimal do sensor que será utilizado na comunicação I2C precisa ser informado na função *beginTransmission(endereço)*, da biblioteca *Wire* do Arduino, pois só assim será possível obter os valores de cada sensor em separado. O endereço de cada sensor específico pode ser encontrado através do código aberto chamado I2C Scanner, disponibilizado em diversos sites de conteúdo para Arduino. Dos sensores do módulo GY-80, o endereço hexadecimal do acelerômetro ADXL345 é 0x53 e do giroscópio L3G4200D é 0x69.

4.3.1 Acelerômetro de 3 eixos

Os dados de cada eixo do ADXL345 estão divididos em dois registradores com endereços diferentes para cada um dos 3 eixos do sensor, sendo necessário utilizar a função *write(endereço)* seguida da função *read()* para obter os valores de cada um deles. Os 10 bits de resolução estão divididos entre dois registradores, um contendo os 2 bits mais significativos e o outro 8 bits menos significativos. Portanto, é necessário utilizar um operador lógico bit-a-bit (<<) para deslocar os dados mais significativos para os primeiros 2 bits e somar com os 8 bits menos significativos para obter o valor real informado pelo sensor.

A visualização da velocidade angular foi feita com a ferramenta *Monitor Serial* usando a função *print(valor)*. Nas configurações escolhidas no teste a aceleração obtida tem valor nominal de 256 unidades para cada 1g (aceleração da gravidade, em m/s^2), sendo necessário dividi-lo por 256 para ter a aceleração em g. Utilizando apenas uma das hélices do eixo X, e variando manualmente a angulação da estrutura, foi possível obter 42 amostras de aceleração, ilustrados na Figura 27, que não foram convertidos para g, ou seja, não foram divididos por 256.

Figura 27 – Aceleração obtida em testes variando a posição do VANT



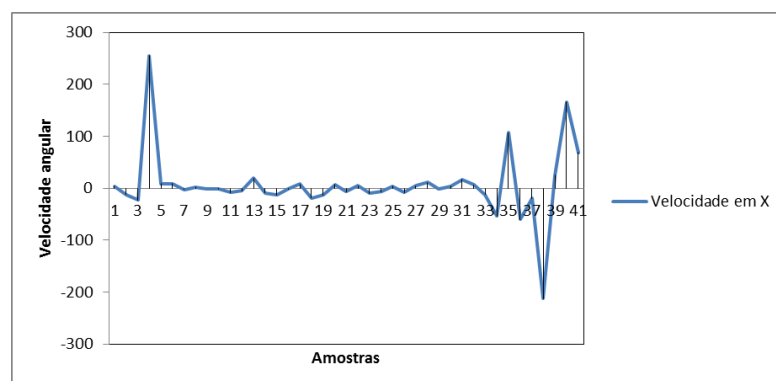
Fonte: Autoria própria.

4.3.2 Giroscópio de 3 eixos

As configurações para obter os valores do L3G4200D também utilizam a função `write(endereço)` seguido da função `read()`, com dois registradores por eixo igual ao acelerômetro. A resolução do L3G4200D contém 16 bits, que estão divididos entre dois registradores contendo 8 bits cada. Utilizando o operador lógico bit-a-bit (`<<`) foi possível obter a velocidade angular com a junção do valor obtido nos dois registradores.

A visualização da velocidade angular foi feita com a ferramenta *Monitor Serial* usando a função `print(valor)`. As 42 amostras de velocidade da Figura 28 foram obtidas utilizando apenas uma das hélices do eixo X e variando manualmente sua angulação, assim como no teste realizado no acelerômetro.

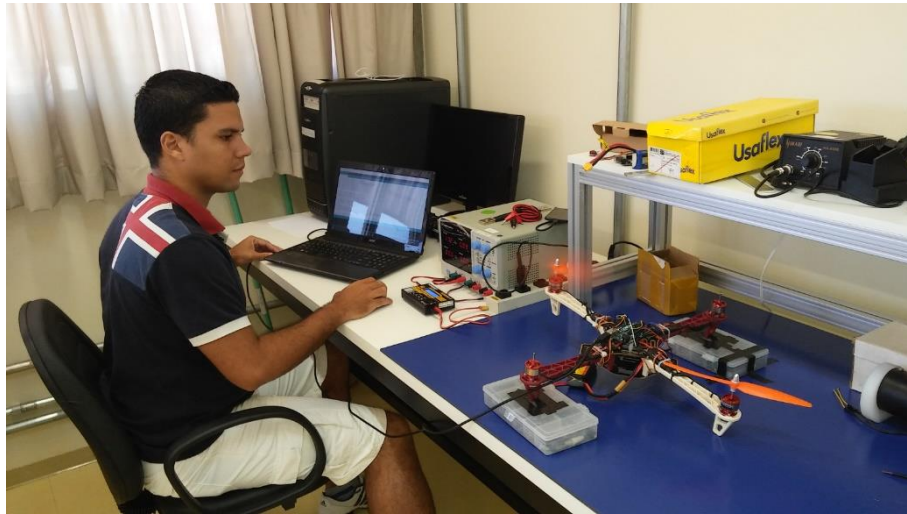
Figura 28 – Velocidade obtida em testes variando a posição do VANT



Fonte: Autoria própria.

Os testes foram realizados no Lab Control da UTFPR, campus Cornélio Procópio, conforme observado Figura 29.

Figura 29 – Testes realizados com o módulo GY-80



Fonte: Autoria própria.

4.3 Controle de velocidade dos motores

Os primeiros testes foram realizados utilizando a função `analogWrite(saída, vel)`, em que “vel” pode variar de 0 a 255 e saída se trata do pino PWM do Arduino. Foram utilizadas as saídas digitais do Arduino de número 3, 5, 6 e 9 que respectivamente controlam os motores 1, 2, 3 e 4 através de seus *ESCs*. Aumentando-se o valor de “vel” pouco a pouco desde 0 (zero), os motores só iniciaram o giro a partir de 135. Após o início do giro foi possível perceber um incremento suave de velocidade até “vel” atingir 255. Portanto, verdadeiramente esta função tem uma faixa de utilização de apenas 120 unidades.

Visando ter várias opções para o acionamento dos motores, alguns testes foram feitos utilizando a biblioteca *Servo* do Arduino. Ela contém duas funções amplamente utilizadas para acionamento de motores *brushless*, sendo a primeira `write()` e a segunda `writeMicroseconds()`. A configuração básica para essas duas funções é a atribuição da saída digital a ser usada, que deve ser indicada na função `attach(saída)`.

Com o comando `servo.write(entrada)`, no qual um valor de entrada de 0 a 180 proporciona um sinal na saída digital indicada (ARDUINO, 2016), os motores só começaram a girar com um valor superior a 50, não havendo mudança de velocidade após 150. Assim, esta função originalmente proporciona apenas uma

faixa de 100 unidades. Mesmo após fragmentar o valor de entrada, proporcionando um aumento da faixa de utilização, não houve incremento na precisão da variação de velocidade. Utilizando a função *writeMicroseconds(entrada)* um sinal é gerado na saída digital associada de acordo com o valor de entrada, que pode variar de 1000 a 2000, porém, os motores iniciaram giro pra um valor de entrada a partir de 1300. A resposta dos motores foi proporcional ao aumento do valor de entrada, inclusive em pequenas variações do valor de entrada. Com uma faixa de utilização maior e um aumento de velocidade gradual, será uma das funções que podem ser utilizadas no controle de velocidade dos motores do VANT.

4.4 Implementação do código de controle da plataforma

Com o intuito de realizar testes iniciais de controle da plataforma, um código de controle foi implementado considerando os métodos descritos na fundamentação teórica, realizando os cálculos necessários para executar o controle PID. Dado que os motores 1 e 3 estão dispostos sobre o eixo Y e os motores 2 e 4 sobre o eixo X, o sinal PWM que é enviado para os ESCs efetuarem o controle de velocidade dos motores, iniciada com um valor definido como *velini*, é calculado no código da seguinte maneira:

$$PWM1 = velini - (Kp * pro_erroy + Kd * der_erroy + Ki * int_erroy)$$

$$PWM2 = velini - (Kp * pro_errox + Kd * der_errox + Ki * int_errox)$$

$$PWM3 = velini + (Kp * pro_erroy + Kd * der_erroy + Ki * int_erroy)$$

$$PWM4 = velini + (Kp * pro_errox + Kd * der_errox + Ki * int_errox)$$

No qual os erros atuais do sistema são calculados no código pelas equações abaixo:

$$pro_errox = AX_out - AXoffset$$

$$pro_erroy = AY_out - AYoffset$$

Em que:

- *AX_out* e *AY_out* são a aceleração em X e Y lidas pelo sensor;
- *AXoffset* e *AYoffset* são os valores de aceleração em X e Y no qual o VANT se encontra alinhado horizontalmente, ou seja, são utilizados

para calibrar o sensor devido a uma possível inclinação do componente fixado.

Já as variações dos erros são calculadas pela equação abaixo, em que $Aerrox_ant$ e $Aerroy_ant$ são os erros anteriores:

$$der_errox = pro_errox - Aerrox_ant$$

$$der_erroy = pro_erroy - Aerroy_ant$$

As variáveis int_errox e int_erroy são a somatória dos erros do eixo X e Y e $velini$ é a velocidade inicial.

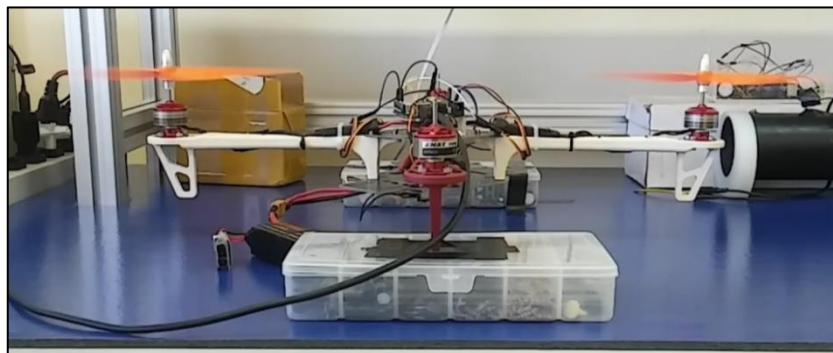
$$int_errox = int_errox + pro_errox$$

$$int_erroy = int_erroy + pro_erroy$$

4.5 Testes do controle PID na plataforma

Almejando realizar os testes do código implementado, o controle PID foi executado em dois motores do VANT, no qual deixou-se o eixo dos motores com certo grau de liberdade para tentativa de estabilidade como na Figura 30. Durante os testes foi utilizada uma função para envio das informações para o serial monitor do Arduino, visando guardar os dados para comparação e geração dos gráficos.

Figura 30 – Teste do controle de estabilidade com 2 motores

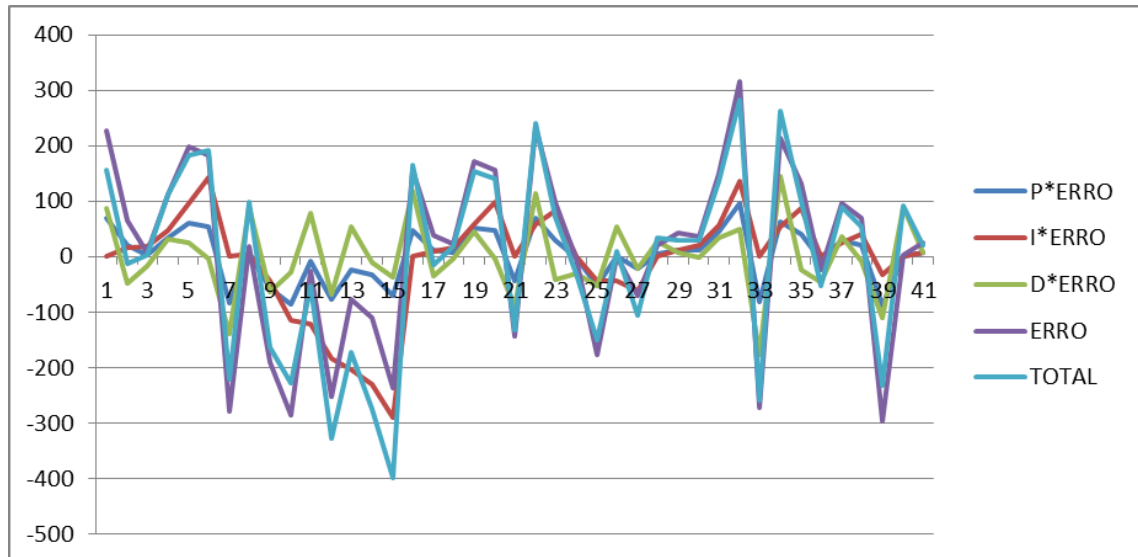


Fonte: A autoria própria

A função utilizada no controle de velocidade dos motores foi a `writeMicroseconds(entrada)`, com o valor de entrada igual a 1400 para o testes, por ser um valor com empuxo suficiente para inclinar a estrutura. Para não ocasionar mudanças bruscas de velocidade, ao realizar o controle de estabilidade, foram

definidas velocidades mínima e máxima, sendo respectivamente 1340 e 1460. Utilizando valores de K_p , K_i e K_d valendo 0,3, 0,25 e 0,3 respectivamente, os testes realizados com controle o PID proporcionaram dados que serviram de base para o gráfico da Figura 31, o qual utilizou-se 41 amostras para a geração da mesma.

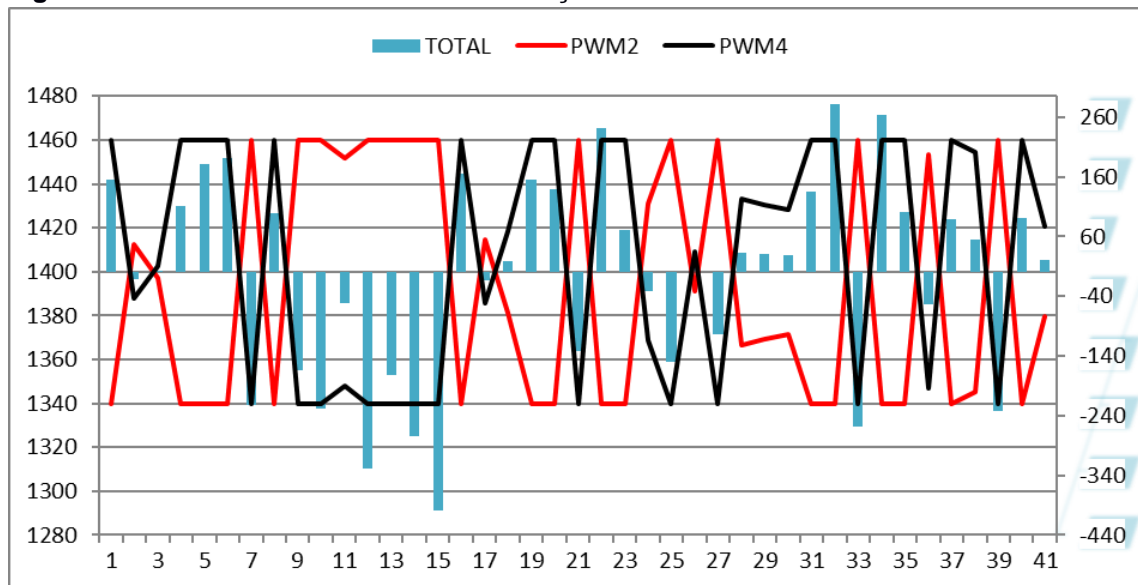
Figura 31 – Teste utilizando Controle PID com 2 motores



Fonte: Autoria própria

Os valores recebidos do acelerômetro nos testes não foram convertidos em g (m/s^2), sendo assim, cada 256 unidades correspondem a 1 g . Utilizando o valor de “TOTAL”, que refere-se ao sinal de controle PID ($K_p * pro_erroy + K_d * der_erroy + K_i * int_erroy$), pode-se calcular o valor do sinal PWM1 e PWM3 que correspondem respectivamente aos motores ligados nas saídas 5 e 9 do UNO. Em consequência de os motores nessas saídas serem opostos, ao mesmo tempo que o sinal de controle PID aumenta a velocidade do motor da saída 3, o motor da saída 6 tem sua velocidade diminuída, conforme gráfico da Figura 32.

Figura 32 – Sinal PWM dos motores em função do Controle PID



Fonte: Autoria própria

Os resultados obtidos foram satisfatórios pois, ainda que a velocidade dos motores permanecesse em seus limites (1340 e 1460) em algumas amostras, refletiu as pequenas oscilações que o VANT apresentou, que não foram tão bruscas nem sequer causaram instabilidade durante os testes.

Para melhoraria da estabilidade encontrada nos testes faz-se necessário a utilização de uma metodologia de sintonia do controle PID, assim como o método de Ziegler e Nichols, com o propósito de tornar o sistema estável inclusive perante perturbações que possam ser impostas manualmente.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A estrutura física F450 se mostrou bastante resistente nos testes, pois apenas em rotações muito altas que a mesma começou a vibrar além do normal, não suportando o empuxo e propulsão exercidos pelos motores CF2822 com as hélices de 9x5 da GWS. Caso fosse realizado uma propulsão vertical brusca, com velocidade alta nos motores, a estrutura poderia prejudicar o método de controle implementado, em consequência das vibrações exercidas nos eixos que iriam induzir erros nas leituras dos sensores.

A placa microcontroladora Arduino UNO operou de forma aceitável durante os testes realizados, visto que o código final tem por finalidade buscar a estabilidade do VANT e não realiza operações matemáticas para as funções de rolagem, guinada ou arfagem.

O módulo multissensor GY-80 teve tempo de resposta alto e sem variações inesperadas, pois mesmo ao realizar pequenas inclinações os valores foram refletidos proporcionalmente e de maneira rápida, em virtude de sua taxa de saída de dados ser de 800Hz.

Uma das células da bateria *Turnigy* de 2200mAh entrou em curto, após a realização constante de testes nos motores e do controle PID. O problema não ocorreu durante testes com velocidades altas, muito menos por culpa da utilização de um carregador de pouca qualidade, já que o modelo *Accucel 6* foi o único utilizado após sua obtenção. Em contrapartida, a bateria *Zippy Flightmax* de 4000mAh suportou uma quantidade de cargas e descargas bastante superior à da *Turnigy*, sem nenhuma mudança perceptível em sua durabilidade.

Os motores CF2822 da *EMAX* permaneceram funcionando sem folgas nos eixos ou problemas com rolamentos, mesmo com demasiados testes em rotações altas não tiveram aumento de temperatura relevante. Apesar disso, algumas hélices 9x5 da GWS utilizadas no motor ficaram empenadas após uma certa quantidade de utilizações, causando vibrações além do normal e interferências nos sensores.

Os resultados obtidos nos testes foram satisfatórios, pois demonstram a capacidade da plataforma tornar possível a utilização de métodos de controle para VANTs, através da programação de uma placa microcontroladora e realizando a leitura dos sensores. Para utilização dos sensores da plataforma não será necessário nenhum desenvolvimento além do realizado neste trabalho, bastando

aplicar as partes do código desenvolvido que correspondem respectivamente a cada tipo de sensor.

Portanto, como sugestão para trabalhos futuros, sugere-se implementar um controle completo de um VANT na plataforma, que realize todas as funções e movimentos, além de realizar comunicação entre um rádio transmissor e receptor, poderá ser necessária uma placa microcontroladora com mais capacidade de processamento, por causa do aumento na quantidade de dados.

Além disso, recomenda-se a utilização de hélices com diâmetro de 8 polegadas para um melhor desempenho na estrutura F450, pois a plataforma irá vibrar menos devido a interferência entre as hélices ser menor, uma vez que a distância entre as áreas de atuação das hélices seria maior. Outro ponto a ser melhorado é a fixação do módulo multissensor na plataforma, que deve ser fixado com anéis de borracha para absorção da vibração, afim de evitar erros de leitura em altas rotações.

REFERÊNCIAS

ANALOG. **Digital accelerometer ADXL345**. Disponível em: <<http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>>. Acesso em: 18 fev. 2016.

ARDUINO. **Servo library**. Disponível em: <<https://www.arduino.cc/en/Reference/Servo>>. Acesso em: 12 abr. 2016.
 _____. **Wire library**. Disponível em: <<https://www.arduino.cc/en/Reference/Wire>>. Acesso em: 12 abr. 2016.

ARDUINO E CIA. **Sensor GY-80 - Acelerômetro Giroscópio Magnetômetro e Barômetro**. Disponível em: <<http://www.arduinoocia.com.br/2014/09/sensor-gy-80-acelerometro-bussola-barometro.html>>. Acesso em: 18 fev. 2016.

ARDUINO PWM. **PWM**. Disponível em: <<https://www.arduino.cc/en/Tutorial/PWM>>. Acesso em: 15 maio 2016.

ARDUINO SERIAL. **Serial**. Disponível em: <<https://www.arduino.cc/en/Reference/Serial>>. Acesso em: 12 abr. 2016.

ARDUINO-UA. **Рама для квадрокоптера F450**. Disponível em: <<https://arduino-ua.com/prod726-Rama-dlya-kvadrokoptera-F450>>. Acesso em: 18 abr. 2016.

ARDUINO UNO. **Arduino UNO & Genuino UNO**. 2016. Disponível em: <<https://www.arduino.cc/en/main/arduinoBoardUno>>. Acesso em: 12 abr. 2016.

BRASIL. Ministério da Defesa. Comando da Aeronáutica. Departamento do Controle do Espaço Aéreo. **Circular de Informações Aeronáuticas (AIC) N 24\18**. Aeronaves remotamente pilotadas para uso exclusivo em operações dos órgãos de segurança pública, da defesa civil e de fiscalização da receita federal, 2018. Disponível em: <<https://publicacoes.decea.gov.br/download.cfm?d=4784>> Acesso em: 29 abr. 2019

BENIGNO Tayara CP. **Modelagem matemática e controle de atitude e posição do quadrotor**. 2015. 72f. Dissertação (Mestrado)- Pós graduação em Sistemas de Comunicação e Automação, Universidade Federal Rural do Semi-árido, Mossoró, 2015.

BOTELHO, Rodrigo. **Estudo de estabilização de um veículo quadricóptero não tripulado com carga pendular**. 2018.166f. Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânica, São Paulo, 2018. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3152/tde-04022019-153802/publico/RodrigodeCastroBakerBotelhoCorr18.pdf>>. Acesso em: 15 abr. 2019.

D'AUSILIO, Alessandro. **Arduino: A low-cost multipurpose lab equipment**. Disponível em: <<http://link.springer.com/content/pdf/10.3758%2Fs13428-011-0163-z.pdf>>. Acesso em: 14 abr. 2016.

DJI. **FlameWheel 450 user manual V2.1. 2013**. Disponível em: <http://dl.djicdn.com/downloads/flamewheel/en/F450_User_Manual_v2.1_en.pdf>. Acesso em: 27 maio 2019.

_____. **FlameWheel 450 user manual V2.2. 2015**. Disponível em: <http://dl.djicdn.com/downloads/flamewheel/en/F450_User_Manual_v2.2_en.pdf>. Acesso em: 27 maio 2019.

EMAX. **EMAX Simon Series 20A for muti-copter**. Disponível em: <<http://www.emaxmodel.com/esc/simonk-series/emax-simon-series-20a-for-muti-copter.html>>. Acesso em: 18 abr. 2016.

EMAX BRUSHLESS MOTORS. **EMAX brushless motors CF2822**. Disponível em: <<http://www.emaxmodel.com/brushless-motor/cf-series-motors/cf2822.html>>. Acesso em: 18 abr. 2016.

HARDGRAVE. **O pioneirismo com o VANT**. Estados Unidos da América, 2005. Disponível em: <<http://www.ctie.monash.edu.au/hargrave/>> Acesso em: 29 abr. 2019

INCOMETAL. **Barra retangular de alumínio**. Disponível em: <<http://www.incometal.com.br/produtos/aluminio-para-industria/barras-de-aluminio/barra-retangular-de-aluminio/>>. Acesso em: 18 abr. 2016.

JAXA-MALACHOWSKI, Ricardo. **Designing process of unmanned aerial vehicles**. In: INTERNATIONAL COUNCIL OF AERONAUTICAL SCIENCES CONGRESS, paper 4.2.3, 1996. p. 391-397.

KENDOUL, F. **Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems**. Journal of Field Robotics, v. 29, n. 2, p. 315–378, 2012. ISSN 1556-4967.

MARINESCU, Liana-Eleonora. **Autonomous quadcopter for noninvasive wildlife surveillance and analysis**. 2013. 50 f. Bachelor Thesis (Faculty of Automatic Control and Computers) – Automatic Control and Systems Engineering Department, University POLITEHNICA of Bucharest, Bucharest, 2013. Disponível em: <http://acse.pub.ro/wp-content/uploads/2013/07/Licenta_Marinescu_Liana_341B31.pdf>. Acesso em: 13 abr. 2016.

MAZZINI, Humberto M. **Métodos de sintonia de controladores PID**. Disponível em: <<http://www.ppgel.net.br/mazzini/sintonia.pdf>>. Acesso em: 22 maio 2016.

MELO, Alexandre S. de. **Implementação de um Quadrotor como Plataforma de Desenvolvimento para Algoritmos de Controle**. 2010. 114 f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2010. Disponível em:

<http://portais4.ufes.br/posgrad/teses/tese_2872_DissertacaoMestradoAlexandreSecchindeMelo.pdf>. Acesso em: 18 fev. 2016.

MPS. **Brushless DC motor fundamentals application note**. Disponível em: <<https://www.monolithicpower.com/Portals/0/Documents/Products/Documents/appnotes/Brushless%20DC%20Motor%20Fundamentals.pdf>>. Acesso em: 21 abr. 2016.

NASCIMENTO, David A. do. **Análise de requisitos de hardware em projeto de UAV quadrotor**. 2011. 72 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Engenharia de Computação – Universidade de Pernambuco, Recife, 2011. Disponível em: <http://tcc.ecomp.poli.br/20112/david_alain_uav_quadrotor.pdf>. Acesso em: 22 maio 2016.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Editora Pearson / Prentice-Hall, 2011.

OLIVEIRA, Flavio Araripe. **CTA e o Projeto VANT**. In: 1º Seminário Internacional de Vant. São José dos Campos, 2005. Palestra proferida no Centro Tecnológico da Aeronáutica. Disponível em: <https://www.defesa.gov.br/arquivos/pdf/ciencia_tecnologia/7_seminarioC_T/7_quarta/DCTA.pdf> Acesso em: 29 abr. 2019

PFEIFER Erick. **Projeto e controle de um UAV quadrirotor**. 2013. 132f. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, 2013. Disponível em: <https://www.teses.usp.br/teses/disponiveis/3/3139/tde-06072014223705/publico/Dissertacao_Erick.pdf>. Acesso em: 15 abr. 2019

RODRIGUES, João P. Q. **Gestão inteligente da carga das baterias de um robô futebolista**. 2011. 99 f. Dissertação (Mestrado em Engenharia Eletrônica Industrial e Computadores) – Escola de Engenharia, Universidade do Minho, Braga, 2010. Disponível em: <<http://intranet.dei.uminho.pt/gdmi/galeria/temas/pdf/50073.pdf>>. Acesso em: 17 maio 2016.

SÁ, Rejane C. **Construção, modelagem dinâmica e controle PID para estabilidade de um veículo aéreo não tripulado do tipo quadrirotor**. 2012. 94 f. Dissertação (Mestrado em Engenharia de Teleinformática) – Programa de Pós Graduação em Engenharia de Teleinformática, Universidade Federal de Fortaleza, Fortaleza, 2012. Disponível em: <http://www.repositorio.ufc.br/bitstream/riufc/4097/1/2012_dis_rcsa.pdf>. Acesso em: 19 abr. 2016.

SANTANA Lucas Vago. **Sistemas de navegação e controle para veículos aéreos não tripulados e suas aplicações**. 2016. 153f. Tese (Doutorado em Engenharia Elétrica)- Programa de Pós graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2016. Disponível em: <http://repositorio.ufes.br/jspui/bitstream/10/9685/1/tese_10333_Tese%20de%20Doutorado%20-%20Lucas%20Vago%20Santana.pdf>. Acesso em: 15 abr. 2019.

ST. **Digital output gyroscope L3G4200D**. 2010. Disponível em: <<http://www.st.com/content/ccc/resource/technical/document/datasheet/04/46/d6/00/>>

be/d9/46/ae/CD00265057.pdf/files/CD00265057.pdf/jcr:content/translations/en.CD00265057.pdf>. Acesso em: 18 fev. 2016.

TURNIGY. **Chargers**. Disponível em: <<http://www.turnigy.com/chargers/>>. Acesso em: 13 abr. 2016.

TURNIGY BLUE. **Turnigy blue battery**. Disponível em: <<http://www.turnigy.com/batteries/turnigy-blue/>>. Acesso em: 13 abr. 2016.

THOMPSON, L.A.; BIL, C. **The design and flight trials of a multi-purpose autonomous flight vehicle system**. 1994. Disponível em: <https://www.icas.org/ICAS_ARCHIVE/ICAS1994/ICAS-94-6.3.3.pdf>. Acesso em: 29 abr. 2019.

VICENTE, João P. N. **Diferenciadores estratégicos da Guerra Aérea Remota**. Disponível em: <<http://www.scielo.mec.pt/pdf/ri/n40/n40a10.pdf>>. Acesso em: 22 maio 2016.

WIDMAIER, Klaus. **Algoritmo genético aplicado à otimização de asas de material compósito de veículos aéreos não tripulados, 2005, 181f**. Dissertação (Mestrado) - Escola de Engenharia de São Carlos da Universidade de São Paulo. São Carlos – SP, 2005.

APÊNDICE A – Código do Controlador PID de Estabilidade do VANT

//Código desenvolvido para teste dos sensores e do controle de estabilidade do VANT utilizando controle PID

//Os códigos para leitura dos sensores foram adaptados de alguns códigos abertos encontrados em fóruns de Arduino, que são bastante utilizados para leitura dos dados do módulo GY-80

```
#include <Servo.h>
```

```
#include <Wire.h>
```

```
#define BAddress 0x1E // Endereço I2C da bussola
```

```
#define L3G4200D_Address 105 // Endereço I2C do giroscopio
```

```
#define CTRL_REG1 0x20 // Endereço do Registrador 1 do giroscopio
```

```
#define CTRL_REG2 0x21 // Endereço do Registrador 2 do giroscopio
```

```
#define CTRL_REG3 0x22 // Endereço do Registrador 3 do giroscopio
```

```
#define CTRL_REG4 0x23 // Endereço do Registrador 4 do giroscopio
```

```
#define CTRL_REG5 0x24 // Endereço do Registrador 5 do giroscopio
```

```
#define Register_ID 0 // Endereço I2C do ID do dispositivo
```

```
#define ADXAddress 0x53 // Endereço I2C do acelerômetro
```

```
#define Register_2D 0x2D // Endereço I2C da economia de energia do acelerômetro
```

```
#define Register_AX0 0x32 // Endereço I2C do Dado 0 da aceleração em X
```

```
#define Register_AX1 0x33 // Endereço I2C do Dado 1 da aceleração em X
```

```
#define Register_AY0 0x34 // Endereço I2C do Dado 0 da aceleração em Y
```

```
#define Register_AY1 0x35 // Endereço I2C do Dado 1 da aceleração em Y
```

```
#define Register_AZ0 0x36 // Endereço I2C do Dado 0 da aceleração em Z
```

```
#define Register_AZ1 0x37 // Endereço I2C do Dado 1 da aceleração em Z
```

```
Servo ESC1; //Habilita as funções da biblioteca Servo pra ESC1
```

```
Servo ESC2; //Habilita as funções da biblioteca Servo pra ESC2
```

```
Servo ESC3; //Habilita as funções da biblioteca Servo pra ESC3
```

```
Servo ESC4; //Habilita as funções da biblioteca Servo pra ESC4
```

```
/* Inicialização de variáveis */
```

```
int AX0, AX1, AX_out; //Variaveis utilizadas no acelerometro
```

```
int AY0, AY1, AY_out; //Variaveis utilizadas no acelerometro
```

```
int AZ0, AZ1, AZ_out; //Variaveis utilizadas no acelerometro
```

```
int i=1;
```

```

double P=0.4, I=0.08, D=0.5; // P/I > 5
//double P=0, I=0, D=0; // para testes
double AXoffset=4, AYoffset=2;
double AXf, AYf, AZf, AXant=0, AYant=0, AZant=0, Aerrox, Aerroy, Aerroz;
//Variaveis utilizadas no acelerometro
double AXmax=0, AYmax=0, AZmax=0, AXmin=100, AYmin=100, AZmin=100;
//Variaveis utilizadas no acelerometro

double int_errox, int_erroy, int_erroz, der_errox, der_erroy, der_erroz; // Variaveis
utilizadas no controle PID
double pro_errox, pro_erroy, pro_erroz, Aerrox_ant=0, Aerroy_ant=0, Aerroz_ant=0;
// Variaveis utilizadas no controle PID

int BX, BY, BZ; //Variaveis utilizadas na bussola
double BXant=0, BYant=0, BZant=0, Berrox, Berroy, Berroz; //Variaveis utilizadas na
bussola
double BXmax=0, BYmax=0, BZmax=0, BXmin=100, BYmin=100, BZmin=100;
//Variaveis utilizadas na bussola

int GX, GY, GZ; //Variaveis utilizadas no giroscópio
double GXant=0, GYant=0, GZant=0, Gerrox, Gerroy, Gerroz; //Variaveis utilizadas
no giroscópio
double GXmax=0, GYmax=0, GZmax=0, GXmin=100, GYmin=100, GZmin=100;
//Variaveis utilizadas no giroscópio

double velini = 1400; //Velocidade angular inicial dos motores (iniciam o giro em
aproximadamente 1000)
double velmin = velini-60;
double velmax = velini+60; //Velocidade maxima que sera aplicada nos motores,
sendo 2000 o maximo possível
double PWM1 = velini;
double PWM2 = velini;
double PWM3 = velini;
double PWM4 = velini;

```

```

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  /* Configuração e Inicialiação do giroscópio*/
  setupL3G4200D(2000); // Configura o L3G4200 para 200, 500 ou 2000 graus/seg

  /* Configuração e Inicialiação do acelerômetro*/
  Wire.beginTransmission(ADXAddress); //Inicia a comunicação com o acelerometro
  Wire.write(Register_2D); //Inicia o modo de economia de energia
  Wire.write(8);
  Wire.endTransmission(); // Termina a configuração do acelerometro

  /*Configuração e inicialiação da bussola */
  Wire.begin();
  Wire.beginTransmission(BAddress); // Inicia a comunicação com a bussola
  Wire.write(0x02); // Seleciona o modo de medição contínua
  Wire.write(0x00); // Seleciona o modo de medição contínua
  Wire.endTransmission();

  /* Configuração das saídas PWM do Arduino */
  ESC1.attach(3); //Atribui o Pino 3 (PWM digital) como saída pro ESC1
  ESC2.attach(5); //Atribui o Pino 5 (PWM digital) como saída pro ESC2
  ESC3.attach(6); //Atribui o Pino 6 (PWM digital) como saída pro ESC3
  ESC4.attach(9); //Atribui o Pino 9 (PWM digital) como saída pro ESC4
}

void print_errox(){
  Serial.print("\t");
  Serial.print(P*pro_errox);
  Serial.print("\t\t");
  Serial.print(I*int_errox);
  Serial.print("\t\t");
  Serial.print(D*der_errox);
  Serial.print("\t\t");
}

```

```

Serial.print(AXf);
Serial.print("\t\t");
Serial.print(AXmax);
Serial.print("\t\t");
Serial.print(P*pro_errox + D*der_errox + I*int_errox);
Serial.print("\n");
}
void print_erroy(){
  Serial.print("\t");
  Serial.print(P*pro_erroy);
  Serial.print("\t\t");
  Serial.print(I*int_erroy);
  Serial.print("\t\t");
  Serial.print(D*der_erroy);
  Serial.print("\n");
}
void print_ancel(){
// Imprime os valores do acelerometro no serial monitor
  acelerometro();
  Serial.print("Erro X = ");
  Serial.println(AXf);
  Serial.print("Erro Y = ");
  Serial.println(AYf);
  Serial.print("Erro Z = ");
  Serial.println(AZf);
  Serial.print("\n\n\n");
}
void print_esc(){
Serial.print("\t\t\t\t\tMotor 1 graus = ");
Serial.println(PWM1);
Serial.print("\nMotor 4 graus = ");
Serial.print(PWM4);
Serial.print("\t\t\t\t\tMotor 2 graus = ");
Serial.println(PWM2);

```

```

Serial.print("\n\t\t\t\t\tMotor 3 graus = ");
Serial.println(PWM3);
Serial.print("\n\n\n");
}

/////////////////////////////////////////////////////////////////
                        /* Acelerometro */
void acelerometro(){
//-----X
Wire.beginTransmission(ADXAddress); // transmit to device
Wire.write(Register_AX0);
Wire.write(Register_AX1);
Wire.endTransmission();
Wire.requestFrom(ADXAddress,2);
if(Wire.available()<=2)
{
  AX0 = Wire.read();
  AX1 = Wire.read();
  AX1=AX1<<8; //desloca AX1 8 bits a esquerda (mesmo que multiplicar por 2^8)
  AX_out=AX0+AX1;
}

//-----Y
Wire.beginTransmission(ADXAddress); // transmit to device
Wire.write(Register_AY0);
Wire.write(Register_AY1);
Wire.endTransmission();
Wire.requestFrom(ADXAddress,2);
if(Wire.available()<=2)
{
  AY0 = Wire.read();
  AY1 = Wire.read();
  AY1=AY1<<8; //desloca AY1 8 bits a esquerda (mesmo que multiplicar por 2^8)
  AY_out=AY0+AY1;
}

```

```

}
//-----Z
Wire.beginTransaction(ADXAddress); // transmit to device
Wire.write(Register_AZ0);
Wire.write(Register_AZ1);
Wire.endTransmission();
Wire.requestFrom(ADXAddress,2);
if(Wire.available()<=2)
{
  AZ0 = Wire.read();
  AZ1 = Wire.read();
  AZ1=AZ1<<8; //desloca AZ1 8 bits a esquerda (mesmo que multiplicar por 2^8)
  AZ_out=AZ0+AZ1;
}
//
AXf=AX_out - AXoffset; // Calcula a aceleração em X, sendo 1g a cada 256bits
AYf=AY_out - AYoffset; // Calcula a aceleração em Y, sendo 1g a cada 256bits
AZf=AZ_out; // Calcula a aceleração em Z, sendo 1g a cada 256bits
Aerrox = AXf - AXant; //Calcula a variação da aceleração no eixo X
Aerroy = AYf - AYant; //Calcula a variação da aceleração no eixo Y
Aerroz = AZf - AZant; //Calcula a variação da aceleração no eixo Z
if (Aerrox>=AXmax && Aerrox!=AXf)AXmax = Aerrox; //Verifica se a variação em X
é maxima
if (Aerroy>=AYmax && Aerroy!=AYf)AYmax = Aerroy; //Verifica se a variação em Y
é maxima
if (Aerroz>=AZmax && Aerroz!=AZf)AZmax = Aerroz; //Verifica se a variação em Z
é maxima
if (Aerrox<=AXmin)AXmin = Aerrox; //Verifica se a variação em X é minima
if (Aerroy<=AYmin)AYmin = Aerroy; //Verifica se a variação em Y é minima
if (Aerroz<=AZmin)AZmin = Aerroz; //Verifica se a variação em Z é minima
AXant = AXf;
AYant = AYf;
AZant = AZf;

```

```

if(AXf>100) AXf = 100; //Limita em 60 o máximo erro em X (máximo da mesa de
testes com o motor 4 abaixado)
if(AXf<-55) AXf = -55; //Limita em -60 o mínimo erro em X (mínimo da mesa de
testes com o motor 2 abaixado)
pro_errox = AXf;
pro_erroy = AYf;
pro_erroz = AZf;
int_errox = int_errox + AXf; //Calcula a integral do erro em X do acelerometro
int_erroy = int_erroy + AYf; //Calcula a integral do erro em Y do acelerometro
int_erroz = int_erroz + AZf; //Calcula a integral do erro em Z do acelerometro
if (int_errox>0 && AXf < 0 ) int_errox = 0; // Zera a parte integrativa do eixo X após
o cruzamento de eixo
if (int_erroy>0 && AYf < 0 ) int_erroy = 0; // Zera a parte integrativa do eixo Y após
o cruzamento de eixo
if (int_erroz>0 && AZf < 0 ) int_erroz = 0; // Zera a parte integrativa do eixo Z após o
cruzamento de eixo
if (int_errox<0 && AXf > 0 ) int_errox = 0; // Zera a parte integrativa do eixo X após
o cruzamento de eixo
if (int_erroy<0 && AYf > 0 ) int_erroy = 0; // Zera a parte integrativa do eixo Y após
o cruzamento de eixo
if (int_erroz<0 && AZf > 0 ) int_erroz = 0; // Zera a parte integrativa do eixo Z após o
cruzamento de eixo
if (i==1){
    der_errox = AXf; //Calcula a variação inicial do erro em X do acelerometro
    der_erroy = AYf; //Calcula a variação inicial do erro em Y do acelerometro
    der_erroz = AZf; //Calcula a variação inicial do erro em Z do acelerometro
    i=i+1;
}
else{
    der_errox = AXf - Aerrox_ant; //Calcula a variação do erro em X do acelerometro
    der_erroy = AYf - Aerroy_ant; //Calcula a variação do erro em Y do acelerometro
    der_erroz = AZf - Aerroz_ant; //Calcula a variação do erro em Z do acelerometro
}
Aerrox_ant = AXf;

```

```

Aerroy_ant = AYf;
Aerroz_ant = AZf;
}

////////////////////////////////////
                        /* Bussola */
void bussola(){
  Wire.beginTransmission(BAddress); // Inicia a leitura
  Wire.write(0x03); //select register 3, X MSB register
  Wire.endTransmission();

  // Le os dados de cada eixo, 2 registradores por eixo
  Wire.requestFrom(BAddress, 6); //Recebe a leitura de bit a bit da bussola, de 6
bytes
  if(6<=Wire.available())
  {
    BX = Wire.read()<<8; //Bits mais significativos da coordenada X
    BX |= Wire.read(); //Bits menos significativos da coordenada X
    BZ = Wire.read()<<8; //Bits mais significativos da coordenada Z
    BZ |= Wire.read(); //Bits menos significativos da coordenada Z
    BY = Wire.read()<<8; //Bits mais significativos da coordenada Y
    BY |= Wire.read(); //Bits menos significativos da coordenada Y
  }
  Berrox = BX - BXant; //Calcula a variação do angulo no eixo X
  Berroy = BY - BYant; //Calcula a variação do angulo no eixo Y
  Berroz = BZ - BZant; //Calcula a variação do angulo no eixo Z
  if (Berrox>=BXmax && Berrox!=BX)BXmax = Berrox; //Verifica se a variação
angular em X é maxima
  if (Berroy>=BYmax && Berroy!=BY)BYmax = Berroy; //Verifica se a variação
angular em Y é maxima
  if (Berroz>=BZmax && Berroz!=BZ)BZmax = Berroz; //Verifica se a variação
angular em Z é maxima
  if (Berrox<=BXmin)BXmin = Berrox; //Verifica se a variação angular em X é minima
  if (Berroy<=BYmin)BYmin = Berroy; //Verifica se a variação angular em Y é minima

```



```

void giroscopio()
{
    // Rotina para leitura dos valores de X, Y e Z
    byte xMSB = readRegister(L3G4200D_Address, 0x29);
    byte xLSB = readRegister(L3G4200D_Address, 0x28);
    GX = ((xMSB << 8) | xLSB);

    byte yMSB = readRegister(L3G4200D_Address, 0x2B);
    byte yLSB = readRegister(L3G4200D_Address, 0x2A);
    GY = ((yMSB << 8) | yLSB);

    byte zMSB = readRegister(L3G4200D_Address, 0x2D);
    byte zLSB = readRegister(L3G4200D_Address, 0x2C);
    GZ = ((zMSB << 8) | zLSB);
    Gerrox = GX - GXant; //Calcula a variação da velocidade angular no eixo X
    Gerroy = GY - GYant; //Calcula a variação da velocidade angular no eixo Y
    Gerroz = GZ - GZant; //Calcula a variação da velocidade angular no eixo Z
    if (Gerrox>=GXmax && Gerrox!=GX)GXmax = Gerrox; //Verifica se a variação da
    velocidade angular em X é maxima
    if (Gerroy>=GYmax && Gerroy!=GY)GYmax = Gerroy; //Verifica se a variação da
    velocidade angular em Y é maxima
    if (Gerroz>=GZmax && Gerroz!=GZ)GZmax = Gerroz; //Verifica se a variação da
    velocidade angular em Z é maxima
    if (Gerrox<=GXmin)GXmin = Gerrox; //Verifica se a variação da velocidade angular
    em X é minima
    if (Gerroy<=GYmin)GYmin = Gerroy; //Verifica se a variação da velocidade angular
    em Y é minima
    if (Gerroz<=GZmin)GZmin = Gerroz; //Verifica se a variação da velocidade angular
    em Z é minima
    GXant = GX;
    GYant = GY;
    GZant = GZ;
    Serial.print("Giroscopio 3 Eixos - valor em graus/s \n");
    Serial.print("X = ");

```

```

Serial.print(GX);
Serial.print(" graus/s, Variacao de X = ");
Serial.print(Gerrox);
Serial.print(" graus/s, Variacao max de X = ");
Serial.print(GXmax);
Serial.print(" graus/s, Variacao min de X = ");
Serial.print(GXmin);
Serial.print("\nY = ");
Serial.print(GY);
Serial.print(" graus/s, Variacao de Y = ");
Serial.print(Gerroy);
Serial.print(" graus/s, Variacao max de Y = ");
Serial.print(GYmax);
Serial.print(" graus/s, Variacao min de Y = ");
Serial.print(GYmin);
Serial.print("\nZ = ");
Serial.print(GZ);
Serial.print(" graus/s, Variacao de Z = ");
Serial.print(Gerroz);
Serial.print(" graus/s, Variacao max de Z = ");
Serial.print(GZmax);
Serial.print(" graus/s, Variacao min de Z = ");
Serial.print(GZmin);
Serial.print("\n\n");
}

int setupL3G4200D(int scale)
{
  //From Jim Lindblom of Sparkfun's code

  // Enable x, y, z and turn off power down:
  writeRegister(L3G4200D_Address, CTRL_REG1, 0b00001111);
}

```

// If you'd like to adjust/use the HPF, you can edit the line below to configure CTRL_REG2:

```
writeRegister(L3G4200D_Address, CTRL_REG2, 0b00000000);
```

// Configure CTRL_REG3 to generate data ready interrupt on INT2

// No interrupts used on INT1, if you'd like to configure INT1

// or INT2 otherwise, consult the datasheet:

```
writeRegister(L3G4200D_Address, CTRL_REG3, 0b00001000);
```

// CTRL_REG4 controls the full-scale range, among other things:

```
if(scale == 250){
```

```
    writeRegister(L3G4200D_Address, CTRL_REG4, 0b00000000);
```

```
}else if(scale == 500){
```

```
    writeRegister(L3G4200D_Address, CTRL_REG4, 0b00010000);
```

```
}else{
```

```
    writeRegister(L3G4200D_Address, CTRL_REG4, 0b00110000);
```

```
}
```

// CTRL_REG5 controls high-pass filtering of outputs, use it

// if you'd like:

```
writeRegister(L3G4200D_Address, CTRL_REG5, 0b00000000);
```

```
}
```

```
void writeRegister(int deviceAddress, byte address, byte val)
```

```
{
```

```
    Wire.beginTransaction(deviceAddress); // start transmission to device
```

```
    Wire.write(address); // send register address
```

```
    Wire.write(val); // send value to write
```

```
    Wire.endTransmission(); // end transmission
```

```
}
```

```
int readRegister(int deviceAddress, byte address)
```

```
{
```

```
    int v;
```

```

Wire.beginTransmission(deviceAddress);
Wire.write(address); // register to read
Wire.endTransmission();

Wire.requestFrom(deviceAddress, 1); // read a byte

while(!Wire.available())
{
    // waiting
}
v = Wire.read();
return v;
}
/////Função para realizar testes de velocidade em rampa/////
void motortest(){
    PWM1= PWM1+1;
    PWM2= PWM2+1;
    PWM3= PWM3+1;
    PWM4= PWM4+1;
    delay(100);
    if (PWM1>velmax) PWM1 = velini;
    if (PWM2>velmax) PWM2 = velini;
    if (PWM3>velmax) PWM3 = velini;
    if (PWM4>velmax) PWM4 = velini;
    ESC1.writeMicroseconds(PWM1);
    ESC2.writeMicroseconds(PWM2);
    ESC3.writeMicroseconds(PWM3);
    ESC4.writeMicroseconds(PWM4);
}
/////Função do Controle PID/////
void controle(){
    PWM1 = velini - (P*pro_erroy + D*der_erroy + I*int_erroy);
    PWM2 = velini - (P*pro_errox + D*der_errox + I*int_errox);
    PWM3 = velini + (P*pro_erroy + D*der_erroy + I*int_erroy);

```

```
PWM4 = velini + (P*pro_errox + D*der_errox + I*int_errox);
if (PWM1>velmax) PWM1 = velmax;
if (PWM2>velmax) PWM2 = velmax;
if (PWM3>velmax) PWM3 = velmax;
if (PWM4>velmax) PWM4 = velmax;
if (PWM1<velmin) PWM1 = velmin;
if (PWM2<velmin) PWM2 = velmin;
if (PWM3<velmin) PWM3 = velmin;
if (PWM4<velmin) PWM4 = velmin;
ESC1.writeMicroseconds(PWM1);
ESC2.writeMicroseconds(PWM2);
ESC3.writeMicroseconds(PWM3);
ESC4.writeMicroseconds(PWM4);
}

void loop(){
while(1){
acelerometro(); //Função que inicializa o sensor acelerometro
giroscopio(); //Função que inicializa o sensor giroscópio
//motortest(); //Função de testes em rampa
controle(); //Função do controle PID
//print_errox();//Envia os valores do erro no eixo para serial monitor
//print_esc();//Envia os valores utilizados nos ESCs para serial monitor
print_acel(); //Envia os valores do acelerometro para serial monitor
}
}
```