



**MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
INSTITUTO MILITAR DE ENGENHARIA
(Real Academia de Artilharia, Fortificação e Desenho, 1792)
SEÇÃO DE ENGENHARIA ELÉTRICA**

**EDUARDO HENRIQUE WIESER
GUILHERME ANGELO LEITE
VICTOR LEAL VITOR**

**Desenvolvimento de um Quadricóptero para Monitoração
Remota: Montagem, Programação e Sensoriamento**

**RIO DE JANEIRO
2014**

INSTITUTO MILITAR DE ENGENHARIA

**EDUARDO HENRIQUE WIESER
GUILHERME ANGELO LEITE
VICTOR LEAL VITOR**

**DESENVOLVIMENTO DE UM QUADRICÓPTERO PARA
MONITORAÇÃO REMOTA: MONTAGEM, PROGRAMAÇÃO
E SENSORIAMENTO**

Monografia apresentada ao curso de graduação em Engenharia Eletrônica para o Projeto de Fim de Curso.

Orientador: Antonio Eduardo Carrilho da Cunha, Dr.
Eng.

Rio de Janeiro
2014

c2014

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmар ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

Leite, G.A. Vitor, V.L. Wieser, E.H.

Desenvolvimento de um Quadricóptero para Monitoração Remota: Montagem, Programação e Sensoriamento.

– Rio de Janeiro: Instituto Militar de Engenharia, 2014.
46p.: il., tab.

Monografia (Iniciação à Pesquisa) – Instituto Militar de Engenharia – Rio de Janeiro, 2014.

1. Quadricóptero; 2. Controle. I. Título II. Instituto Militar de Engenharia.

CDD 629.892

INSTITUTO MILITAR DE ENGENHARIA

Desenvolvimento de um Quadricóptero para Monitoração Remota: Montagem, Programação e Sensoriamento

Relatório apresentado ao curso de graduação em Engenharia Eletrônica para a conclusão da disciplina de Projeto de Fim de Curso.

Orientador: Antonio Eduardo **Carrilho** da Cunha, Dr. Eng.

Aprovado em 04 de Abril de 2014 pela seguinte Banca Examinadora :

Marcus Vinícius dos Santos Fernandes, Dr. Eng.

Antonio Eduardo **Carrilho** da Cunha, Dr. Eng.

Alberto Mota **Simões**, Dr. ISAE

Rio de Janeiro
2014

RESUMO

Este projeto enquadra-se no desenvolvimento de um quadricóptero para monitoração remota por câmera de vídeo, com as seguintes especificações: que possa ser pilotado remotamente, que realize gravação de vídeo e que possua hardware disponível para o controle automático de atitude. Especificamente este projeto destina-se ao dimensionamento e seleção de peças e componentes; a montagem do hardware e dos sistemas de sensores e atuadores; e a programação do sistema embarcado.

ABSTRACT

This project fits into the development of a quadcopter for remote surveillance by video camera, with the following specifications: that can be piloted remotely, conduct video recording and have hardware available for the automatic control of attitude. Specifically this project aims to design and selection of parts and components; mounting hardware and systems of sensors and actuators; and embedded system programming.

SUMÁRIO

| | |
|---|-----------|
| LISTA DE ILUSTRAÇÕES | vii |
| 1 INTRODUÇÃO | 1 |
| 1.1 Contextualização | 1 |
| 1.2 Motivação | 2 |
| 1.3 Objetivos | 3 |
| 1.4 Metodologia | 3 |
| 1.5 Estrutura da monografia | 3 |
| 2 PROPOSTA DE PROJETO | 5 |
| 3 ESCOLHA E DIMENSIONAMENTO DE PEÇAS | 7 |
| 3.1 Hélices | 7 |
| 3.2 Motores | 8 |
| 3.2.1 Motores com escovas | 8 |
| 3.2.2 Motores <i>brushless</i> | 9 |
| 3.3 Dimensionamento do par motor-hélice | 10 |
| 3.4 ESC | 11 |
| 3.5 Frame | 12 |
| 3.6 Sensores | 13 |
| 3.7 Placa Controle | 14 |
| 3.8 Bateria | 15 |
| 3.9 Peças Adquiridas | 16 |
| 4 CONDICIONAMENTO DAS PEÇAS | 17 |
| 4.1 Hélices | 17 |
| 4.2 Motor | 19 |
| 4.3 ESC | 20 |
| 4.4 <i>Frame</i> | 21 |

| | | |
|----------|--|-----------|
| 4.5 | Controlador | 22 |
| 5 | MODELAGEM E CONTROLE | 23 |
| 5.1 | Modelo Físico | 23 |
| 5.2 | Algoritmo de Controle | 25 |
| 5.3 | Controlador PID | 27 |
| 5.3.1 | Análise da malha PID para <i>ROLL</i> e <i>PITCH</i> | 28 |
| 5.3.2 | Análise da malha PID para <i>YAW</i> | 30 |
| 6 | IMPLEMENTAÇÃO DO CONTROLE | 34 |
| 6.1 | Programação da Placa | 34 |
| 6.2 | Programa de Comunicação | 35 |
| 6.3 | Programa de interface com os sensores | 35 |
| 6.4 | Programa de Controle | 36 |
| 6.5 | Programa de interface com os motores | 37 |
| 7 | TESTES | 42 |
| 7.1 | Testes Simulados | 42 |
| 7.2 | Testes Práticos | 43 |
| 8 | CONCLUSÕES | 44 |
| 8.1 | Pontos abordados e contribuições | 44 |
| 8.2 | Limitações | 44 |
| 8.3 | Perspectivas de trabalhos futuros | 45 |
| 9 | REFERÊNCIAS | 46 |

LISTA DE ILUSTRAÇÕES

| | | |
|---------|--|----|
| FIG.2.1 | Layout do projeto | 6 |
| FIG.3.1 | Fases do motor CC <i>brushed</i> | 8 |
| FIG.3.2 | Esquemático do motor CC <i>brushless</i> | 9 |
| FIG.3.3 | <i>Electronic Speed Controller</i> - AE-25A da Turnigy | 12 |
| FIG.3.4 | Frame da Xfly | 13 |
| FIG.3.5 | Placa controladora MultiWii | 14 |
| FIG.3.6 | Bateria adquirida | 15 |
| FIG.4.1 | Balanceador de hélices | 18 |
| FIG.4.2 | Exemplo do balanceador de motores utilizado | 20 |
| FIG.5.1 | Modelo X e numeração dos motores | 24 |
| FIG.5.2 | Diagrama de blocos de um controlador PID básico (Fonte: <i>ClassicalPIDControl</i>) | 27 |
| FIG.5.3 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos | 29 |
| FIG.5.4 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos, componentes pro- porcional, derivativa e integral | 30 |
| FIG.5.5 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos, malha de ajuste dos sinais proporcional e integral | 30 |
| FIG.5.6 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos, malha de com- posição de ajuste dos sinais proporcional e integral | 32 |
| FIG.5.7 | Representação do algoritmo de controle PID para <i>YAW</i> em diagrama de blocos | 33 |
| FIG.6.1 | Representação do algoritmo de controle PID para <i>ROLL</i> | |

| | | |
|---------|--|----|
| | e <i>PITCH</i> em diagrama de blocos | 38 |
| FIG.6.2 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos, componentes pro- porcional, derivativa e integral | 38 |
| FIG.6.3 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos, malha de ajuste dos sinais proporcional e integral | 39 |
| FIG.6.4 | Representação do algoritmo de controle PID para <i>ROLL</i> e <i>PITCH</i> em diagrama de blocos, malha de com- posição de ajuste dos sinais proporcional e integral | 39 |
| FIG.6.5 | Representação do algoritmo de controle PID para <i>YAW</i> em diagrama de blocos | 40 |
| FIG.6.6 | Numeração das hélices do quadricóptero no programa | 40 |
| FIG.7.1 | Configuração que foram realizados os testes | 43 |

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Desde os primórdios a humanidade busca formas de gerar um maior conforto e comodidade para as pessoas. Com este caráter é mais que natural as tentativas de desenvolver dispositivos que façam o trabalho dos homens. Com isso em mente foi desenvolvido o estudo da robótica, circuitos mecatrônicos controlados.

A primeira aplicação da robótica foi na indústria, tendo como o pai da robótica industrial George Devol⁽¹⁾, com a invenção do primeiro robô industrial⁽²⁾, um braço mecânico chamado Unimate .

As aplicações industriais da robótica são as mais visadas até hoje, pois permite a substituição da mão-de-obra humana, o que permite aumentar o lucro e produtividade das empresas. Sendo assim, surgiu uma nova área de estudos mais focados para esta vertente, a mecatrônica.

Saindo do escopo industrial, a robótica é uma solução explorada para substituir humanos em situações em que sua vida estariam em risco. O maior exemplo disso e que está sendo empregado neste trabalho é o desenvolvimento de UAV (*unmanned aerial vehicle*) ou VANT (veículo aéreo não tripulado).

Os UAV foram idealizados para fins militares e emprego na guerra. Imaginava-se poder poupar vida dos pilotos controlando-se remotamente as aeronaves. As primeiras tentativas datam de 1916 com o *Aerial Target* de Archibald Montgomery Low. Porém, o desenvolvimento desse tipo de aeronave se deu mesmo durante o curso da 1ª e 2ª guerra

mundial, em que a corrida por tecnologia dos exércitos permitiu um melhor desenvolvimento.

A história dos VANT no Brasil começou com o BQM1BR⁽³⁾, cujo protótipo produzido pela CBT(Companhia Brasileira de Tratores), levantou voo em 1983. Atualmente existem vários VANT produzidos pelo Brasil, inclusive um deles desenvolvido no IME (Instituto Militar de Engenharia).

Porém, existem missões nas quais são necessárias melhor habilidade de manobra, assim como rápida decolagem. Por isso, vem aumentando as pesquisas sobre quadricópteros, ou helicópteros quadrirrotores. A história desses veículos remonta a 1920⁽⁴⁾, quando Etienne Oehmichen dentre várias tentativas de projeto para um helicóptero, propôs um com quatro hélices.

Por realizar missões que nenhuma outra aeronave consegue, a pesquisa no campo de quadricópteros é grande, principalmente na área militar. Porém, muitas empresas comercializam quadricópteros principalmente para filmagens ou simplesmente como um *hobby*. Pode-se citar como exemplo o Parrot *AR Drone*⁽⁵⁾, que pode ser controlado via tablet ou smartphones e o *Aero-Quad*⁽⁶⁾, projeto código-aberto de hardware e software baseado em *Arduino*[®] para construção de quadricópteros.

1.2 MOTIVAÇÃO

Com base em monografias, artigos e diversos sites sobre o assunto, observamos diversas aplicações para um quadricóptero. Entre elas, a capacidade de monitorar remotamente uma região graças a grande estabilidade proporcionada pelos quatro rotores. Esta habilidade seria de grande valia para missões de reconhecimento em terrenos fechados, com muitos obstáculos e de grande risco. Além disso, o projeto de um quadricóptero abrange

diversas áreas de importância nos dias de hoje como controle, microcontroladores, automação e monitoramento remoto. Portanto, o grande potencial do quadricóptero e suas contribuições para o avanço tecnológico do país foram os grandes motivantes deste projeto.

1.3 OBJETIVOS

O projeto tem por finalidade construir um helicóptero que se utiliza de quatro motores para se locomover de maneira estável. Este controle será realizado por componentes eletrônicos programados com um algoritmo desenvolvido pelo grupo. Além disso, busca-se o aprimoramento acadêmico dos alunos nas áreas do conhecimento envolvidas, controle, sensoramento, modelagem e programação de microcontroladores.

1.4 METODOLOGIA

A partir de pesquisas sobre os projetos existentes para a montagem de um quadricóptero, determinamos os componentes essenciais para a construção deste. Providenciou-se, então, os materiais escolhidos com financiamento do Instituto Militar de Engenharia (IME). Com posse dos componentes, realizaram-se nestes diversos testes e ajustes antes de se montar a aeronave. Em paralelo a isso, estudamos como modelar e controlar o sistema físico que representa o quadricóptero. Por fim, integramos os algoritmos matemáticos ao protótipo por meio de um *software* que controla os motores em resposta aos valores obtidos por sensores.

1.5 ESTRUTURA DA MONOGRAFIA

Este trabalho está estruturado em oito capítulos.

O primeiro capítulo se refere a introdução dos conteúdos a serem abordados nesta monografia.

O segundo capítulo resume a proposta da monografia

O terceiro apresenta os componentes essenciais para a fabricação de um quadricóptero.

No quarto capítulo consta da descrição da montagem da aeronave.

O quinto capítulo discorre como foi desenvolvido o algoritmo de controle do quadricóptero.

O sexto capítulo apresenta a aplicação desenvolvida durante o trabalho detalhando o *software* feito.

O sétimo consta os teste realizados para a montagem e regulagem do quadricóptero

O oitavo capítulo apresenta as conclusões e sugestões de trabalhos futuros.

2 PROPOSTA DE PROJETO

O projeto objetivado consiste na montagem de um veículo aéreo quadricóptero não tripulado (Quadricóptero) cujo comando será feito indiretamente por um computador acoplado a um controle remoto de um Xbox 360.

O quadricóptero irá conter os seguintes itens: um sistema de sensoria-mento, uma bateria para sua autonomia, uma placa controladora, quatro atuadores (Sua forma de propulsão, no caso são quatro motores e hélices) e um sistema de comunicação com o computador que irá comandá-lo.

Deve-se mencionar que ainda existe grande espaço para adições ao projeto como um sistema de localização por GPS, inclusão de câmeras de vídeo para realização de monitoramentos à distância, dentre outros.

Um layout geral do projeto pode ser visto na FIG2.1

Explicando de forma simples, o controle do quadricóptero será feito da seguinte forma: os comandos de movimento serão feitos no controle remoto de Xbox 360, interpretados pelo programa no computador e enviados pelo sistema de comunicação de dados até o veículo. A placa controladora irá executar os comandos de movimento manipulando os atuadores em paralelo com um sistema de controle de estabilidade que constantemente mantém o veículo no ar e com o mínimo de perturbação possível.

O algoritmo e programa usados para a realização do controle e da esta-bilização, bem como cada parte do quadricóptero representada na FIG2.1 será tratada nesse trabalho com exceção da parte de comunicação (Realizada pelos XBees vistos na FIG2.1) que será desenvolvida por um outro projeto

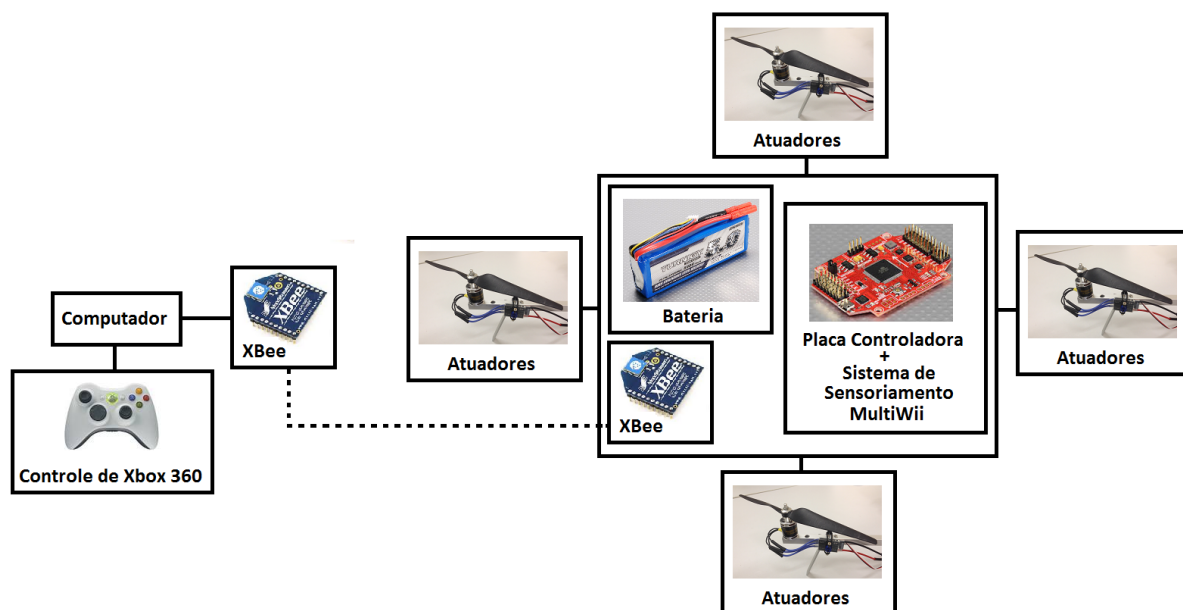


FIG. 2.1: Layout do projeto

paralelo dedicado a parte de comunicação, monitoramento e transferência de dados juntamente com a inclusão de um sistema transmissão de vídeo.

3 ESCOLHA E DIMENSIONAMENTO DE PEÇAS

A escolha do tipo de motor e o dimensionamento da hélice são os fatores de maior importância na montagem de aeronaves. Primordialmente, o conjunto motor-hélice é responsável por garantir empuxo suficiente para que a aeronave se mantenha no ar. Entretanto, para o melhor controle do veículo, são necessárias pequenas variações no empuxo. Porém, o zero lógico está associado ao zero de empuxo e o maior valor lógico ao empuxo máximo, o que implica que a distância entre cada nível lógico depende exclusivamente da escolha da potência do motor. Por isso, o conjunto deve ser escolhido tendo em mente o nível de controle, o peso da aeronave, a carga a ser transportada e a velocidade máxima a ser alcançada. Neste projeto, buscamos montar um quadricóptero para patrulhamento, atividade que exige muita estabilidade, baixas velocidades e cargas leves.

3.1 HÉLICES

As hélices são as peças mais importantes de um aeromodelo. As pás destas peças possuem uma pequena angulação para poder forçar o fluxo de ar ao girar. Outro fator importante em seu funcionamento é a área da pá e seu comprimento. Quanto maior a área, maior a massa de ar deslocada numa mesma velocidade e quanto mais longa mais a hélice é indiferente perante correntes externas de ar. Portanto, para projetos de baixa velocidade e grande estabilidade, normalmente se utiliza hélices longas e finas.

Entretanto, para se utilizar todo o potencial de uma hélice, é necessário que o motor forneça um determinado torque e atinja certas velocidades de rotação. Portanto, antes de dimensionar a hélice, é necessário ter em mente

qual o peso da aeronave, qual o torque do motor e a velocidade máxima desejada.

3.2 MOTORES

Os motores são compostos por duas partes: o rotor e o estator. O estator é composto por um ímã permanente enquanto o rotor possui bobinas energizadas de forma a estarem sempre com cargas magnéticas opostas ao estator. Com isso, as cargas do rotor repelem as do estator gerando movimento. A partir desse princípio, foram desenvolvidos vários modelos de motores elétricos. Destes, dois tipos de motores de corrente contínua (motores CC) são muito utilizados em projetos de modelismo, os *brushless* (sem escova) e os *brushed* (com contato por escovas).

3.2.1 MOTORES COM ESCOVAS

Os motores com escovas são montados de forma a manter o estator fixo e movimentar o rotor. Como a corrente no rotor precisa mudar de sentido, esse modelo se utiliza do giro do rotor para realizar esta alternância enquanto mantém o contato elétrico por meio de escovas de carbono. O funcionamento e as fases do motor com escovas podem ser visualizados na FIG 3.1.

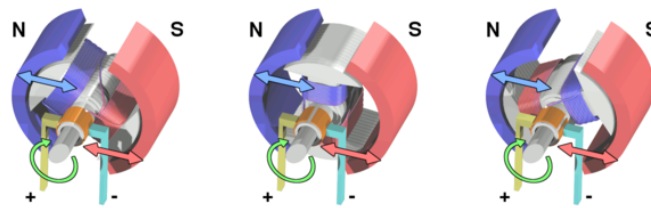


FIG. 3.1: Fases do motor CC *brushed*

Por ser muito simples, é um motor barato e fácil de utilizar, tornando-o muito popular em pequenos projetos. Entretanto, suas escovas o tornam

um motor pouco eficiente e pouco durável. Além disso, evita-se o uso desses motores em projetos de aeromodelismo, pois o contato da escova com o rotor gera vibrações, dificultando o controle da aeronave.

3.2.2 MOTORES *BRUSHLESS*

Os motores *brushless*, ao contrário dos com escovas, movimentam o rotor e fixam o estator, por isso não necessitam de escovas para o contato. Contudo, a alternância de corrente no rotor deve ser feita externamente ao motor. A figura FIG 3.2 mostra um esquema simplificado de um motor *brushless* de três fios.

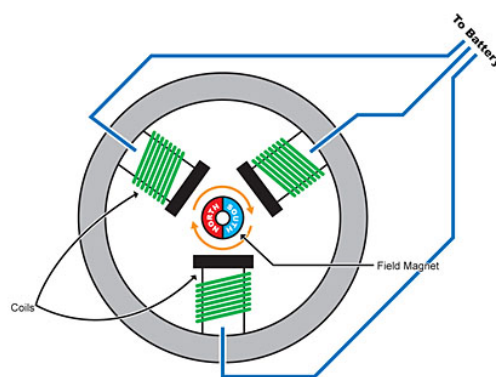


FIG. 3.2: Esquemático do motor CC *brushless*

Como seu funcionamento é similar ao de um motor de corrente alternada, costuma-se utilizar nos motores *brushless* um componente intermediário, o ESC, *Electronic Speed Controller*, para realizar a alternância da corrente. Apesar de mais complexos, como não utilizam escovas, esses motores não perdem energia por atrito, logo são muito mais eficientes e duráveis que os *brushed*. Por isso, este tipo de motor vem sendo cada vez mais utilizado, principalmente em projetos de aeromodelismo.

3.3 DIMENSIONAMENTO DO PAR MOTOR-HÉLICE

Os motores CC têm sua velocidade de rotação proporcional à corrente e seu torque proporcional à tensão de entrada. Ao serem fabricados, esses motores são projetados para obter um determinado torque sobre uma tensão fixa. Portanto, a relação torque por voltagem determina a potência máxima do motor.

Para melhor aproximar a potência ideal do motor, seriam necessários diversos testes práticos. Porém, pode-se utilizar a TAB 3.1⁽⁹⁾ para obter uma relação aproximada da potência do motor com o peso da aeronave. Estimando o peso da nossa aeronave em 1100g, calcula-se uma potência de aproximadamente 180 W. Escolhemos, então, trabalhar com motores de 11,1V e de até 20A

| Tipo de Voo | Relação Peso x Potência (W/kg) |
|--------------------------------|--------------------------------|
| alar com baixa carga | 110 a 154 |
| Voo Lento | 154 a 198 |
| Acrobático de Voo Rápido | 198 a 243 |
| Acrobáticos de Alta Velocidade | 243 a 287 |
| 3D c/ baixa carga alar | 287 a 331 |
| 3D de desempenho ilimitado | 331 a 441 |

TAB. 3.1: Tabela com relação potência-peso do site Aeromodelismo Online⁽⁹⁾

Entretanto, apenas a escolha do motor não garante que o empuxo gerado pelo conjunto motor-hélice será o desejado. Variando-se as dimensões da hélice, altera-se não só o empuxo gerado como a estabilidade do quadricóptero. Porém, como a relação precisa do empuxo com as dimensões da hélice são empíricas, normalmente utilizam-se tabelas para determinar o par motor-hélice desejado.

Tendo o peso aproximado da aeronave projetada (1100g), optou-se por buscar um par motor-hélice com empuxo próximo ao do peso. Com isto, busca-se uma boa relação controle-velocidade como a apresentada na monografia *quadricóptero*⁽⁸⁾. O par utilizado foi escolhido com base na tabela disponibilizada pelo fornecedor do motor⁽⁷⁾ (TAB 3.2) com as estimativas de empuxo para certas hélices fornecidas pelo mesmo.

| | Dados de Desempenho | | |
|--------|---------------------|--------|--------|
| Tensão | Corrente | hélice | Empuxo |
| 11.1V | 12.3A | 10x60 | 800g |
| 11.1V | 16.3A | 10x70 | 970g |
| 11.1V | 17.9A | 11x47 | 1040g |

TAB. 3.2: Tabela com relação corrente, hélice, empuxo do site Hobby King⁽⁷⁾

3.4 ESC

O ESC, *Eletronic Speed Controller*, controla a velocidade do motor CC a partir de uma modulação PWM da corrente. Com um microcontrolador e chaves eletrônicas (transistores), o ESC liga e desliga a corrente em intervalos regulares para alterar controladamente a sua média. Além disso, o ESC é responsável por manter constante a tensão correta do motor, por isso o ESC deve ser corretamente configurado para o motor escolhido.

Outro ponto importante é o tipo de motor a ser controlado. Como os motores *brushed* e *brushless* funcionam de maneiras diferentes, necessitam de controladores específicos. Portanto, cabe aos ESC's de motores *brushless* coordenar e sincronizar as correntes nos terminais do motor.

Com base em outros projetos e em dados obtidos com o fornecedor, optou-se por adquirir o ESC “Turnigy AE-25A Brushless ESC”(FIG 3.3), o qual é leve, compatível com o motor previamente escolhido e possui modos de

operação específicos para helicópteros (Transições suaves de velocidade e partida controlada do motor).

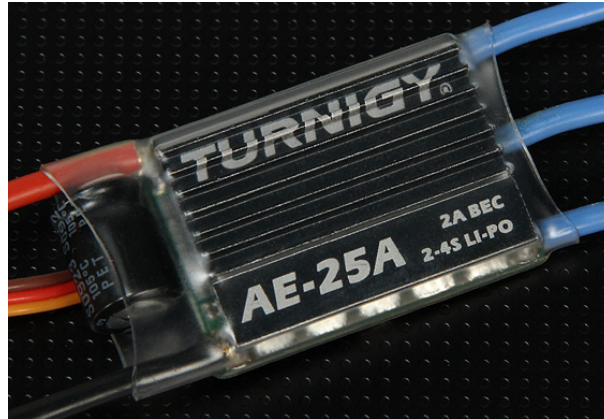


FIG. 3.3: *Eletronic Speed Controller* - AE-25A da Turnigy

3.5 FRAME

O frame de um quadricóptero é responsável por servir de base para a fixação dos demais componentes. Ela é uma das peças mais pesadas do conjunto, juntamente com a bateria. Por isso, é importante que o frame seja de um material resistente, porém leve. Outro aspecto relevante é a vibração.

Um frame mal balanceado irá gerar vibração, o que prejudica consideravelmente o controle da aeronave. Além disso, a má fixação de uma peça também gera distúrbios no controle. Portanto, é de extrema importância para um projeto que o frame esteja bem balanceado e com as peças fixadas de maneira estável.

Outro fator importante é a forma do frame. Existem pequenas variações no formato dos frames, o que leva a pequenas alterações na dinâmica do sistema físico. Dentre os aspectos físicos mais relevantes, está a distância entre os motores. Quanto maior esta distância, menor o efeito do fluxo de ar gerado

por uma hélice sobre a outra, contribuindo para uma maior estabilidade do quadricóptero.

Para o projeto, escolhemos o “FRAME QUADRICÓPTERO SEM PROTEÇÕES” (FIG 3.4) da Xfly com distância entre eixos dos motores de 550mm e que pesa 340g.



FIG. 3.4: Frame da Xfly

Este frame foi escolhido por ser um produto nacional e barato. Outra vantagem que influenciou na escolha deste produto foi que o frame já vem com toda a furação para a fixação dos motores, da bateria e da placa controladora, facilitando a montagem e o balanceamento das peças.

3.6 SENSORES

Os sensores são os componentes que permitem ao controlador reconhecer o estado atual do ambiente. Com esses dados, é possível realimentar os sistemas de controle e garantir a estabilidade de um processo. No projeto de uma aeronave, alguns sensores são essenciais: o girômetro, que indica a inclinação, e o acelerômetro, que fornece um referencial vertical. Outros sensores importantes para a navegação aérea são o barômetro, responsável por confirmar a altitude, e o magnetômetro, que fornece um referencial horizontal.

3.7 PLACA CONTROLE

A placa controladora é responsável por processar os dados coletados pelos sensores e com estes definir a nova velocidade dos motores. Portanto, é de suma importância que a placa seja compatível com todos os sensores do projeto, além de ser capaz de enviar informação aos motores.

Normalmente, a comunicação com os motores é feita através de modulação PWM. Por isso, o microcontrolador escolhido deve possuir ao menos um terminal, para cada motor, que possibilite esta modulação.

Para a placa controladora, escolheu-se importar a “MultiWii PRO Flight Controller with MTK GPS Module” (FIG 3.5) de 70x50x12mm que pesa 16g.

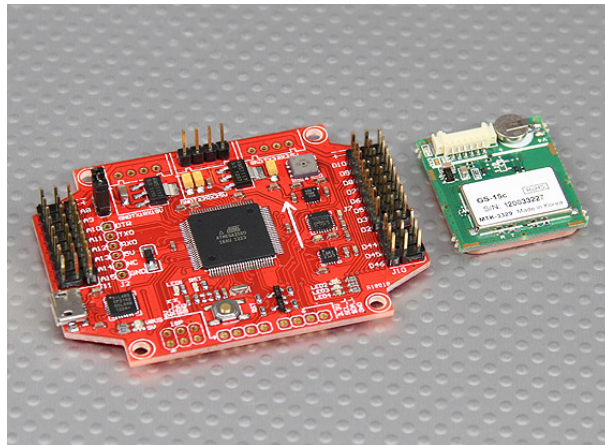


FIG. 3.5: Placa controladora MultiWii

A placa MultiWii já vem integrada com diversos sensores: girômetro de três eixos, acelerômetro, barômetro e magnetômetro. Além disso, inclui portas para controle de servos motores, placa GPS e USB para programação “*On-board*”. Portanto, a razão da escolha desta placa no projeto foi o fato de todos os componentes já virem integrados, reduzindo o peso e possíveis problemas de compatibilidade entre os sensores e a placa.

3.8 BATERIA

A fonte de energia do quadricóptero, assim como os demais componentes, devem ser leves e eficientes. Dentro desses requisitos, as baterias de íon de Lítio, ou LiPo, vêm se destacando. Além de leves, as baterias LiPo são recarregáveis e capazes de armazenar muita energia.

Para o cálculo da bateria, existem duas características relevantes: a carga de uma célula, usualmente medida em mAh, e o número de células que ela possui. Com o produto desses dados, estima-se a corrente máxima fornecida pela bateria. Como utilizamos quatro motores de 20A, precisamos de uma bateria de ao menos 4000mAh com 20C (20 células). Neste projeto escolheu-se trabalhar com a “Turnigy 5000mAh 3S 20C Lipo Pack” (FIG 3.6)



FIG. 3.6: Bateria adquirida

3.9 PEÇAS ADQUIRIDAS

Os componentes escolhidos foram dispostos na tabela TAB 3.3.

| Tipo de Peça | Especificação | Quantidade | Peso por peça (g) |
|----------------|---|------------|-------------------|
| Frame | Frame sem proteção da Xfly | 1 | 340 |
| ESC | Turnigy AE-25A Brushless ESC | 4 | 25 |
| Motor | Turnigy 2217 20turn 860kv 22A Outrunner | 4 | 72 |
| Bateria | Turnigy 5000mAh 3S 20C Lipo Pack | 1 | 312 |
| Placa Controle | MultiWii PRO Flight Controller | 1 | 16 |
| | | Peso total | 981 |

TAB. 3.3: Tabela com os componetes comprados e seus respectivos pesos

4 CONDICIONAMENTO DAS PEÇAS

Mesmo comprando peças feitas para aeromodelismo, alguns ajustes devem ser feitos antes do primeiro voo. Entre esses ajustes estão o balanceamento das peças e a configuração dos componentes eletrônicos.

A vibração é um dos fatores que mais atrapalham o desempenho de um projeto de controle. Para tentar reduzir ao máximo esse fator, deve-se certificar que todas as peças têm seu centro de massa no ponto esperado, ou seja, estejam balanceados.

Além disso, os componentes eletrônicos costumam ter configurações específicas para cada tipo de aeromodelo, visando otimizar o desempenho do sistema. Portanto, mesmo que não seja essencial para todos os projetos, recomenda-se ajustar todos os eletrônicos para o modo ideal de operação. Com isso, pode-se economizar energia, reduzir as vibrações e simplificar o modelo de controle.

4.1 HÉLICES

Normalmente, hélices de menor qualidade não vêm com o centro de massa exatamente no seu centro. Esse desbalanceio força a hélice a rotacionar em um ponto diferente do eixo do motor, gerando vibrações irregulares e indesejadas. Para solucionar isso, utilizam-se técnicas e equipamentos específicos para o balanceamento de hélices.

Nesse projeto, balanceamos as hélices com auxílio do suporte da FIG 4.1. Neste equipamento, fixa-se a hélice a uma haste com liberdade para rotacionar. Caso o centro de massa não se encontre exatamente alinhado com

a haste, a hélice irá rodar em busca de um ponto de maior estabilidade. Ou seja, a hélice só será considerada balanceada se esta permanecer imóvel, independente da posição de suas pás.



FIG. 4.1: Balanceador de hélices

Para corrigir a irregularidade do centro de massa da hélice, deve-se lixar suas pás. Entretanto, se a pá for lixada de maneira errônea, altera-se sua aerodinâmica de maneira irregular aumentando as vibrações. Por isso, deve-se utilizar uma lixa fina no lado da hélice que não entrará em contato com o vento (parte debaixo no caso de helicópteros).

Técnicas mais avançadas de balanceamento de hélices, além de lixar, utilizam contrapesos para melhorar ainda mais o balanceamento. Contudo, pela inexperiência dos operadores com estas técnicas, optou-se por não utilizar contra pesos no eixo da hélice.

4.2 MOTOR

Assim como as hélices, motores mais baratos não são adquiridos balanceados. Como seu centro de massa não está alinhado com o eixo do motor, ao girar, o motor tende a vibrar. Essas vibrações podem forçar os parafusos de fixação do motor, o encaixe da hélice e, em alguns casos, até entortar o eixo. Portanto, não deve ser ignorado o balanceamento dos motores, mesmo em projetos que não exijam grande controle.

Para corrigir o centro de massa, normalmente utilizam-se pedaços de fita de vinil. Além de ter um peso relevante, porém não muito elevado, é um material de fácil obtenção e fixação. Outra opção são fitas de chumbo e/ou colas epóxi.

Existem diversas técnicas para se balancear um motor, mas a maioria delas exige equipamentos eletrônicos sofisticados. Contudo, optou-se nesse projeto por uma técnica sem eletrônicos, o que a torna menos precisa. Porém, mais econômica e simples.

Nesta técnica, prende-se o motor ao extremo de um suporte levemente flexível. Também, neste suporte, fixa-se um pequeno espelho levemente angulado como na FIG 4.2. Depois, direciona-se um feixe de laser ao centro do espelho, liga-se o motor e observa-se o reflexo do laser. Como o motor está desbalanceado, o suporte irá vibrar, assim como o espelho o que irá mudar levemente o ângulo de reflexão. Quando o reflexo do laser não se alterar com o giro do motor, então este estará balanceado.

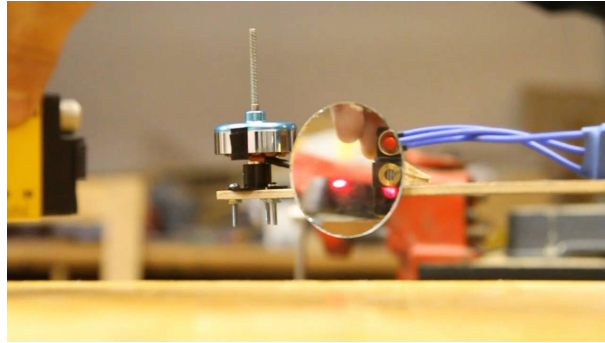


FIG. 4.2: Exemplo do balanceador de motores utilizado

4.3 ESC

O ESC, é um componente eletrônico utilizado em diversos tipos de aeromodelos, por isso, possui inúmeras configurações. Analisando especificamente o ESC adquirido para esse projeto, deve-se configurar cinco parâmetros: freio, *timing*, *cutoff threshold*, modo de inicialização e velocidade de resposta.

O parâmetro freio indica se o ESC deverá parar as hélices ou deixá-las rodar livremente quando recebido um sinal de zero. Esse modo foi projetado principalmente para aviões de alta velocidade, os quais poderiam danificar suas hélices se essas acidentalmente acertassem o chão durante o pouso. Portanto, esse parâmetro foi deixado no seu padrão, desabilitado.

Outro parâmetro importante é o *timing*. Esse parâmetro define o tempo de inicialização dos motores. Normalmente, utiliza-se um tempo curto ou médio para motores de baixa velocidade e um tempo longo para motores de alta velocidade. Segundo o fabricante do ESC, o uso do tempo curto em motores de 2 polos e do tempo médio em motores de 6 polos são mais eficientes. Por isso, escolheu-se usar o modo *tempo médio*.

O *cutoff threshold* define qual tensão de fonte a bateria é considerada como descarregada. Assim, para cada tipo de bateria, o ESC possui um valor pré-definido e transparente ao usuário que ele define como alto, médio ou baixo. No projeto, utilizou-se a configuração padrão, ao atingir a tensão média entre em modo *cutoff*.

Outra variável a se configurar é o modo de inicialização. Neste, pode-se definir a suavidade da aceleração dos motores na inicialização da aeronave. Utiliza-se para helicópteros o modo super-suave para facilitar o controle do veículo.

A última variável a se configurar é a velocidade de resposta do controlador. Para modelos acrobáticos pode-se escolher reduzir o atraso do controlador, porém isso normalmente aumenta o consumo energético. Como optou-se por um modelo de baixa velocidade e alto controle, definiu-se esse parâmetro como normal para um menor consumo de energia.

Após definir todos os modos de operação, criou-se um código C que simula o controle e configura o ESC com os parâmetros desejados. Este código inicia enviando o valor máximo e mínimo do PWM ao ESC para que esse reconheça os limites trabalhados. Em seguida, realiza uma sequência pré-definida pelo manual do ESC para reconfigurar as variáveis.

4.4 *FRAME*

Após balancear e configurar os outros componentes, pode-se, então, fixá-los no *frame*. Para isso, deve-se atentar, além da boa montagem do *frame*, a alguns detalhes como o posicionamento dos componentes, o tipo de furo e a simetria do modelo.

Antes de fixar qualquer componente, é preciso definir como cada um será preso ao *frame*. Normalmente, os motores e a placa controladora são parafusados, para uma maior estabilidade, e os demais componentes são presos por fitas, elásticos ou velcros, por não possuírem espaço destinados a parafusos. Para a parte parafusada, são necessários furos precisos e condizentes com as peças escolhidas. Portanto, costuma-se pedir para o fabricante do *frame* fazer os furos para as peças escolhidas com ferramentas de alta precisão.

Além disso, para manter o quadricóptero balanceado é preciso garantir que seu centro de gravidade está no centro do *frame*. O modo mais simples de atingir essa meta é tentar manter a aeronave simétrica e corrigir as imperfeições com o posicionamento da bateria. portanto, posiciona-se os componentes dos braços (ESC, motor e outros opcionais como leds) aproximadamente a mesma distância do centro do modelo.

Outro fator importante é a influência de um componente no outro. Como os ESCs e os motores causam interferência no sistema de comunicação, eles devem estar posicionados o mais distante o possível. A bateria, por outro lado, atinge altas temperatura, devendo, então, ser posicionada em uma placa diferente da que abriga a placa controladora.

4.5 CONTROLADOR

Apesar de escolhido por sua fácil programação, o controlador *Arduino* exige alguns *softwares* específicos para sua programação. Para se comunicar com a placa, é necessário configurar uma porta USB do computador para simular o funcionamento de uma porta serial. Outra peculiaridade do controlador escolhido é que este necessita do *software* de programação *Arduino* ou o *Arduino Uno* disponíveis gratuitamente na internet.

5 MODELAGEM E CONTROLE

O controle de um quadricóptero se resume a controlar um sistema físico de quatro entradas e quatro saídas. A partir de algumas simplificações, é possível modelar o quadricóptero em um conjunto de equações matemáticas. Com estas, um método se baseia em montar um modelo linear no entorno de um ponto de estabilidade e, então, cria-se um controlador para esse modelo. Portanto, neste caso, deve-se tomar o cuidado de não simplificar demais as equações físicas a ponto de distanciar significativamente o modelo do real.

Outro ponto importante é a realimentação. O controlador trabalha com um modelo aproximado, o que insere um erro proveniente de suas aproximações. Contudo, esse erro pode ser minimizado com a comparação do estado desejado com o estado real, através de realimentação. Na prática, isso implica no uso de sensores e estimadores de estados que captam diversas variáveis do sistema real para os inserir no programa de controle.

Além disso, os quadricópteros possuem dois modelos básicos de voo, o modelo X e o modelo +. O modelo + tem os braços do *frame* perpendiculares e um deles é considerado a frente do modelo. O modelo X, considera a bissetriz de dois braços como a frente da aeronave. Por isso, cada um utiliza um conjunto diferente de motores para se inclinar para frente, diferenciando as equações e os controladores.

5.1 MODELO FÍSICO

Neste projeto, escolheu-se trabalhar com o modelo X de quadricóptero. Desta forma, a frente da aeronave está entre os motores 1 e 2. Definiu-se,

também, que os motores pares serão os que giram no sentido anti-horário. O esquema do modelo escolhido está representado na FIG 5.1.

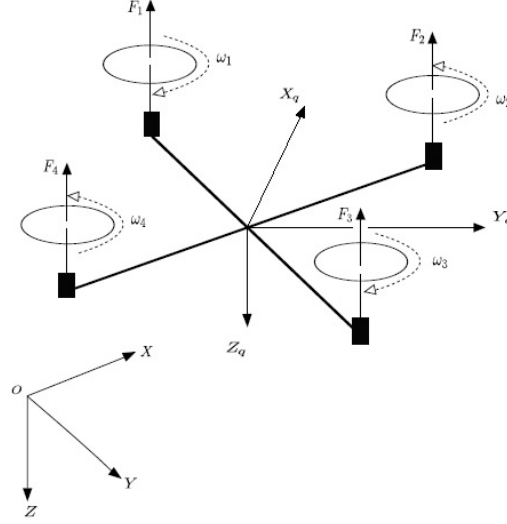


FIG. 5.1: Modelo X e numeração dos motores

A partir desse modelo, é fácil aferir as condições para os quatro movimentos básicos de uma aeronave. Aumentando a velocidade de todos os motores a altitude tende a aumentar.

Considerando F_i o empuxo de cada motor, obteve-se equações físicas similares as do artigo da CBA⁽¹⁰⁾.

$$\begin{aligned}\ddot{\phi} &= \frac{L}{J_x}(F_1 - F_2 - F_3 + F_4) \\ \ddot{\theta} &= \frac{L}{J_y}(F_1 + F_2 - F_3 - F_4) \\ \ddot{\psi} &= \frac{K}{J_z}(F_1 - F_2 + F_3 - F_4) \\ \ddot{Z} &= \frac{\cos(\phi) \cos(\theta)}{m}(F_1 + F_2 + F_3 + F_4) - g\end{aligned}$$

Onde ϕ , θ e ψ são os ângulos de rolagem, arfagem e guinada; Z é a altitude do centro de massa da aeronave; m é a massa do quadricóptero; L é o tamanho

de um braço do frame; K é a constante que relaciona os empuxos com os torques de guinada; J_x, J_y e J_z são os momentos de inércia em torno dos eixos Xq, Yq e Zq, respectivamente e g é aceleração da gravidade.

5.2 ALGORITMO DE CONTROLE

Uma vez que as relações físicas que regem a movimentação do quadricóptero projetado são compreendidas, o próximo passo é escolher um algoritmo de controle a ser implementado na placa controladora. O sistema de controle visa manter a estabilidade em voo, executando a compensação de eventuais perturbações que possam interferir no equilíbrio ou trajetória do veículo, sejam de natureza do ambiente (Vento e colisões leves) ou de próprias imperfeições inerentes a estrutura (Vibrações mecânicas, ruído elétrico e assimetria da distribuição de massa). Ao mesmo tempo o sistema deve permitir que o quadricóptero se locomova como planejado sem se desestabilizar e perder capacidade de se manter no ar.

O algoritmo de controle utilizado é um PID (*Proportional – integral – derivative*). Esse algoritmo calcula a resposta de um sistema através da entrada e uma realimentação que nesse trabalho são compostas respectivamente pelo sinal de comando dado a placa controladora e pela leitura dos sensores, algo que a maioria dos sistemas de controle já faz. Porém o que caracteriza um PID é o cálculo de três ações componentes a serem integradas na saída: a ação proporcional, a ação integral e a ação derivativa.

A componente proporcional, vista na FIG 5.2 pelo bloco de ganho K_c é dependente apenas da diferença entre o ponto de ajuste e a variável de processo, ou seja, ele representa um ajuste direto e linear entre dois valores: um proveniente de um sensor e um proveniente da entrada de comando, sendo essa diferença chamada de termo do erro do sistema. Sua aplicação pode ser feita após as composições dos outros dois termos (Derivativo e integral) ou

antes, mas obviamente sempre após o cálculo do termo do erro feito na FIG 5.2 pelo primeiro somador (Com somente duas entradas, sendo uma delas subtrativa). A magnitude do ganho proporcional determina a velocidade de resposta do sistema, isto é, quanto maior o valor de K_c , mais rápido o sistema irá responder, porém deve-se tomar cuidado pois um aumento exagerado desse mesmo termo pode provocar uma resposta oscilatória muito brusca causando instabilidade no sistema. Uma solução de compromisso é recomendada.

A componente integral do controlador PID executa uma soma cumulativa do erro com o tempo, isso embora aparente ser desvantajoso é, na verdade, uma maneira de conduzir o erro de estado estacionário do sistema para zero, pois força o controlador a corrigir qualquer desvio que poderia passar sem ser detectado, uma vez que mesmo um pequeno erro fará com que a componente integral aumente. Na FIG 5.2 o termo integral é resultado do bloco integrador em série com um bloco de ganho $1/T_d$. A fim de que a componente integral nem aumente indefinidamente e provoque uma resposta muito brusca por parte da componente proporcional que poderia desestabilizar o sistema é recomendável a introdução de um saturador no bloco de integração, para definir um limite máximo que o erro pode se acumular.

Por fim, temos a componente derivativa, na FIG 5.2 ela resulta do ramo composto por um bloco de ganho T_d em série com um derivador. Normalmente o termo derivativo tem caráter de antecipação da velocidade da resposta do sistema, ou seja, seu crescimento fará com que o sistema reaja mais agressivamente a variações nos parâmetros do erro. Infelizmente isso também, tornará o sistema extra sensível a ruído e poderá torná-lo instável. Assim, a maioria dos sistemas com termo derivativo é projetado de forma a ter seu ganho T_d baixo.

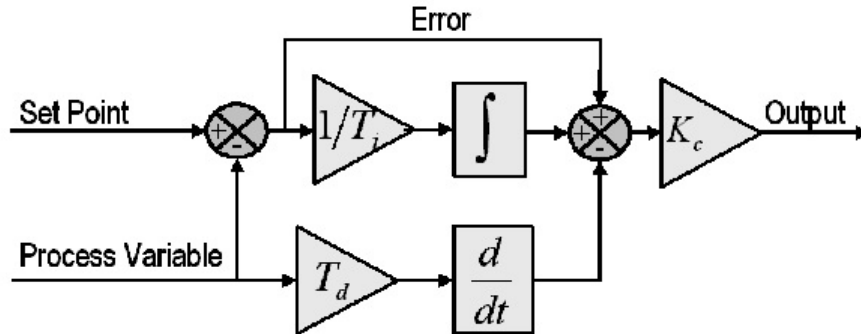


FIG. 5.2: Diagrama de blocos de um controlador PID básico (Fonte: *ClassicalPIDControl*)

5.3 CONTROLADOR PID

O sistema de controle implementado na placa MultiWii se utiliza de 2 algoritmos PID. Apesar da sua implementação ser realizada na forma de um programa cujo código se encontra em anexo a este relatório, os algoritmos utilizados foram colocados na forma de diagrama de blocos para facilitar seu entendimento e podem ser visualizados nas figuras 5.3 e 5.7.

Esses dois diagramas de blocos representam 3 sistemas que foram implementados computacionalmente no código de programação da placa de controle (MultiWii), cada sistema tem como entrada os dados de comando relativos aos movimentos a serem executados pelo quadricóptero e as leituras dos sensores periodicamente e suas saídas são sinais associados respectivamente aos ângulos de *ROLL* e *PITCH* para o sistema representado na FIG 5.3, e *YAW* para o sistema representado na figura 5.7.

Antes de analisar o comportamento de cada sistema devem ficar esclarecidos certos detalhes:

É importante ressaltar a adimensionalidade dos sinais envolvidos. Os diagramas de blocos são meras representações do funcionamento da lógica do algoritmo de controle PID utilizado e de forma alguma representam sinais elétricos reais em portas lógicas. Dessa forma, os números e ordens de grandeza neles representados são inerentes a lógica do programa.

Da mesma forma deve-se ressaltar que a saída de cada sistema é um sinal associado a um dos três ângulos de *ROLL*, *PITCH* e *YAW* e não o valor do ângulo em si. O sinal de saída é uma variável computacional adimensional que será posteriormente trabalhada pelo programa de controle e enviada na forma de um sinal PWM para os ESCs de forma a controlar os motores do quadricóptero fisicamente.

Por fim, vê-se que a malha da figura 5.3 é demasiadamente complexa e a fim de facilitar seu entendimento ela foi dividida em três malhas vistas nas figuras 5.4, 5.5 e 5.6.

5.3.1 ANÁLISE DA MALHA PID PARA *ROLL* E *PITCH*

A figura 5.3 representa a malha de controle associada aos parâmetros *PITCH* e *ROLL* do quadricóptero. Ambos utilizam o mesmo controlador PID, porém sua complexidade exigiu que a malha fosse analisada em três diferentes diagramas de blocos vistos nas figuras 5.4, 5.5 e 5.6.

Iniciando-se pela malha representada na figura 5.4, temos um sistema com uma entrada de referência e uma realimentação associada a velocidade angular. Essa malha utiliza-se de um somador para calcular a diferença desses dois termos e compor o erro. As saídas do sistema são os termos proporcional (PTerm_vel), integral (ITerm_vel) e derivativo (DTerm_vel) que são calculados respectivamente a partir do sinal de entrada, do sinal do erro e do

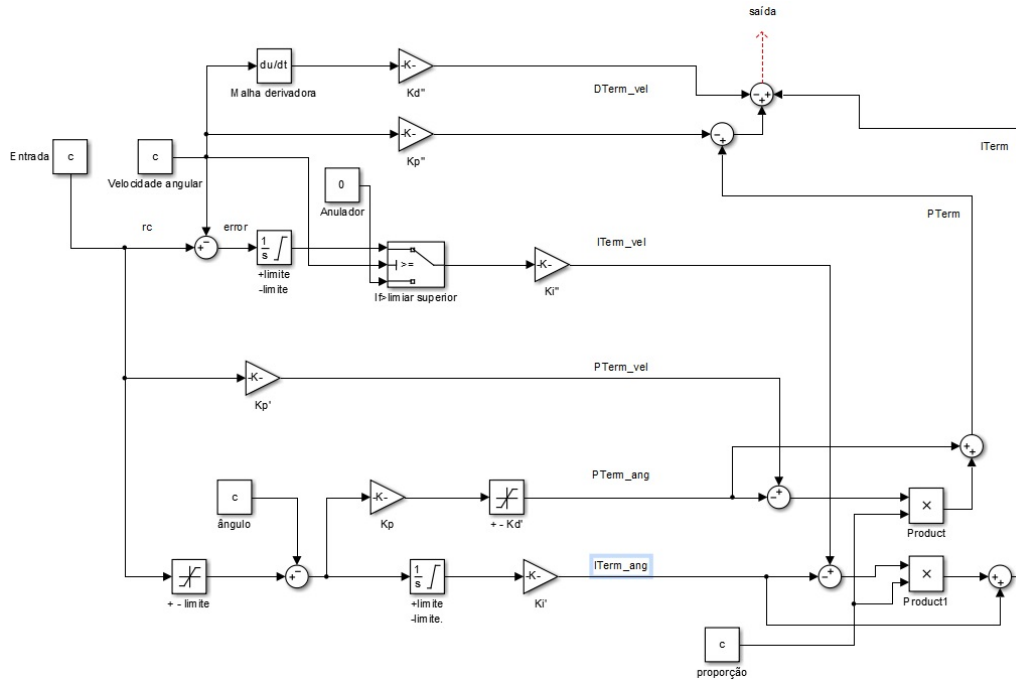


FIG. 5.3: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos

signal de realimentação. Observa-se também que o termo integral é limitado por um saturador para evitar um superacúmulo do erro com o tempo.

A malha vista pela figura 5.5 é responsável pelos ajustes mais finos do sistema e está relacionada a variações dos ângulos medidos pelos sensores. Ela só entra em operação quando a variação exigida pelo sistema é menor do que um limiar superior. Isso é garantido pelo saturador na entrada da malha e pelo anulador na malha da figura 5.4 analisada anteriormente. O erro angular é utilizado para gerar duas saídas: uma proporcional ($PTerm_ang$) e uma integral ($ITerm_ang$), sendo a integral novamente controlada por um saturador.

Por último, a malha da figura 5.6 combina as saídas das malhas anteriores em uma média ponderada através de uma entrada proporcional, utilizada

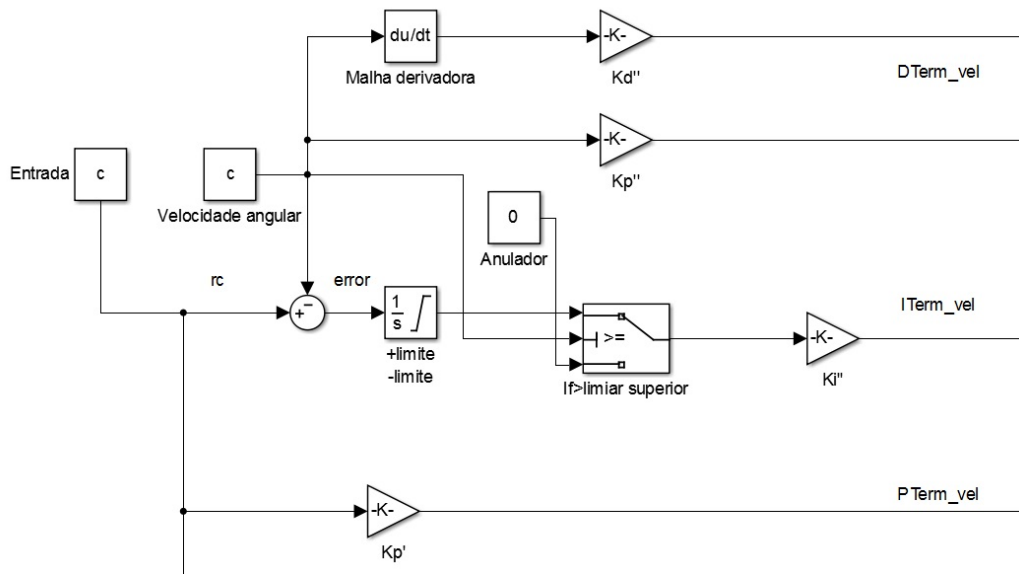


FIG. 5.4: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos, componentes proporcional, derivativa e integral

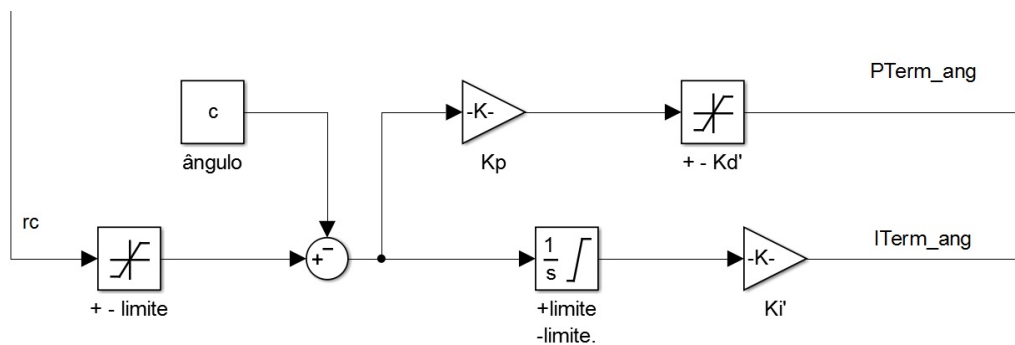


FIG. 5.5: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos, malha de ajuste dos sinais proporcional e integral

como peso, dois multiplicadores e seis somadores, sendo um de três entradas para realizar a composição final do sinal de saída.

5.3.2 ANÁLISE DA MALHA PID PARA *YAW*

A figura 5.7 representa o algoritmo de controle PID que está associado ao parâmetro *YAW*. O sistema possui uma entrada e uma realimentação tam-

bém associada a velocidade angular. O sinal de erro é calculado através da diferença entre a entrada e a realimentação e é utilizado para gerar as componente integral e proporcional da saída, sendo a integral também limitada via saturador. A componente proporcional (PTerm) e a integral (ITerm) são combinadas através de um somador e o sinal de saída é gerado.

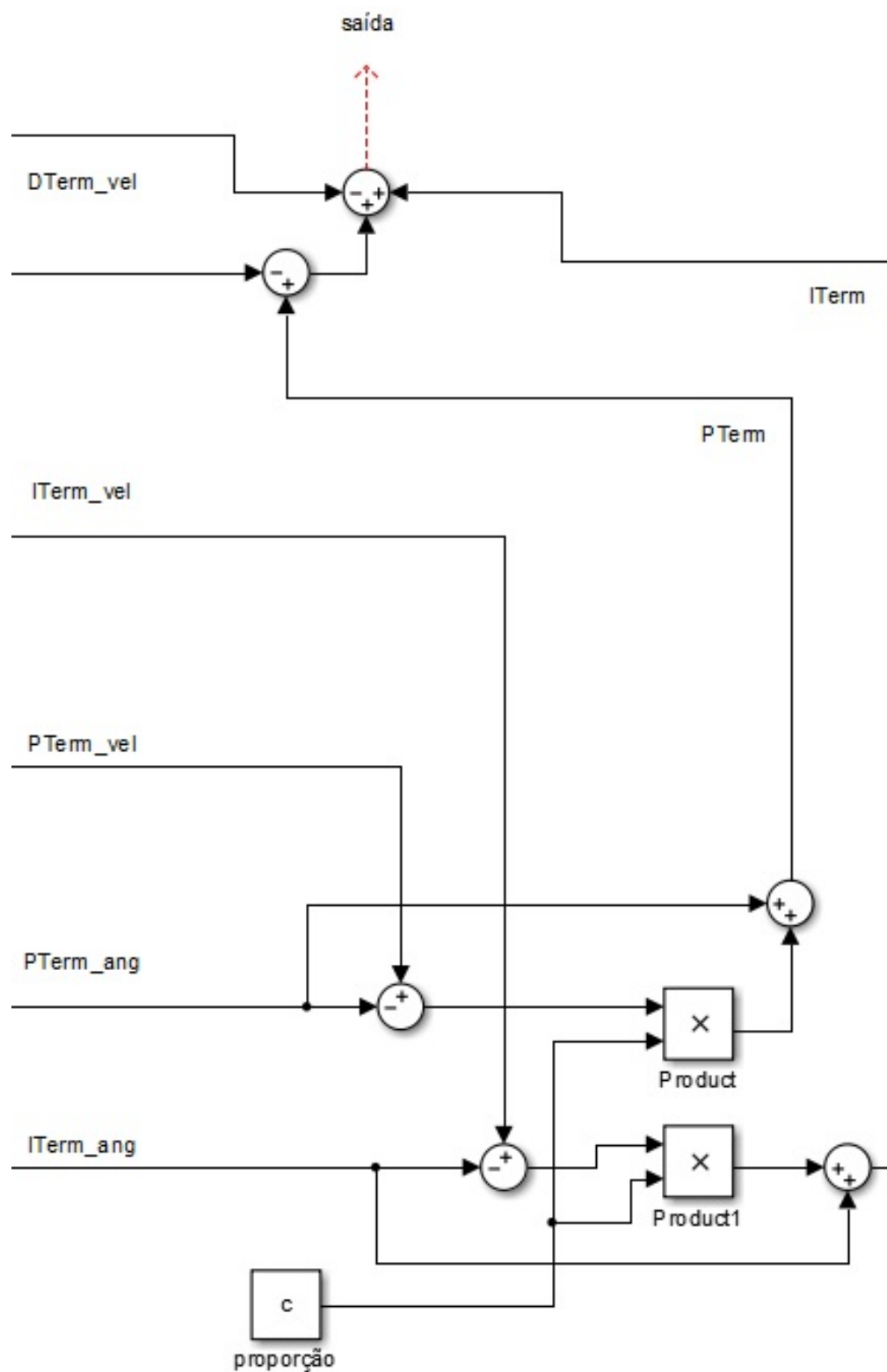


FIG. 5.6: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos, malha de composição de ajuste dos sinais proporcional e integral

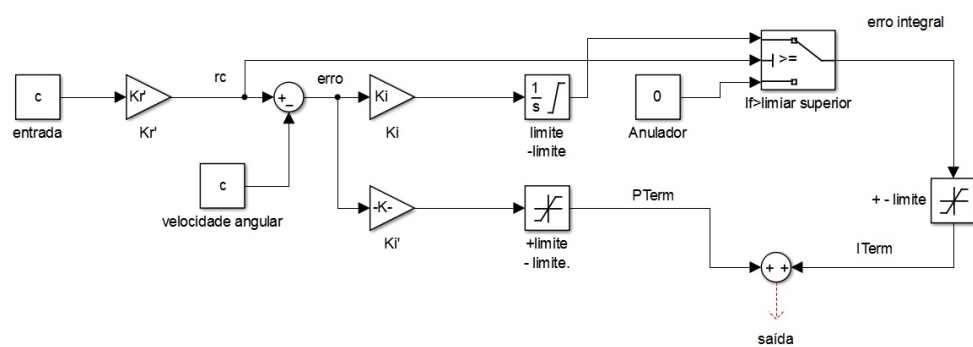


FIG. 5.7: Representação do algoritmo de controle PID para *YAW* em diagrama de blocos

6 IMPLEMENTAÇÃO DO CONTROLE

6.1 PROGRAMAÇÃO DA PLACA

Como mencionado anteriormente, a placa controladora exigiu o uso do *software* de programação *Arduino*. Este *software* compila programas na linguagem C com uma biblioteca específica, o *Ardunio.h*, orientada a microcontroladores. Esta biblioteca apenas define algumas funções de interface com a placa como configurar os pinos como entrada ou saída, ajustar *timers* internos e configurar funções de interrupções.

A partir desta biblioteca e comandos C, desenvolveu-se um programa para cumprir determinadas atribuições. As principais atribuições a serem realizadas são: a leitura dos sensores, o envio dos sinais aos motores, a comunicação com a estação e o algoritmo de controle.

Devido a complexidade computacional do projeto, optou-se por encapsular o código em arquivos fontes diferentes. Em cada arquivo fonte agrupou-se um conjunto de funções e variáveis globais referentes a uma atribuição.

Outra peculiaridade do programa é a orientação a microcontrolador. Devido as suas limitações, o microcontrolador necessita de um programa com poucos níveis de sub-rotinas, número controlado de variáveis e preferencialmente, aritmética com inteiros de 16 bits. Para isso, opta-se por aumentar a complexidade da compilação, utilizar aproximações matemáticas e comandos de baixo nível.

Além disso, decidiu-se por utilizar o código *open source Multiwii*⁽¹¹⁾ para acelerar a realização das atribuições. Esse código já possui arquivos fontes

para quase todas as atribuições do projeto precisando, contudo, de pequenas adaptações.

6.2 PROGRAMA DE COMUNICAÇÃO

O programa de comunicação será o responsável por capturar e interpretar a mensagem que chegar através do *XBee* na porta serial da placa controladora. Esta parte do programa está em dois arquivos distintos: o *RX.cpp* e o *PROTOCOL.cpp*. A parte do programa contida em *RX.cpp* é a responsável por fazer a comunicação serial com o *XBee*, enquanto o *PROTOCOL.cpp* define uma sequência de comandos para observar, através da porta micro usb, sinais importantes do quadricóptero. Contudo, outro grupo de pesquisa se responsabilizou por realizar a implementação da parte de comunicação e telemetria deste projeto.

6.3 PROGRAMA DE INTERFACE COM OS SENSORES

A placa controladora *MultiWii* vem com diversos sensores, a fim de ser obter as variáveis de estados utilizadas na malha de controle.

Nos blocos de programa *config.h* e *def.h*, conjuntamente, são definidos os sensores da placa que serão utilizados. No caso foram selecionados o giroscópio, acelerômetro, barômetro e magnetômetro.

Os dados obtidos nestes sensores são muito ruidosos, por isso não servem diretamente para realizar o controle do quadricóptero. Para o tratamento destes, é feito em *imu.cpp* por médias móveis, e armazenado no vetor global *imu.gyroData[axis]*, o qual guarda os valores que serão utilizados pelo programa de controle.

6.4 PROGRAMA DE CONTROLE

O programa de controle está localizado em *MultWii.cpp*. Ele implementa a malha PID descrita na sessão 4.3.

Como foi ressaltado, a análise da malha da figura 6.1 será dividida em três partes vistas nas figuras 6.2, 6.3 e 6.4.

Iniciando-se pela malha da figura 5.4: o sinal de entrada representado pela variável `rcCommand[AXIS]` passa por um ganho de 2 gerando o sinal `rc`. Esse sinal por sua vez entra num somador, onde sua diferença com o sinal `imu.gyroData[axis]`, proveniente dos sensores filtrados de velocidade angular *PITCH* ou *ROLL* dependendo do valor assumido por `axis`, é calculada gerando o sinal `error`, que é o erro do sistema.

O sinal de erro então é acumulado em uma variável e amplificado gerando a componente integral do sistema (`ITerm`). O integrador também contém um saturador para limitar o acúmulo máximo do `error` entre 16000 e -16000 . Além disso, passa-se esse sinal por um bloco anulador que zera a parcela integral da saída caso o sensor envie um sinal demasiadamente pequeno de variação, o que evita que o acúmulo do erro se torne maior do que o ajuste do comando de entrada em si.

Já a parcela proporcional `PTerm` é obtida amostrando-se `rc` e o passando por um bloco de ganho.

Por fim, a passagem do sinal `imu.gyroData[axis]` por um bloco derivador discreto e um bloco de ganho geram a parcela derivativa do sinal de saída. A parcela derivativa já compõe a saída do sistema, porém os sinais `ITerm` e `PTerm` das parcelas integral e proporcional antes passam por um ajuste extra feito nas malhas das figuras 6.3 e 6.4, que será analisado adiante. Após esse

ajuste os sinais ITerm e PTerm são adicionados e o sinal DTerm é subtraído ao sinal de saída `axisPID[axis]` que está associado ao sinal elétrico PWM posteriormente enviado aos motores pelo ESC.

Analizando agora a malha vista na figura 5.5: a entrada dessa malha de controle também é o sinal `rc`. Porém visto que essa malha representa os ajustes finos do sistema temos que o sinal `rc` imediatamente entra em um bloco saturador, para limitar sua magnitude entre 500 e -500 . O sinal resultante passa por outro somador onde sua diferença com o sinal `att.angle[axis]` é calculada gerando mais um erro. Vê se que esse novo erro é usado para calcular mais dois parâmetros, sendo um proporcional proveniente do ramo superior com o ganho K_p' e outro integral proveniente do ramo inferior. Ambos os sinais gerados são limitados por saturadores e encaminham-se para a terceira e última malha do sistema vista na figura 6.4.

O sinal PID que será enviado aos motores após a lei de controle ser aplicada, ficará contido no vetor `axisPID[axis]` onde cada posição contém o sinal de um eixo.

6.5 PROGRAMA DE INTERFACE COM OS MOTORES

As funções de interação com o motor e as variáveis relevantes foram agrupadas no arquivo *Output.cpp*.

Este bloco de programa pegará os sinais PID gerado no programa de controle e aplicar nos atuadores. A função responsável por isso é a chamada *mixTable()*.

Esta função juntará o sinal gerado no PID correspondente a cada ângulo no atuador correto. A fim de obter o efeito desejado. Para isso foi definida a função *PIDMIX(X,Y,Z)* que fará a operação `rcCommand[THROTTLE] +`

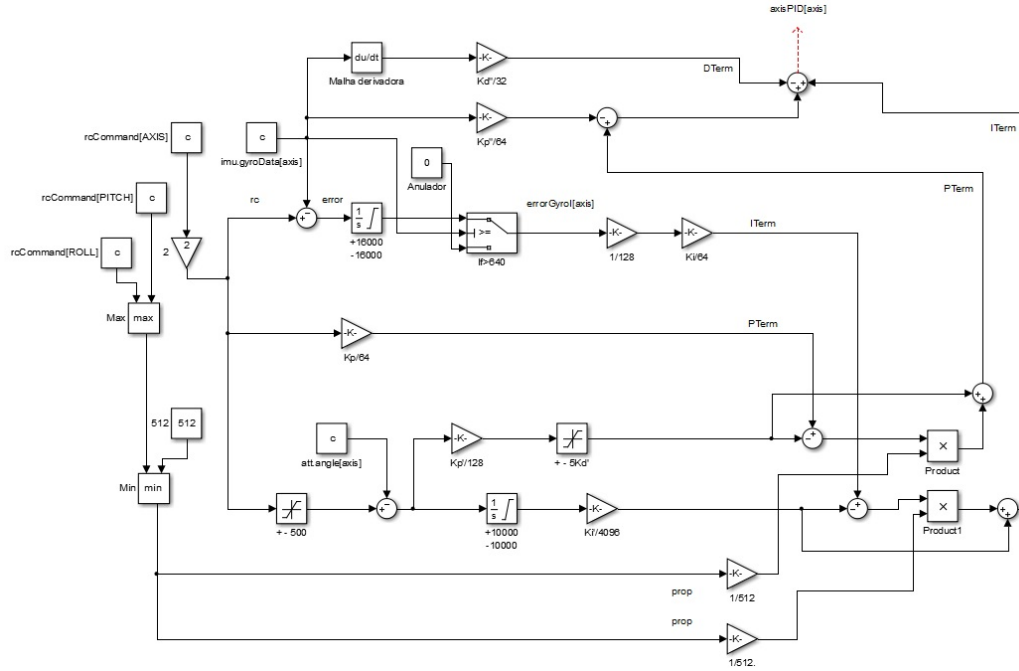


FIG. 6.1: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos

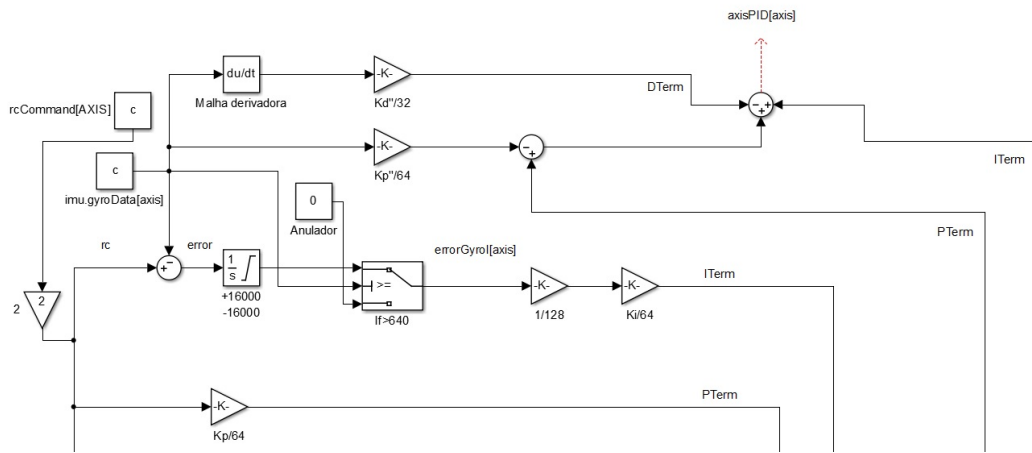


FIG. 6.2: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos, componentes proporcional, derivativa e integral

$$axisPID[ROLL]*X + axisPID[PITCH]*Y + YAW_DIRECTIONS * axisPID[YAW]*Z.$$

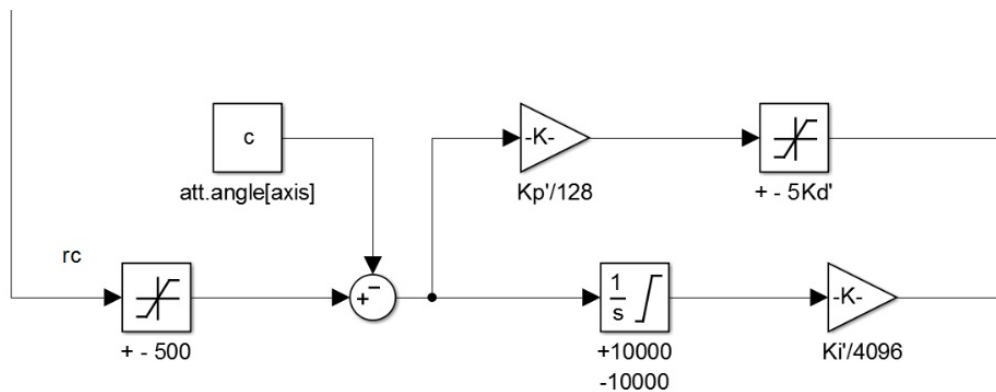


FIG. 6.3: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos, malha de ajuste dos sinais proporcional e integral

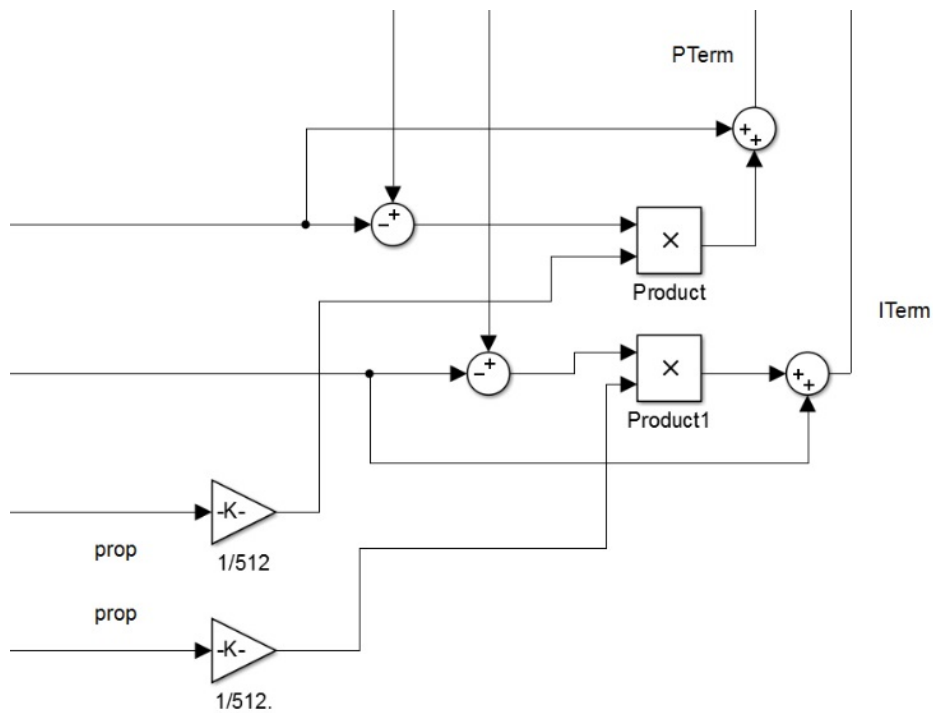


FIG. 6.4: Representação do algoritmo de controle PID para *ROLL* e *PITCH* em diagrama de blocos, malha de composição de ajuste dos sinais proporcional e integral

Dessa forma, cada motor receberá o seguinte sinal *PWM*:

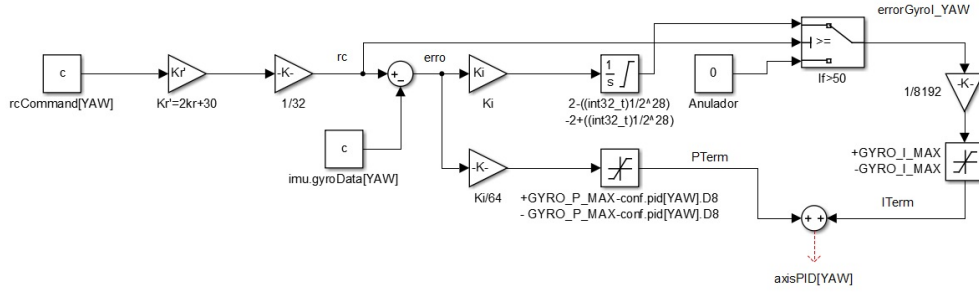


FIG. 6.5: Representação do algoritmo de controle PID para *YAW* em diagrama de blocos

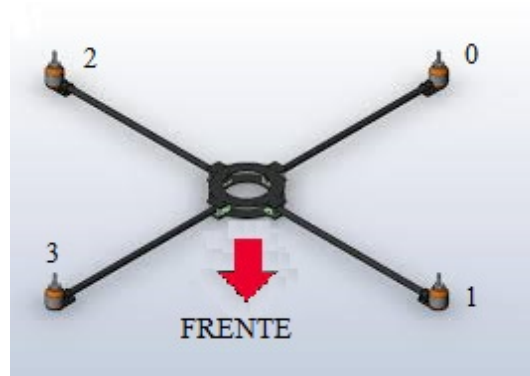


FIG. 6.6: Numeração das hélices do quadricóptero no programa

$motor[0] = PIDMIX(-1,+1,-1);$

$motor[1] = PIDMIX(-1,-1,+1);$

$motor[2] = PIDMIX(+1,+1,+1);$

$motor[3] = PIDMIX(+1,-1,-1);$

Ou seja, se o *DRONE* tentar mergulhar e rotacionar no sentido horário, será gerado um *axisPID[ROLL]* positivo, acelerando assim o motor 2 e 3 e

diminuindo 0 e 1. E o *axisPID[PITCH]* será negativo, acarretando a desaceleração do motor 0 e 2 e aumentando o 1 e 3.

7 TESTES

A fim de testar se a malha de controle é eficiente, testes foram realizados para ver a resposta desta em diversas situações. Foram realizados dois tipos de testes que serão descritos a seguir.

7.1 TESTES SIMULADOS

Para apenas verificar se os sinais gerados no PID eram os desejados para cada tipo de inclinação do quadricóptero foi realizado o seguinte teste.

Desconecta-se os motores da placa controladora e ligou-se ela ao computador, com o objetivo de, com a ajuda de um *software* específico, colher os dados do sinal PWM que foi enviado a cada um do motor.

Assim ao inclinar o bico do veículo para baixo, esperava-se que os sinais que iriam para os motores da frente aumentassem e os que iam para os motores detrás diminuíssem.

Porém, nos primeiros testes houveram muitas discrepâncias entre o que deveria acontecer e o que de fato acontecia.

Para resolver este problema o cabo de alimentação que ia da bateria até os *ESC* foram afastado da placa controladora. O fato de por ele circular uma corrente muito alta estava interferindo no correto do funcionamento dos sensores, causando um ruído grande que estava atrapalhando o controle.

7.2 TESTES PRÁTICOS

Após verificar com os testes anteriores que a malha de controle gera respostas satisfatórias, foi testado se após colocado numa posição e ligado, ele conseguiria ficar horizontal nesta.

Para isto ele foi preso ao teto por cordas e também ao chão (este último para segurança do equipamento e pessoal envolvido no teste).



FIG. 7.1: Configuração que foram realizados os testes

8 CONCLUSÕES

8.1 PONTOS ABORDADOS E CONTRIBUIÇÕES

Para realizar o projeto de desenvolvimento de um quadricóptero foi necessário o aprendizado de diversos pontos.

O primeiro deles foi a escolha das peças que eram necessárias às necessidades de nosso projeto. Seguido pela correta montagem e configuração dos diversos componentes.

A malha PID que foi implementada na placa controladora teve que ser adaptada para o nosso modelo de quadricóptero.

A fim de testar se a malha PID implementava um correto controle, testes tiveram que ser elaborados no ambiente do laboratório.

Por fim foi possível, com o projeto, ver limitações de material do laboratório que virão a ser sanadas no futuro.

8.2 LIMITAÇÕES

O grande limitador do trabalho foi quanto a obtenção de *Hardware*.

Grande parte das peças que eram necessárias a plena realização do projeto só eram obtidas no exterior, inclusive com escassez de alguns materiais.

Alguns componentes foram perdidos no frete ou mandados errados, o que atrasou mais ainda a montagem do quadricóptero.

Outra dificuldade encontrada se refere ao laboratório em que trabalhamos, por ser ainda novo e em desenvolvimento, não possuía todas as ferramentas necessárias para a montagem de nosso projeto. O que foi facilmente solucionado com a ajuda dos outros diversos laboratórios do IME, porém gerou uma certa barreira.

8.3 PERSPECTIVAS DE TRABALHOS FUTUROS

Para a evolução do projeto, um primeiro passo será os comandos de controle do quadricóptero serem dados por rádio-frequência. Ao invés de executar um programa que só tenta estabilizá-lo em uma posição.

Passado esta etapa, a perspectiva seria integrar ao projeto a orientação do quadricóptero via GPS. O que eliminaria a necessidade de uma pessoa enviando comando para ele o tempo todo, aumentando sua autonomia e a aplicabilidade para monitoração remota.

9 REFERÊNCIAS

- [1] Biografia de George Devol; http://en.wikipedia.org/wiki/George_Devol; Acessado em novembro de 2013.
- [2] Invent; http://www.invent.org/2011induction/1_3_11_induction_devol.asp; Acessado em novembro de 2013.
- [3] Defesa Brasileira; <http://www.defesabr.com/Fab/BQM1BR.pdf>; Acessado em novembro de 2013.
- [4] Flight Global; <http://www.flightglobal.com/pdfarchive/view/1924/1924%20-%200047.html>; Acessado em novembro de 2013.
- [5] Ardrone; <http://ardrone2.parrot.com>; Acessado em novembro de 2013.
- [6] SlashGear; <http://www.slashgear.com/diy-quadcopters-quaduino-ng-and-aeroquad-videos-1369771/>; Acessado em novembro de 2013.
- [7] HobbyKing; <http://www.hobbyking.com>; Acessado em dezembro de 2013.
- [8] Gerson Luiz S. Filho, Gustavo T. Rudiger, Jonathan P. M. Do Nascimento; *Quadricóptero*; Universidade Tecnológica Federal do Paraná, Engenharia de Computação, 2011.
- [9] Aeromodelismo Online; <http://www.aeromodelismoonline.com.br/como-descobrir-o-peso-motor-e-bateria-para-seu-modelo-elétrico-calculadora/>; Acessado em março de 2014.
- [10] João Paulo Ferreira Guimarães, Armando Sanca Sanca, Michel Santana de Deus, Tania Luna Laura, Alessandro Nakoneczny Schildt, Adelarado Adelino Dantas de Medeiros, Pablo Javier Alsina; *ESTRATÉGIA DE ESTABILIZAÇÃO PARA UM HELICÓPTERO QUADRI-ROTOR*; Instituto Anais do XIX Congresso Brasileiro de Automática, CBA 2012.
- [11] Multiwii; <http://www.multiwii.com/>; Acessado em abril de 2014.
- [12] *ClassicalPIDControl* by Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado; Prentice Hall, 2000.