



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO



Escola de Engenharia Elétrica Mecânica e Computação

Construção de um quadricóptero com microcontrolador
ATMEGA328P

Glauco Buarque da Silva

Orientador: Prof. Dr. Fernando Ferreira de Melo

Goiânia

2018

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR
VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE
GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC nº 1204/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG):

Nome completo do autor: *Glauco Buarque da Silva*

Título do trabalho: *Construindo um quadricóptero com o microcontrolador ATMEGA 328 P*

2. Informações de acesso ao documento:

Concorda com a liberação total do documento SIM NÃO¹

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF do TCCG.

Glauco Buarque da Silva
(Glauco Buarque da Silva)

Ciente e de acordo:

Fernando Ferreira de Melo
(Fernando Ferreira de Melo)

Data: 17 / 12 / 18

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Glauco Buarque da Silva

**Construção de um quadricóptero com
microcontrolador ATMEGA328P**

Monografia apresentada à escola de Engenharia Elétrica, Mecânica e Computação da Universidade Federal de Goiás como requisito para finalização do curso de Engenharia Elétrica.

Orientador: Prof. Dr. Fernando Ferreira de Melo

Goiânia

2018

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

SILVA, Glauco Buarque da
Construindo um quadricóptero com o microcontrolador
ATMEGA328P [manuscrito] / Glauco Buarque da SILVA. - 2018.
XCIII, 93 f.: il.

Orientador: Prof. Dr. Fernando Ferreira de MELO.
Trabalho de Conclusão de Curso (Graduação) - Universidade
Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de
Computação (EMC), Engenharia Elétrica, Goiânia, 2018.
Bibliografia. Anexos. Apêndice.
Inclui siglas, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Drone, Quadricóptero. 2. Filtro Mahony. 3. Filtro digital IIR. 4.
Controlador PID. 5. AVR. I. MELO, Fernando Ferreira de, orient. II.
Título.

CDU 621.3



ATA DE AVALIAÇÃO DE PROJETO FINAL

Aos 17 dias do mês de dezembro do ano de 2018,

foi apresentado e defendido o Projeto Final intitulado Construção de um
Quadróptero com Microcontrolador ATMEGA328P

perante a banca examinadora composta pelos membros:

1. Fernando Ferreira de Melo, orientador e presidente;
2. Prof. Dr. José Wilson Lima Merys; 3. Msc. Rubens Antônio ALVES.

Após a exposição do trabalho por parte do(s) autor(es), aluno(s) do curso de Engenharia Elétrica, foram lhe(s) atribuídas as seguintes notas pelos membros da banca:

Aluno(s)	Membro 1	Membro 2	Membro 3
<u>Glauco Buarque da Silva</u>	<u>10,0</u>	<u>10,0</u>	<u>9,5</u>

Nada mais havendo a registrar, eu, Fernando Ferreira de Melo, designado secretário "ad hoc" da banca examinadora, lavrei a presente Ata do ocorrido, a qual, lida e considerada conforme, vai assinada por mim e pelos membros da banca.

Goiânia, 17 de dezembro de 2018.

Fernando Ferreira de Melo
José Wilson Lima Merys
Rubens Antônio Alves

Dedicatória

Dedico o seguinte trabalho aos meus pais, que me apoiaram desde o início, mesmo após dificuldades foram meus pilares, garantindo saúde, atenção, e me deram essa oportunidade de realização.

Dedico também aos meus amigos, que me auxiliaram no processo.

Agradecimentos

Começo agradecendo aos meus pais pelo apoio moral, meus amigos e familiares que depositaram em min confiança para a conclusão desta etapa na minha vida. Um sonho de cursar um curso superior na área de tecnologia.

Agradeço aos professores da UFG, por me fornecer conhecimento, suporte e orientação, nesse caminho longo trilhado.

Resumo

O presente trabalho demonstra a construção de um quadricóptero (veículo aéreo quadrimotor), sua construção física e implementações para software. É um sistema solidário (*strapdown*) pois seus sensores estão fixados na estrutura, reduzindo a necessidade de algumas calibrações e melhorando a confiabilidade, já que reduz o número de partes móveis. Proveniente de uma realimentação negativa, e através de um módulo IMU (3x acelerômetros, 3x giroscópios, 3x magnetômetros) o software (firmware) para controle da estabilização foi desenvolvido utilizando a técnica do controle discreto PID, com adição de um mecanismo para deslocamento do mesmo. Os algoritmos utilizados para contornar o problema do ruído elétrico (lido nos sensores) devido a característica físicas e mecânicas da estrutura, é um filtro digital (IIR) passa-baixos de segunda ordem biquadrático, e “Filtro Madgwick-Mahony” onde este faz uso dos ‘quatérnios’ como variável de estado e fusão dos dados sensoriais para posteriormente calcular os ângulos euclidianos: arfagem, rolagem e guinada (*pitch*, *roll*, *yaw*). Também foi realizado a análise de Fourier para detectar o limiar do filtro passa-baixos digital. Devido ao projeto utilizar o microcontrolador de 8-bits AVR ATMEGA328-PB, fez-se necessário algumas otimizações e aproximações, devido ao limitado poder de processamento. Sua CPU não possui FPU (*floating point unit*) para lidar com cálculos de ponto flutuante (são emulados por software, o compilador AVR-GCC faz automaticamente). A IDE usada para desenvolvimento do software do microcontrolador foi o “Atmel Studio 7.0”. O software de telemetria e comando foi implementado em C# (C Sharp) no “Microsoft Visual Studio 2017”, para fins de monitoramento, medição e simulação (do filtro).

Palavras-chave: Quadricóptero, filtro Mahony, arquitetura AVR, controle PID, quatérnios, drone, filtro digital IIR, transformação biquadrática, C# (C Sharp).

Abstract

The present work demonstrates the construction of a quadcopter (air vehicle quad motor), from its physical construction to its implantation by software. It is a strapdown system because its sensors are strapped to the structure, removing the need for some calibrations and increasing its reliability by reducing moving parts. It comes from a negative feedback and through an inertial sensor (containing 3x accelerometers, 3x gyroscopes, 3x magnetometers, in an exclusive board) the control stabilization software was developed using a discrete PID control technique, with addition of a mechanism for displacement. The algorithm used to solve electrical noise problem (read in sensors) is Digital IIR 2nd order bi-quad low-pass filter, and 'Filter Madgwick-Mahony', which uses 'quaternions' to further calculate Euclidean angles: pitch, roll and yaw. Was also performed the Fourier analysis of frequency spectrum to set up the filter cut-off frequency. The use of an ATMEGA328-PB AVR 8-bit microcontroller has processing capability limitations and was necessary to use some optimizations and approximations. Its CPU does not have a FPU (floating point unit) to deal with floating point calculations (they are emulated by software, the compiler AVR-GCC does it automatically). The IDE used to develop the microcontroller firmware was "Atmel Studio 7.0". Telemetry and control software was developed on C# (C Sharp) in "Microsoft Visual Studio 2017", for purposes of monitoring, measurement and simulation (filter response).

Keywords: Quadcopter, Mahony filter, AVR architecture, PID control, quaternions, drone, IIR digital filter, biquadratic transformation, C#.

Índice de figuras

FIGURA 1 - BATERIA UTILIZADA COMO FONTE PRINCIPAL DE ENERGIA	17
FIGURA 2 - CARREGADOR DE BATERIAS MICROCONTROLADO UTILIZADO	18
FIGURA 3 - MINI CONVERSOR DC-DC BUCK UTILIZADO	18
FIGURA 4 - TRANSCEPTOR NRF24L01	20
FIGURA 5 - PINOS DO TRANSCEPTOR NRF24L01	21
FIGURA 6 - EIXOS DE ORIENTAÇÃO DO ACELERÔMETRO E GIROSCÓPIO	22
FIGURA 7 - EIXOS DE ORIENTAÇÃO DO MAGNETÔMETRO	22
FIGURA 8 - ESC UTILIZADA.....	26
FIGURA 9 - COMANDO MÍNIMO PARA A ESC (0%).....	27
FIGURA 10 - COMANDO DE 50% PARA A ESC	27
FIGURA 11 - COMANDO MÁXIMO PARA A ESC (100%)	27
FIGURA 12 - SAÍDA DAS ESCs, PARA OS MOTORES BRUSHLESS	28
FIGURA 13 - MEDIÇÃO DO PESO DO MOTOR.....	29
FIGURA 14 - MOTOR BRUSHLESS (SENTIDO HORÁRIO).....	29
FIGURA 15 - MEDIÇÃO DA FORÇA EMPUXO DO MOTOR BRUSHLESS	29
FIGURA 16 - MOTOR BRUSHLESS (SENTIDO ANTI-HORÁRIO).....	29
FIGURA 17 - ESTRUTURA DO DRONE, VISTA LATERAL	30
FIGURA 18 - ESTRUTURA DO DRONE, VISTA SUPERIOR	30
FIGURA 19 - ÂNGULOS DE EULER.....	32
FIGURA 20 - VISTA SUPERIOR DA ROTAÇÃO DOS MOTORES	35
FIGURA 21 - ROTAÇÃO DO CORPO, A NECESSIDADE DA COMPENSAÇÃO	36
FIGURA 22 - ESTRUTURA GERAL DE FILTROS IIR (FONTE: (WEEKS, 2007)).....	37
FIGURA 23 - MÉTRICAS DOS ACELERÔMETROS EM ESTADO OCIOSO	38
FIGURA 24 - - ESPECTRO DE FREQUÊNCIAS DO ACELERÔMETRO Z COM MOTORES OCIOSOS	39
FIGURA 25 - MÉTRICAS DOS ACELERÔMETROS COM MOTORES A 50% DE POTÊNCIA	39
FIGURA 26 - ESPECTRO DE FREQUÊNCIAS DO ACELERÔMETRO Z COM MOTORES A 50% DE POTÊNCIA	39
FIGURA 27 - MÉTRICAS DOS GIROSCÓPIOS COM MOTORES OCIOSOS	40
FIGURA 28 – ESPECTRO UNILATERAL DE FREQUÊNCIAS DO GIROSCÓPIO X PARA MOTORES OCIOSOS	41
FIGURA 29 - MÉTRICAS DOS GIROSCÓPIOS COM MOTORES A 50% DE POTÊNCIA.....	41

FIGURA 30 – ESPECTRO UNILATERAL DE FREQUÊNCIAS DO GIROSCÓPIO X PARA MOTORES A 50% DE POTÊNCIA.....	41
FIGURA 31 - POSICIONAMENTO DO MÓDULO IMU NA PLACA DE CIRCUITO IMPRESSO.....	42
FIGURA 32 - LEITURAS DOS MAGNETÔMETROS COM MOTORES EM ESTADO OCIOSO.....	43
FIGURA 33 - ESPECTRO DE FREQUÊNCIAS DO MAGNETÔMETRO X (MOTORES OCIOSOS).....	43
FIGURA 34 - LEITURA DOS MAGNETÔMETROS COM MOTORES EM 50% DE POTÊNCIA.....	44
FIGURA 35 - ESPECTRO DE FREQUÊNCIAS DO MAGNETÔMETRO X COM MOTORES A 50% DE POTÊNCIA.....	44
FIGURA 36 - ESTRUTURA BÁSICA DE UM CONTROLADOR PID.....	47
FIGURA 37 - ESQUEMÁTICO DA PCI.....	51
FIGURA 38 - LAYOUT DA PCI.....	52
FIGURA 39 - PLACA PCI CONFECCIONADA.....	53
FIGURA 40 - FIXAÇÃO DA PCI.....	53
FIGURA 41 - CONTROLE FÍSICO UTILIZADO (FORMATO PS2).....	55
FIGURA 42 - FORMA DA CLASSE PRINCIPAL DO SOFTWARE DE TELEMETRIA.....	56
FIGURA 43 - FORM AUXILIAR PARA PLOTAR DADOS.....	56
FIGURA 44 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS FILTRADOS, MOTORES A 15% DE POTÊNCIA... ..	58
FIGURA 45 - AJUSTE DE LINHA PARA O SINAL DO ACELERÔMETRO X FILTRADO.....	59
FIGURA 46 - PLOTAGEM DO AJUSTE DE LINHA PARA OS TRÊS EIXOS, SINAL FILTRADO.....	60
FIGURA 47 - FIXAÇÃO DO DRONE PARA REALIZAÇÃO DE TESTES E CALIBRAÇÕES.....	61
FIGURA 48 – RESPOSTA AO DEGRAU COM GANHOS AJUSTADOS.....	61
FIGURA 49 - RESPOSTA AO DEGRAU COM CONSTANTE DE INTEGRAÇÃO PEQUENA (K_I).....	62
FIGURA 50 - RESPOSTA AO DEGRAU COM CONSTANTE DE PROPORCIONALIDADE ALTA (K_P).....	62
FIGURA 51 - RESPOSTA AO DEGRAU DO DRONE COM CONTROLADOR PID ANGULAR.....	63
FIGURA 52 - DIVERGÊNCIA NA SAÍDA DO FILTRO DIGITAL PASSA-BAIXOS.....	64
FIGURA 53 - FLUXOGRAMA DO FIRMWARE DO MICROCONTROLADOR.....	71
FIGURA 54 - LEITURA DO MAGNETÔMETRO X. EM AZUL, DADOS BRUTOS, EM VERMELHO, DADOS FILTRADOS. O EIXO Y ESTÁ EM MILGAUSS.....	73
FIGURA 55 - ESPECTRO DE FREQUÊNCIAS DO MAGNETÔMETRO X, ANTES E APÓS FILTRAGEM.....	73
FIGURA 56 - LEITURA DOS GIROSCÓPIOS COM MOTORES OCIOSOS SEM FILTRO.....	74
FIGURA 57 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS COM MOTORES OCIOSOS SEM FILTRO.....	74

FIGURA 58 - LEITURA FILTRADA DOS GIROSCÓPIOS, MOTORES OCIOSOS	75
FIGURA 59 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS FILTRADOS, MOTORES OCIOSOS	75
FIGURA 60 - LEITURA DOS GIROSCÓPIOS COM MOTORES A 15% DE POTÊNCIA	76
FIGURA 61 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS COM MOTORES A 15% DE POTÊNCIA	76
FIGURA 62 - LEITURA FILTRADA DOS GIROSCÓPIOS, MOTORES A 15% DE POTÊNCIA.....	77
FIGURA 63 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS FILTRADOS, MOTORES A 15% DE POTÊNCIA ...	77
FIGURA 64 - LEITURA DOS GIROSCÓPIOS COM MOTORES A 40% DE POTÊNCIA	78
FIGURA 65 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS COM MOTORES A 40% DE POTÊNCIA	78
FIGURA 66 - LEITURA FILTRADA DOS GIROSCÓPIOS, MOTORES A 40%	79
FIGURA 67 - ESPECTRO DE FREQUÊNCIAS DOS GIROSCÓPIOS FILTRADOS, MOTORES A 40% DE POTÊNCIA ...	79
FIGURA 68 - LEITURA DOS ACELERÔMETROS, MOTORES A 5% DE POTÊNCIA, SEM FILTRO	80
FIGURA 69 - ESPECTRO DE FREQUÊNCIAS DOS ACELERÔMETROS, MOTORES A 5% DE POTÊNCIA, SEM FILTRO	80
FIGURA 70 - LEITURA FILTRADA DOS ACELERÔMETROS, MOTORES A 5% DE POTÊNCIA	81
FIGURA 71 - ESPECTRO DE FREQUÊNCIAS DOS ACELERÔMETROS FILTRADOS, MOTORES A 5% DE POTÊNCIA	81
FIGURA 72 - LEITURA DOS ACELERÔMETROS, MOTORES A 30% DE POTÊNCIA, SEM FILTRO	82
FIGURA 73 - ESPECTRO DE FREQUÊNCIAS DOS ACELERÔMETROS, MOTORES A 30% DE POTÊNCIA, SEM FILTRO	82
FIGURA 74 - LEITURA FILTRADA DOS ACELERÔMETROS, MOTORES A 30%	83
FIGURA 75 - ESPECTRO DE FREQUÊNCIAS DOS ACELERÔMETROS FILTRADOS, MOTORES A 30% DE POTÊNCIA	83
FIGURA 76 - LEITURA DOS ACELERÔMETROS, MOTORES A 60% DE POTÊNCIA, SEM FILTRO	84
FIGURA 77 - ESPECTRO DE FREQUÊNCIAS DOS ACELERÔMETROS, MOTORES A 60% DE POTÊNCIA, SEM FILTRO	84
FIGURA 78 - LEITURA FILTRADA DOS ACELERÔMETROS, MOTORES A 60% DE POTÊNCIA	85
FIGURA 79 - ESPECTRO DE FREQUÊNCIAS DOS ACELERÔMETROS FILTRADOS, MOTORES A 60% DE POTÊNCIA	85

Lista de tabelas

TABELA 1 - RUÍDO (RMS) DOS ACELERÔMETROS	58
TABELA 2 - RUÍDO (RMS) DOS GIROSCÓPIOS	59

Lista de abreviações

FPV	First Person View
AVR	Advanced Virtual Risc
MEMS	Micro-Electro-Mechanical Systems
RAM	Random Access Memory
ROM	Read Only Memory
MIPS	Millions Instructions Per Second
FPU	Floating Point Unit
CPU	Central Processor Unit
PID	Proportional Integral and Derivative
PWM	Pulse Width Modulation
ESCs	Eletronic Speed Controller
LiPo	Lithium Polymer
BEC	Battery Eliminator Circuit
DC	Direct Current
PCI	Placa de Circuito Impresso
IMU	Inertial Measurement Unit
RISC	Reduced Instruction Set Computer
CMOS	Complementary Metal-Oxide-Semiconductor
EEPROM	Electrically-Erasable Programmable Read-Only Memory
SMD	Surface Mounting Device
C.I.	Circuito Integrado
IC	Integrated Circuit
ADC	Analog Digital Conversor
SPI	Serial Peripheral Interface

USART	Universal Synchronous Asynchronous Receiver
Transceiver	
I2C	2-wire Serial Interface
CPOL	Clock Polarity
CPHA	Clock Phase
MOSI	Master output slave input
MISO	Master input slave output
IIR	Infinity Impulse Response
GUI	Graphical User Interface
IDE	Integrated Development Environment
DOF	Degrees of Freedom
AHRS	Attitude and Heading Reference System
GPS	Global Position System

Sumário

Sumário.....	11
Introdução.....	14
Objetivos.....	16
Capítulo 1 – Periféricos.....	17
1.1 Alimentação.....	17
1.2 Microcontrolador.....	19
1.3 Transceptor.....	19
1.4 Módulo IMU.....	22
1.4.1 Acelerômetro.....	23
1.4.2 Giroscópios.....	23
1.4.3 Magnetômetros.....	24
1.5 ESCs.....	25
1.6 Motores.....	28
1.7 Estrutura.....	29
1.8 Controle remoto.....	30
Capítulo 2 – Métodos de controle, orientações e referenciais.....	31
2.1 Justificativa da escolha ângulos de Euler e quatérnios.....	31
2.2 Ângulos de Euler.....	32
2.3 Quatérnios.....	32
2.4 Ação de controle dos motores.....	34
2.4.1 O controle da guinada (<i>yaw</i>).....	35

	12
2.4.2 O controle de rolagem e arfagem	36
2.4.3 O controle da vertical	36
Capítulo 3 – Análise dos sinais e filtros	37
3.1 Necessidade de um filtro digital passa-baixos.....	38
3.1.1 Acelerômetros.....	38
3.1.2 Giroscópios.....	40
3.1.3 Magnetômetros	43
3.2 Filtro digital passa-baixos (Biquadrático IIR).....	45
3.3 Controlador PID discreto.....	47
3.4 Filtro de fusão de dados Madgwick & Mahony	49
Capítulo 4 – Placa de circuito impresso (PCI).....	51
4.1 O esquemático	51
4.2 O layout	51
4.3 A placa confeccionada.....	53
Capítulo 5 – Software de Telemetria e Comando.....	54
Capítulo 6 – Resultados	57
6.1 Efeito após aplicação do filtro digital IIR passa-baixos.....	57
6.2 Respostas obtidas.....	61
6.2.1 PID da posição angular.....	61
6.2.2 PID da velocidade angular.....	63
6.3 Divergência do filtro passa-baixos	63
6.4 Custo temporal das funções principais	64

Conclusões	65
Possíveis melhorias.....	67
Referências	69
Apêndice A – Fluxograma do firmware da controladora	71
Apêndice B – Métricas dos sensores e seus espectros de Fourier	72
B.1 Magnetômetro X	72
B.2 Giroscópios	74
B.3 Acelerômetros	80
Apêndice C – Funções para cálculos do filtro e PID.....	86
Anexo A – Filtro de fusão de dados Madgwick-Mahony.....	88
Anexo B – Funções rápidas de cálculo aproximado.....	91

Introdução

Drones, ou veículos aéreos quadricópteros possuem importância no cotidiano das pessoas de hoje, principalmente em produtoras, onde são utilizados para filmagem de eventos no lugar de helicópteros ou grandes estruturas metálicas para obtenção de imagens aéreas. Vários *hobbistas* adotam drones de corrida, de alta performance como passatempo ou forma de competição, utilizando óculos de visão em primeira pessoa com transmissão de vídeo analógica (famoso FPV). São ainda utilizados por bombeiros em resgate de pessoas em zonas próximas a rios, lagos e praias, com lançamento de boias.

Devido vantagem da estrutura física e operação dos drones, em relação à helicópteros e aviões, estão ganhando cada vez mais espaço no mundo, pois: aviões não se mantem em uma posição fixa no espaço (i.e. parados no ar) e helicópteros convencionais utilizam-se de complexos controles mecânicos para estabilidade, o controle é feito por inclinações nos ângulos de ataque das hélices do motor principal (McGowen, 2005). Já os drones, utilizam um controle “simples”, basicamente aplicando um torque contrário ao movimento em relação ao seu centro de massa para estabilização horizontal e, na guinada (‘yaw’) aumentando a velocidade de motores gêmeos (i.e., motores com mesmo sentido de rotação) varia-se o momento do corpo, e com isso é aplicado um torque em relação a seu centro.

É um sistema solidário (*strapdown*) pois seus sensores estão fixados na estrutura, reduzindo a necessidade de algumas calibrações e melhorando a confiabilidade, já que reduz o número de partes móveis. Sistemas *strapdown* possuem altas taxas de atualização (maiores que 60Hz, geralmente acima de 1kHz) e um alcance dinâmico de medida. Isso é necessário para se ter uma alta precisão.

Os drones em relação a outros aeromodelos são sistemas controlados por leituras elétricas de sensores inerciais, e isso engloba o problema do ruído elétrico. A vibração

mecânica gerada pela rotação dos motores e a variação da força empuxo por eles aplicada é a principal fonte de ruídos pois, gera mini vibrações captadas pelos acelerômetros e giroscópios feitos de tecnologia MEMS. O ruído de maior evidência, conhecido como ruído browniano (ou marrom) é o principal fator da necessidade de se utilizar filtros digitais na controladora. O ruído branco também está presente, embora sua contribuição para o caos do controle seja insignificante, devido ao fato de sua intensidade ser constante no espectro de frequências, e também este depende da temperatura (Davenport & Root, 1987), um fator de variação pequena no sistema eletromecânico abordado.

A controladora, um microcontrolador AVR de 8 bits modelo ATMEGA328-PB, foi selecionada pelo seu baixo custo, facilidade de obtenção. Entretanto, por ser um microcontrolador de baixo nível de processamento (~1 MIPS/MHz), CPU sem FPU, limitada quantidade de memória RAM (2kBytes) e ROM (32kBytes), este, mostrou-se capaz de realizar tal tarefa, envolvendo cálculos de pontos flutuantes emulados por software (o próprio compilador cuida disso) e funções trigonométricas. Houve a necessidade de implementar funções mais rápidas e algumas desconsiderações, de casas decimais e variáveis para melhorar o desempenho do código, como por exemplo a expansão em série de Taylor da função arco seno.

O processo controle, foi implementado baseado na técnica PID discreta, é a parte menos intuitiva e abstrata, pois, o controle utiliza-se de duas etapas PID, a primeira e mais importante, tem como base os giroscópios, e a segunda, utiliza-se da fusão sensorial dos giroscópios e acelerômetros, por meio do filtro Mahony. O controle mecânico é realizado por motores brushless, modelos importados e convencionais.

O processo de envio de controle, é realizado por PWM para as ESCs, existem dois protocolos padrões discutidos na respectiva seção, o utilizado neste projeto é o Standard PWM 1-2ms. Posteriormente os motores brushless necessitam comandos comutados

adequadamente em cada uma de suas três fases para adequado funcionamento (Brown Ward; Microchip Technology Inc, 2002), é onde entra a saída das ESCs, pulsos de alta frequência e potência na saída, com entrada de controle PWM. A escolha do uso de ESCs foi devido reduzir carga de processamento do microcontrolador principal, e facilitar o controle dos motores.

A telemetria implementada por software, mostra os dados em tempo real na tela do computador, seus ângulos euclidianos e potência dos motores.

Objetivos

O objetivo principal deste trabalho é concluir um mecanismo protótipo (quadricóptero, drone) com estabilidade, para posterior aperfeiçoamento.

Uma interface de telemetria no computador para monitorar a situação em tempo real das variáveis de estado e seu desempenho.

A implementação de um controle do usuário.

Capítulo 1 – Periféricos

Neste capítulo será apresentado os periféricos utilizados, bem como suas configurações, consumo e finalidade.

1.1 Alimentação

A alimentação é feita por uma bateria de LiPo (polímeros de lítio) de 3 células (11.1V), 2200mAh de capacidade de corrente e 25C (~55A) de taxa de descarga contínua (corrente). Cada célula tem por convenção 3.7V, e seus limiares são 4.2V quando carregada e 3.2V totalmente descarregada. É importante ressaltar utilizar um carregador que não ultrapasse a magnitude máxima de tensão como exemplo o da Figura 2, um carregador de vários tipos de baterias (chumbo, níquel-cádmio, polímero de lítio etc...) microcontrolado. Esse carregador controla corrente e tensão de saída de forma ajustável pelo usuário, porém, está implementado em seu firmware os níveis máximos de tensão de cada tipo de bateria, e não é ajustável pelo usuário. Ele amostra continuamente com seu conversor ADC embutido a tensão de cada célula para não subir acima do máximo (pois pode danificá-la permanentemente) e ajusta o processo de carregamento, descarregando a célula com maior tensão para equilibrar todas em uma mesma magnitude.

As baterias de polímero de lítio são as mais usadas nessa aplicação devido sua alta capacidade de descarga de corrente, já que os motores brushless possuem altíssimas potências.



Figura 1 - Bateria utilizada como fonte principal de energia



Figura 2 - Carregador de baterias microcontrolado utilizado

A seguir, existem duas fontes reguladores de tensão:

- A chamada BEC, é um circuito composto principalmente pelo C.I. 7805-SMD interno às ESCs. Usam a regulação linear de tensão para obter 5Vdc fixos na saída e capacidade de 2A (variável, depende do modelo, mais ou menos corrente). Sua finalidade é alimentar o microcontrolador e o conversor no próximo item descrito.
- Um pequeno conversor DC-DC buck de entrada 4.75V~23V e saída 1~17V, corrente de 1.8A, onde por facilidade de implementação na PCI, sua entrada é 5V (a partir das BECs) e sua saída foi ajustada manualmente para 3V. Sua finalidade é alimentar o módulo IMU (onde contém os acelerômetros e giroscópios) e o rádio transceptor.



Figura 3 - Mini conversor DC-DC buck utilizado

1.2 Microcontrolador

O microcontrolador, é um C.I. de 28 pinos, AVR ATMEGA328-PB de alta performance e baixa potência de 8-bits fabricado com a tecnologia CMOS, com 32kBytes de memória ROM, 2kBytes de RAM, 1kByte de EEPROM e clock máximo de 20MHz. Sua arquitetura RISC avançada possibilita 131 instruções de máquina e, aproximadamente 1MIPS por MHz de clock. Sua CPU não possui unidade de cálculo de ponto flutuante separada, no entanto é emulada por software e custa muito tempo de processamento, sendo assim, utilizados somente onde não é possível contornar com outra solução. Possui resumidamente dois timers de 8-bits (T0 e T2), um timer de 16-bits (T1), um timer *watchdog* (serve para monitorar se o software/CPU travou), um módulo conversor analógico digital (ADC), dois módulos SPI, uma USART, um I2C, um comparador analógico, e pinos com suporte a interrupção *on-change* (i.e., se o nível lógico do pino mudar, ele chama uma interrupção). Mais detalhes sobre o hardware interno, encontra-se no seu datasheet (Microchip, Atmega 328/P Datasheet, 2016).

O microcontrolador foi configurado para utilizar seus 3 timers, os dois T0 e T2 para gerar o PWM dos motores e o T1 para contagem do tempo. O módulo SPI foi configurado para 1Mbits de velocidade e CPOL, CPHA como 0 0 respectivamente (amostra na subida e travamento do dado na descida), a justificativa da escolha do módulo SPI vs I2C é sua velocidade, superior ao I2C, economizando tempo. Por fim, seu clock de entrada é sintonizado por um cristal de quartzo de 16MHz.

1.3 Transceptor

Os transceptores são módulos feitos baseados no C.I. nRF24L01, que possui as seguintes características:

- Banda de operação em 2.4GHz & 0dBm de potência máxima.

- Taxas de transferências no ar: 250kbps, 1Mbps, 2Mbps.
- Alimentação máxima de 3.6V, nível alto à 3.3V porém seus pinos suportam entradas de 5V, dispensando conversores de nível, para tanto é de extrema importância que a alimentação do C.I. não ultrapasse 3.3V, de acordo com seu datasheet.
- Tecnologia *Enhanced ShockBurst™*, gerencia a montagem e desmontagem de pacotes de dados automaticamente, e pode implementar comunicação *half-duplex* (um canal para TX e RX, um sentido por vez).
- Possui até 6 camadas por canal (nós).
- Comunicação somente por um canal, ou seja, pode ser configurado como TX ou RX, no entanto, se corretamente configurada, pode ser implementada a comunicação *half-duplex*.
- 125 canais selecionáveis, com largura de banda aproximada de 1MHz, no entanto se selecionado a taxa de transferência de 2Mbps, a largura de banda sobe para aproximadamente 2MHz.
- Pino para interrupção, chegada, envio ou estouro do limite de reenvio.

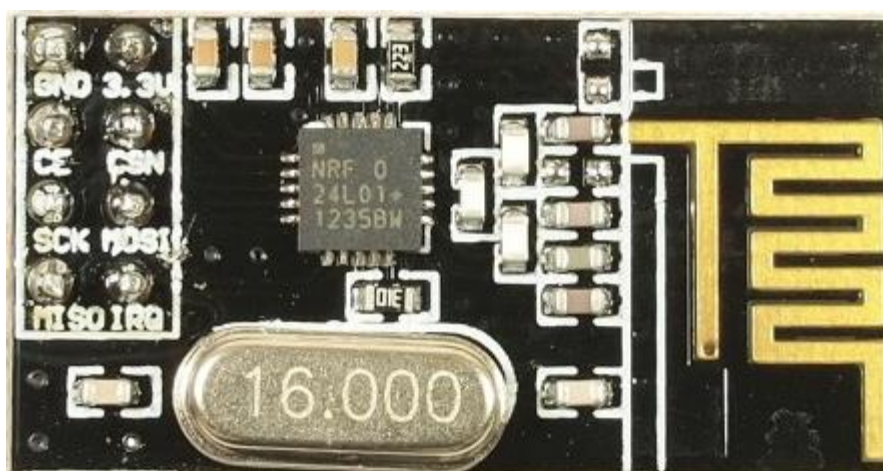


Figura 4 - Transceptor nRF24L01

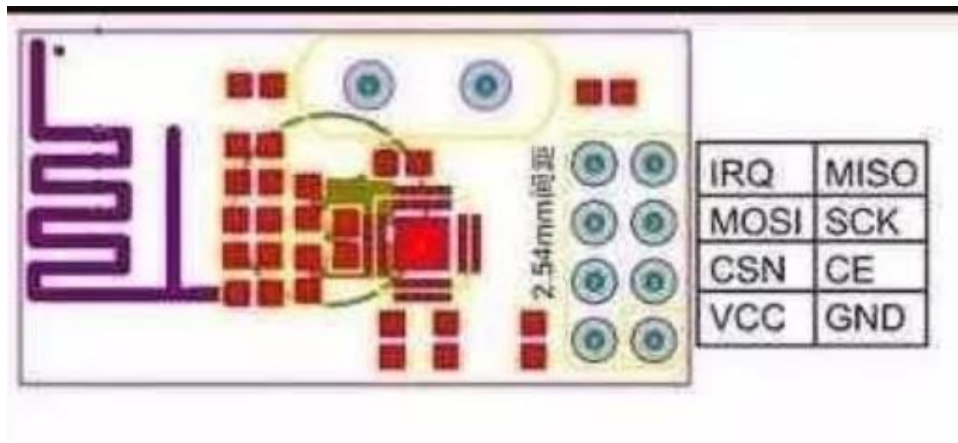


Figura 5 - Pinos do transceptor nRF24L01

A pinagem do transceptor é como a Figura 5 - Pinos do transceptor nRF24L01, onde:

- IRQ: *Interrupt request queue* (fila de requisição de interrupção), o pino é ativo em baixo.
- MISO: *Master input, slave output* (Entrada no mestre, saída do escravo), é o pino onde trafegam os dados do módulo para o microcontrolador.
- MOSI: *Master output, slave input* (Saída do mestre, entrada no escravo), é o pino cujo sentido do tráfego é do microcontrolador para o módulo.
- SCK: *Serial clock*, é o pino de clock.
- CSN: *Chip select*, é o pino ativo em baixo, que deve ser utilizado para iniciar a comunicação SPI.
- CE: *Chip enable*, é um pino utilizado para configurar o estado ativo do módulo.
- VCC: Alimentação positiva, como o módulo possui regulador interno, pode ser até 12V. No presente projeto, alimenta-se com 3V.
- GND: Pino de referência, ou terra (*ground*).

Os transceptores foram configurados para usar a tecnologia *Enhanced ShockBurst™*, o reconhecimento de pacote automático sem retransmissão (o TX sabe se o pacote chegou, mas não reenvia o pacote), taxa de transferência de 2Mbps, potência máxima de envio (0dBm) e sem máscaras de interrupção, o microcontrolador é encarregado da checagem.

O transceptor do drone foi configurado como função primária RX, e ele envia um pacote de dados de seu estado atual para o TX sempre que solicitado por meio do auto reconhecimento dinâmico, uma característica manipulada pela *Enhanced ShockBurst™*. Já o transceptor do computador foi configurado como função primária TX, ele envia um pacote de dados para o RX a cada 2ms aproximadamente. São enviados pacotes vazios com objetivo de não perder o link de comunicação.

1.4 Módulo IMU

O módulo de medidas inerciais MPU-9255 é uma versão atualizada do módulo MPU9250 da fabricante InvenSense, é a primeira geração de módulos com 9 eixos (InvenSense Store, 2018), com suporte a I2C e SPI, onde foi implantado um magnetômetro (AK8963) no barramento I2C interno. O nível lógico alto dos pinos é 3.3V. Consiste de 3 acelerômetros, 3 giroscópios e 3 magnetômetros, todas perpendiculares entre si, entretanto o magnetômetro possui orientações diferentes como nas figuras abaixo:

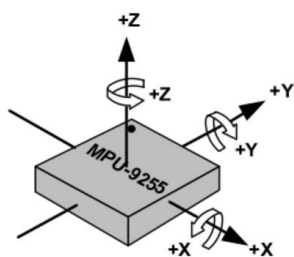


Figura 6 - Eixos de orientação do acelerômetro e giroscópio

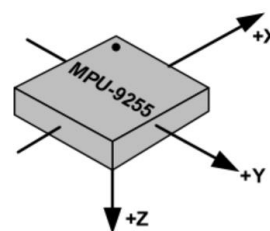


Figura 7 - Eixos de orientação do magnetômetro

1.4.1 Acelerômetro

O acelerômetro é um sensor inercial que mede a aceleração que está submetido, e quando fixado a um corpo mede a aceleração ao qual esse corpo está submetido. Quando em repouso no planeta Terra, sua medida indica uma aceleração de 1g para cima (onde ‘g’ significa a aceleração da gravidade) pois, em relação à um objeto em queda livre, o sensor está acelerado a 1g para cima.

Os acelerômetros deste módulo são sistemas MEMS e possuem quatro escalas de medida, $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$. Internamente possui um filtro digital passa baixas de frequência de corte pré-configurada para remoção de ruído, e sua taxa máxima de amostragem é 4kHz sem filtro e 1kHz com filtro. A saída é coletada no tamanho de 2 bytes, ou ainda, um inteiro com sinal de 16 bits (*signed int*).

1.4.2 Giroscópios

Os giroscópios também são sistemas MEMS e medem a velocidade angular que está submetido. São altamente sensíveis a vibrações mecânicas. No presente projeto eles serão utilizados para verificar a mudança de direção.

O giroscópio possui também 4 escalas de medidas, $\pm 250dps$, $\pm 500dps$, $\pm 1000dps$, $\pm 2000dps$, onde “dps” significa degraus por segundo. Internamente possui um filtro digital passa baixas com frequências de corte pré-configuradas, e diferente do acelerômetro, sua taxa de amostragem é incrivelmente superior, a máxima chega a 32kHz sem filtro e a mínima 1kHz, possui uma peculiaridade, caso o filtro digital esteja configurado para a frequência de 250Hz, sua taxa de amostragem é 8kHz. A saída deste também é coletada no tamanho de 2 bytes, ou ainda, um inteiro com sinal de 16 bits

(*signed int*). Mais informações no datasheet (InvenSense, D.S., 2014) e no mapa de registros (InvenSense, R.M., 2015)

1.4.3 Magnetômetros

O magnetômetro é um sensor usado para medir a intensidade, direção e sentido do campo magnético em que está submetido, no caso o da Terra.

O magnetômetro possui somente uma escala de medida $\pm 4800\mu\text{T}$, duas resoluções de saída, 14 e 16 bits, taxa de amostragem de 8Hz ou 100Hz, sem filtro digital. É um C.I. a separado, utiliza-se da comunicação I2C, entretanto o módulo MPU9255 possui um mestre i2c dentro do chip, a qual foi configurado para comunicar-se com o magnetômetro e adquirir os dados sem a necessidade de intervenção do microcontrolador. Mais informações no datasheet (AsahiKASEI, 2013).

O módulo foi configurado para taxa de amostragem de 1kHz, filtro digital em ambos os sensores na frequência de 184Hz para acelerômetro e 92Hz para o giroscópio. O magnetômetro foi configurado para uma saída de 100Hz, e resolução de 16 bits.

Como finalidade, o módulo é responsável pelas medidas que alimentam os algoritmos de controle e estabilização. Com intuito de não haver redundâncias de figuras, observe a Figura 31 onde mostra o módulo fixado à placa de circuito impresso.

Os pinos são:

- VCC: Alimentação positiva, possui um regulador linear ‘*low drop-out voltage*’ de 3.3V para garantir a estabilidade, portanto é possível aplicar uma tensão maior à 3.3V no pino VCC.
- GND: *Ground*, ou pino de referência (terra).
- SCK: *Serial clock*, pino onde entra o clock da comunicação, tanto para I2C ou SPI.

- SDA ou SDI: *Serial data input*, é o pino de MOSI para comunicação SPI ou o pino bidirecional para comunicação I2C.
- EDA: Pino do barramento interno de comunicação I2C, equivalente a SDA, não utilizado neste projeto.
- ECL: Pino do barramento interno de comunicação I2C, equivalente a SCL, não utilizado neste projeto.
- AD0 ou SDO: *Serial data output*, é o pino de MISO para comunicação SPI, na comunicação I2C ele define o bit final do endereço do módulo.
- INT: Interrupção, pino utilizado para mostrar por hardware novos dados.
- NCS: *Chip select*, o pino que inicia comunicação SPI.
- FSYNC: Pino utilizado para sincronizar medidas, caso existam outros sensores, este pino segura o disparo dos conversores internos. Não utilizado neste projeto.

1.5 ESCs

As controladoras eletrônicas de velocidade são pequenos circuitos com 8 fios de conexões, onde 3 deles conectam-se ao motor brushless, outros dois vermelho e preto mais espessos à bateria, e os três fios coloridos são os sinais de controle e alimentação, na seguinte ordem: Marrom – terra; Vermelho – saída da BEC, alimentação positiva 5V; Laranja: Sinal de PWM, ou entrada de comando. Essa ESC possui uma BEC de 2A, e corrente de pico de saída 30A. A característica “LIPO 2S-4S” refere-se à quantidade de células da bateria de alimentação suportada, ou seja, ela suporta baterias de 2 a 4 células (7,4V a 14,8V).



Figura 8 - ESC utilizada

Os dois padrões aceitos de comando para essa ESCs são PWM modificados, no aspecto de não necessitar uma frequência constante de atualização, pois, os controladores internos agem nas bordas de subida e descida do sinal. Tal sinal de controle necessita de um pulso com aproximadamente 1ms de largura para ser reconhecido como “novo comando de velocidade” e outro pulso concatenado de no máximo 1ms também, ou seja, para a velocidade máxima de controle, um pulso de 2ms, para a mínima, um pulso de 1ms. As velocidades intermediárias são linearmente relacionadas com a largura do pulso, e é necessário uma *dead-band* (zona morta) entre um sinal e outro de pelo menos 4 μ s. O outro padrão segue na mesma lógica que o descrito acima, entretanto, dividido por 8 as larguras dos pulsos, resultando em no mínimo 125 μ s e máximo de 250 μ s. Neste projeto utilizou-se do comando Standard.

- O primeiro padrão aceita comandos de até ~490Hz, chamado de Standard PWM.
- O segundo padrão, conhecido como ONE-SHOT, aceita comandos de até ~3.93kHz.

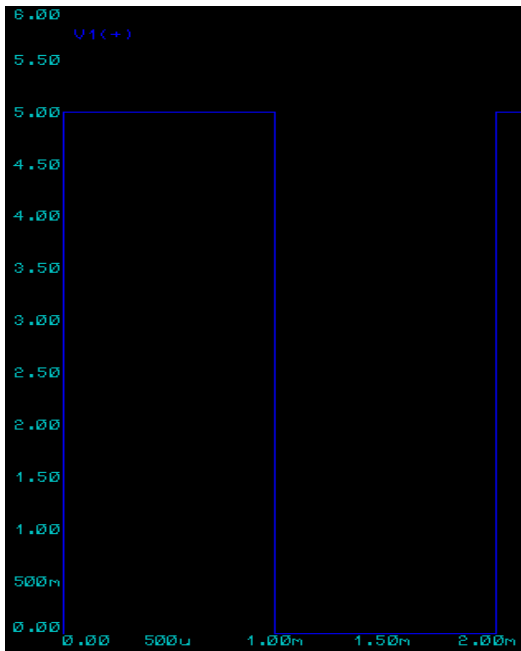


Figura 9 - Comando mínimo para a ESC (0%)

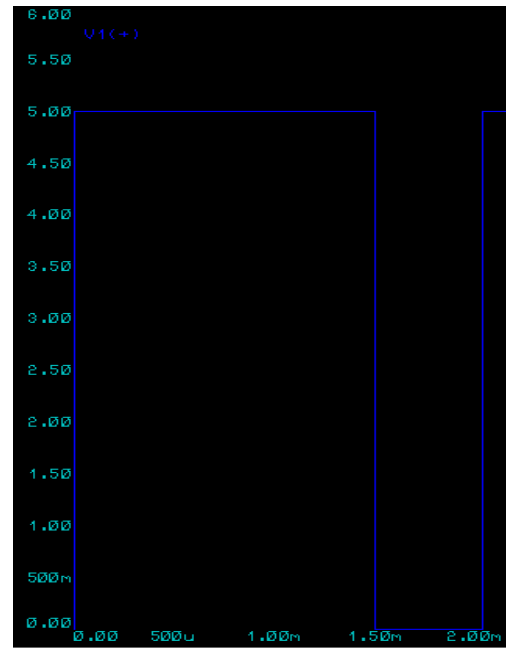


Figura 10 - Comando de 50% para a ESC

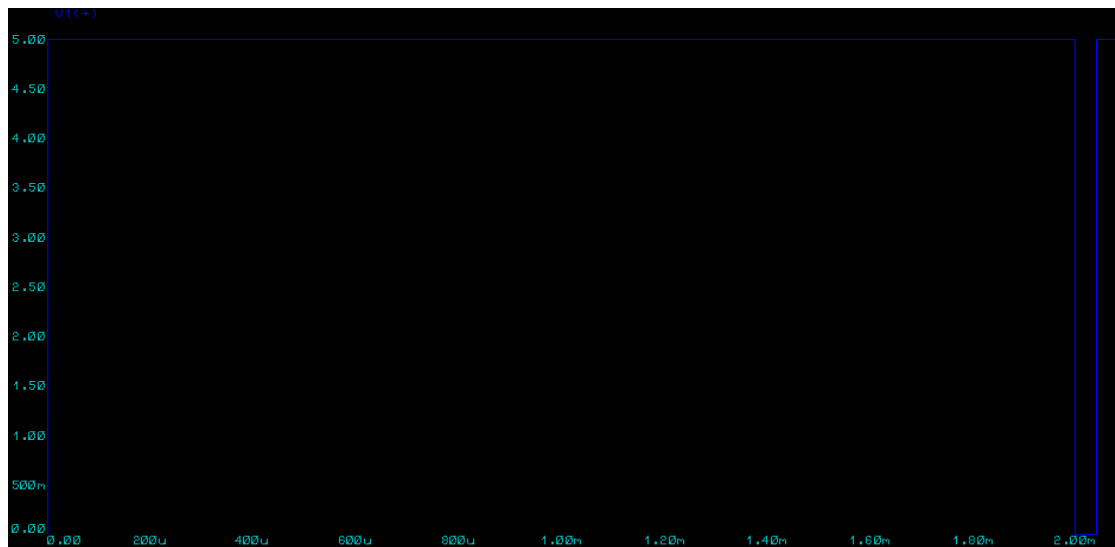


Figura 11 - Comando máximo para a ESC (100%)

Nota-se a *dead-band* na Figura 11 de aproximadamente $4\mu\text{s}$, caso não haja essa zona morta, a controladora descarta o comando ou age imprevisivelmente.

Independentemente da frequência de entrada, a saída é um PWM de 8kHz de frequência, e amplitude do valor da alimentação, e possui uma *dead-band* fixa de 15° de

acordo com o manual de usuário (BLHELI User Manual), como mostra a Figura 12, pulsos de $10\mu\text{s}$ de largura afastados por 15° elétricos (equivalente a $\sim 5.21\mu\text{s}$).

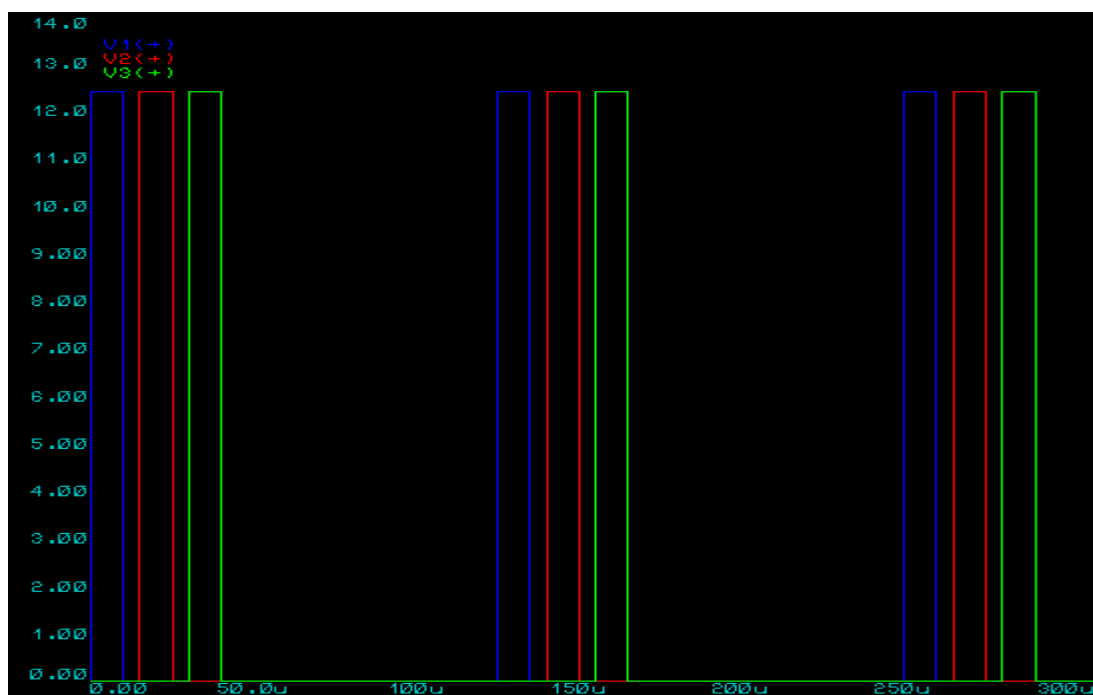


Figura 12 - Saída das ESCs, para os motores brushless

1.6 Motores

Os motores são brushless modelo 2212 920KV, dois pares, horário e anti-horário. A medida KV não significa *Quilo Volt*, e sim rpm/V, ou seja, a cada 1 volt aplicado, ele incrementa sua rotação por 920rpm (mas apenas a partir da tensão inicial de funcionamento). Sua tensão de operação é entre 7 e 13V. A força de empuxo de cada motor é aproximadamente 607g utilizando hélices 1045 (10mm por 4.5" de abertura) e a bateria completamente carregada. Seu peso é de aproximadamente 58g. Os motores com presilha de hélice prateada são de sentido de giro anti-horário e os de presilha preta são de giro horário.



Figura 13 - Medição do peso do motor



Figura 15 - Medição da força empuxo do motor brushless



Figura 14 - Motor brushless (sentido horário)



Figura 16 - Motor brushless (sentido anti-horário)

1.7 Estrutura

A estrutura é simples, feita de fibra de carbono e plástico. Possui trem de pouso. A distância entre dois motores de mesmo sentido (distância diagonal) é aproximadamente 75cm, e a distância entre dois motores de sentidos reversos (distância lateral) é aproximadamente 60cm. O peso total é cerca de 960g.



Figura 17 - Estrutura do drone, vista lateral

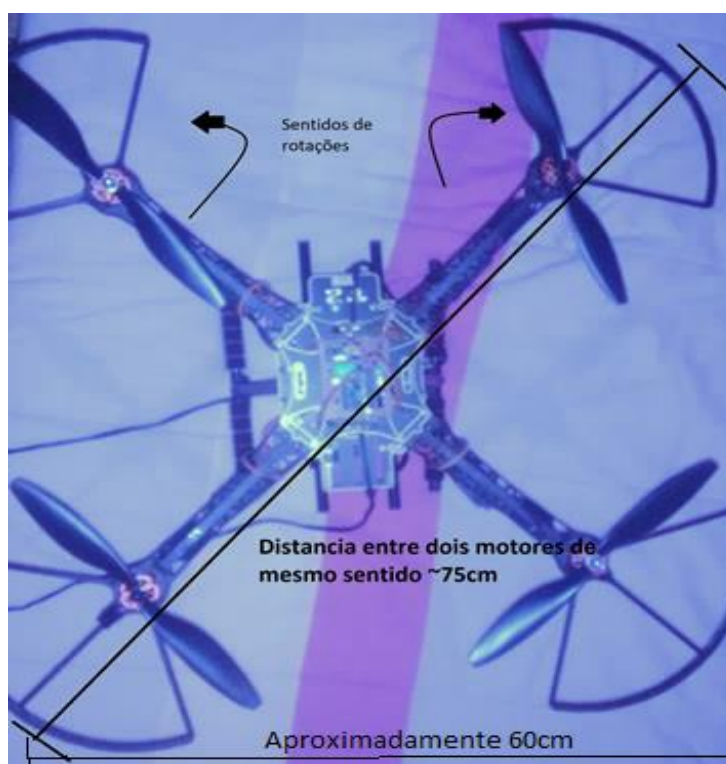


Figura 18 - Estrutura do drone, vista superior

1.8 Controle remoto

O controle remoto é feito com auxílio de um controle convencional de videogame modelo PS2 com conexão USB, realiza todos os comandos implementados na interface de telemetria e comando (Capítulo 5 – Software de Telemetria e Comando).

Capítulo 2 – Métodos de controle, orientações e referenciais

Nessa seção será abordado os métodos de controle, as referências do quadricóptero (ângulos de inclinação em relação ao referencial terrestre por meio dos quatérnios), as ações que devem ser tomadas pelo controlador para obter estabilidade física (horizontal, no espaço), e por fim, o atuador PID discreto no tempo.

2.1 Justificativa da escolha ângulos de Euler e quatérnios

Neste projeto optou-se por utilizar ângulos de Euler com auxílio de quatérnios, com justificativa de evitar matriz de rotação cartesiana (dos eixos coordenados e modo convenção aeronáutico XYZ), pois, necessitam de interpolação linear, cálculos envolvendo matrizes (e esses são caros computacionalmente) devido a múltiplas rotações, isso gera uma transição arbitrária e imprevisível. Outro fator para essa escolha foi contornar o problema do “*gimbal lock*”, no qual perde-se um grau de liberdade toda vez que uma rotação de 90° é realizada em torno de algum eixo coordenado.

Para contornar o problema de transições imprevisíveis, realiza-se somente uma rotação simples, “fato (demonstrado por Euler) que sempre é possível chegar de uma orientação a outra através de uma rotação simples, ao redor de um único eixo. [...]. Esse eixo, porém, não é necessariamente um dos eixos coordenados.” (Biasi & Gattass, 2002) por isso a adoção dos ângulos de Euler, uma rotação simples em torno de um eixo arbitrário gera uma transição suave. Já para contornar o problema das rotações diretas nos eixos cartesianos (que levam a equações extremamente complexas e pouco intuitivas), utiliza-se o sistema de representação dos quatérnios, que é bem mais adequado para operar sobre rotações (Biasi & Gattass, 2002).

2.2 Ângulos de Euler

Os ângulos de Euler foram formulados por *Leonard Euler* para descreverem a orientação de um corpo rígido girante em um espaço euclidiano tridimensional. São de simples compreensão, utilizam o referencial terrestre (fixo) e o referencial do próprio corpo (girante). São três ângulos independentes, de rotação anti-horária para especificar a orientação do corpo em relação ao referencial terrestre. Podem ser visualizados como segue:

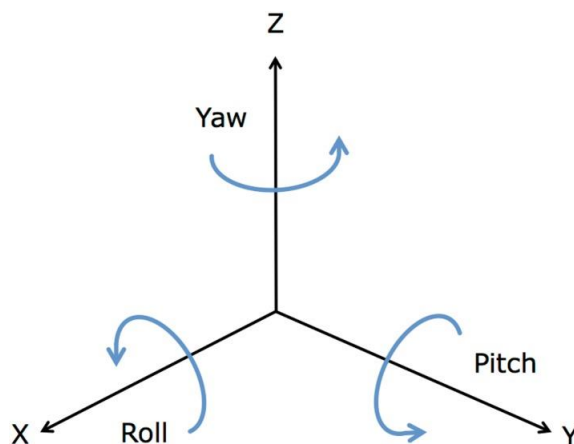


Figura 19 - Ângulos de Euler

Onde:

- *Yaw*: Guinada
- *Pitch*: Arfagem
- *Roll*: Rolagem

2.3 Quatérnios

Quatérnios, ou ainda, quatérnios, são uma generalização dos números complexos para o espaço tridimensional. Parece intuitivo, entretanto, essa generalização não incorpora somente mais um parâmetro, mas sim dois. No plano complexo bidimensional só existe um grau de liberdade angular, a rotação em torno da origem, pois, para parametrizar uma rotação, só precisamos especificar seu ângulo (o eixo de rotação sempre

é perpendicular ao plano complexo). Os dois parâmetros introduzidos advêm do fato que ao acrescentar uma dimensão, aumenta-se o número de graus de liberdade de rotação em dois (Biasi & Gattass, 2002).

Os quatérnios são números reais geralmente denotados pôr a , b , c , e d . E ainda com três números imaginários \mathbf{i} , \mathbf{j} e \mathbf{k} . De tal forma que um quatérnio:

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}. \quad (1)$$

As propriedades de \mathbf{i} , \mathbf{j} e \mathbf{k} são generalizações da propriedade \mathbf{i} dos números complexos bidimensionais:

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

Diferente dos números complexos de duas dimensões, existe a necessidade da multiplicação entre números imaginários:

$$\mathbf{ij} = \mathbf{k} \quad \mathbf{jk} = \mathbf{i} \quad \mathbf{ki} = \mathbf{j} \quad (3)$$

$$\mathbf{ji} = -\mathbf{k} \quad \mathbf{kj} = -\mathbf{i} \quad \mathbf{ik} = -\mathbf{j}$$

Seja $q = (s, \vec{w})$ onde \vec{w} é um vetor tridimensional representando b , c , e d .

$$q_1 * q_2 = (s_1, \vec{w}_1) * (s_2, \vec{w}_2) \quad (4)$$

$$= (s_1s_2 - \vec{w}_1 \cdot \vec{w}_2, s_1 \vec{w}_2 + s_2 \vec{w}_1 + \vec{w}_1 \times \vec{w}_2)$$

O conjugado de um quatérnio q é representado por:

$$\bar{q} = (s, -\vec{w}) \quad (5)$$

Seu inverso q^{-1} é definido de tal forma a:

$$qq^{-1} = 1 \quad (6)$$

Como q é unitário, seu inverso é seu conjugado:

$$q^{-1} = \bar{q} \quad (7)$$

Dadas essas definições, um ponto arbitrário $\vec{r} = (r_x, r_y, r_z)$ ao qual será executada uma rotação, será representado por um quatérnio $p = (0, \vec{r})$ de parte real nula. A rotação aplicada a \vec{r} será representada por um quatérnio $q = (s, \vec{w})$. A rotação $R_q(p)$ significa rotação de p por q , e obtida:

$$R_q(\mathbf{p}) = \mathbf{p}q\bar{q}^{-1} = \mathbf{p}q\bar{q} \quad (8)$$

Resolvendo a eq. (8):

$$\mathbf{R}_q(\mathbf{p}) = (\mathbf{0}, s^2\vec{\mathbf{w}} - (\vec{\mathbf{w}} \cdot \vec{\mathbf{w}}) \cdot \vec{\mathbf{r}} + 2(\vec{\mathbf{w}} \cdot \vec{\mathbf{r}}) \cdot \vec{\mathbf{w}} + 2s\vec{\mathbf{w}} \times \vec{\mathbf{r}} \quad (9)$$

O módulo (magnitude) de um quatérnio \mathbf{q} é:

$$\begin{aligned} |\mathbf{q}|^2 = \mathbf{q}\bar{\mathbf{q}} &= (s^2 - \vec{\mathbf{w}} \cdot -\vec{\mathbf{w}}, -s\vec{\mathbf{w}} + s\vec{\mathbf{w}} + \vec{\mathbf{w}} \times -\vec{\mathbf{w}}) \\ &= (s^2 + \vec{\mathbf{w}} \cdot \vec{\mathbf{w}}, \mathbf{0}) \\ &= s^2 + |\vec{\mathbf{w}}|^2 = 1 \end{aligned} \quad (10)$$

Como sua magnitude é 1, da eq. (10) um quatérnio pode ser representado por:

$$\mathbf{q} = (\cos \theta, \sin \theta \vec{\mathbf{h}}) \quad \text{e} \quad |\vec{\mathbf{h}}| = 1 \quad (11)$$

Na eq. (11) $\vec{\mathbf{h}}$ pode ser interpretado como vetor normalizado do eixo a ser rotacionado, e θ como ângulo a ser rotacionado de forma anti-horária nesse eixo.

Substituindo (11) em (9) e resolvendo:

$$\mathbf{R}_q(\mathbf{p}) = (\mathbf{0}, (\cos 2\theta)\vec{\mathbf{r}} + (1 - \cos 2\theta)(\vec{\mathbf{h}} \cdot \vec{\mathbf{r}})\vec{\mathbf{h}} + (\sin 2\theta) \vec{\mathbf{h}} \times \vec{\mathbf{r}}) \quad (12)$$

A eq. (12) pode parecer também complicada, no entanto, a rotação de uma rotação envolve a multiplicação de dois quatérnios (eq. 4) e depois aplicando a eq. (12) no quatérnio resultante, facilitando assim, as rotações de corpo livre matematicamente (Biasi & Gattass, 2002).

2.4 Ação de controle dos motores

Na Figura 20 podemos identificar o sentido de rotação individual de cada motor que pode ser operado separadamente. Observe a rotação dos motores em torno dos eixos horizontais, a soma da conservação do momento angular L é constante e o torque resultante é zero se todos os motores estiverem rotacionando na mesma velocidade.

O momento angular é definido como a quantidade de movimento angular de uma partícula e é dado por $L = \vec{\mathbf{r}} \times \vec{\mathbf{p}}$, onde $\vec{\mathbf{r}}$ é o vetor distancia a origem e $\vec{\mathbf{p}}$ é o vetor momento linear da partícula. “para possuir momento angular em relação a origem a

partícula não precisa estar girando em torno dela” (Halliday, Resnick, & Walker, 2008). A lei de conservação do momento angular diz que, se uma parte do corpo parado começa a se mover, a outra precisa rodar no sentido oposto para a conservação do momento, pois, caso não existe nenhuma força externa, o momento angular do corpo deve ser conservado. O torque é definido como a taxa de variação do momento angular em relação ao tempo.

2.4.1 O controle da guinada (*yaw*)

O controle da rotação do plano do quadricóptero em torno de seu eixo vertical é feito pelo aumento da rotação de motores de mesmo sentido (i.e., M1 e M3 ou M2 e M4, como na Figura 20) de tal forma a utilizar-se da conservação de momento angular. Por exemplo: Se o objetivo for mover o corpo no sentido anti-horário (eixo vertical saindo do papel), inicialmente temos que o momento angular total do corpo é nulo, dessa forma deve-se aumentar a rotação dos motores M2 e M4, pois assim, o momento angular total será desbalanceado (note que ele aponta para baixo), e pela lei da conservação do momento haverá um torque interno no sentido oposto ao momento angular desbalanceado, e o corpo girará para que seu momento se conserve (continue zero). Esse torque de sentido oposto aponta para cima, e de acordo com a regra da mão direita, a rotação se dá no sentido anti-horário.

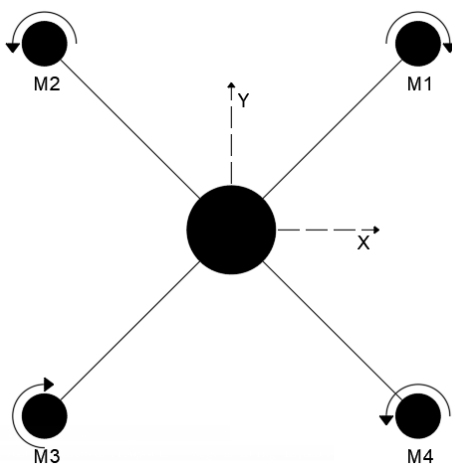


Figura 20 - Vista superior da rotação dos motores

2.4.2 O controle de rolagem e arfagem

O controle desses são bem intuitivos, é necessária uma simples aplicação de torque pela dupla de motores com rotações opostas, lateral ao eixo.

Para controlar a arfagem (o ângulo em torno do eixo “y”), aplica-se um controle de aumento de velocidade nos motores M2 e M3 para produzir torque positivo (em relação a y) ou M1 e M4 para produzir torque negativo.

De maneira análoga a arfagem, a rolagem utiliza-se dos motores M1 e M2 para torque positivo e, M3 e M4 para torque negativo, sempre em relação ao eixo em questão.

2.4.3 O controle da vertical

Para realizar este controle, realiza-se um aumento de velocidade em todos os motores, resultando assim, em uma força empuxo de maior magnitude e conseqüente elevação do quadricóptero. Caso o objetivo seja a descida, uma redução na velocidade de todos os motores se faz necessária.

Observe na Figura 21, quando o controle da arfagem ou rolagem for realizado, é necessário a compensação da força empuxo pois com a inclinação da força empuxo \vec{F}_E , a componente vertical é reduzida para $\cos \theta * \vec{F}_E$. Caso não seja realizada a compensação, o corpo tenderia a uma trajetória curvilínea em direção ao solo sempre que houver uma inclinação dessas. Basta dividir a força empuxo pelo cosseno do ângulo de inclinação.

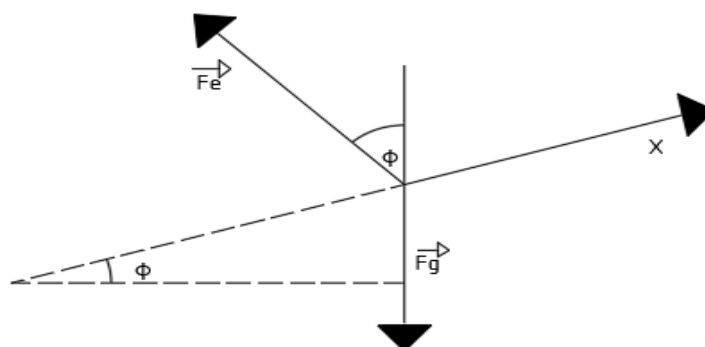


Figura 21 - Rotação do corpo, a necessidade da compensação

Capítulo 3 – Análise dos sinais e filtros

Neste capítulo será abordado o tratamento dos dados lidos dos sensores, bem como suas métricas e espectros de frequências.

Filtros IIR são filtros de *resposta ao impulso de duração infinito* que utilizam da realimentação de sinal (*feed-back*) (Weeks, 2007). Quanto maior a ordem do filtro IIR mais registradores são necessários (é baseado em valores passados, por isso a realimentação) e pode ocasionar instabilidade devido sua sensibilidade. Os filtros de 2nd ordem (segunda) em geral possuem desempenho satisfatório e raramente apresentam instabilidade. A escolha da topologia IIR é pela sua característica de excelente resposta em magnitude com banda passante estreita (Mersereau & Smith, 1993).

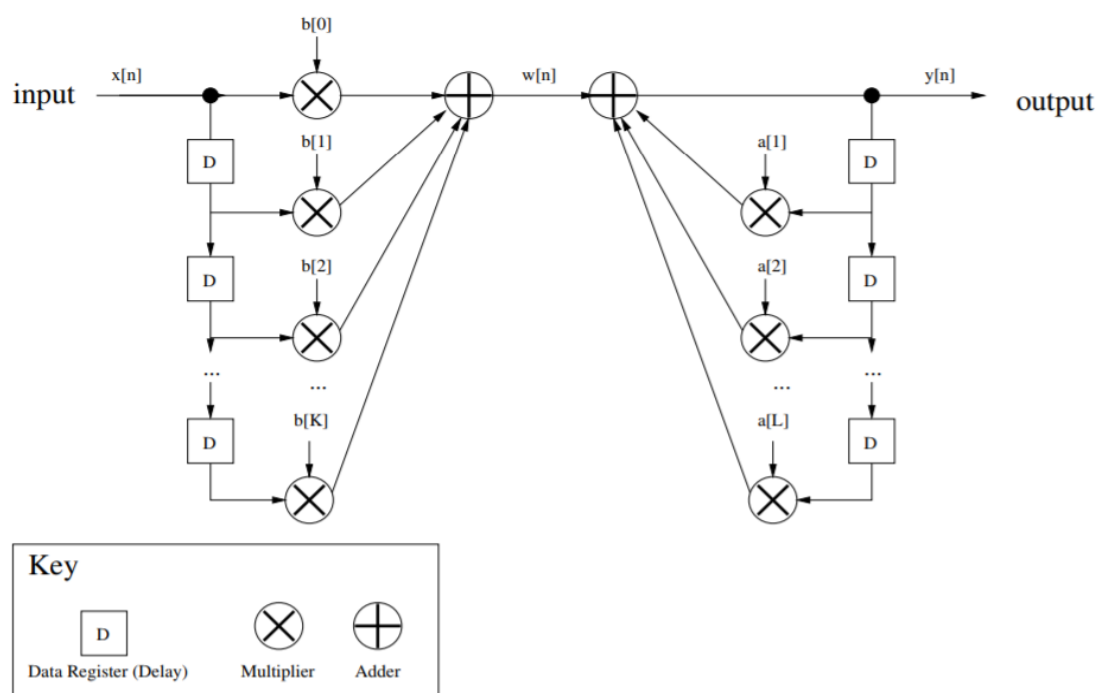


Figura 22 - Estrutura geral de filtros IIR (fonte: (Weeks, 2007))

Um filtro IIR tem a forma de uma equação de diferenças com termos no presente e no passado (nunca no futuro). Os filtros analógicos e os filtros digitais IIR por possuírem resposta ao impulso de duração infinita podem ser relacionados.

3.1 Necessidade de um filtro digital passa-baixos

Como mencionado anteriormente, os sensores inerciais foram configurados com um filtro passa-baixos de primeira ordem embutidos no módulo IMU (92Hz para acelerômetros e 92Hz para giroscópios). No entanto, mesmo com a atenuação primária desse filtro, o sinal de saída apresenta-se ruidoso e inapto para estabilização mecânica do quadricóptero. Para mostrar isso, foram realizadas aquisições de dados com o firmware do microcontrolador modificado, somente realizando a leitura e comando dos motores, e fixado a uma taxa de 1000 amostras/segundo (ou ainda $F_s = 1\text{kHz}$) para velocidades diferentes dos motores e em estado ocioso. O software de telemetria realizou a coleta de dados. A análise de Fourier foi efetuada com auxílio do software MATLAB. A finalidade de aplicar diferentes velocidades nos motores advém do fato que quanto maior a vibração mecânica maior o ruído captado pelos sensores.

As figuras a seguir mostram graficamente os sinais adquiridos. O eixo vertical (ordena) depende do sensor e o eixo horizontal (abscissa) é o tempo.

3.1.1 Acelerômetros

A análise de Fourier foi feita a partir de 5000 amostras, ou ainda, 5 segundos.

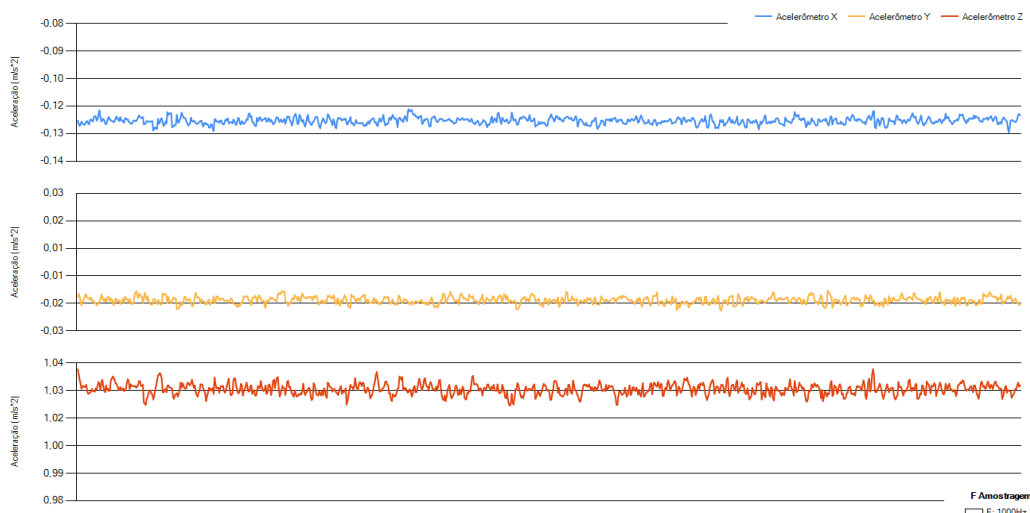


Figura 23 - Métricas dos acelerômetros em estado ocioso

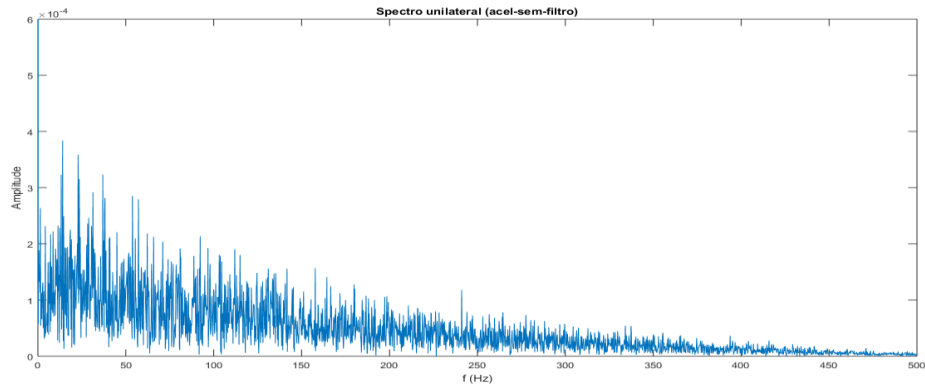


Figura 24 - - Espectro de frequências do acelerômetro Z com motores ociosos

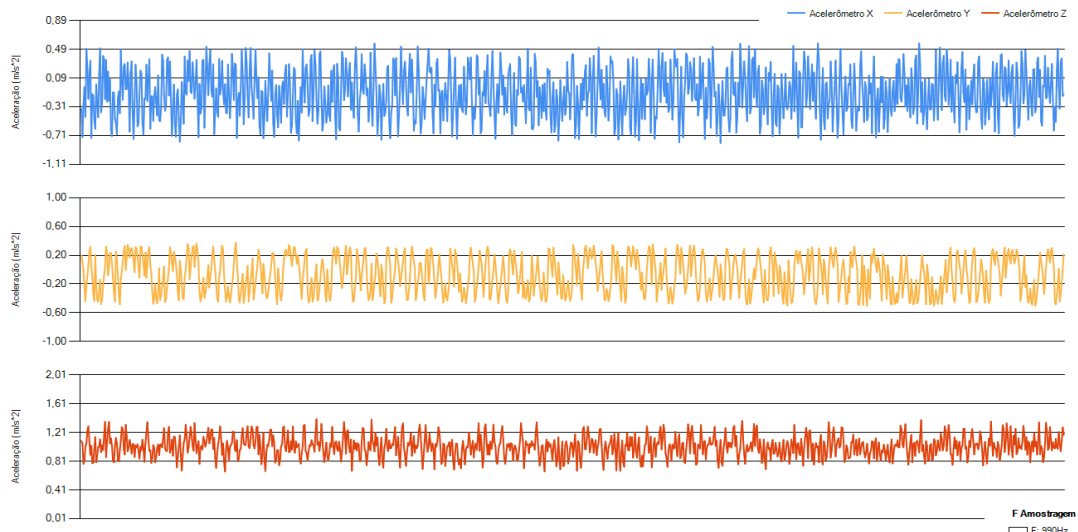


Figura 25 - Métricas dos acelerômetros com motores a 50% de potência

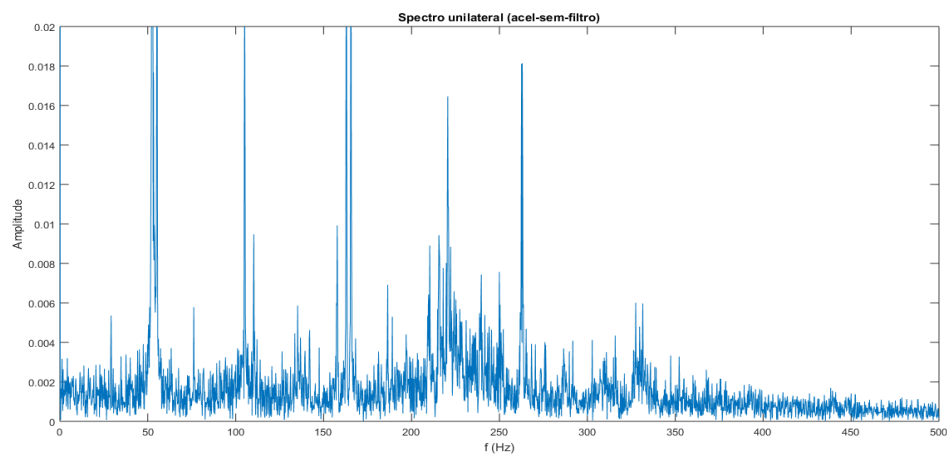


Figura 26 - Espectro de frequências do acelerômetro Z com motores a 50% de potência

Obs.: Para obter as métricas com motores a 50% de potência foi acoplado uma carga de 5kg no pé da estrutura do quadricóptero, pois, à essa potência o quadricóptero já está voando.

Dos gráficos, pode-se concluir o aumento do ruído e um nível CC em cada sensor. O nível CC é facilmente contornado obtendo-se a média de um grande número de amostras (~1k) e posteriormente subtraindo esse valor das amostras posteriores (calibração inicial). O nível de ruído aumenta significativamente com o aumento da potência dos motores (aumento da vibração mecânica) de $\pm 10^{-2} m/s^2$ no estado ocioso para $\pm 20^{-1} m/s^2$ a 50% de potência. Lembrando ainda, que os acelerômetros passam por um filtro de 1ª ordem diretamente no módulo IMU, na frequência de 92Hz.

3.1.2 Giroscópios

A análise de Fourier foi feita a partir de 5000 amostras, espaçadas de 1ms cada.

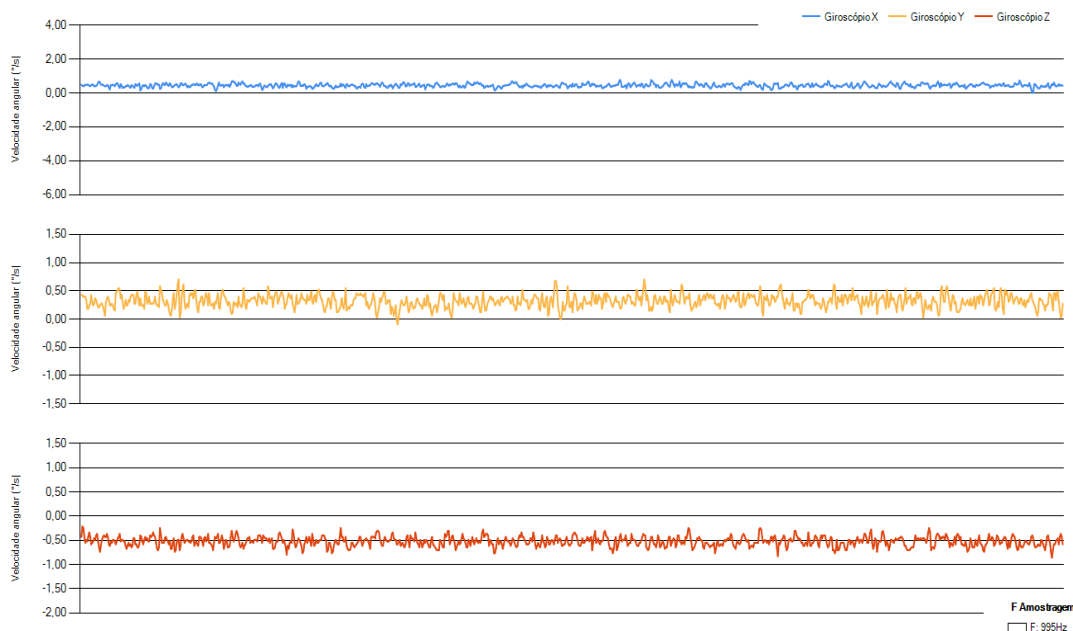


Figura 27 - Métricas dos giroscópios com motores ociosos

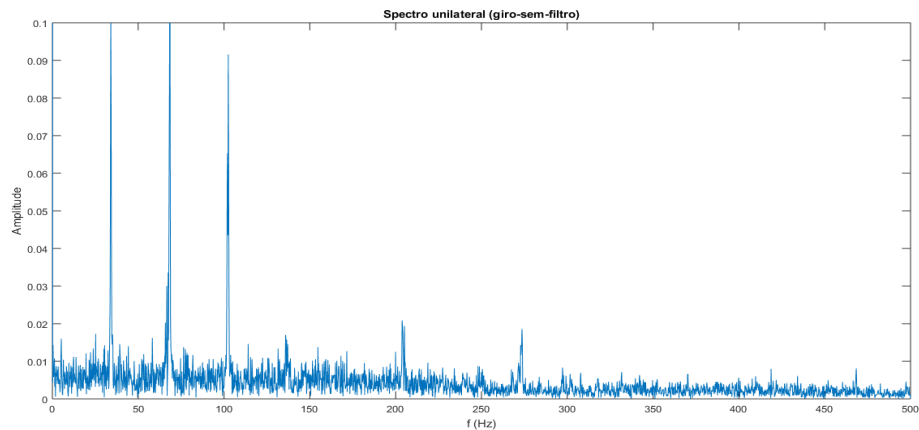


Figura 28 – Espectro unilateral de frequências do giroscópio X para motores ociosos

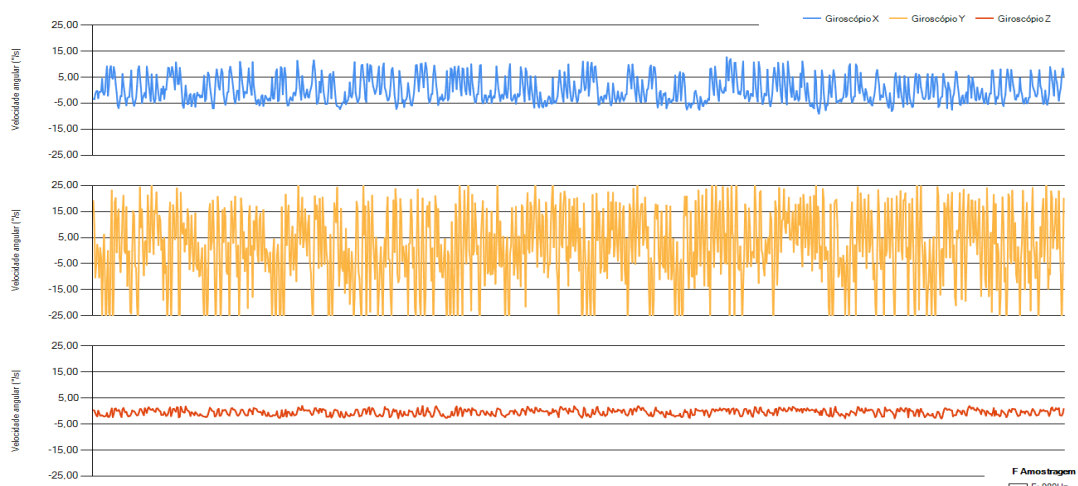


Figura 29 - Métricas dos giroscópios com motores a 50% de potência

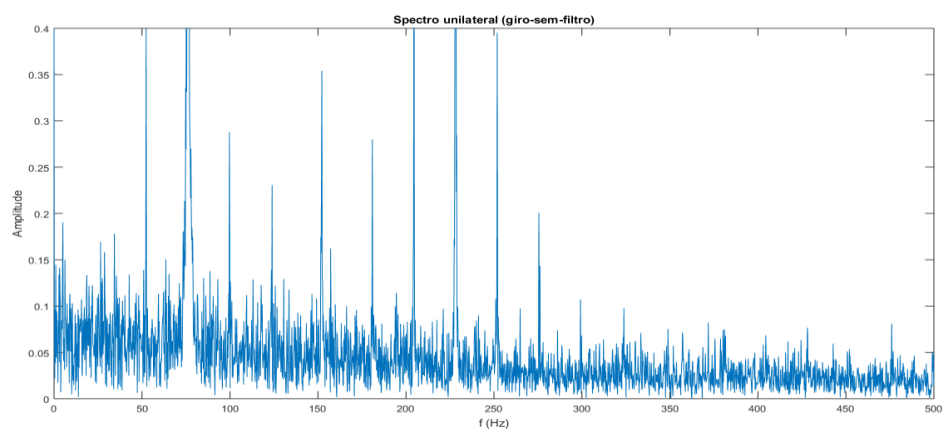


Figura 30 – Espectro unilateral de frequências do giroscópio X para motores a 50% de potência

Assim como nos acelerômetros conclui-se o aumento do ruído e um nível CC em cada sensor. O nível CC também será removido pela calibração inicial. O nível de ruído aumenta significativamente com o aumento da potência dos motores (aumento da vibração mecânica). Diferente dos acelerômetros, os giroscópios são bem mais sensíveis a vibrações, e a faixa de ruído aumenta diferentemente para o eixo ortogonal ao plano **XY**. O giroscópio Z apresenta o menor nível de ruído, e é devido a sua posição, onde grande parte das vibrações emitidas pelos motores são “torques em relação ao centro da estrutura” e, o giroscópio Z faz leituras na “horizontal” (aponte o dedo polegar da mão direita para cima, gire a mão, essa é a leitura captada por ele), logo, são pouco captadas.

O espectro de frequências dos giroscópios foi analisado inicialmente pelo eixo **X**. Da análise das figuras dos espectros, nota-se o aumento acentuado de ruído a partir de das frequências maiores que 100Hz com o aumento da velocidade dos motores. Também o aumento das frequências mais baixas (< 100Hz, vibração), que posteriormente necessitará de um filtro passa-baixos (no código da controladora) para eliminar o ruído e suavizar o sinal do giroscópio. Daqui temos um limiar: Filtrar frequências acima de 100Hz, pois, abaixo disso são justamente os movimentos mecânicos.

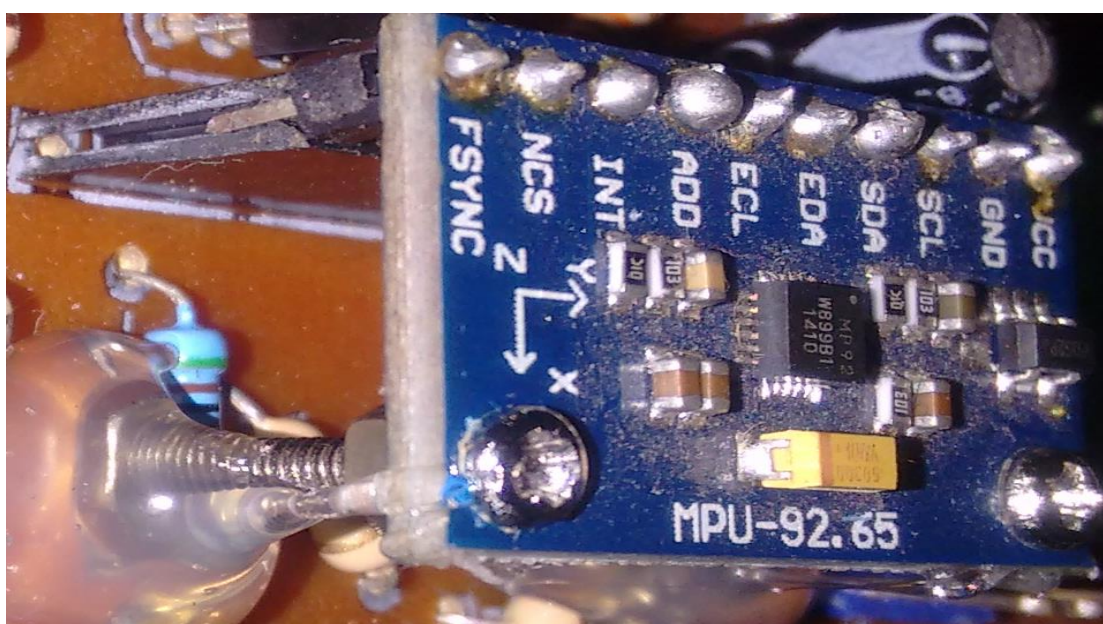


Figura 31 - Posicionamento do módulo IMU na placa de circuito impresso

3.1.3 Magnetômetros

Diferentemente dos outros sensores, a taxa máxima de saída dos magnetômetros é 100Hz. Os valores máximos de leitura são $\pm 4192\mu\text{T}$ (escala de $0.15\mu\text{T}$ por bit), entretanto, a seguir na leitura dos magnetômetros, a densidade de campo magnético está em Miligauss (mG), para facilitar cálculos internos ao microcontrolador ($1\text{Tesla} = 10000\text{Gauss}$, para obter o valor em μT , basta dividir por 10 o valor lido no gráfico).

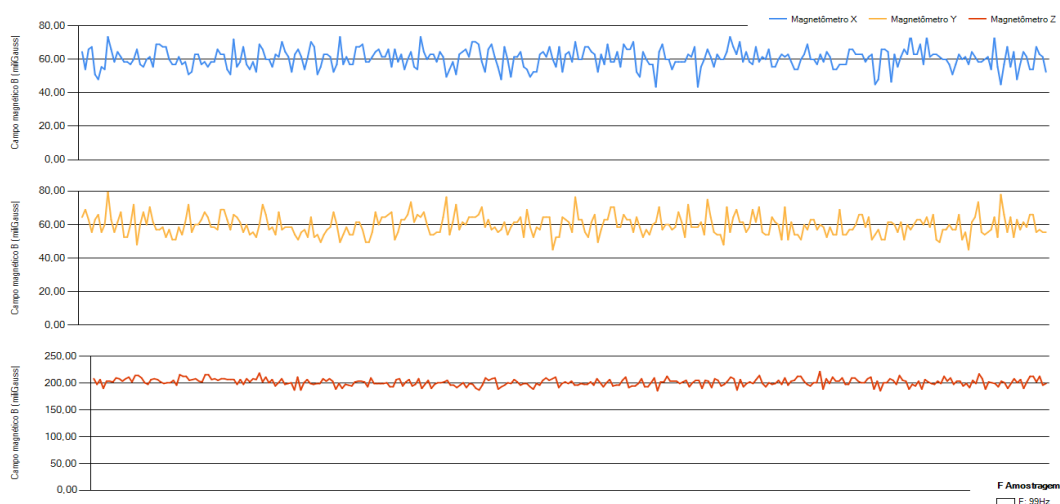


Figura 32 - Leituras dos magnetômetros com motores em estado ocioso

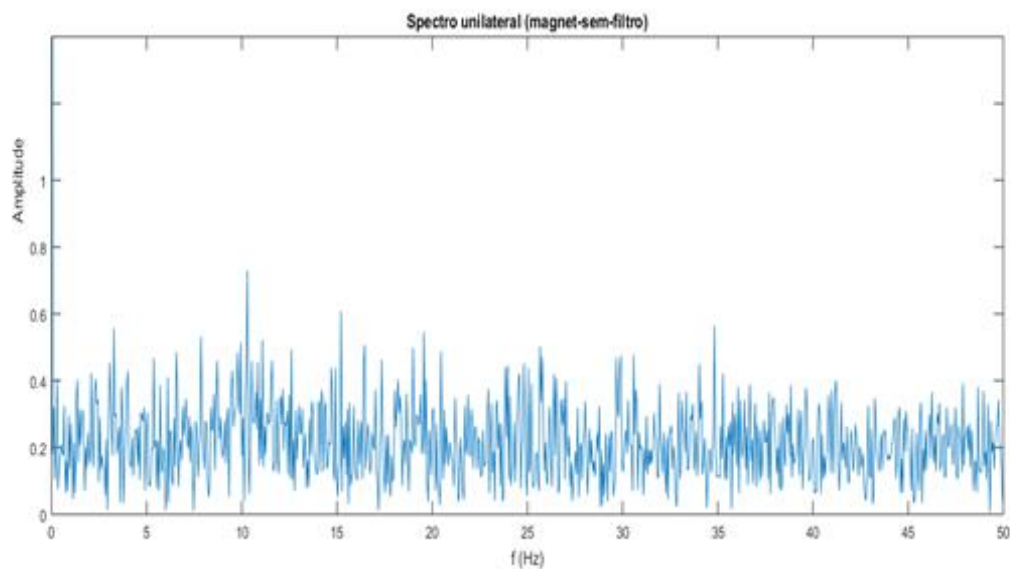


Figura 33 - Espectro de frequências do magnetômetro X (motores ociosos)

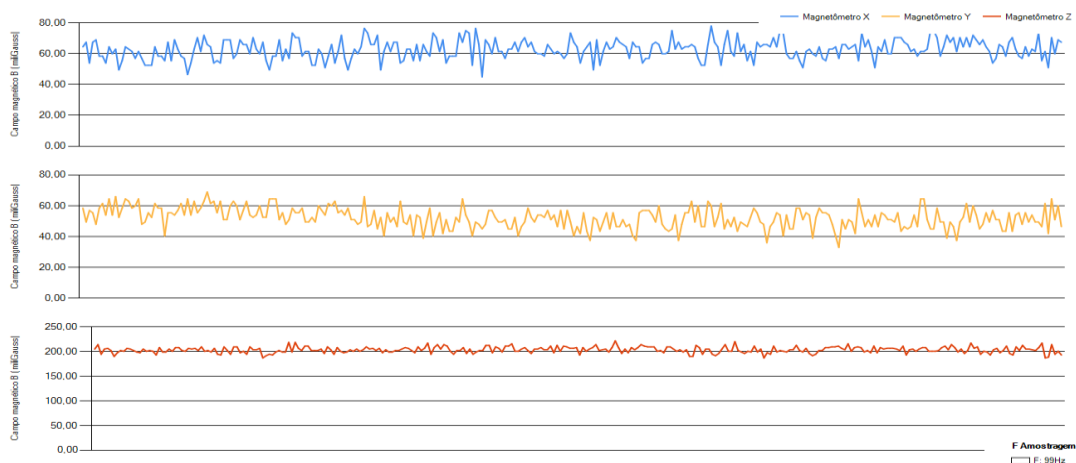


Figura 34 - Leitura dos magnetômetros com motores em 50% de potência

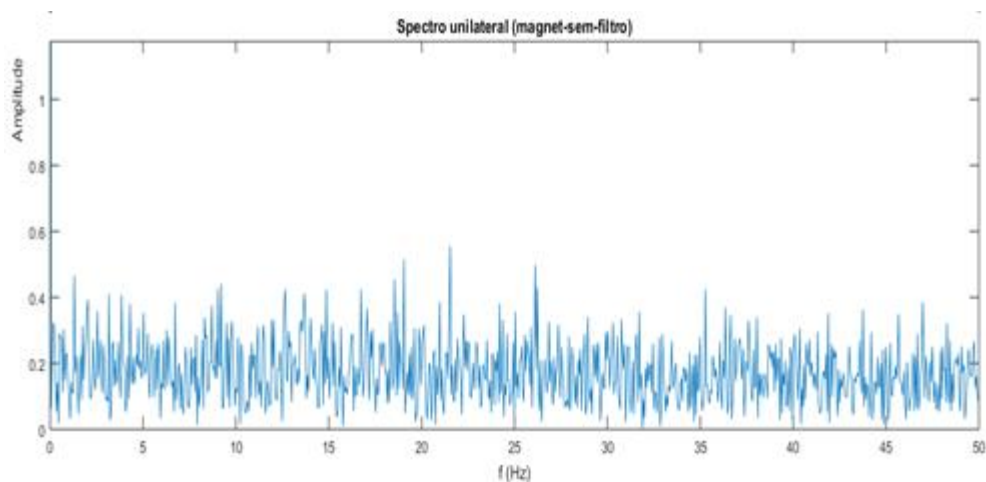


Figura 35 - Espectro de frequências do magnetômetro X com motores a 50% de potência

Observa-se uma pequena suscetibilidade à ruído de vibração dos magnetômetros mesmo com os motores a 50% de potência, não se nota nenhuma alteração considerável. Por esse fato, a controladora não filtrará os dados colhidos dos magnetômetros. A análise de Fourier mostra nos espectros de frequência valores elevados para a frequência 0 que foi cortado dos gráficos para obter uma ampliação e ficar visível todas as contribuições das frequências. Isso quer dizer que existe um *bias* (valor CC) nos valores lidos e por isso, é necessário também uma calibração inicial.

3.2 Filtro digital passa-baixos (Biquadrático IIR)

O filtro digital foi elaborado baseado no filtro analógico butterworth, por ter uma resposta linear nas frequências passantes. A técnica selecionada para o filtro em questão foi a de um filtro digital IIR (resposta ao impulso de duração infinita) pela transformação biquadrática.

O filtro biquadrático passa-baixos foi desenvolvido através da transformação biquadrática de um filtro analógico butterworth com objetivo de suavizar o ruído presente na leitura dos sensores, mais especificamente nos giroscópios.

A transformação biquadrática é uma operação matemática de mapeamento de variáveis. Em filtros digitais é um método padrão de mapear s ou plano analógico para o plano z discreto (Soliman & Srinath, 1998). Transforma filtros analógicos desenvolvidos com técnicas clássicas, em filtros discretos equivalentes.

O mapeamento ocorre por:

$$H(z) = H(s)$$

$$s = 2 * f_s * \frac{z - 1}{z + 1} \quad (13)$$

Onde f_s é a frequência de amostragem.

Mapeando assim o eixo imaginário $j\Omega$ (s) em volta do círculo unitário de $-\frac{f_s}{2}$ até $\frac{f_s}{2}$ (para $z = e^{-j\omega}$ de $-\pi$ até $+\pi$, onde $\omega = 2 * \pi * \frac{f}{f_s}$).

Resolvendo a equação (13) para z :

$$z = \frac{1 + \frac{s}{2 * f_s}}{1 - \frac{s}{2 * f_s}} \quad (14)$$

A equação do filtro butterworth de segunda ordem analógico e polos p_1 e p_2 :

$$H(s) = \frac{1}{(s - p_1)(s - p_2)} \quad (15)$$

A equação (15) possui 2 zeros em ∞ , logo seu mapeamento se dá por $z = -1$. E agora pode ser escrita como:

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} = \frac{(z + 1)(z + 1)}{(z - p_1)(z - p_2)} \\ &= K * \frac{b_0 + b_1 * z^{-1} + b_2 * z^{-2}}{a_0 + a_1 * z^{-1} + a_2 * z^{-2}} \end{aligned} \quad (16)$$

Onde:

- $b_0 = 1$
- $b_1 = 2$
- $b_2 = 1$
- a_0, a_1 e a_2 dependem da frequência de amostragem escolhida e os polos.
- K é uma constante escolhida para o ganho ser unitário, da seguinte forma:

$$K = \frac{\sum a}{\sum b} = \frac{\sum a}{4} \quad (17)$$

Por fim, realizando a transformada inversa Z na equação (16):

$$y[n] = x[n] * B_0 + x[n - 1] * B_1 + x[n - 2] * B_2 - y[n - 1] * A_1 - y[n - 2] * A_2 \quad (18)$$

Os novos termos são:

- $B_0 = \frac{b_0}{a_0} * K$
- $B_1 = \frac{b_1}{a_0} * K$
- $B_2 = \frac{b_2}{a_0} * K$
- $A_1 = \frac{a_1}{a_0}$

- $A_2 = \frac{a_2}{a_0}$

Obs.: Note que a transformada Z inversa foi realizada supondo que o sinal é um impulso, condição dada no início do desenvolvimento.

A equação (18) é a discretização de um filtro passa-baixos butterworth de segunda ordem, onde agora pode ser facilmente implementada em um microcontrolador.

3.3 Controlador PID discreto

O controlador PID (Proporcional, Integral e Derivativo) é parte do sistema de estabilização e controle do quadricóptero. Sua função é alimentar o atuador dos motores para obter o resultado desejado.

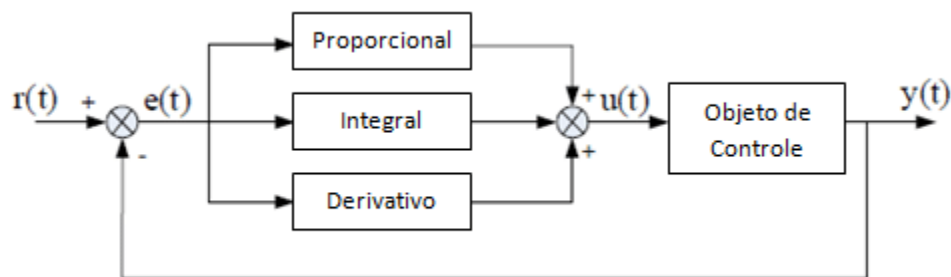


Figura 36 - Estrutura básica de um controlador PID

Da teoria de controle, dado um controlador PID contínuo:

$$y(t) = Kp * e(t) + Ki * \int e(\tau)dt + Kd * \frac{de(t)}{dt} \quad (19)$$

Pode ser representado no domínio da frequência s como:

$$Y(s) = Kp * E(s) + Ki \frac{E(s)}{s} + Kd * E(s) * s$$

$$\frac{Y(s)}{E(s)} = Kp + \frac{Ki}{s} + Kd * s \quad (20)$$

Onde:

- Kp = coeficiente de ação proporcional

- K_i = coeficiente de ação integral
- K_d = coeficiente de ação derivativa
- t = instante do estado do sistema
- $y(t)$ = sinal de saída do controlador no tempo t
- $e(t)$ = sinal de erro, de entrada no controlador no tempo t
- s = variável complexa $\sigma + j\omega$

A expressão (20) pode ser utilizada para programar um computador digital para emular o compensador (Nise, 2013). Para conseguir programar um emulador no computador, é necessário que a função de transferência esteja no domínio discreto (esteja na forma de equação de diferenças). Logo, o uso da transformada Z e sua inversa se faz necessário. Para tanto, utilizando-se das equações (19) e (20), transformação biquadrática, obtém-se:

$$\frac{Y(z)}{E(z)} = Kp + Ki * \frac{\Delta t}{2} * \frac{z + 1}{z - 1} + Kd * \frac{1}{\Delta t} * \frac{z - 1}{z} \quad (21)$$

Reorganizando:

$$\frac{Y(z)}{E(z)} = \frac{b_0 + b_1 * z^{-1} + b_2 z^{-2}}{a_0 + a_1 * z^{-1} + a_2 z^{-2}} \quad (22)$$

Onde:

- $b_0 = Kp + Ki * \frac{\Delta t}{2} + Kd * \frac{1}{\Delta t}$
- $b_1 = -Kp + Ki * \frac{\Delta t}{2} - 2 * Kd * \frac{1}{\Delta t}$
- $b_2 = Kd * \frac{1}{\Delta t}$
- $a_0 = 1$
- $a_1 = -1$
- $a_2 = 0$

Obtendo a equação de diferenças aplicando a transformada inversa Z na eq. (22) (ver (Nise, 2013) tabela 13.1, pág. 1046):

$$Y(z) * (a_0 + a_1 * z^{-1} + a_2 z^{-2}) = E(z) * (b_0 + b_1 * z^{-1} + b_2 z^{-2}) \quad (23)$$

$$y[n] = e[n] * b_0 + e[n - 1] * b_1 + e[n - 2] * b_2 - y[n - 1] * a_1 - y[n - 2] * a_2$$

$$y[n] = e[n] * b_0 + e[n - 1] * b_1 + e[n - 2] * b_2 - y[n - 1] * a_1 \quad (24)$$

Por fim, note que a equação (22)(23) é um caso especial do filtro digital IIR de segunda ordem contendo um polo e dois zeros (Poley, 2003).

A escolha desse método de implementação do controlador PID foi devido seu certo grau de sofisticação (desempenho melhor que o básico) e contorno de problemas de estouro do termo integrativo. Diferentemente do método de implementação comum, onde existe problemas de “*wind-up*” do termo integrativo onde o sistema entra em oscilação permanente, problemas na integral divergindo seu valor, e necessidade de um saturador.

3.4 Filtro de fusão de dados Madgwick & Mahony

Em 2009 Sebastian O.H. Madgwick desenvolveu um algoritmo de fusão de dados AHRS (sistema de referência e direcionamento da posição) e IMU, como parte de sua pesquisa de Ph.D. na Universidade de Bristol (Madgwick & Technologies, 2012).

Em 29/09/2011 Sebastian O.H. Madgwick lançou a versão inicial de um algoritmo de fusão sensorial de 9DOF (nove graus de liberdade), baseado no filtro complementar não linear DCM (matriz de cosseno direto) de Robert Mahony (Baldwin, Mahony, Trumppf, Hamel, & Cheviron, 2007) e (Mahony, Hamel, & Pflimlin, 2008), na forma de quatérnios.

Este algoritmo utiliza leituras de acelerômetros, giroscópios e magnetômetros para obter a rotação do corpo em relação ao referencial fixo terrestre (i.e., utiliza o vetor

aceleração da gravidade como referência). A ideia por trás dele é sua realimentação, sua atualização diferentemente do conhecido filtro de Kalman (preditivo) usa valores do passado como referência. O filtro também lida com níveis CC dos sensores através de um controlador PI (proporcional e integral), a ideia de R. Mahony é substituir os valores lidos por valores providos do controlador PI.

Basicamente, o algoritmo realiza:

- Obtém os dados dos sensores, ω , a , e B , normaliza a aceleração a e campo magnético B .
- Determina o vetor aceleração da gravidade d através da estimativa atual da sua postura (posição).
- Calcula o erro estimado $e = a \times d$
- Aplica o controlador PI para obter a correção da taxa de rotação: $\delta\omega = e * (Kp + Ki * \frac{1}{s})$ onde Kp é a constante de proporcionalidade, Ki é a constante de integração e s é a variável complexa $\sigma + j\omega$ de tempo contínuo.
- Calcula a taxa de rotação corrigida $\omega' = \omega + \omega\delta$
- Integra a taxa de mudança usando ω'
- Repete tudo

Este algoritmo implementado por S. Madgwick pode ser obtido na página (Madgwick & Technologies, 2012), onde é um repositório de código-fonte aberto mantida pela *x-io Technologies*.

Capítulo 4 – Placa de circuito impresso (PCI)

A placa de circuito impresso foi confeccionada em uma placa de fenolite de uma camada (*single layer*) com medidas 7.5cm x 5.5cm. O método de confecção foi a transferência térmica, utilizando uma impressão à laser em papel gloss.

4.1 O esquemático

O esquemático foi desenvolvido no software CAD Proteus 8. Foi desenvolvido a placa mínima necessária, pois, devido ao método de confecção, limita-se a complexidade de uma camada.

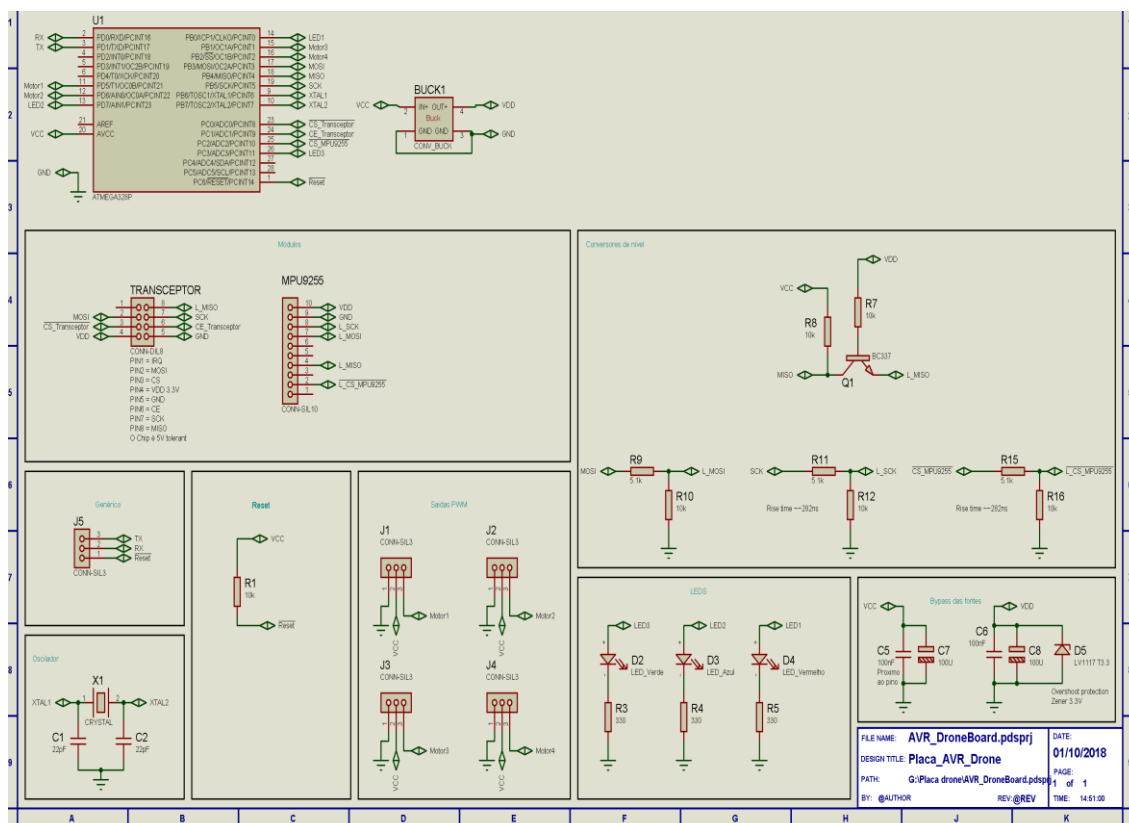


Figura 37 - Esquemático da PCI

4.2 O layout

O layout foi desenvolvido visando o menor número de jumpers e alocação de todos os componentes na placa. Em primeiro plano, o circuito integrado dos sensores

IMU foi colocado de modo a centralizar-se na PCI. Em segundo plano, a redução da distância percorrida pelas trilhas do barramento SPI para evitar o efeito passa-baixos que ocorre com seu aumento. Não foi utilizado a técnica de *power planes* porquê aumentaria a capacitância entre as trilhas, ocasionando uma possível perda adicional na comunicação. De acordo com os *datasheets* do módulo IMU e do transceptor, a capacitância de teste no barramento que gerou uma comunicação sem perdas foram: 400pF para o IMU a 1MHz e 100pF para o transceptor a 5MHz.

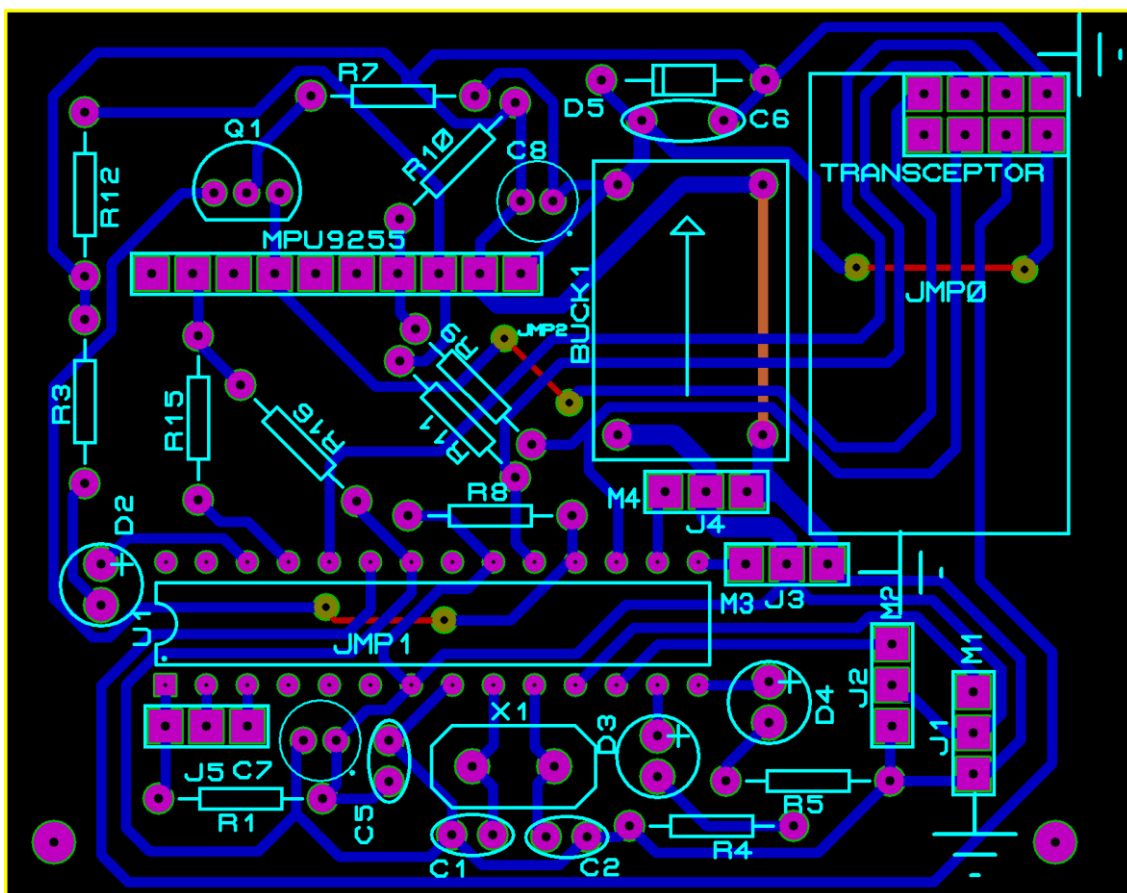


Figura 38 - Layout da PCI

Em vermelho temos os jumpers, em azul o cobre da camada inferior, e o marrom é a conexão *GND* do modelo do conversor buck. Os jacks J1, J2, J3, e J4 são as saídas para as ESCs, o jack J5 é para programar e depurar o microcontrolador. Os dois furos nas extremidades inferiores são para o propósito de alinhamento na confecção.

4.3 A placa confeccionada

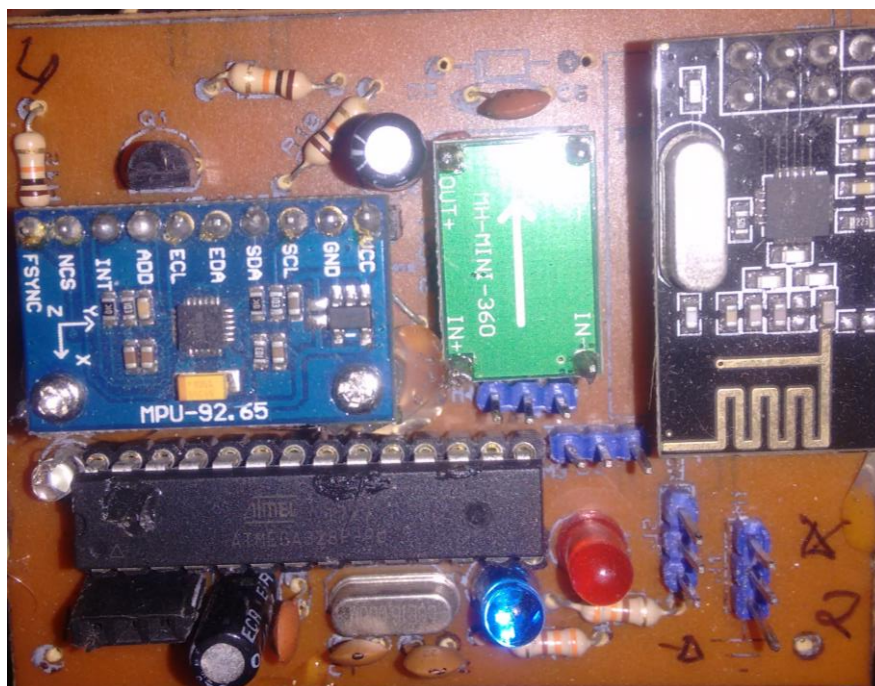


Figura 39 - Placa PCI confeccionada

A fixação da PCI na estrutura foi feita através de cola quente e amortecedores de espuma como mostra a Figura 40.

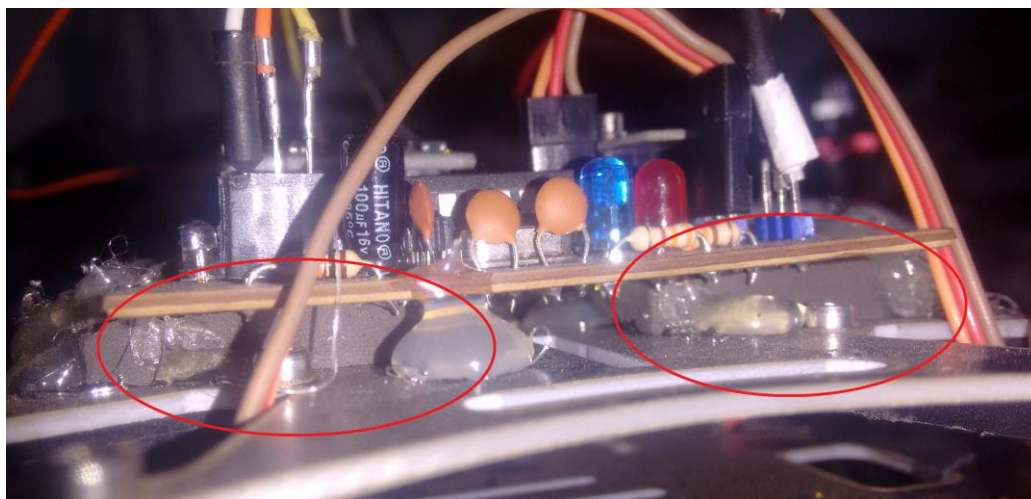


Figura 40 - Fixação da PCI

A justificativa dessa fixação na parte superior é porque não houve espaço para alocação no compartimento inferior da estrutura, devido à altura final da placa.

Capítulo 5 – Software de Telemetria e Comando

O software de telemetria e comando foi desenvolvido no ambiente de desenvolvimento integrado “Microsoft Visual Studio 2017” na linguagem de programação C#, aplicativo *Windows Forms* e estrutura de destino *.NET FRAMEWORK 4.0*. Esse software expõe, de forma gráfica, os dados telemétricos utilizando uma comunicação serial através de uma porta COM (emulada pelo sistema operacional) para se comunicar com uma placa Arduino que, intermedia a comunicação SPI com o transceptor nRF24L01. Dentre os comandos implementados estão: Alteração das constantes dos controladores PID, desligamento do sistema, seleção do modo de estabilização (ligado ou desligado), seleção do modo de voo.

A comunicação é feita por pacotes, dois tipos deles: Comando e Dados. O pacote Comando é enviado pelo usuário à máquina e é constituído de 14 bytes, 7 bytes espelhados como forma de codificação a fim de reduzir a taxa de erro na comunicação. O pacote de Dados é enviado da máquina ao usuário, seu tamanho é de 30 bytes no total e envolve as variáveis de estado, porcentagem da potência solicitada às ESCs, valores de saída do PID e valores das medições dos sensores. O pacote de dados foi implementado utilizando a habilidade da tecnologia *Enhanced ShockBurst™* de enviar um sinal de reconhecimento de recebimento de pacote (*acknowledgement*), onde no envio desse sinal pode-se enviar um pacote de dados com tamanho máximo de 32bytes. Por conter limitações no tamanho desse pacote, e ainda, a quantidade de informação por pacote ser grande (em comparação ao tamanho máximo), não foi implementada nenhuma forma de codificação neste.

Quando um pacote é recebido pelo transceptor, a placa Arduino utilizada como interface serial adiciona uma palavra de sincronismo na frente de cada pacote para não haver dados entrelaçados. Na interface de telemetria o pacote é recebido pelo objeto

SerialPort e é disparado um evento no qual o pacote é desmembrado e armazenado em uma fila (*queue*). Posteriormente, para não haver *cross thread calls* no disparo dos eventos de objetos *Timers*, os dados nessas filas são convertidos (por exemplo dois bytes para inteiro) e inseridos nos objetos *Charts*. Não foi possível a inserção de todos os pacotes de dados recebidos, por alguma razão, o tempo de redesenho do objeto *Chart* é longo demais. Esse tempo acarretaria no aumento da fila de dados, entretanto, existe outro *Timer* que realiza as limpezas, esse *Timer* quando disparado seu evento retira todos os pontos nas posições iniciais dos *Charts* até restar aproximadamente 500 pontos, e ainda realiza a limpeza da fila de dados (basta mudar o ponteiro do início da fila).

No âmbito de controle, utilizando da biblioteca padrão da Microsoft *directInput*, realiza-se a leitura do estado do joystick (controle físico, Figura 41) a cada 15ms e então seus valores são convertidos, empacotados, inserido uma palavra de sincronismo e enviados pela comunicação serial ao transceptor. Os pacotes são enviados a cada 2ms aproximadamente, no entanto, são enviados também pacotes “vazios” com finalidade de não perder o link de comunicação dos transceptores. Existe também os modos de controle por software (por exemplo resposta ao impulso). Nestes modos, o próprio software se encarrega de enviar os pacotes de comando até obter a resposta esperada (Figura 51) ou o usuário cancelar a operação.



Figura 41 - Controle físico utilizado (formato PS2)

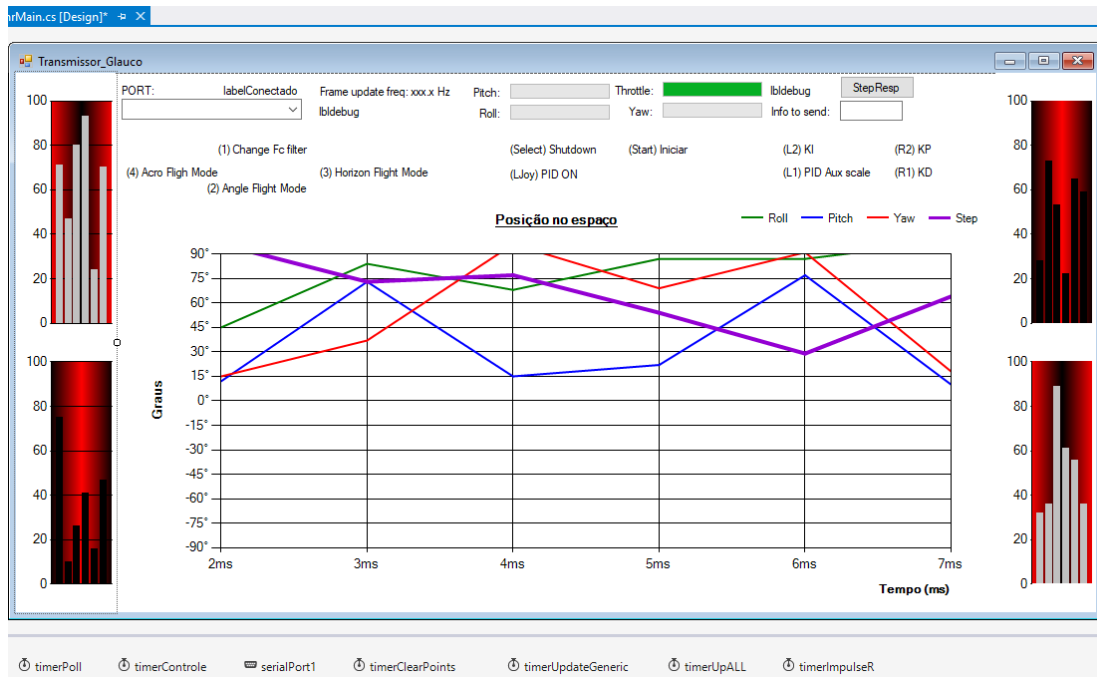


Figura 42 - Forma da classe principal do software de telemetria

Os gráficos em barras nas laterais demonstram o *duty cycle* do PWM enviado aos motores; as barras de progresso na parte superior mostram a posição do joystick; o gráfico ao centro mostra o estado atual do quadricóptero, ou seja, os ângulos euclidianos.

Existe outro objeto *Form* como o da Figura 43, esse *form* por exemplo plota os dados dos giroscópios e do controlador PID. Como a implementação desse *form* foi genérica, abre possibilidade da implementação da transformada de Fourier.

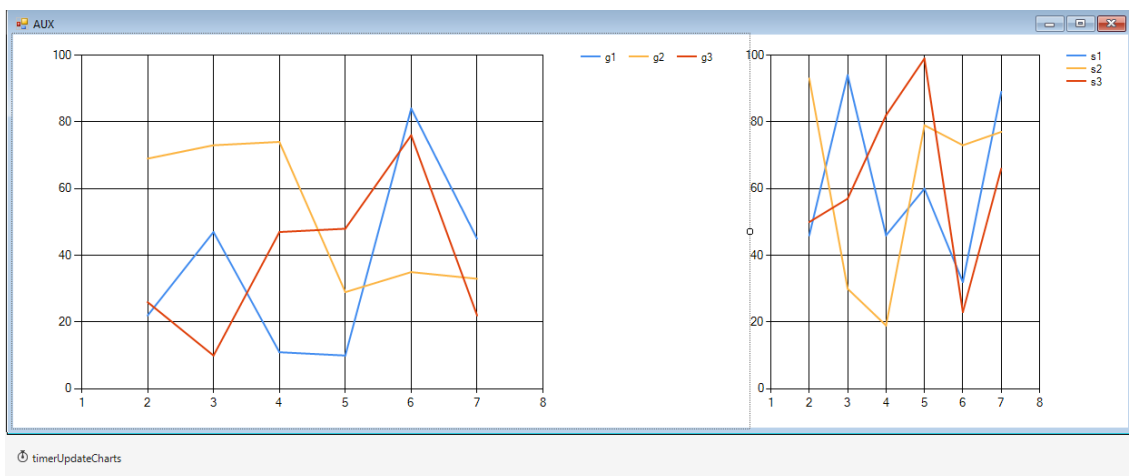


Figura 43 - Form auxiliar para plotar dados

Capítulo 6 – Resultados

Neste capítulo será mostrado os resultados obtidos através da aplicação dos filtros e do controlador PID. Também será foco deste, os resultados em testes de voo.

A controladora, por ter um processamento limitado não se mostrou capaz de atualizar e calcular os dados necessários (i.e., filtrar, cálculo dos ângulos) em taxas altas, portanto foi fixado a uma taxa de atualização de 300Hz.

Devido a modificação da taxa de atualização, os filtros foram calculados para uma frequência de amostragem (f_s) de 300Hz e frequência de corte (f_c) de 30Hz. Esses valores foram escolhidos pois, na aquisição de dados no Capítulo 3 – Análise dos sinais e filtros a taxa de amostragem era 1kHz e o limiar verificado foi 100Hz, entretanto, como a taxa de amostragem reduziu para $\frac{3}{10}$ do valor inicial, a frequência de corte também foi reduzida na mesma proporção.

Já para os magnetômetros, como a taxa máxima de amostragem é 100Hz, utilizouse um filtro com frequência de corte em 4Hz somente para ilustração. O filtro foi implementado, mas não utilizado nos magnetômetros.

O software para utilizado para análise foi o MATLAB, e para coleta dos dados foi utilizado o de telemetria.

6.1 Efeito após aplicação do filtro digital IIR passa-baixos

Na **Figura 44**, nota-se o efeito do filtro passa-baixos aplicado reduzindo as altas frequências do sinal amostrado, e conseqüentemente deixando-o suave para então ser processado pelo controlador PID. Várias medições e seus espectros estão no **Apêndice B – Métricas dos sensores e seus espectros de Fourier**.

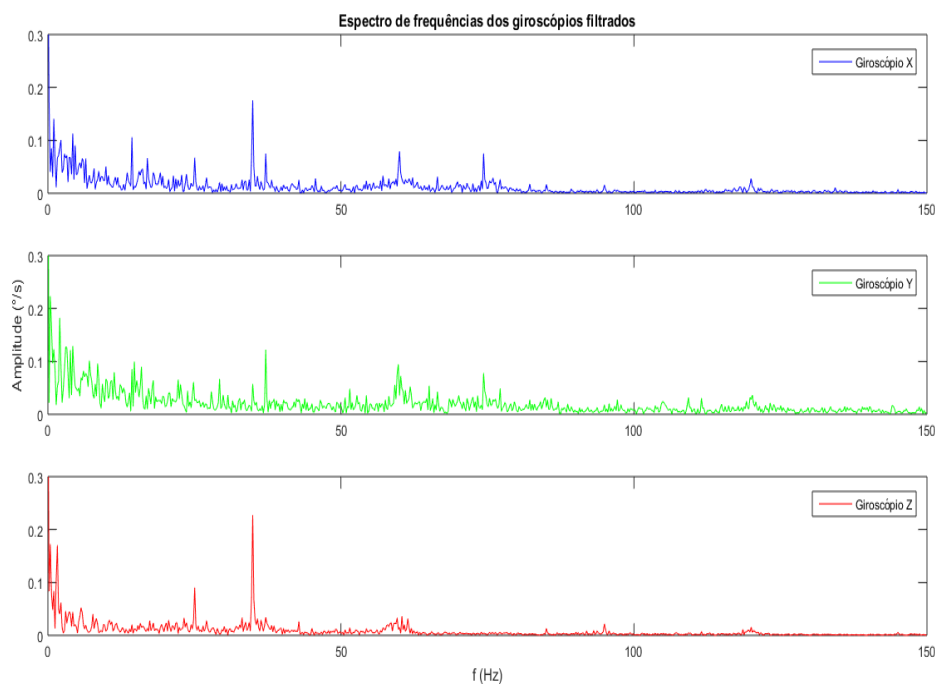


Figura 44 - Espectro de frequências dos giroscópios filtrados, motores a 15% de potência

Para demonstrar por meio de números, as tabelas abaixo sintetizam o efeito do filtro aplicado. O valor obtido nas tabelas refere-se ao valor RMS do espectro (ou ainda, o ruído presente na coleção) retirando o valor da frequência 0 (valor CC da coleção).

Potência dos motores	Ruído (RMS) dos acelerômetros (m/s^2)			
	0%	5%	20%	50%
<i>X bruto</i>	4,34E-05	3,15E-03	9,21E-03	6,52E-02
<i>X filtrado</i>	5,02E-05	9,65E-04	2,29E-03	2,54E-02
<i>Y bruto</i>	4,46E-05	2,34E-03	6,77E-03	3,72E-02
<i>Y filtrado</i>	4,40E-05	9,61E-04	1,72E-03	1,49E-02
<i>Z bruto</i>	6,55E-05	1,58E-03	6,33E-03	3,64E-02
<i>Z filtrado</i>	5,65E-04	1,99E-03	2,75E-03	1,31E-02

Tabela 1 - Ruído (RMS) dos acelerômetros

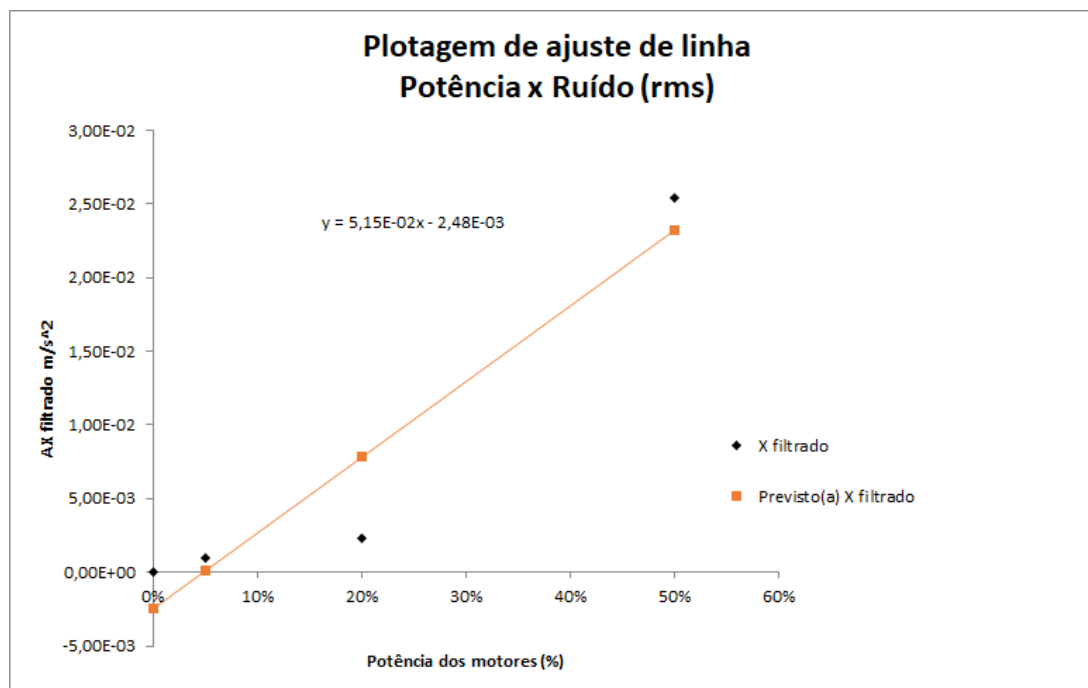


Figura 45 - Ajuste de linha para o sinal do acelerômetro X filtrado.

O ruído apresentado pelo acelerômetro com motores ligados à 60% de potência foi reduzido na faixa de $> 50\%$, entretanto, note as células em destaque, como o acelerômetro está configurado para uma resolução de $\pm 8g$ e o registrador de 16 bits é um inteiro com sinal, seu valor máximo é 32768, logo $8 * 9,81 / 32768 = 2,40E-03$ que corresponde ao limiar da escala de sensibilidade nessa resolução. Ruídos dessa magnitude abaixo do limiar são de pouca contribuição para o sinal final.

Potência dos motores ->	Ruído (RMS) dos giroscópios ($^{\circ}/s$)					
	0%	10%	14%	25%	41%	50%
<i>X bruto</i>	4,49E-03	7,66E-03	7,69E-03	5,83E-02	1,35E-01	8,33E-01
<i>X filtrado</i>	2,00E-03	4,46E-03	4,95E-03	2,16E-02	1,37E-01	1,65E-01
<i>Y bruto</i>	4,99E-03	2,71E-02	3,44E-02	4,99E-02	4,57E-01	4,07E-01
<i>Y filtrado</i>	2,27E-03	1,22E-02	1,45E-02	3,01E-02	2,76E-01	1,40E-01
<i>Z bruto</i>	4,09E-03	4,75E-03	5,92E-03	4,22E-02	2,96E-01	2,43E-01
<i>Z filtrado</i>	2,03E-03	3,02E-03	3,43E-03	1,83E-02	1,29E-01	1,80E-01

Tabela 2 - Ruído (RMS) dos giroscópios

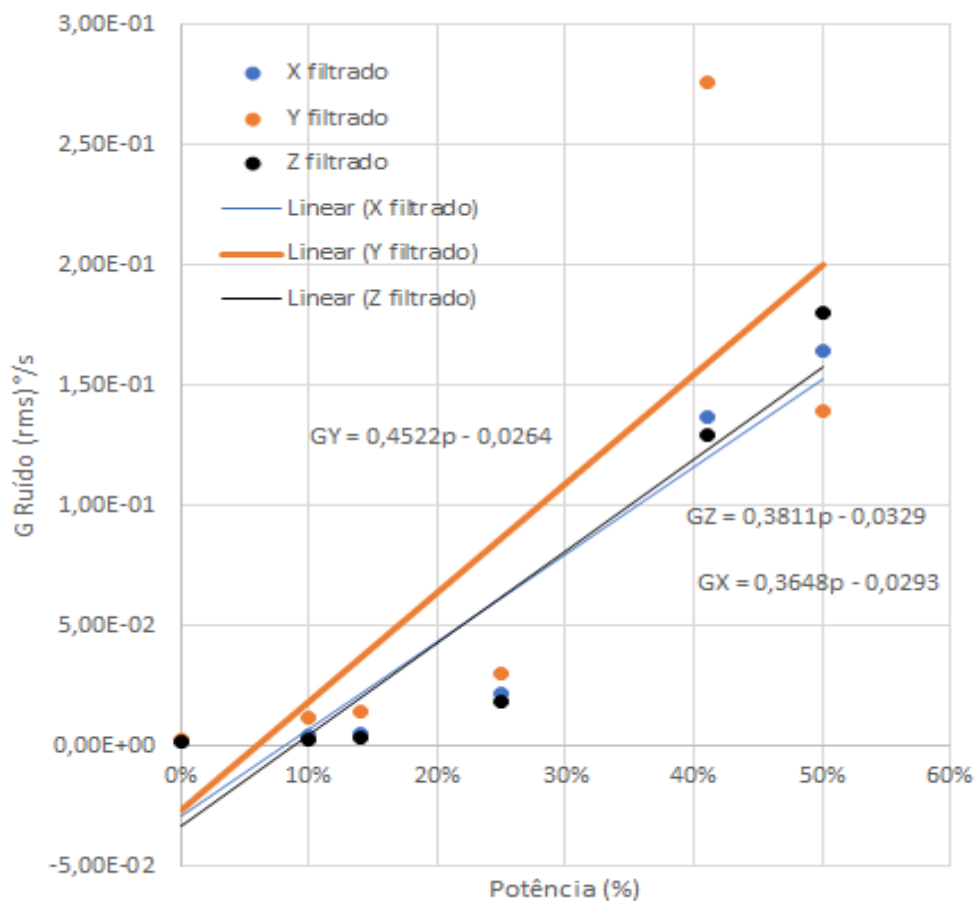


Figura 46 - Plotagem do ajuste de linha para os três eixos, sinal filtrado.

Os giroscópios sofreram uma redução no ruído (rms) da ordem $> 25\%$ após a aplicação do filtro, menor que as dos acelerômetros, ressaltando ainda mais a necessidade de filtra-los por serem mais sensíveis. O limiar da escala de sensibilidade pode ser calculado por: $1000/32768 = 3,05E-02$. Estão em destaque na Tabela 2 as magnitudes inferiores à escala de sensibilidade do giroscópio.

6.2 Respostas obtidas

Foram realizados testes controlados, isto é, foi fixado um eixo do drone por meio de cordões (Figura 47) para testar as respostas dos controladores PID um por vez. Cada controlador PID (o primeiro de base angular e segundo com base a velocidade angular) teve suas constantes de ganho ajustadas manualmente, começando pela proporcional, seguindo a integradora, e pôr fim a derivativa.



Figura 47 - Fixação do drone para realização de testes e calibrações

6.2.1 PID da posição angular

A seguir, respostas do modo de voo com base na posição angular somente.

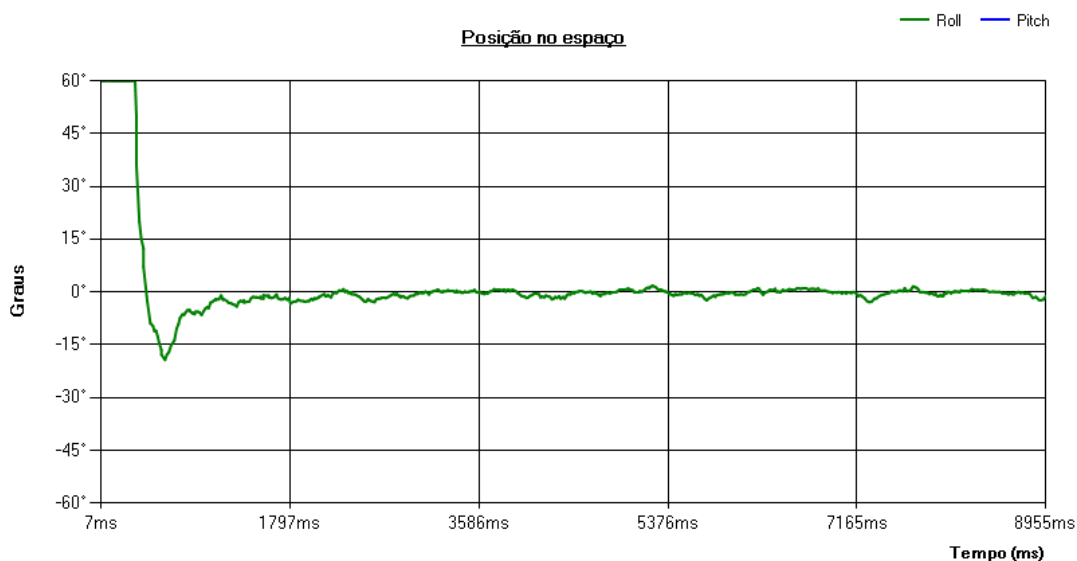


Figura 48 – Resposta ao degrau com ganhos ajustados

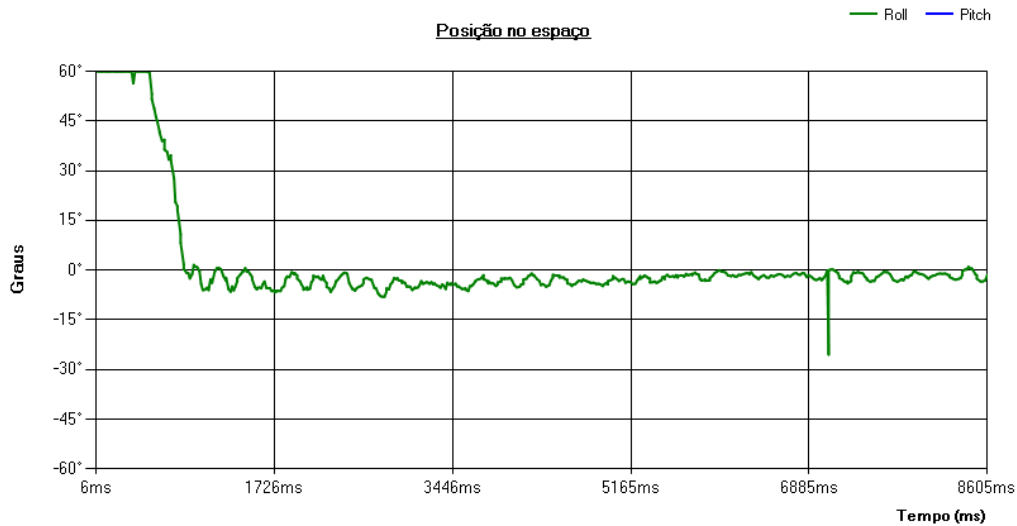


Figura 49 - Resposta ao degrau com constante de integração pequena (K_i)

- Os picos apresentados nas figuras são devido a taxa de erro de bit do transceptor (0,26%), mesmo após a codificação de espelhamento. Especificamente no canal USART transceptor → computador.

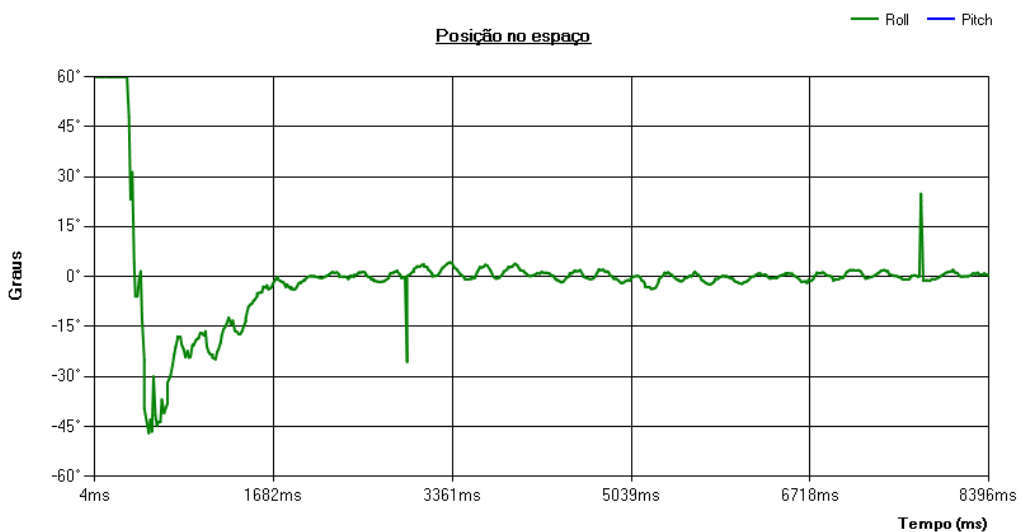


Figura 50 - Resposta ao degrau com constante de proporcionalidade alta (K_p)

Todo o processo de ajuste não foi documentado pois, a resposta com diferentes constantes de ganho não ficaram evidentes graficamente.

6.2.2 PID da velocidade angular

Esse PID tem como base a velocidade angular do corpo (giroscópios). Ele é responsável pela estabilidade giratória do corpo. A resposta ao degrau aplicada foi da seguinte forma: o software de controle envia um sinal para o PID girar no sentido anti-horário, e quando na resposta obtiver um valor acima de 40° o software de controle envia outro sinal para girar no sentido horário até 0° , pausando por 1000ms (na posição de 0°), e então realizando o mesmo movimento no outro sentido (horário, anti-horário).



Figura 51 - Resposta ao degrau do drone com controlador PID angular

6.3 Divergência do filtro passa-baixos

Devido a limitação de cálculos de ponto flutuantes (precisão finita) do microcontrolador, quando a frequência de corte selecionada para o filtro é alta suficiente o filtro apresenta dois comportamentos inesperados: após certo tempo (aleatório) cresce sem limite até estourar o limite do tipo da variável de alocação; O valor fica constantemente nulo como na Figura 52 - Divergência na saída do filtro digital passa-baixos Figura 52. A saída do PID, está convertida e escalada para introdução aos motores, e mostra o efeito desta divergência.

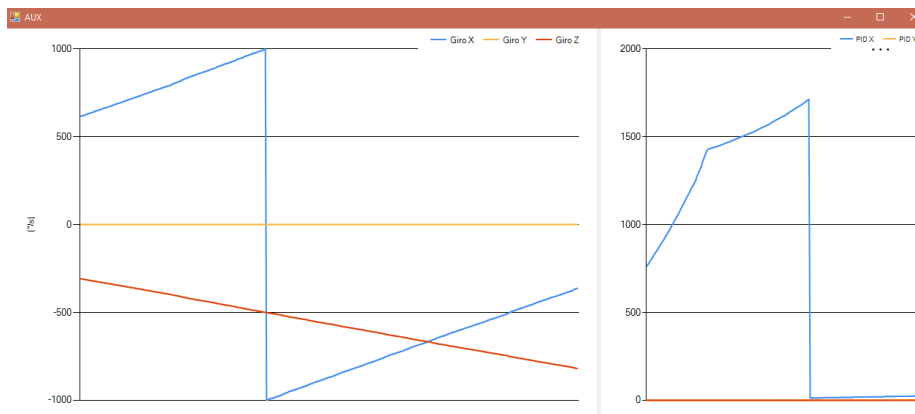


Figura 52 - Divergência na saída do filtro digital passa-baixos

6.4 Custo temporal das funções principais

A seguir os custos temporais médios de cada função principal, essa medição foi realizada utilizando o *timer1* no modo clock sem *prescaler* (x1) para contar os ciclos de máquina necessários para entrar e sair da função. Posteriormente este valor foi dividido por 16 e assim obtém-se o tempo de execução em microssegundos.

- Aquisição de dados dos sensores: $66\mu s$
- Filtro digital passa-baixos: $297\mu s$
- Filtro Madgwick-Mahony: $2055\mu s$
- Cada controlador PID: $244\mu s$
- Envio de telemetria: $133\mu s$

Existe uma reserva de tempo em cada laço de aproximadamente $200\mu s$ caso haja a necessidade de executar algum comando requerido pelo usuário, sem interferir na taxa de amostragem.

Conclusões

O sistema montado foi capaz de realizar o propósito (estabilidade), embora não satisfatoriamente (lento, mesmo após várias combinações de constantes de ganho) devido suas limitações. O quadricóptero **não** foi capaz de voar livremente pois, sua resposta a estabilidade está muito lenta.

O problema de maior peso encontrado foi a divergência do filtro passa-baixos, por ser realizado em ponto flutuante emulado (precisão simples, 7 dígitos após casa decimal) não há precisão suficiente. Para contornar a situação, a frequência de corte foi reduzida, levando a um atraso no sinal e posterior “enrijecimento” do sistema, em outras palavras, ficou mais “rígido” o controle do drone, e ainda aritmética de ponto fixo, uma vez que se conhece os valores máximos e mínimos de saída.

O filtro *Madgwick-Mahony* respondeu muito bem, foi alimentado pelas leituras filtradas dos giroscópios e pela leitura pura dos acelerômetros e magnetômetro. O ruído quase não se mostra em sua saída.

A interface de telemetria mostrou-se adequada para uma posterior implementação da transformada de Fourier em tempo real, pois consegue transmitir os dados de telemetria a cada quadro de atualização (máquina → usuário). Possui uma taxa de erro de bits na comunicação aérea de 0,26%, a alta velocidade definida de 2Mbaud/s contribui para isso.

O uso de componentes genéricos como as ESCs e os motores mostrou ser plausível do uso, entretanto, componentes específicos para essa aplicação podem apresentar desempenho superior. Os motores são diferentes entre si, cada um possui certa rigidez facilmente detectada pela rotação manual com a mão. As ESCs possuem limitações nas frequências de saída e somente dois modos de sinal de entrada, atribuindo um *delay* de 1ms à cada novo comando.

O controlador PID demonstrou bom desempenho, apesar da parcela derivativa inserir ruído no sinal de controle, é necessário somente reduzir a constante derivativa K_d . Assim como o problema mencionado da divergência do filtro passa-baixos, o controlador PID também é suscetível à tal problema, embora em momento algum tenha apresentado esta falha.

O microcontrolador ATMEGA 328-PB foi capaz de realizar todo o processamento, entretanto não satisfatoriamente. Seu maior ponto fraco é o tamanho dos registradores, apenas 8 bits, inviabilizando cálculos de alta precisão. O clock de 16Mhz não demonstrou ser um ponto fraco, pois mesmo com emulações por software de tipos maiores (*float*, *long*, *int*), o mesmo foi capaz de atingir uma taxa de atualização superior à 300Hz.

Por fim, o projeto demonstrou ser uma oportunidade única de aprendizado, multidisciplinar para um projeto de engenharia, envolvendo teoria de controle, sistemas digitais, processamento de sinais, circuitos elétricos, eletrônica aplicada, programação em diferentes linguagens (C, C++, C#), telecomunicações e microeletrônica. Existe muitas áreas de aplicação para o projeto proposto e isso foi um dos motivos da escolha do mesmo, o outro motivo foi a solidificação de conhecimentos adquiridos na graduação.

Possíveis melhorias

Neste projeto, devido a taxa de processamento do microcontrolador, pode-se futuramente troca-lo por outro de maior poder, mais especificamente, um microcontrolador de 32-bits ou mais com FPU, que por padrão possui clock maior, maior número de instruções, e conseqüentemente maior taxa de processamento. Assim sendo, o horizonte mostra a possibilidade de rodar o algoritmo em uma taxa de atualização superior, possivelmente até a taxa máxima de amostragem dos giroscópios (32kHz), um aumento de aproximadamente 65 vezes na taxa de atualização. Existe também a possibilidade de otimizar o código, multiplexando o cálculo dos ângulos euclidianos (filtro Mahony), já que essa função demanda muito tempo, dividi-la em partes pode aumentar a performance, sem comprometer o sistema, pois as leituras base dessa rotina são as dos acelerômetros, e elas não são filtradas, logo não é necessário uma taxa fixa de amostragem.

Outra possibilidade é a incorporação de materiais amortecedores como fitas de silicone plásticas autoadesivas na base dos motores, uma vez que são responsáveis pelas vibrações inseridas no drone, reduzindo-as.

A questão das hélices, podem ser trocadas por hélices de fibra de carbono mais leves e resistentes, e claro, realizar seu balanceamento.

No que diz respeito a sensores, a incorporação de medidores barométricos (de pressão) para estabilização da altura, sensores ultrassônicos para complementar a medição de altura. Ainda, como finalidade principal dos sensores ultrassônicos tem-se o desvio e prevenção de choque contra objetos. A inclusão de sensores de distância por luz infravermelha, pode complementar o futuro sistema de desvio dos sensores ultrassônicos e o futuro sistema de estabilização de altitude devido seu alto alcance. Um sensor de temperatura para realizar a compensação térmica instruída pelos fabricantes.

A incorporação de sensores de precipitação e presença, podem ser aliados em sistemas de defesa (ou autopreservação), pois quando há chuva ou uma presença detectada, o sistema encarregado pode tomar atitudes defensivas como por exemplo: descer ao chão e desligar os motores, subir a uma altura em que deixe de ser detectado presença.

A inserção de um módulo GPS tem como possível melhoramento, aliado na inserção de um sistema de *failsafe*, caso perca sinal de rádio, o drone voa a uma altitude suficiente e retorna ao ponto inicial quando foi ligado. Outras melhorias seriam: o voo monitorado e pedido de socorro (caso a bateria atinja um limiar, o drone pousa, e fica emitindo o sinal da sua localização).

Por fim, a inclusão de um rádio transceptor de maior potência faz-se necessária devido ao pequeno alcance obtido com o módulo presente nesse projeto (nRFL2401).

Referências

- AsahiKASEI. (Outubro de 2013). Fonte: <Acesso em 2018>:
<https://www.akm.com/akm/en/file/datasheet/AK8963C.pdf>
- Baldwin, G., Mahony, R., Trumppf, J., Hamel, T., & Cheviron, T. (July de 2007).
 Complementary filter design on the Special Euclidean group SE(3). *2007 European Control Conference (ECC)*.
- Biasi, S. C., & Gattass, M. (25 de Fevereiro de 2002). *PUC-RJ*. Fonte: <Acesso em 2018>: <https://webserver2.tecgraf.puc-rio.br/~mgattass/Quaternios.pdf>
- BLHELI User Manual. (s.d.). Fonte: YIN YAN MODEL TECH. MFT:
https://img.banggood.com/images/upload/2014/12/Emax%20BLHeli%20ESC%20Manual.pdf?admitad_uid=debd661f8ac649d6d730aa8915e03be9
- Brown Ward; Microchip Technology Inc. (01 de Agosto de 2002). *Application Notes*.
 Fonte: <Acesso em 2018>:
<http://ww1.microchip.com/downloads/en/appnotes/00857a.pdf>
- Davenport, W. B., & Root, W. L. (1987). *An Introduction to the Theory of Random Signals and Noise*. New York: IEEE Press.
- Halliday, D., Resnick, R., & Walker, J. (2008). *Fundamentos de física, volume 1*. Rio de Janeiro: LTC.
- InvenSense Store. (2018). <Acesso em 2018>. Fonte: Loja virtual:
<https://store.invensense.com/Products/Detail/MPU9255-TDK-InvenSense/520231/>
- InvenSense, D.S. (Setembro de 2014). *Numero do documento: DS-000007 Rev. 1.0*.
 Fonte: <Acesso em 2018>:
<https://store.invensense.com/datasheets/invensense/PS-MPU-9255.pdf>

- InvenSense, R.M. (Janeiro de 2015). *Document Number: RM-MPU-9250A-00. Revision: 1.6.* Fonte: <Acesso em 2018>: <http://www.invensense.com/wp-content/uploads/2017/11/RM-MPU-9250A-00-v1.6.pdf>
- Madgwick, S., & Technologies, x.-i. (31 de July de 2012). *Open source IMU and AHRS algorithms.* Fonte: <http://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>
- Mahony, R., Hamel, T., & Pflimlin, J.-M. (26 de August de 2008). Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control* (Volume: 53 , Issue: 5 , June 2008), pp. 1203 - 1218.
- McGowen, S. S. (2005). *Helicopters: An Illustrated History of Their Impact.* ABC-CLIO.
- Mersereau, R. M., & Smith, M. J. (1993). *Digital Filtering, A Computer Laboratory TextBook.* GeorgiaTech.
- Microchip, Atmega 328/P Datasheet. (Nov de 2016). Fonte: <Acesso em 2018>: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
- Nise, N. S. (2013). *Engenharia de Sistemas de Controle* (Vol. 6 ed.). Rio de Janeiro: LTC.
- Parks, W., T., & Burrus, S. (1987). *Digital Filter Design.* New York: John Wiley & Sons.
Fonte: <https://www.mathworks.com/help/signal/ref/bilinear.html>
- Poley, R. (2003). *Introduction to digital control using digital signal processors.*
- Soliman, S. S., & Srinath, M. D. (1998). *Continuous and Discrete Signals and Systems.* New Jersey: Prentice Hall Inc.
- Weeks, M. (2007). *Digital Signal Processing Using MATLAB and Wavelets.* Hingham, Massachusetts: Infinity Science Press LLC.

Apêndice B – Métricas dos sensores e seus espectros de Fourier

Aqui foram realizadas com auxílio do software *MATLAB* análises dos espectros de Fourier dos sinais lidos e sinais filtrados. As figuras que seguem, contém ambas as leituras e espectro em frequência.

A taxa de amostragem foi fixada em 300Hz no microcontrolador (um firmware de coleta, filtragem e, controle de motores desenvolvido para aquisição rápida de dados), e então colhida 1500 amostras (aproximadamente 5 segundos) de leituras dos sensores, e posteriormente salva em um arquivo *.log* (similar a *.txt*). Após o colhimento dos dados, a função do *MATLAB* utilizada para encontrar o espectro de Fourier foi *FFT*, entretanto, essa função retorna o espectro bilateral no qual foi devidamente tratado e separado sua parte positiva de frequências.

B.1 Magnetômetro X

Por razões ilustrativas, a medição por 18 segundos da saída do magnetômetro **X** e seu respectivo sinal filtrado. A razão por traz de um único eixo medido é, o magnetômetro por sua resolução ser alta ($\pm 4912\mu T$), é praticamente imune ao ruído de vibração da estrutura, mas sofrem com o efeito *hard-iron* no qual ao aproximar-se um metal próximo do sensor, altera suas medidas pela interação do campo magnético com o metal. Pelo fato dos motores estarem fixos, não há aproximação de metal no sensor.

A amplitude do eixo **Y** está em Miligauss.

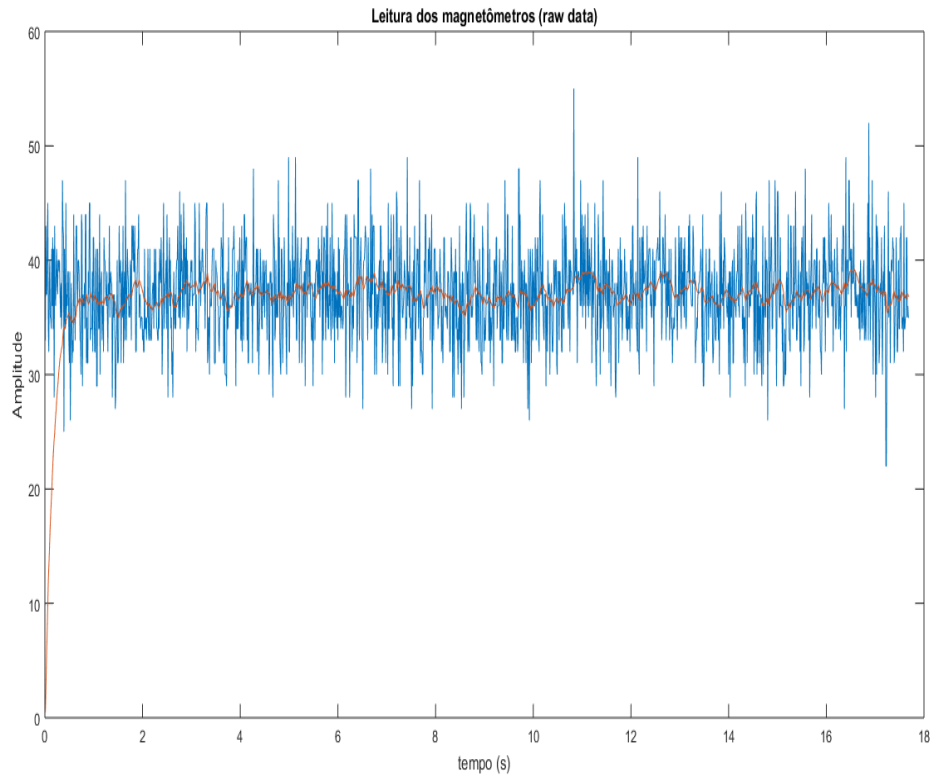


Figura 54 - Leitura do magnetômetro X. Em azul, dados brutos, em vermelho, dados filtrados. O eixo Y está em Miligauss.

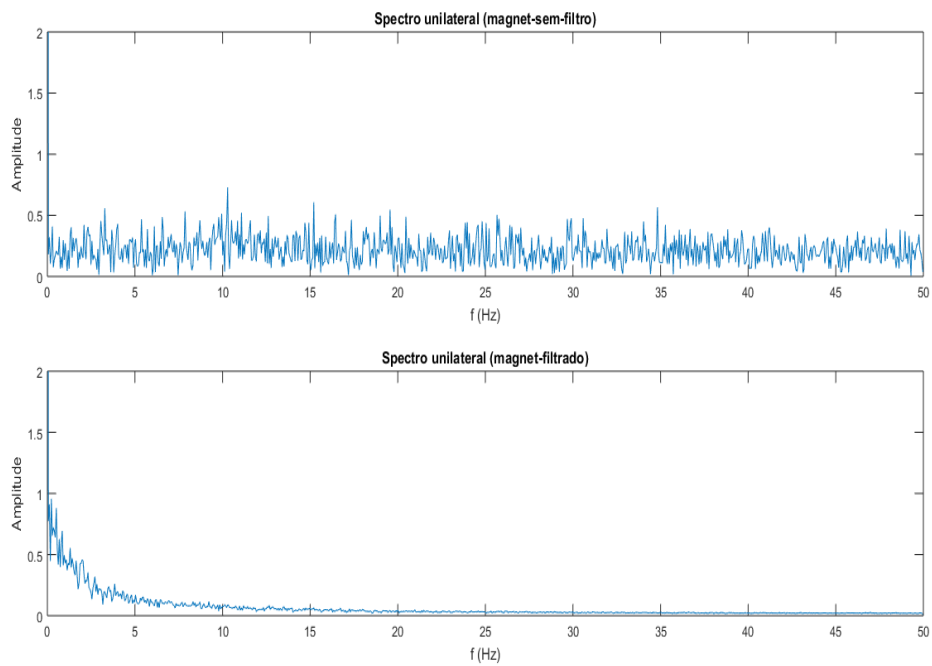


Figura 55 - Espectro de frequências do magnetômetro X, antes e após filtragem.

B.2 Giroscópios

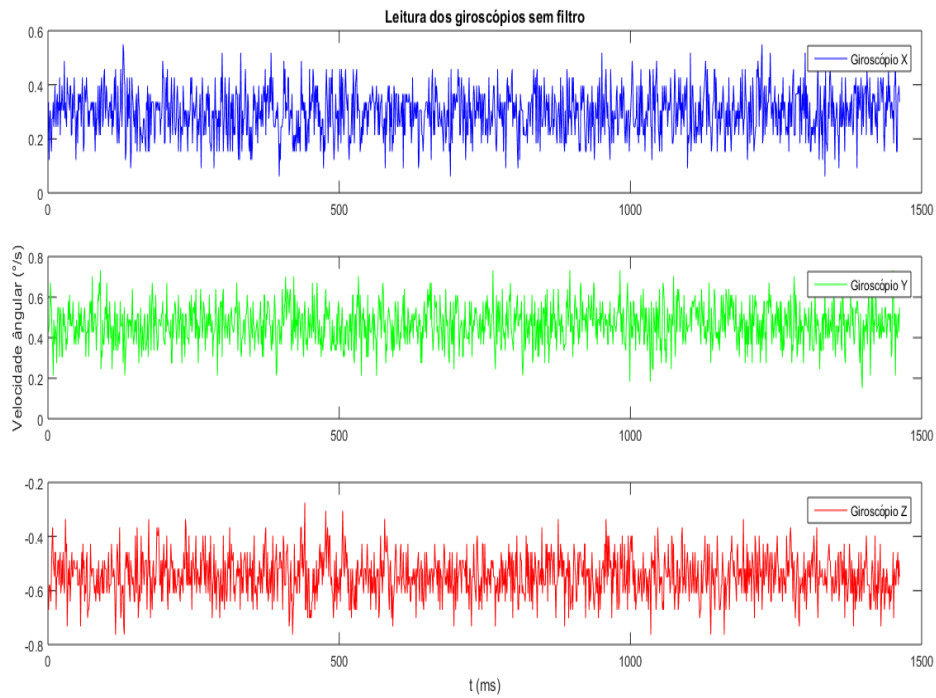


Figura 56 - Leitura dos giroscópios com motores ociosos sem filtro

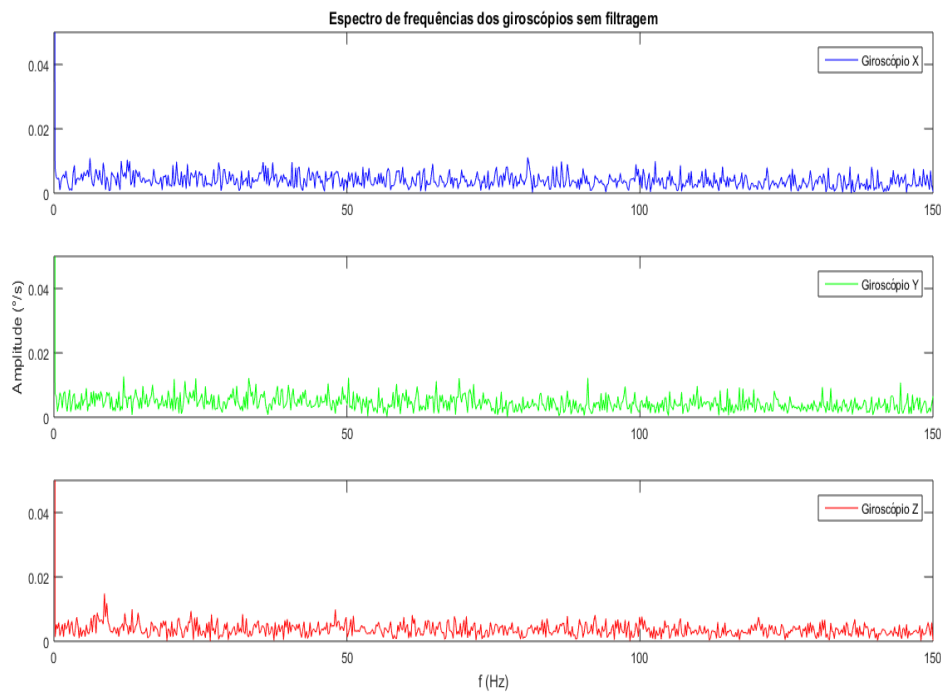


Figura 57 - Espectro de frequências dos giroscópios com motores ociosos sem filtro

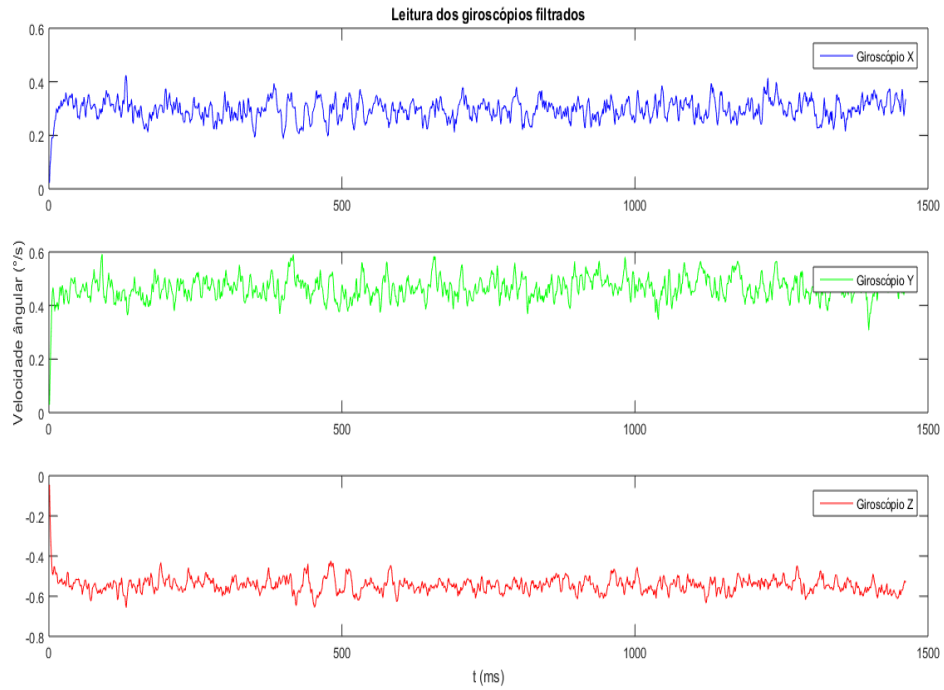


Figura 58 - Leitura filtrada dos giroscópios, motores ociosos

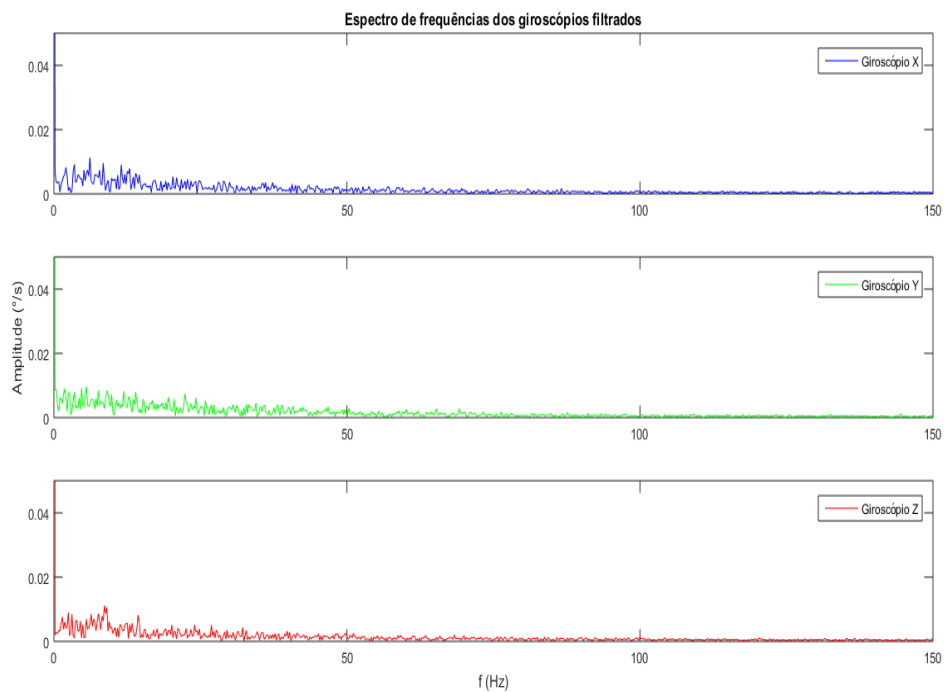


Figura 59 - Espectro de frequências dos giroscópios filtrados, motores ociosos

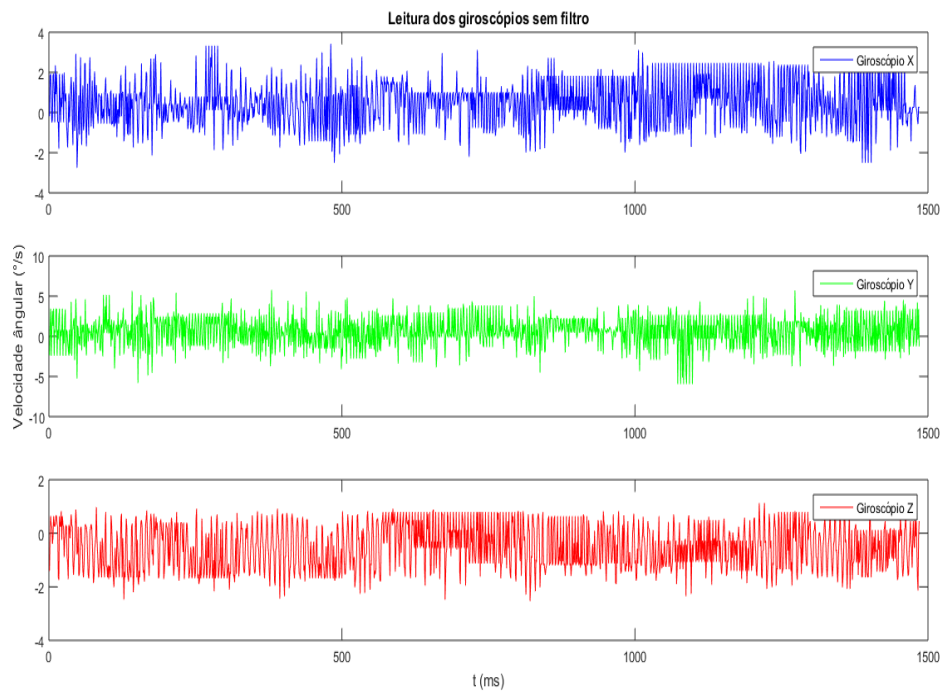


Figura 60 - Leitura dos giroscópios com motores a 15% de potência

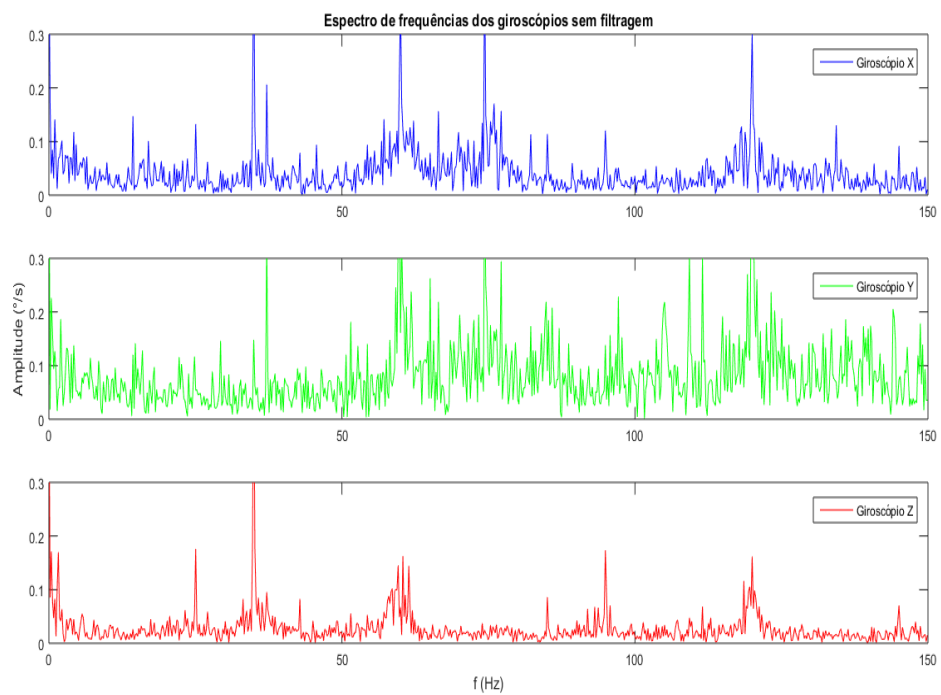


Figura 61 - Espectro de frequências dos giroscópios com motores a 15% de potência

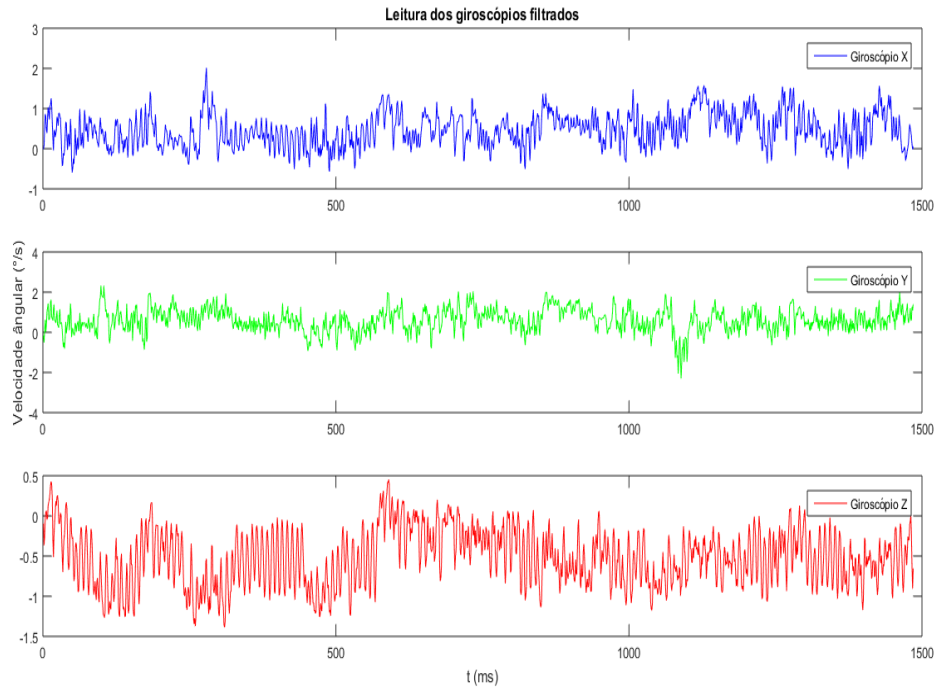


Figura 62 - Leitura filtrada dos giroscópios, motores a 15% de potência

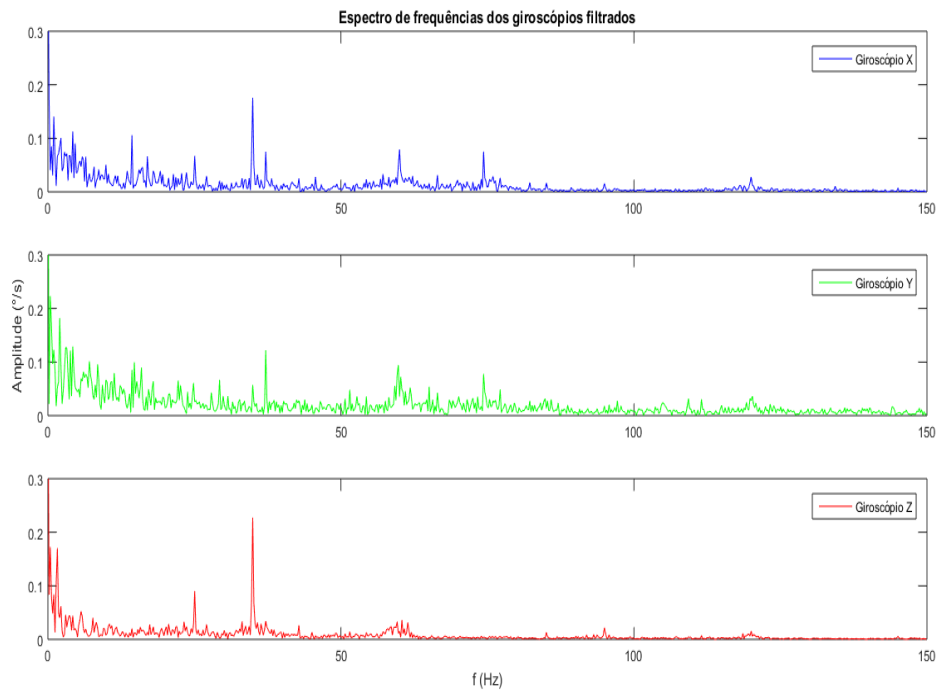


Figura 63 - Espectro de frequências dos giroscópios filtrados, motores a 15% de potência

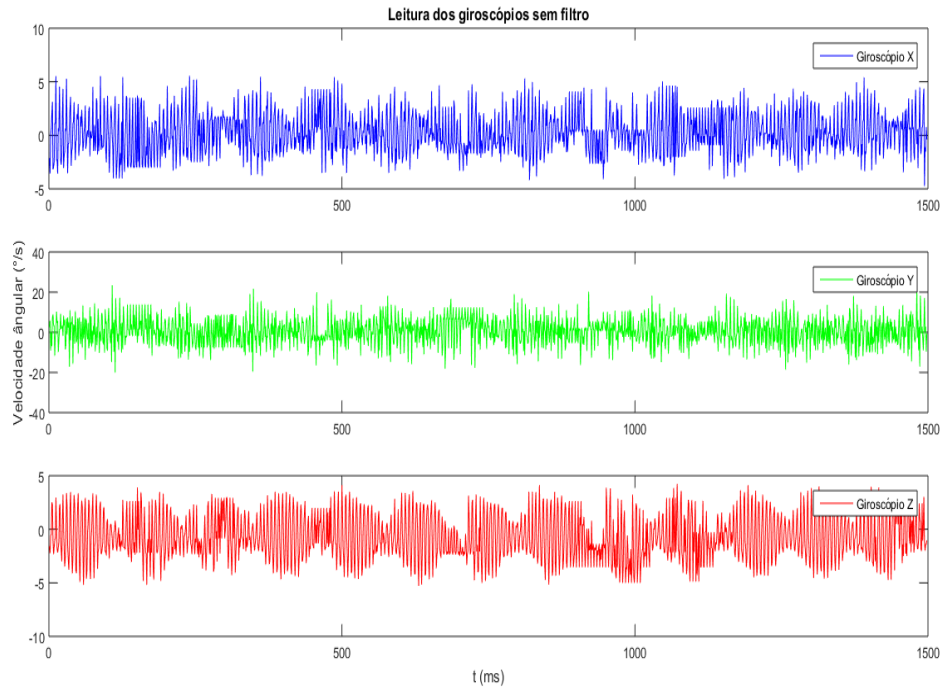


Figura 64 - Leitura dos giroscópios com motores a 40% de potência

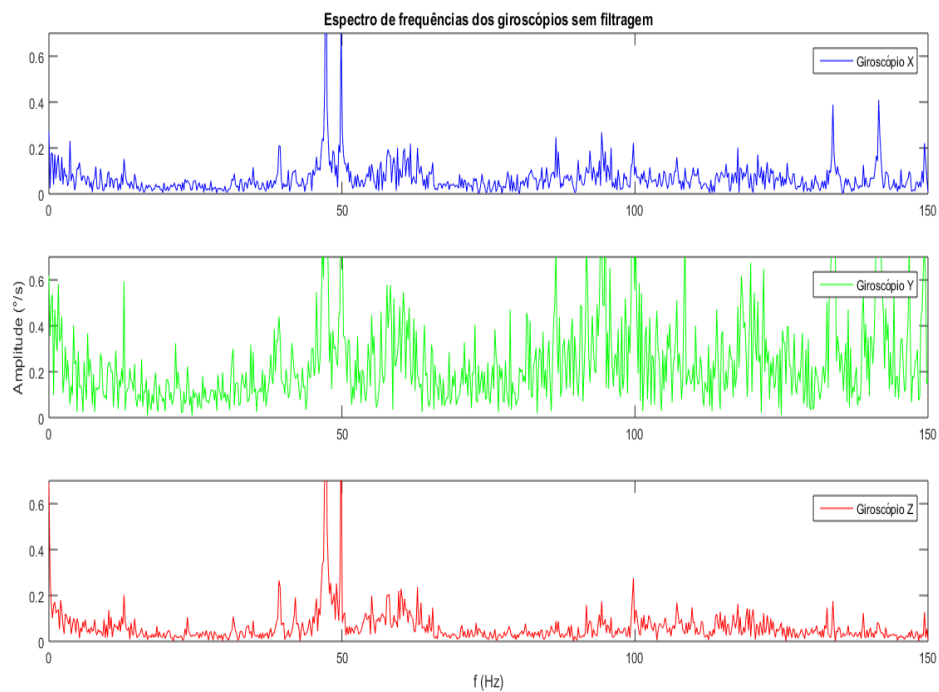


Figura 65 - Espectro de frequências dos giroscópios com motores a 40% de potência

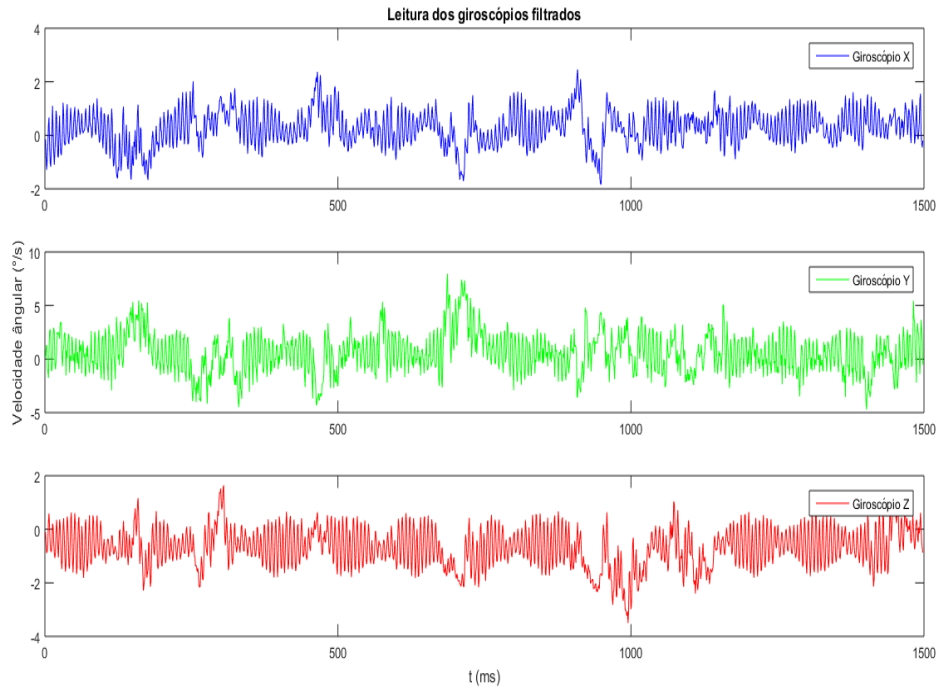


Figura 66 - Leitura filtrada dos giroscópios, motores a 40%

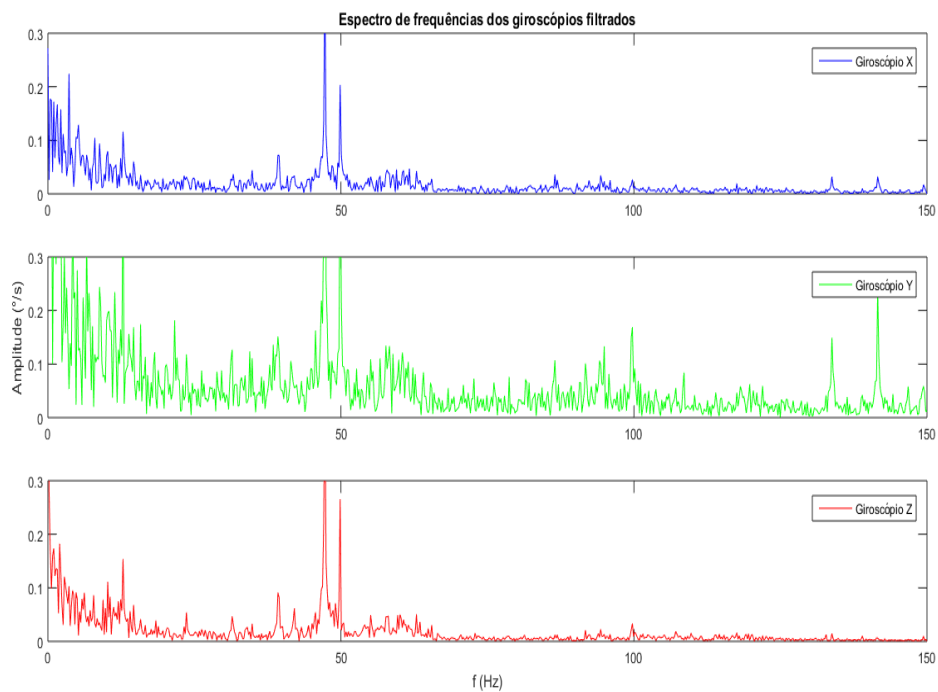


Figura 67 - Espectro de frequências dos giroscópios filtrados, motores a 40% de potência

B.3 Acelerômetros

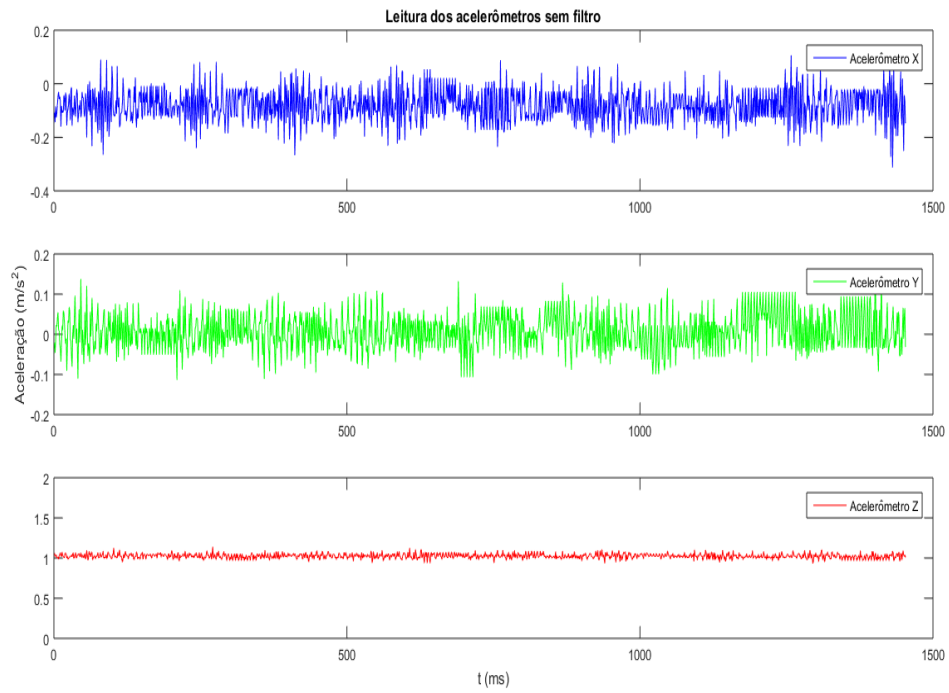


Figura 68 - Leitura dos acelerômetros, motores a 5% de potência, sem filtro

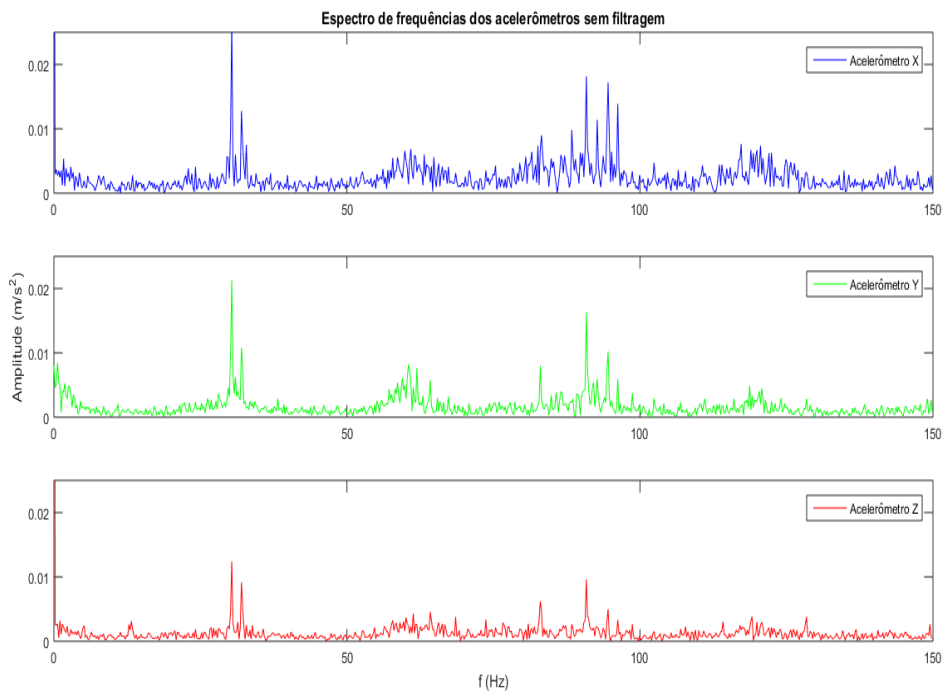


Figura 69 - Espectro de frequências dos acelerômetros, motores a 5% de potência, sem filtro

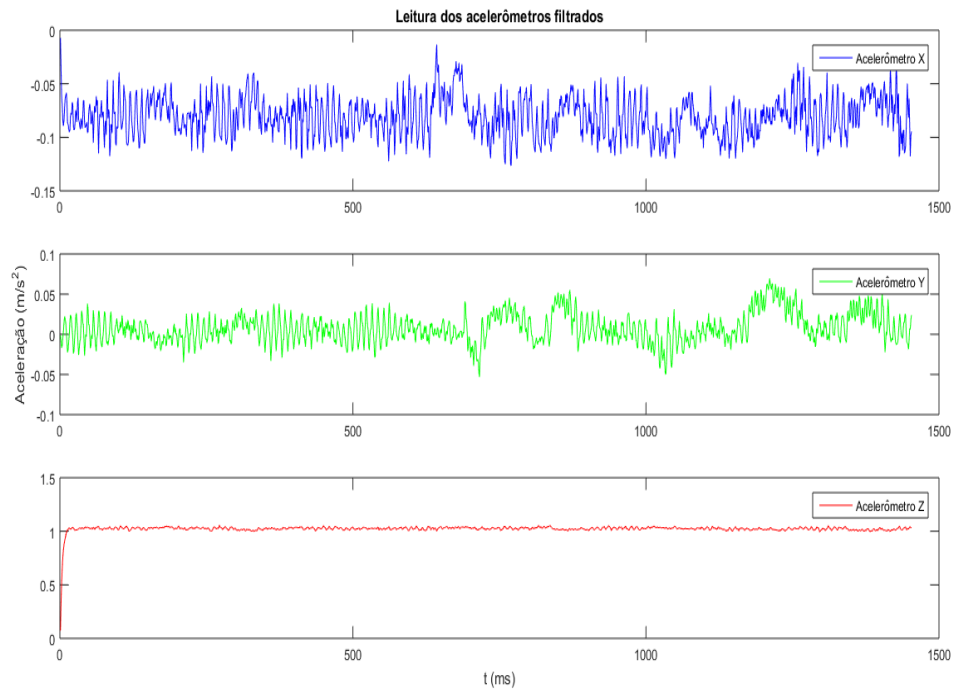


Figura 70 - Leitura filtrada dos acelerômetros, motores a 5% de potência

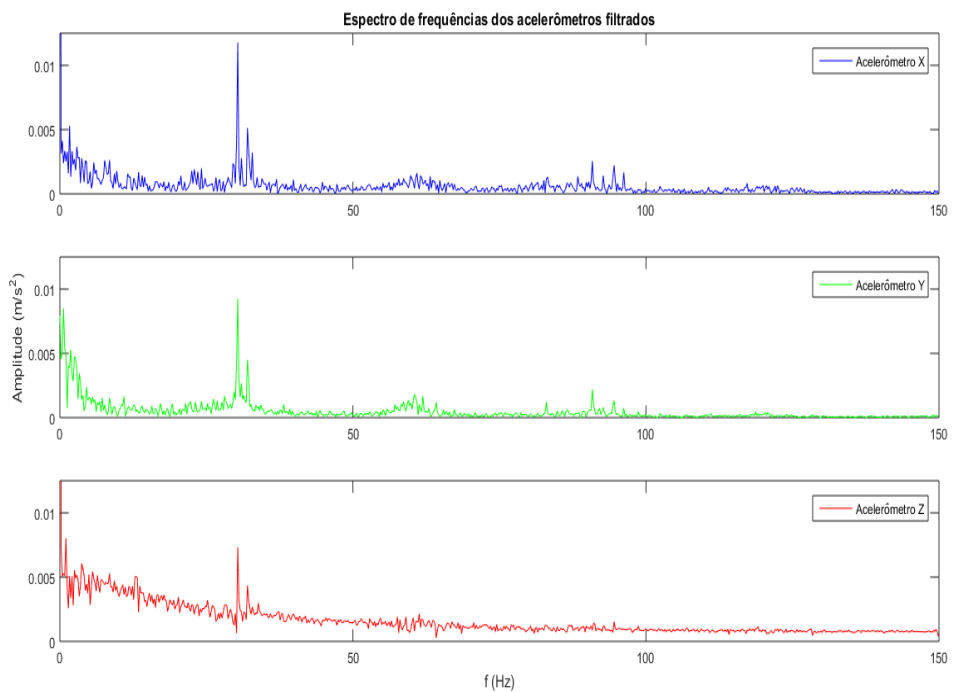


Figura 71 - Espectro de frequências dos acelerômetros filtrados, motores a 5% de potência

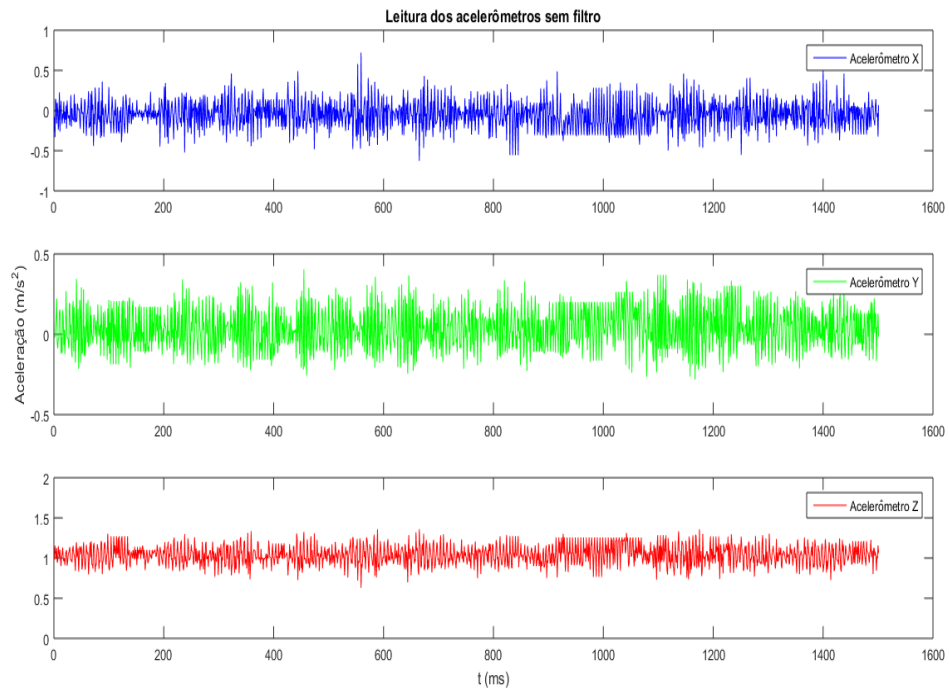


Figura 72 - Leitura dos acelerômetros, motores a 30% de potência, sem filtro

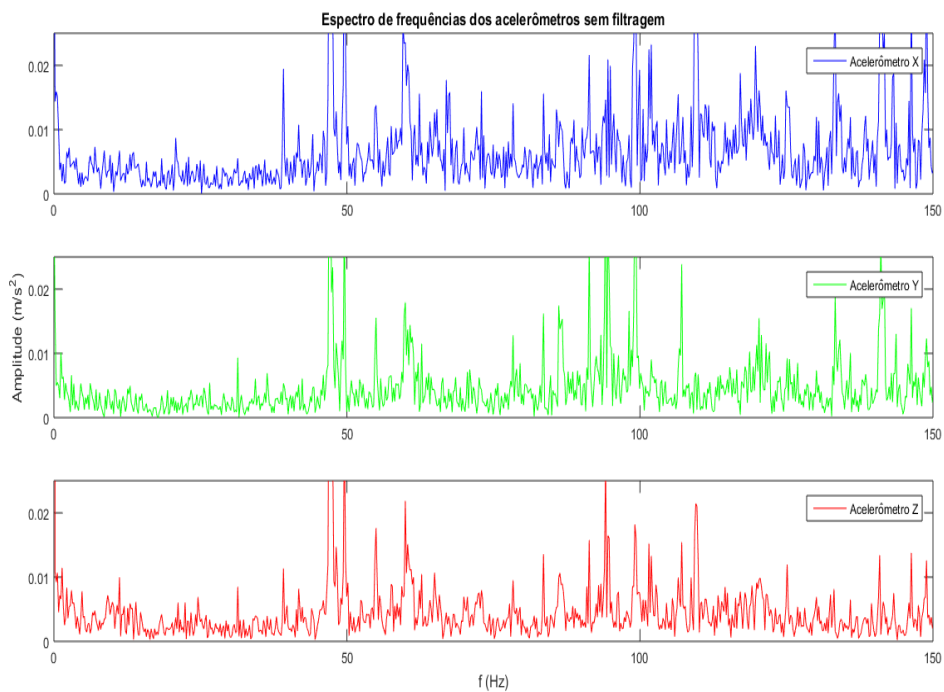


Figura 73 - Espectro de frequências dos acelerômetros, motores a 30% de potência, sem filtro

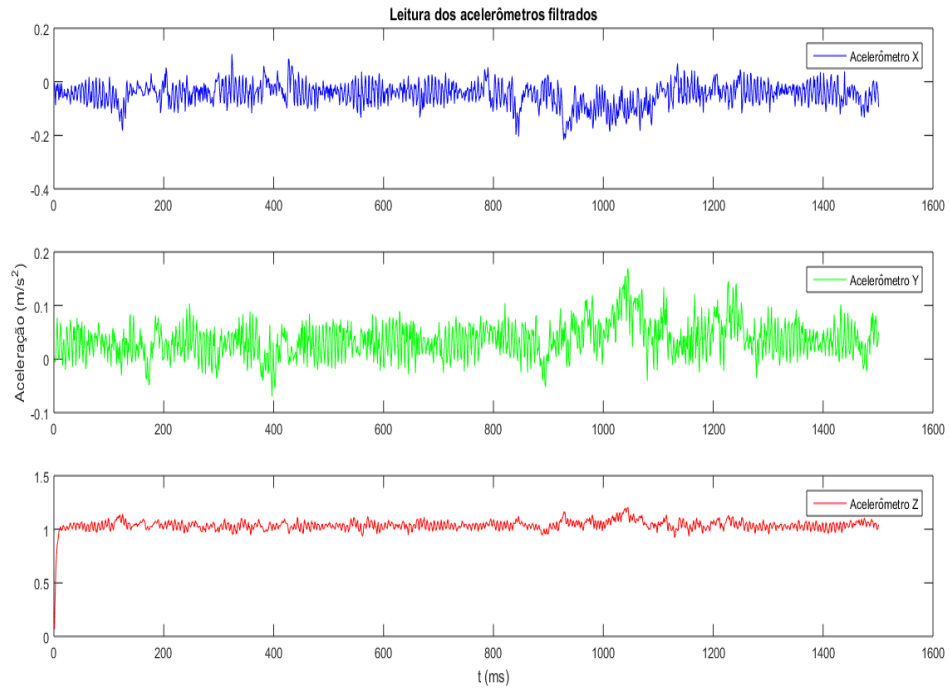


Figura 74 - Leitura filtrada dos acelerômetros, motores a 30%

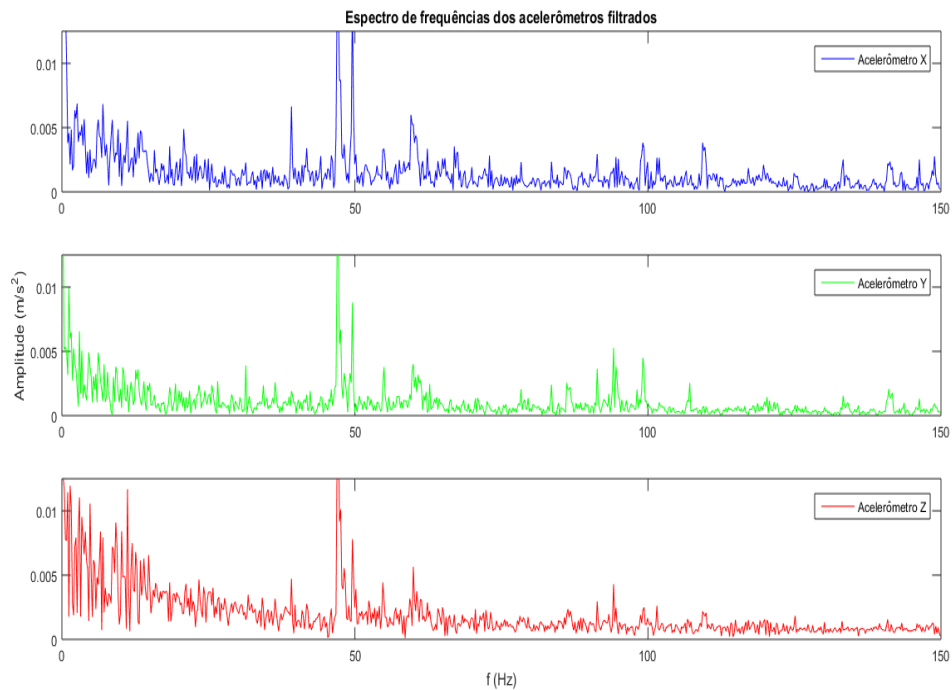


Figura 75 - Espectro de frequências dos acelerômetros filtrados, motores a 30% de potência

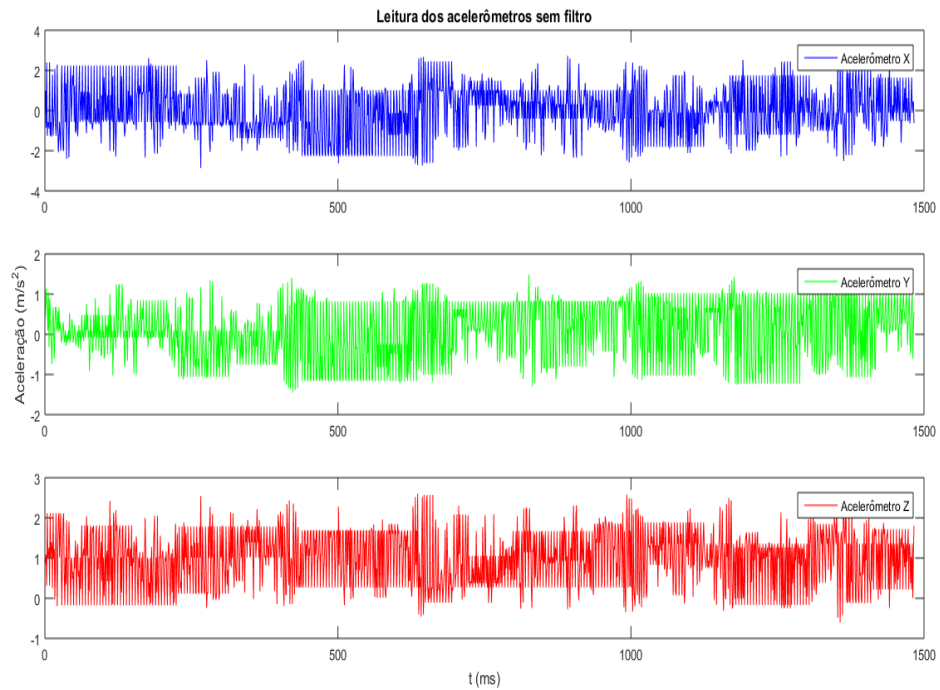


Figura 76 - Leitura dos acelerômetros, motores a 60% de potência, sem filtro

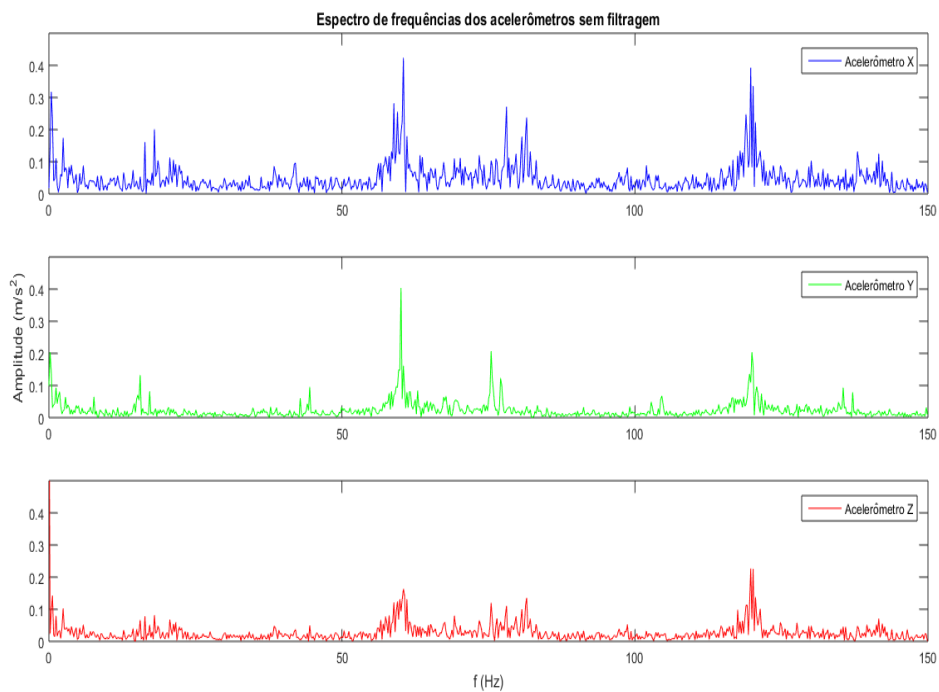


Figura 77 - Espectro de frequências dos acelerômetros, motores a 60% de potência, sem filtro

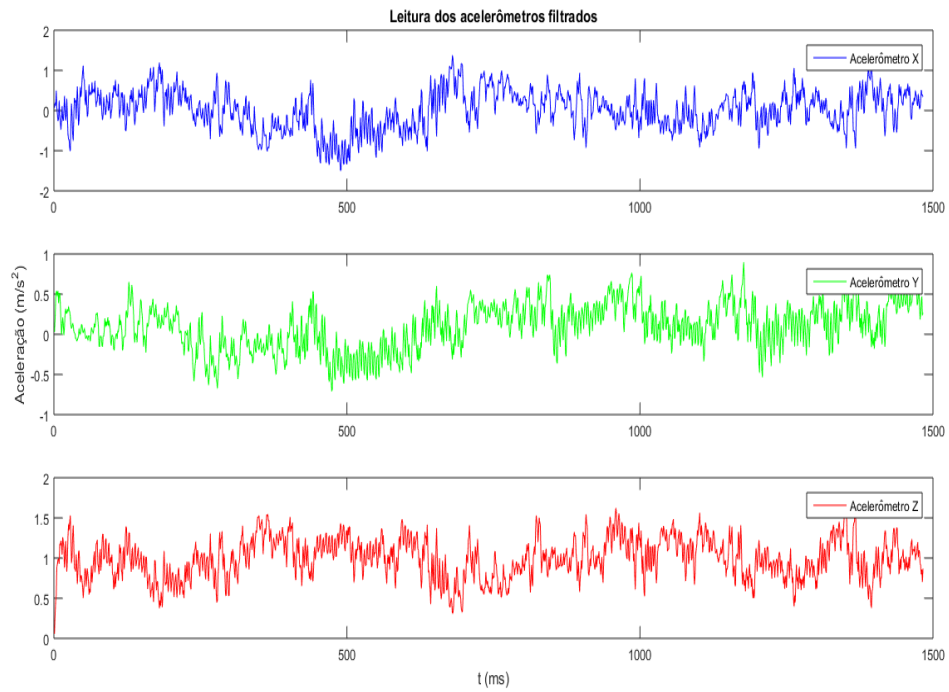


Figura 78 - Leitura filtrada dos acelerômetros, motores a 60% de potência

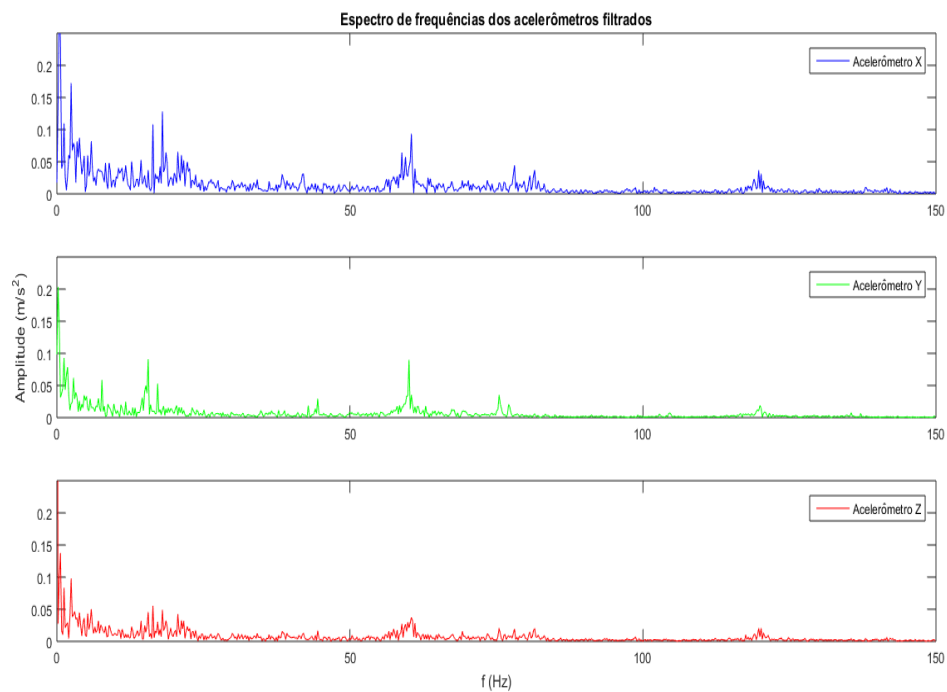


Figura 79 - Espectro de frequências dos acelerômetros filtrados, motores a 60% de potência

Apêndice C – Funções para cálculos do filtro e PID

```

typedef struct biquad_filter_s {
    float wn ; //actual
    float wn1; //past 1
    float wn2; //past 2
    float yn ; //output
}biquad_filter_t;

/*
 * Computes the new value of given xn sample
 */

float Biquad_filter(biquad_filter_t* var, float xn)
{
    var->wn = xn - a1 * var->wn1 - a2 * var->wn2;
    var->yn = b0 * var->wn + b1 * var->wn1 + b2 * var->wn2;
    var->wn1 = var->wn;
    var->wn2 = var->wn1;
    return var->yn ;
}

/*
 * Faster Fc calculation (~288us)
 * Designed for this firmware
 */
#define fsample      (300.0)
void Biquad_setNewFC(unsigned int Fc)
{
    float omega0, cosOmega, alfa, a0;
    #define Q 0.7071068 // 1/sqrt(2) butterworth Q
    omega0 = 0.0209440 / Fc; //2 * PI * Fc / fsample;
    cosOmega = cos(omega0);
    alfa = sin(omega0)*Q; // sinOmega/(2*Q)

    a0 = (1.0+alfa);
    a1 = (-2.0*cosOmega)/a0;
    a2 = (1.0-alfa)/a0;
    b0 = (1.0 - cosOmega)/2.0/a0;
    b1 = b0*2.0;
    b2 = b0;
}

```

```
typedef struct pid2order_discrete_s {
    float x1; //past 1
    float x2; //past 2
    float y1; //past 1
}pid2order_discrete_t;

/*
 * Computes PID output from given xn error
 */

signed int Pid2Order_Calc(pid2order_discrete_t* var, float xn)

{
    var->y1 = xn*B0 + var->x1*B1 + var->x2*B2 + var->y1;
    var->x2 = var->x1;
    var->x1 = xn;
    //implement anti blow-up
    // if((int)var->y1 > MAXOUT) return MAXOUT;
    // if((int)var->y1 < -MAXOUT) return -MAXOUT;
    return var->y1;
}
```

Anexo A – Filtro de fusão de dados Madgwick-Mahony

O seguinte algoritmo utiliza a leitura dos sensores para atualizar os quatérnios (variáveis de estado). A unidade de medida de entrada é:

- Giroscópios: rad/s
- Acelerômetros: m/s^2
- Magnetômetros: Miligauss

```
//-----
// Definitions

#define sampleFreq      300.0f      // sample frequency in Hz
#define twoKpDef  (2.0f * 0.5f)    // 2 * proportional gain
#define twoKiDef  (2.0f * 0.0f)    // 2 * integral gain

//-----
// Variable definitions

volatile float twoKp = twoKpDef;    // 2 * proportional gain (Kp)
volatile float twoKi = twoKiDef;    // 2 * integral gain (Ki)
    // quaternion of sensor frame relative to auxiliary frame
volatile float q0 = 1.0f, q1 = 0.0f, q2 = 0.0f, q3 = 0.0f;
    // integral error terms scaled by Ki
volatile float integralFBx = 0.0f, integralFBy = 0.0f, integralFBz =
0.0f;

void MahonyQuaternionUpdate(float gx, float gy, float gz, float ax,
float ay, float az, float mx, float my, float mz) {
    float recipNorm;
    float q0q0, q0q1, q0q2, q0q3, q1q1, q1q2, q1q3, q2q2, q2q3,
q3q3;
    float hx, hy, bx, bz;
    float halfvx, halfvy, halfvz, halfwx, halfwy, halfwz;
    float halfex, halfey, halfez;
    float qa, qb, qc;

    // Use IMU algorithm if magnetometer measurement invalid (avoids
NaN in magnetometer normalisation)
    if((mx == 0.0f) && (my == 0.0f) && (mz == 0.0f)) {
        ReadSensors_G_A(gx, gy, gz, ax, ay, az);
        return;
    }

    // Compute feedback only if accelerometer measurement valid
(avoids NaN in accelerometer normalisation)
    if(!((ax == 0.0f) && (ay == 0.0f) && (az == 0.0f))) {
```

```

// Normalise accelerometer measurement
recipNorm = 1.0/sqrt(ax * ax + ay * ay + az * az);
ax *= recipNorm;
ay *= recipNorm;
az *= recipNorm;

// Normalise magnetometer measurement
recipNorm = 1.0/sqrt(mx * mx + my * my + mz * mz);
mx *= recipNorm;
my *= recipNorm;
mz *= recipNorm;

// Auxiliary variables to avoid repeated arithmetic
q0q0 = q0 * q0;
q0q1 = q0 * q1;
q0q2 = q0 * q2;
q0q3 = q0 * q3;
q1q1 = q1 * q1;
q1q2 = q1 * q2;
q1q3 = q1 * q3;
q2q2 = q2 * q2;
q2q3 = q2 * q3;
q3q3 = q3 * q3;

// Reference direction of Earth's magnetic field
hx = 2.0f * (mx * (0.5f - q2q2 - q3q3) + my * (q1q2 -
q0q3) + mz * (q1q3 + q0q2));
hy = 2.0f * (mx * (q1q2 + q0q3) + my * (0.5f - q1q1 -
q3q3) + mz * (q2q3 - q0q1));
bx = sqrt(hx * hx + hy * hy);
bz = 2.0f * (mx * (q1q3 - q0q2) + my * (q2q3 + q0q1) + mz
* (0.5f - q1q1 - q2q2));

// Estimated direction of gravity and magnetic field
halfvx = q1q3 - q0q2;
halfvy = q0q1 + q2q3;
halfvz = q0q0 - 0.5f + q3q3;
halfwx = bx * (0.5f - q2q2 - q3q3) + bz * (q1q3 - q0q2);
halfwy = bx * (q1q2 - q0q3) + bz * (q0q1 + q2q3);
halfwz = bx * (q0q2 + q1q3) + bz * (0.5f - q1q1 - q2q2);

// Error is sum of cross product between estimated
direction and measured direction of field vectors
halfex = (ay * halfvz - az * halfvy) + (my * halfwz - mz *
halfwy);
halfey = (az * halfvx - ax * halfvz) + (mz * halfwx - mx *
halfwz);
halfez = (ax * halfvy - ay * halfvx) + (mx * halfwy - my *
halfwx);

// Compute and apply integral feedback if enabled
if(twoKi > 0.0f) {

```

```

        integralFBx += twoKi * halfex * (1.0f / sampleFreq);
// integral error scaled by Ki
        integralFBy += twoKi * halfey * (1.0f / sampleFreq);
        integralFBz += twoKi * halfez * (1.0f / sampleFreq);
        gx += integralFBx;    // apply integral feedback
        gy += integralFBy;
        gz += integralFBz;
    }
    else {
        integralFBx = 0.0f;    // prevent integral windup
        integralFBy = 0.0f;
        integralFBz = 0.0f;
    }

    // Apply proportional feedback
    gx += twoKp * halfex;
    gy += twoKp * halfey;
    gz += twoKp * halfez;
}

// Integrate rate of change of quaternion
gx *= (0.5f * (1.0f / sampleFreq));    // pre-multiply common
factors
gy *= (0.5f * (1.0f / sampleFreq));
gz *= (0.5f * (1.0f / sampleFreq));
qa = q0;
qb = q1;
qc = q2;
q0 += (-qb * gx - qc * gy - q3 * gz);
q1 += (qa * gx + qc * gz - q3 * gy);
q2 += (qa * gy - qb * gz + q3 * gx);
q3 += (qa * gz + qb * gy - qc * gx);

// Normalise quaternion
recipNorm = 1.0/sqrt(q0 * q0 + q1 * q1 + q2 * q2 + q3 * q3);
q0 *= recipNorm;
q1 *= recipNorm;
q2 *= recipNorm;
q3 *= recipNorm;
}

```

Anexo B – Funções rápidas de cálculo aproximado

A seguir, as funções utilizadas para otimizar o código, ganhando tempo de execução em troca de precisão.

```
//3-4 times faster than arcsin from math.h
// error is negligible under < 90° output
//taylor series with 3 elements
// about 62us
float ArcSino(float x)
{
    float xx = x * x;
    return x*(1 + xx*(0.16666666 + 0.075*xx));
}

//http://pubs.opengroup.org/onlinepubs/009695399/functions/atan2
.html
//Volkan SALMA, fastest to this firmware
// about 180us
float atan2_approximation1(float y, float x)
{
    const float ONEQTR_PI = M_PI / 4.0;
    const float THRQTR_PI = 3.0 * M_PI / 4.0;
    float r, angle;
    float abs_y = fabs(y) + 1e-10f; // kludge to prevent 0/0
condition
    if ( x < 0.0f )
    {
        r = (x + abs_y) / (abs_y - x);
        angle = THRQTR_PI;
    }
    else
    {
        r = (x - abs_y) / (x + abs_y);
        angle = ONEQTR_PI;
    }
    angle += (0.1963f * r * r - 0.9817f) * r;
    if ( y < 0.0f )
    return( -angle ); // negate if in quad III or IV
    else
    return( angle );
}

//Quake III implementation adequated to 8 bits microcontroler
// means 1/sqrt(x)
// about 64us
```

```
float invSqrt(float x) {  
    float halfx = 0.5f * x;  
    float y = x;  
    long i = *(long*)&y;  
    i = 0x5f3759df - (i>>1);  
    y = *(float*)&i;  
    y = y * (1.5f - (halfx * y * y));  
    return y;  
}
```

```
//Quake III implementation adequated  
// about 56us  
// means sqrt() faster  
float SqrtAprox(float number) {//low in fat  
    unsigned long i;  
    float y;  
    y = number;  
    i = * ( long * ) &y;  
    i = 0x5f375a86 - ( i >> 1 );  
    y = * ( float * ) &i;  
    return number * y;  
}
```