



**MARCELLO GONÇALVES MOREIRA**

**IMPLEMENTAÇÃO DE UM SISTEMA DE ATITUDE E  
DIREÇÃO BASEADO EM QUATERNIÕES PARA VEÍCULOS  
AÉREOS NÃO TRIPULADOS**

**LAVRAS – MG**

**2019**

**MARCELLO GONÇALVES MOREIRA**

**IMPLEMENTAÇÃO DE UM SISTEMA DE ATITUDE E DIREÇÃO BASEADO EM  
QUATERNIÕES PARA VEÍCULOS AÉREOS NÃO TRIPULADOS**

Trabalho de Conclusão de Curso apresentado à  
Universidade Federal de Lavras, como parte das  
exigências do Curso de Engenharia de Controle  
e Automação, para a obtenção do título de  
Bacharel.

Prof. DSc. Belizario Nina Huallpa  
Orientador

**LAVRAS – MG**

**2019**

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos  
da Biblioteca Universitária da UFLA**

Moreira, Marcello Gonçalves

Implementação de um sistema de atitude e direção baseado em quatérniões para veículos aéreos não tripulados / Marcello Gonçalves Moreira. – Lavras : UFLA, 2019.

32 p. : il.

TCC (graduação)–Universidade Federal de Lavras, 2019.

Orientador: Prof. DSc. Belizario Nina Huallpa.

Bibliografia.

1. Aeronaves Remotamente Pilotadas. 2. Fusão de Sensores. 3. Filtros Complementares. I. Huallpa, Belizario Nina. II. Título.

CDD-808.066

**MARCELLO GONÇALVES MOREIRA**

**IMPLEMENTAÇÃO DE UM SISTEMA DE ATITUDE E DIREÇÃO BASEADO EM  
QUATERNIÕES PARA VEÍCULOS AÉREOS NÃO TRIPULADOS  
IMPLEMENTATION OF AN ATTITUDE AND ORIENTATION SYSTEM BASED ON  
QUATERNIONS FOR UNMANNED AERIAL VEHICLES**

Trabalho de Conclusão de Curso apresentado à  
Universidade Federal de Lavras, como parte das  
exigências do Curso de Engenharia de Controle  
e Automação, para a obtenção do título de  
Bacharel.

APROVADA em 17 de Junho de 2019.

Prof. DSc. Belizario Nina Huallpa DAT  
Prof. DSc. Leonardo Silveira Paiva DAT  
Prof. Dsc. Danton Diego Ferreira DAT

Prof. DSc. Belizario Nina Huallpa  
Orientador

**LAVRAS – MG  
2019**

*Dedico este trabalho aos meus pais, pelo esforço e empenho que sempre demonstraram.*

## **AGRADECIMENTOS**

Agradeço ao Prof. Gabriel Araújo e Silva Ferraz, pela oportunidade de trabalhar no projeto de agricultura de precisão com Drones.

Agradeço a CAPES pelo auxílio financeiro permitindo a aquisição de materiais para a pesquisa.

Ao colega Eduardo Massayuki Iwasaki pelo apoio e desenvolvimento do projeto mecânico do drone.

Aos grupos de estudo NESA e NEMAPORT por permitirem o uso do espaço de trabalho.

A todos os professores do curso de Engenharia de Controle e Automação.

Aos técnicos administrativos do núcleo da Engenharia de Controle e Automação pelo auxílio durante as fases práticas do projeto.

Finalmente, agradeço a Universidade Federal de Lavras por todo apoio e incentivo à pesquisa científica.

*O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo.  
Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas  
admiráveis.  
(José de Alencar)*

## RESUMO

Aeronaves Remotamente Pilotadas (Remotely Piloted Aircrafts - RPA) apresentam grande potencial de aplicação para diversas áreas da indústria e do setor agrário. A necessidade do voo autônomo desses veículos exige um grande nível de sensoriamento e redundância em caso de falhas. Na aviação clássica, são usados robustos sistemas de referência de atitude e direção conhecidos pela sua sigla em inglês AHRS (Attitude and Heading Reference System). Esse sistema fornece informações de inclinação e direção da aeronave a partir de medições de diversas variáveis físicas. Os algoritmos de AHRS usualmente fazem a fusão de diversos sensores para obter informações mais confiáveis e precisas. Os cálculos que um AHRS realiza envolvem operações com matrizes rotacionais que possuem custo computacional elevado. Esse alto custo computacional representa limitações para processadores embarcados em pequenas aeronaves. Como alternativa, o cálculo rotacional com quatérniões é mais eficiente e tem custo computacional reduzido. O objetivo desse trabalho é comparar e implementar um AHRS baseado em quatérniões para ser utilizado em uma RPA do tipo quadricóptero. Para este trabalho, propõe-se utilizar um Arduino Uno responsável por executar o algoritmo do sistema AHRS e manter a comunicação com os sensores cinético e magnetômetro. Após a implementação e comparação dos algoritmos propostos. Foi percebido que o Filtro de Kalman Estendido apresentou os melhores resultados teóricos. No entanto, não foi possível implementá-lo em um ambiente tão restrito como um Arduino Uno. Como alternativa, o algoritmo de Madgwick apresentou o segundo melhor resultado e pode ser implementado.

**Palavras-chave:** Filtros Complementares. Aeronaves Remotamente Pilotadas. Fusão de Sensores.

## ABSTRACT

Remotely Piloted Aircraft (RPA) show great potential in various industries and agricultural sectors. The necessity of autonomous flight of these vehicles demands a great level of sensing and redundancy in case something fails. At classical aviation, robust navigational systems known as Attitude and Heading Reference Systems (AHRS) are used. These system gives information about orientation relative to ground frame from the measurement of an assortment of sensors. AHRS Algorithms usually fuse these many sensors to obtain reliable and precise information. The calculations a AHRS needs to do involves the use of rotation matrices that have a high computational cost. Those high demanding computations represents limitations regarding embedded systems in small aircrafts. In order to avoid this, the use of quaternion representation is more computationally efficient. The objective of this paper is too compare and implement an AHRS based on quaternion to be used on a RPA of type quad rotor. For this work, it is proposed to use an Arduino Uno to run the algorithm and communicate with inertial and magnetic sensors. After the comparison and implementation, it was noticed that Extended Kalman Filter presents the best results theoretically. However, it could not be implement in such a restricted environment as an Arduino Uno. As an alternative, the Madgwick algorithm had the second best results and could be implemented.

**Keywords:** Complementary Filters. Remotly Piloted Aircrafts. Sensor Fusion.

## LISTA DE FIGURAS

Figura 2.1 – RPA de 4 rotores: DJI Phantom 4 . . . . .	16
Figura 2.2 – Modelo simplificado de uma RPA . . . . .	17
Figura 4.1 – Diagrama de conexão dos arduinos de controle e sensoriamento . . . . .	23

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>11</b>
<b>2.1</b>	<b>Teoria básica aeronáutica</b>	<b>11</b>
<b>2.2</b>	<b>Veículos aéreos não tripulados</b>	<b>11</b>
<b>2.3</b>	<b>Filtros Complementares e o Filtro de Kalman</b>	<b>12</b>
<b>2.4</b>	<b>Sistema de Atitude e Orientação</b>	<b>13</b>
<b>2.5</b>	<b>Quaterniões e Angulos Euler</b>	<b>14</b>
<b>2.6</b>	<b>Modelagem e controle de uma RPA</b>	<b>16</b>
<b>3</b>	<b>CONCLUSÃO</b>	<b>20</b>
	<b>REFERÊNCIAS</b>	<b>21</b>
<b>4</b>	<b>ARTIGO</b>	<b>23</b>

## 1 INTRODUÇÃO

Aeronaves remotamente pilotadas (RPAs), comumente chamadas de veículos aéreos não tripulados (VANT) ou drones, foram ferramentas criadas inicialmente para fins militares, sendo utilizados em missões de reconhecimento onde era demasiado arriscado o envio de pilotos. O custo dos componentes eletrônicos e mecânicos que compõem a construção da aeronave tem reduzido significativamente com o passar dos anos, e a maior acessibilidade e miniaturização desses componentes tem tornado viável a utilização de drones nos meios civil e industrial. Diversos países possuem trabalhos no desenvolvimento de VANTs. No Brasil, esses veículos aéreos não tripulados atualmente são encontrados em atividades de monitoramento urbano, recreação, filmagem de eventos, aquisição de imagens de lavouras, controle de queimadas e inspeção de linhas aéreas de transmissão de energia.

Drones do tipo quadricoptero possuem quatro hélices acionadas independente uma das outras, em forma de “X”. Essa construção permite uma grande versatilidade no posicionamento da carga, além da capacidade de decolagem vertical e rotação sobre o próprio eixo, o que a torna uma excelente opção para aplicações urbanas, uma vez que são capazes de operar em espaços pequenos. No entanto, existe grande complexidade na operação completamente manual dessa aeronave. Como solução são empregados diferentes algoritmos de controle, permitindo assim que uma pessoa inexperiente seja capaz de pilotar o drone.

Esses algoritmos são controladores digitais, que exigem sensoriamento para perceber a posição, velocidade e outras variáveis físicas que influenciam a aeronave. Com base nessas variáveis, eles tomam ações para manter a estabilidade de voo. Para controlar a aeronave, os sensores usados devem ser robustos, resistentes à interferência e, se possível, redundantes. Essas características são comuns na aeronáutica, onde é necessário um alto grau de confiabilidade do sistema.

Porém, quanto mais robusto um sensor, maior o seu custo, que será refletido no preço final da aeronave. Sensores robustos usados na aeronáutica tradicional são demasiadamente grandes e pesados, sendo inviável a sua utilização em drones de pequeno porte. Como alternativa, é possível usar múltiplos sensores de menor qualidade e medir variáveis físicas com maior precisão. A fusão de sensores é uma ferramenta capaz de prever estados futuros de um sistema a partir de seu modelo dinâmico e medidas ruidosas de um ou mais sensores. Contudo, essa técnica pode provocar um maior custo computacional para a leitura de estados do sistema, tornando o ciclo de medições mais lento e conseqüentemente um sistema não controlável. A proposta

desse trabalho é a implementação de técnicas de fusão de sensores para o controle de atitude e direção de um veículo aéreo não tripulado, utilizando sensores inerciais e magnetômetro.

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os conceitos básicos para compreensão das técnicas de fusão de sensores empregadas no experimento, assim como conceitos de controle e modelagem. Também será feita uma revisão bibliográfica com relação a veículos aéreos não tripulados e os principais componentes envolvidos em sua construção.

### 2.1 Teoria básica aeronáutica

Para um melhor entendimento deste trabalho, uma breve apresentação de alguns conceitos de aeronáutica se faz necessária. Segundo o manual de treinamento da Embraer, uma aeronave é capaz de realizar movimentos em torno de três eixos perpendiculares que passam pelo seu centro de massa. O movimento em torno do eixo transversal é chamado de arfagem (BRASIL, 1989), também conhecido pelo seu termo na língua inglesa inglês "yaw" e comumente representado pela letra grega  $\phi$ . Esta ação movimentada o nariz da aeronave para cima e para baixo. O movimento em torno do eixo longitudinal é chamado de rolagem (BRASIL, 1989), também conhecido pelo termo "roll" e representado pela letra  $\theta$ . Por fim, o movimento em torno do eixo vertical é chamado de guinada, também chamado de "yaw" e representado pela letra  $\psi$ . O movimento de guinada está associado a orientação da aeronave, que chama-se proa (heading). Dizemos que a proa de uma aeronave é  $0^\circ$  quando o nariz ou frente da aeronave aponta para o norte geográfico da Terra. A guinada é o ato ou efeito de desviar lateralmente a aeronave, movimentando em torno do eixo vertical (ANTAS, 1979). Denomina-se atitude a situação ou orientação de uma aeronave, seja com respeito ao vento relativo (atitude de voo), seja em relação a um sistema de eixos fixos à Terra (atitude em relação à Terra) (ANTAS, 1979). Neste trabalho, o termo atitude será usado para se referir a orientação em relação à Terra.

### 2.2 Veículos aéreos não tripulados

De acordo com o Regulamento Brasileiro de Aviação Civil Especial (AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL, 2017) o termo aeronave remotamente pilotadas (ARP ou RPA) refere-se a um veículo aéreo não tripulado onde existe um piloto remoto responsável pela operação segura da aeronave.

De acordo com (MEDEIROS et al., 2008), estes equipamentos tiveram sua primeira aparição em 1883, quando Douglas Archibald instalou um anemômetro a um fio em uma "pan-

dorga” para poder medir a velocidade dos ventos em diferentes altitudes. Ainda como Me-deiros, em 1935, Reginald Denny construiu o RP-1, o primeiro veículo aéreo não tripulado rádio-controlado.

Seu uso militar teve início no final dos anos 50, durante a guerra do Vietnã e a Guerra Fria. A pesquisa desses sistemas chegou ao pleno desenvolvimento em 1970 (NONAMI, 2007). O desenvolvimento da eletrônica e miniaturização dos componentes contribuiu para a popularização dessas aeronaves em outros meios além da área militar.

As RPAs são popularmente classificadas entre veículos de asa fixa ou asa rotativa. SILVA et al. (2014) diz que aeronaves de asa fixa tem maior autonomia de voo, por consumirem menos energia que os de asa rotativa. Da Silva ainda fala que as aeronaves de asa rotativa tem menor velocidade e podem obter melhores fotografias sem arrastamento. O modelo de aeronaves de asa rotativa com quatro rotores não é novo, e o primeiro relato de uma aeronave desse tipo foi datado em 1907, como apresentado no trabalho de Guimarães (2012). O autor ainda afirma que este modelo de aeronave está emergindo como plataforma padrão para pesquisa de mobilidade e percepção tridimensional pela sua flexibilidade no campo da robótica.

### 2.3 Filtros Complementares e o Filtro de Kalman

Uma técnica simples que é usada na indústria de controladores de voo para combinar medições de sensores é o filtro complementar (OSDER; ROUSE; YOUNG, 1973). O filtro complementar permite ponderar as saídas de dois ou mais sensores ruidosos, e fundi-las, fornecendo assim informações mais precisas. Uma aplicação típica de filtros complementares é combinar medições de aceleração vertical e velocidade vertical barométrica para obter uma estimativa da velocidade vertical (HIGGINS, 1975). Um filtro complementar básico onde  $x$  e  $y$  são medições ruidosas de um sinal  $z$  e  $\hat{z}$  é a estimativa de  $z$  produzida pelo filtro. Seja que  $y$  possua um ruído de alta frequência, e  $x$  possua um ruído de baixa frequência. Podemos então elaborar um filtro passa baixa  $G(s)$  e conseqüentemente um filtro passa alta  $1 - G(s)$ , tal que

$$\hat{z} = x[1 - G(s)] + y[G(s)] \quad (2.1)$$

A Equação 2.1 representa um filtro complementar que funde duas observações  $x$  e  $y$  com ruído de baixa e alta frequência em uma estimativa que elimina ambos os ruídos. Os filtros de Kalman são baseados no princípio de filtros complementares (HIGGINS, 1975). Proposto em

1960 por Rudolph Kalman, o filtro de Kalman fornece um método de estimar os estados de um sistema linear, tomando como base a leitura dos estados, corrompida por ruídos gaussianos, e o modelo matemático do processo, que também sofre interferência de ruídos gaussianos (NETO, 2017). O filtro de Kalman permite inferir informações faltantes a partir de medições indiretas e ruidosas (GREWAL; ANDREWS, 2001). Para tal, o filtro de Kalman faz uma estimativa do estado de um sistema, com base no estado anterior e o modelo matemático que rege a dinâmica do sistema. Conhecendo a covariância do modelo do processo e de medições, o filtro determina um ganho chamado "Ganho de Kalman". A estimativa dos estados do sistema são então ponderadas pelo ganho de Kalman e as medições dos sensores. O Filtro de Kalman e sua versão não linear, o Filtro de Kalman Estendido são técnicas de filtragem de sinal preditivo que se tornaram a base aceita na maioria dos algoritmos de orientação e sensores comerciais (MADGWICK; HARRISON; VAIDYANATHAN, 2011).

#### **2.4 Sistema de Atitude e Orientação**

O sensoriamento preciso dos estados de um robô móvel é essencial para permitir o seu funcionamento. Em uma RPA, precisamos ser capazes de determinar os ângulos rotacionais do veículo em um determinado instante  $t$  para poder controlar a estabilidade da aeronave. Sabatini (2006) afirma que a maioria das tecnologias de sensoriamento de movimento atuais necessitam de uma fonte externa para determinar a localização. O autor ainda afirma que geralmente essas fontes são transmitidas por uma distância limitada e interferência são problemas adicionais associados a esses sensores. Attitude Heading Reference System (AHRS) é um sistema de referência usado na aviação para orientar a aeronave. Composto por informações de arfagem, rolagem e guinada, em conjunto com a proa da aeronave. Essas quatro informações são essenciais para permitir a pilotagem manual ou autônoma de veículos aéreos. Para obter os dados de orientação podem ser usados sensores giroscópios, acelerômetros e magnetômetros. Giroscópios são sensíveis a movimentos angulares e são capazes de medir a velocidade angular de um corpo. Integrando este valor de velocidade e conhecendo a posição inicial é possível determinar a orientação do corpo. Porém esses sensores determinam a posição a partir da integração do sinal medido pelo tempo. Devido ao ruído da variável medida, isto provoca um deslocamento "drift" do valor integrado, que aumenta com o tempo (SABATINI, 2006). Acelerômetros como o nome sugere, são sensores sensíveis à aceleração. Estes sensores fornecem medições de aceleração incluindo a aceleração gravitacional. A partir da determinação do vetor de aceleração

gravitacional é possível estimar a orientação de um corpo (NETO, 2017). Ambos os sensores funcionam a partir do princípio da inércia. Um Unidade de Medição Inercial (IMU) consiste em giroscópios e acelerômetros que permitem o sensoriamento de movimentos translacionais e rotacionais (MADGWICK; HARRISON; VAIDYANATHAN, 2011). Esse tipo de sensor é imune a interferência e em princípio, seriam capazes de determinar o movimento de um corpo sem restrições (SABATINI, 2006). O sensor MARG (Magnetic, Angular Rate and Gravity) que é um híbrido de um sensor IMU que incorpora um magnetômetro. O magnetômetro fornece informações de campo magnético e permite a detecção do campo magnético terrestre. Técnicas de fusão de sensores permitem utilizar as boas características de cada sensor e contornar os problemas inerentes a cada um deles.

O algoritmo de Mahony é um filtro complementar proposto em 2008 capaz de obter uma estimativa da orientação de corpo no espaço a partir de medições de um IMU. O filtro proposto, chamado de "filtro complementar explícito" requer apenas medição de acelerômetro e giroscópio como entradas (MAHONY; HAMEL; PFLIMLIN, 2008). Madgwick propõe em 2011 um algoritmo de filtro complementar computacionalmente eficiente, com o intuito de ser usado na captura de movimento humano em pacientes em reabilitação. O avanço tecnológico e de poder de processamento de sistemas embarcados permitiu o emprego de técnicas mais complexas para estimativa de orientação (MARINA et al., 2011). Atualmente grande parte dos controladores de voo de RPAs utilizam variações do filtro de Kalman Extendido (MARANTOS; KOVEOS; KYRIAKOPOULOS, 2015). O artigo apresentado no Capítulo 4 estuda a viabilidade de implementar os algoritmos de Mahony, Madgwick e uma versão do Filtro de Kalman Extendido para estimar a orientação de um corpo.

## 2.5 Quaterniões e Angulos Euler

Existem diversas representações matemáticas para orientação e rotações no espaço tridimensional. Uma delas são os quaterniões, um tipo de número pertencente ao conjunto dos números hipercomplexos que pode ser usado para representar rotações e orientação no espaço tridimensional (VICCI, 2001). Esses números foram descobertos por Sir Willian Hamilton em 1843 como uma extensão de números complexos (SHOEMAKE, 1985). Um quaternião  $H$  é representado matematicamente como na equação 2.2 onde  $w, x, y$  e  $z$  são números reais e as equações 2.3 e 2.4 são verdadeiras.

$$H = w + ix + jy + kz \quad (2.2)$$

$$ij = k = -ji \quad (2.3)$$

$$i^2 = j^2 = k^2 = -1 \quad (2.4)$$

Percebe-se que a multiplicação de  $i$  por  $j$  é diferente da multiplicação de  $j$  por  $i$ . Essa característica de não comutação difere da aritmética clássica. A vantagem do uso de quatérnios em contrapartida de meios clássicos como ângulos Euler ou matrizes de rotação é a economia de recursos computacionais ao fazer operações de rotação. Quatérnios podem ser separados em duas partes quando são usados para representar rotações. Considerando o quatérnio  $H$ , separamos  $w$  como a parte escalar, e o conjunto  $x$ ,  $y$  e  $z$  como um vetor 3D (VICCI, 2001). Ao utilizar os quatérnios para representação de rotações, são usados quatérnios normalizados. Uma rotação de graus em torno de um eixo definido pelo vetor  $v$  arbitrário usamos a seguinte equação.

$$Rot_{\alpha,v} = \cos \frac{\alpha}{2} + i \left( x * \sin \frac{\alpha}{2} \right) + j \left( y * \sin \frac{\alpha}{2} \right) + k \left( z * \sin \frac{\alpha}{2} \right) \quad (2.5)$$

$$v = x, y, z \quad (2.6)$$

sendo que  $x$ ,  $y$  e  $z$  representam um vetor unitário  $v$ , e  $\alpha$  é o ângulo da rotação. Para representar a combinação de diferentes rotações apenas multiplicam-se esses quatérnios, lembrando que a ordem da multiplicação é importante. Considere os quatérnio  $P$  e  $Q$ . A multiplicação de  $P$  por  $Q$  é,

$$P \otimes Q = (p_1 + ip_2 + jp_3 + kp_4) \otimes (q_1 + iq_2 + jq_3 + kq_4) \quad (2.7)$$

$$\begin{aligned} P \otimes Q = & (p_1q_1 - p_2q_2 - p_3q_3 - p_4q_4) \\ & + i(p_1q_2 + p_2q_1 + p_3q_4 - p_4q_3) \\ & + j(p_1q_3 + p_3q_1 + p_4q_2 - p_2q_4) \\ & + j(p_1q_4 + p_4q_1 + p_2q_3 - p_3q_2) \end{aligned} \quad (2.8)$$

## 2.6 Modelagem e controle de uma RPA

A RPA de 4 rotores, chamada de quadricóptero é classificada como uma aeronave mais pesada que o ar, que obtém sustentação a partir do movimento de rotação de 4 hélices posicionadas no plano da aeronave e orientadas perpendiculares ao eixo vertical da aeronave (AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL, 2017).

Figura 2.1 – RPA de 4 rotores: DJI Phantom 4

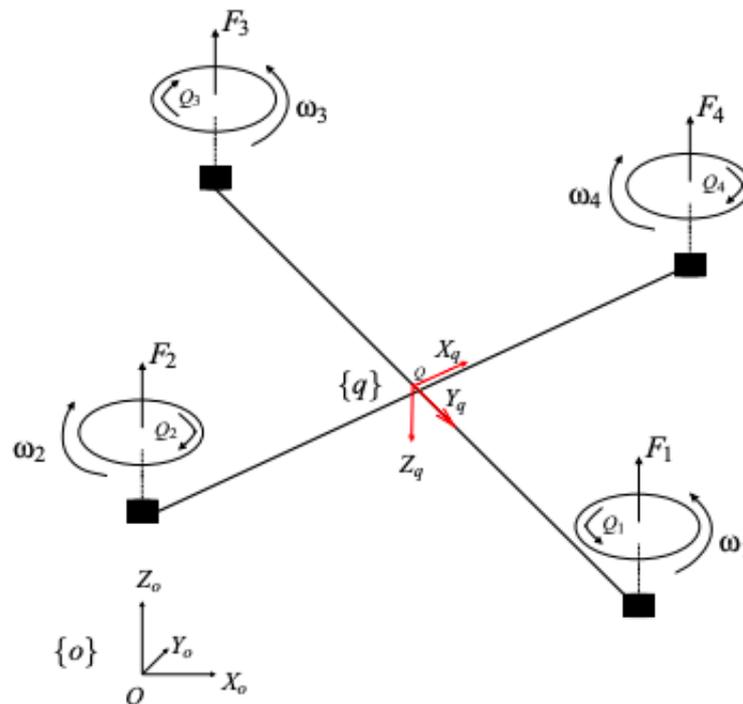


Fonte: DJI (2019)

O princípio de funcionamento da aeronave é explicado no trabalho de Guimarães (2012). Toda a sustentação e movimentação desse modelo de aeronave é feito pela variação da velocidade de rotação dos motores. O Quadricóptero possui duas hélices que giram no sentido horário e duas que giram no sentido anti horário. Deste modo, quando todas as hélices giram com a mesma velocidade, não existe torque rotacionando a aeronave sobre o eixo Z. Quando deseja-se realizar o movimento de guinada, dois motores que giram no mesmo sentido devem aumentar sua rotação, e proporcionalmente, os outros dois motores reduzirem sua rotação, mantendo a sustentação vertical e produzindo um torque diferencial que fará a rotação dessa aeronave. Os movimentos de rolagem e arfagem são obtidos de maneira similar. Aumentando a rotação de um motor e reduzindo a rotação do motor oposto, é possível inclinar a aeronave em determinado eixo. Será exposto em seguida, a modelagem encontrada no trabalho de Rodrigues (2014) e no trabalho de Guimarães (2012). Dois sistemas de coordenadas serão adotados para modelar a dinâmica do veículo. Um referencial inercial local A, usando o sistema de coordenadas NEU ( North-East-Up) onde o eixo X está apontado para o norte geográfico terrestre, o eixo Y apontando para o leste e o eixo Z apontado para cima. Também será usado um sistema de

coordenadas fixo ao corpo da aeronave B, onde a proa da aeronave aponta para o eixo X, o lado direito da aeronave aponta para o eixo Y e a barriga ou parte de baixo da aeronave aponta para o eixo Z. Esse sistema de coordenada do corpo da aeronave é chamado de NED (North-East-Down) e é comumente utilizado nos sensores inerciais de aeronaves.

Figura 2.2 – Modelo simplificado de uma RPA



Fonte: Guimarães (2012)

Determina-se então os vetores de posição e velocidade nos referenciais A e B.

$$\vec{X}_A = (X, Y, Z) \quad (2.9)$$

$$\dot{\vec{X}}_A = (\dot{X}, \dot{Y}, \dot{Z}) \quad (2.10)$$

$$\vec{X}_B = (X, Y, Z) \quad (2.11)$$

$$\dot{\vec{X}}_B = (\dot{X}, \dot{Y}, \dot{Z}) \quad (2.12)$$

Um vetor de velocidade angular no referencial do corpo da aeronave  $\{\omega\}$  e um vetor  $\theta$  contendo os ângulos de Euler  $\phi, \theta, \psi$ .

$$\vec{\omega} = (\omega_1, \omega_2, \omega_3) \quad (2.13)$$

$$\vec{\theta} = (\psi, \theta, \phi) \quad (2.14)$$

A matriz de rotação expressa em relação ao referencial A é dada por (GUIMARÃES, 2012),

$$R(\Phi, \Theta, \Psi) = \begin{pmatrix} \cos\Psi\cos\Theta & \cos\Phi\sin\Theta\sin\Phi - \sin\Psi\cos\Phi & \cos\Psi\sin\Theta\cos\Phi + \sin\Psi\sin\Phi \\ \sin\Psi\cos\Theta & \sin\Psi\sin\Theta\sin\Phi + \cos\Psi\cos\Phi & \sin\Psi\sin\Theta\cos\Phi - \sin\Phi\cos\Psi \\ -\sin\Theta & \cos\Theta\sin\Phi & \cos\Theta\sin\Phi \end{pmatrix} \quad (2.15)$$

A variação dos ângulos no tempo é uma função descontínua, e portanto é diferente da velocidade angular. A relação entre a derivada dos ângulos de Euler e a velocidade angular do veículo é dada pela matriz de transformação exposta no trabalho de Bouabdallah (2007).

$$[\omega] = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \sin\Phi\cos\Theta \\ 0 & -\sin\Phi & \cos\Phi\cos\Theta \end{pmatrix} \quad (2.16)$$

As equações de dinâmica rotacional simplificada do veículo são apresentadas nas equações 2.17 a 2.19

$$\ddot{\Phi} = \frac{d}{J_x} (F_2 - F_4) \quad (2.17)$$

$$\ddot{\Theta} = \frac{d}{J_y} (F_1 - F_3) \quad (2.18)$$

$$\ddot{\Psi} = \frac{b}{J_z} (F_1 - F_2 + F_3 - F_4) \quad (2.19)$$

Sendo as equações 2.17, 2.18 e 2.19 relacionadas aos eixos de arfagem, rolagem e guinada respectivamente. Os termos J equivalem ao momento de inercia da aeronave no eixo indicado pelo seu subscrito. d é a metade da distancia entre dois motores opostos. b é a constante que relaciona os empuxos com os torques de guinada.  $F_{1,2,3,4}$  representam a força de empuxo produzido por cada motor. Expõe-se aqui a modelagem encontrada no trabalho de Guimarães (2012). A partir das equações, propõe-se um novo conjunto de entradas utilizando a Equação 2.20 que relaciona o empuxo do motor i com o sinal PWM de entrada  $\delta$  por uma constante  $k_m$  (GUIMARÃES, 2012).

$$F_i = k_m * \delta_i \quad (2.20)$$

O novo conjunto de entradas representado pelas equações 2.21 a 2.23 relaciona o sinal PWM com o torque produzido em cada eixo da aeronave.

$$\tau_{24} = (\delta_2 - \delta_4)(k_m d) \quad (2.21)$$

$$\tau_{13} = (\delta_1 - \delta_3)(k_m d) \quad (2.22)$$

$$\tau_g = (\delta_1 - \delta_2 + \delta_3 - \delta_4)(k_m b) \quad (2.23)$$

Com esse novo conjunto de entradas, as equações 2.17 a 2.19 possuem apenas uma entrada e uma saída, podendo assim serem controladas de forma desacoplada (KOTTENSTETTE; PORTER, 2009). Três controladores digitais PID são desenvolvidos para controlar os ângulos de arfagem, rolagem e guinada da aeronave. O método de implementação e ajuste dos ganhos de cada um dos controladores está explicitado no trabalho de MOREIRA et al. (2018). Vale ressaltar que, a saída dos controladores são os três sinais  $\tau_{24}$ ,  $\tau_{13}$  e  $\tau_g$ . A saída do sinal de controle que é enviada aos motores é determinada pela Equação 2.24

$$\begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{pmatrix} = k_m \begin{pmatrix} 0 & d & b \\ d & 0 & -b \\ 0 & -d & b \\ -d & 0 & -b \end{pmatrix} \begin{pmatrix} \tau_{24} \\ \tau_{13} \\ \tau_g \end{pmatrix} + T \quad (2.24)$$

Onde T é o valor da aceleração base. Esta aceleração é controlada pelo piloto da aeronave e utilizada para controlar a velocidade vertical. A Equação 2.24 pondera os valores desacoplados da saída de cada controlador PID em um sinal único para cada motor.

### 3 CONCLUSÃO

O objetivo deste trabalho foi apresentar, comparar e implementar métodos de filtragem de sinal e fusão de sensores para obter uma estimativa da orientação de um corpo. O artigo apresentado neste trabalho discursa sobre a implementação de três algoritmos diferentes em um microcontrolador de 8 Bits. Foi utilizado o microcontrolador Arduino Uno para a implementação dos três algoritmos, em conjunto com o módulo de sensores GY-80. Foi identificado que o emprego de algoritmos complexos como o filtro de Kalman, apesar de fornecer estimativas precisas, pode ser impossibilitado devido as restrições de espaço e memória do microcontrolador. Mesmo utilizando a representação em quaterniões para minimizar o custo computacional, o algoritmo de Kalman ocupa mais espaço e demanda mais memória que o Arduino possui. No entanto, o desempenho de algoritmos mais simples como filtros complementares foram bons o suficiente para fornecer estimativas ao controlador de uma RPA. A representação em quaterniões também evitou o efeito de Gimbal Lock que aparece quando rotações são realizadas utilizando ângulos de Euler. As informações fornecidas pelo sistema de orientação (AHRS) proposto no artigo permitem o controle e estabilização da rotação de uma RPA. Um controlador regulador pode então ser usado em conjunto com o sistema AHRS. Como o microcontrolador Arduino não possui paralelismo, é interessante separar o processamento dos filtros e dos loops PID de controle em dois dispositivos diferentes. Isto permite que ambos os dispositivos executem tarefas mais complexas sem sobrecarregar o tempo de execução dos loops PID e o tempo de amostragem do sistema AHRS. O modularismo também permite a integração de um segundo sistema AHRS como backup. Conclui-se que o sistema de atitude e orientação baseado em quaterniões foi implementado com êxito em um microcontrolador para ser usado em uma aeronave remotamente pilotada.

## REFERÊNCIAS

- AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. **Regulamento Brasileiro Da Aviação Civil Especial**. Resolução nº 419. [S.l.], 2017.
- ANTAS, L. M. **Glossário de termos técnicos**. [S.l.]: Traço, 1979.
- BOUABDALLAH, S. **Design and control of quadrotors with application to autonomous flying**. [S.l.], 2007.
- BRASIL. **glossário de termos técnicos de aviação civil**. Rio de Janeiro, 1989.
- DJI. **Phantom 4**. 2019. Disponível em: <<https://www.dji.com/br/phantom-4>>.
- GREWAL, M.; ANDREWS, A. Kalman filtering: theory and practice using matlab. **New York: John Wiley and Sons**, v. 14, 01 2001.
- GUIMARÃES, J. P. F. **Controle de atitude e altitude para um veículo aéreo não tripulado do tipo quadricóptero**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2012.
- HIGGINS, W. T. A comparison of complementary and kalman filtering. **IEEE Transactions on Aerospace and Electronic Systems**, IEEE, n. 3, p. 321–325, 1975.
- KOTTENSTETTE, N.; PORTER, J. Digital passive attitude and altitude control schemes for quadrotor aircraft. In: IEEE. **2009 IEEE International Conference on Control and Automation**. [S.l.], 2009. p. 1761–1768.
- MADGWICK, S. O.; HARRISON, A. J.; VAIDYANATHAN, R. Estimation of imu and marg orientation using a gradient descent algorithm. In: IEEE. **2011 IEEE international conference on rehabilitation robotics**. [S.l.], 2011. p. 1–7.
- MAHONY, R.; HAMEL, T.; PFLIMLIN, J.-M. Nonlinear complementary filters on the special orthogonal group. **IEEE Transactions on automatic control**, v. 53, n. 5, p. 1203–1217, 2008.
- MARANTOS, P.; KOVEOS, Y.; KYRIAKOPOULOS, K. J. Uav state estimation using adaptive complementary filters. **IEEE Transactions on Control Systems Technology**, IEEE, v. 24, n. 4, p. 1214–1226, 2015.
- MARINA, H. G. D. et al. Uav attitude estimation using unscented kalman filter and triad. **IEEE Transactions on Industrial Electronics**, IEEE, v. 59, n. 11, p. 4465–4474, 2011.
- MEDEIROS, F. A. et al. Utilização de um veículo aéreo não-tripulado em atividades de imageamento georeferenciado. **Ciencia rural**, Universidade Federal de Santa Maria, v. 38, n. 8, 2008.
- MOREIRA, M. G. et al. Construção de uma aeronave remotamente pilotada de baixo custo para aplicações na agricultura de precisão. In: CONBAP. **Congresso Brasileiro de Agricultura de Precisão- ConBAP 2018**. [S.l.], 2018. p. 356–361.
- NETO, A. W. **Estudo de filtros de kalman aplicados à estimação de estados em um processo de quatro tanques**. Dissertação (TCC) — Universidade de Brasília, 2017.

NONAMI, K. Prospect and recent research & development for civil use autonomous unmanned aircraft as uav and mav. **Journal of system Design and Dynamics**, The Japan Society of Mechanical Engineers, v. 1, n. 2, p. 120–128, 2007.

OSDER, S. S.; ROUSE, W. E.; YOUNG, L. S. Navigation, guidance, and control systems for v/stol aircraft. 1973.

RODRIGUES, H. K. **Elaboração de Projeto para Construção de Quadricoptero Autônomo Eficiente**. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, nov 2014.

SABATINI, A. M. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. **IEEE Transactions on Biomedical Engineering**, IEEE, v. 53, n. 7, p. 1346–1356, 2006.

SHOEMAKE, K. Animating rotation with quaternion curves. In: ACM. **ACM SIGGRAPH computer graphics**. [S.l.], 1985. v. 19, n. 3, p. 245–254.

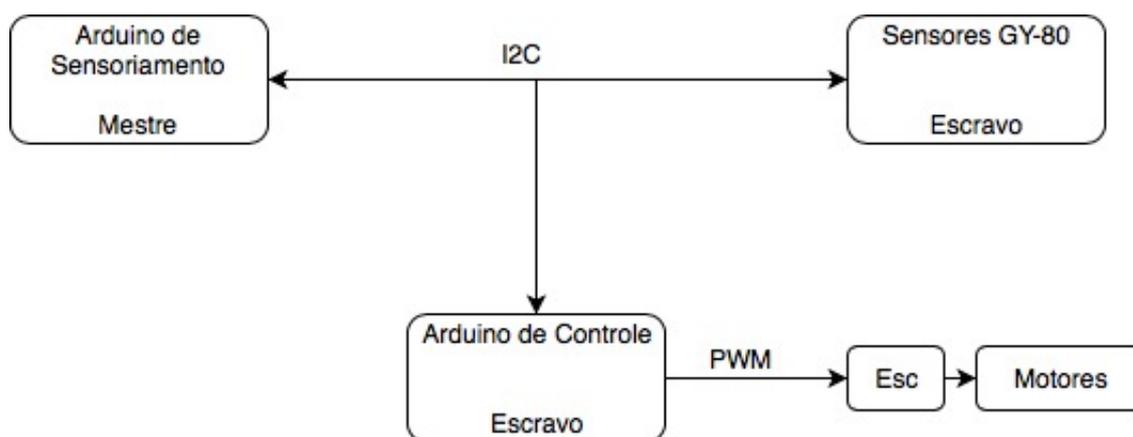
SILVA, D. C. D. et al. Qualidade de ortomosaicos de imagens de vant processados com os softwares aps, pix4d e photoscan. **V Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, Recife-PE**, 2014.

VICCI, L. Quaternions and rotations in 3-space: The algebra and its geometric interpretation. **University of North Carolina at Chapel Hill, Chapel Hill, NC**, Citeseer, 2001.

#### 4 ARTIGO

Neste capítulo é apresentado o artigo "Implementação de um sistema de estimação de atitude e direção baseado em quatérniões para veículos aéreos não tripulados", proposta a ser submetida na revista digital IEEE Sensors. O proposto módulo poderá ser adicionado como um periférico a um segundo Arduino possuindo o código do controlador de uma RPA. O código implementado por MOREIRA et al. (2018) é um controlador básico de estabilidade de drone desenvolvido como parte do projeto de Iniciação Científica da Universidade Federal de Lavras. A implementação consiste na integração do sensor com o protocolo I2C. Esta implementação torna-se extremamente simples com o uso da biblioteca Wire.h. Esta biblioteca nativa do Arduino implementa o protocolo de comunicação I2C e já é usada pelo programa para fazer a comunicação com os sensores propostos no artigo. O protocolo de comunicação I2C possui um dispositivo mestre que gerencia o clock das operações e faz requisições, quantos outros dispositivos escravos que recebem dados, e enviam dados após receber requisições. O diagrama da Figura 4.1 apresenta uma possível estrutura de comunicação entre os Arduinos de controle e sensoriamento. O arduino de controle é responsável por executar os diversos controles PID, e enviar comandos de velocidade aos motores. O Arduino de sensoriamento é responsável por coletar os dados dos sensores, tratar e filtrar esses sinais e então enviá-lo ao Arduino de controle.

Figura 4.1 – Diagrama de conexão dos arduinos de controle e sensoriamento



Fonte: Autoria Própria

## Implementação de um sistema de estimação de atitude e direção baseado em quaterniões para veículos aéreos não tripulados.

Moreira G. Marcello<sup>1</sup>

<sup>1</sup>Engenharia de Controle e Automação, Universidade Federal de Lavras, Lavras – MG, Brasil

**Resumo**— Veículos aéreos não tripulados têm sido utilizados como ferramentas nas áreas de mapeamento e agrimensura. Nos últimos tempos, esses equipamentos vêm se tornando mais populares em diversos setores à medida que o custo e dimensões dos mesmos vêm sendo reduzidos. No entanto, construir um sistema de estabilização robusto e eficiente com um baixo consumo de energia é um grande desafio tecnológico para esses dispositivos. Sensores de tamanho e custo reduzidos são altamente ruidosos. Algoritmos de filtragem de sinal e técnicas de fusão de sensores são necessários para que os valores das respostas das variáveis medidas sejam aceitáveis. No entanto, esses algoritmos demandam um alto custo de processamento para fornecer resultados em tempo real. Além disso, os processadores devem ser de baixo consumo de energia, o qual impõe restrições computacionais. Visando melhorar as restrições de custo computacional, é proposto neste trabalho o uso de quaterniões na representação das rotações espaciais do sistema de atitude e direção da aeronave. A operação de rotação clássica utiliza cálculos matriciais dos ângulos de Euler que são computacionalmente caros comparados às operações com quaterniões. Utilizando a representação de quaterniões, são apresentados três algoritmos que fazem fusão de sensores para estimar orientação de uma aeronave. Estes algoritmos são implementados em um microcontrolador de 8 bits, e é feita a comparação de desempenho entre eles. Para isto primeiro foram comparados no MATLAB três filtros capazes de fundir os dados de sensores cinético e magnético e estimar a orientação da aeronave. Em seguida esses filtros foram implementados em um Arduino Uno para verificar sua taxa de atualização e eficiência. Após a comparação foi percebido que o filtro de Kalman Estendido apresentou os melhores resultados teóricos. Porém no momento a implementação não possível implementado em um ambiente restringido como o Arduino Uno. Como segunda alternativa, o filtro de Madgwick foi utilizado. Apesar de possuir precisão inferior ao Filtro de Kalman Estendido, o filtro de Madgwick é suficientemente preciso para atender as necessidades de estimação de estados para atender uma aeronave remotamente pilotada.

**Palavra-chave**— Filtragem de Sinais, Aeronaves Remotamente Pilotadas, Fusão de Sensores

### I. INTRODUÇÃO

Aeronave remotamente pilotada (do inglês Remotely Piloted Aircraft - RPA) significa uma aeronave não tripulada, pilotada a partir de uma estação remota para uso não recreativo [1]. Estes dispositivos são comumente chamados de Veículos Aéreos não Tripulados (VANT) ou simplesmente drones. Eles foram ferramentas criadas inicialmente para fins militares, sendo utilizados em missões de reconhecimento onde era muito arriscado o envio de pilotos [9]. O desenvolvimento de controladores eletrônicos de tamanho reduzido e baixo consumo de energia permitiu a construção de RPAs com dimensões cada vez menores e de custo acessível. Técnicas avançadas na estabilidade e controle digital implementados no controlador de voo dessas aeronaves permitem a fácil operação por pessoas com pouco ou nenhum conhecimento [5]. Para controlar a aeronave, os sensores usados devem ser robustos, resistentes a interferências e, se possível, redundantes, que são características comuns na aeronáutica, onde é necessário um alto grau de confiabilidade do sistema. Um sistema referencial de atitude e orientação (em inglês *Attitude and Heading Reference System, AHRS*), consiste em um sistema capaz de fornecer informações de orientação nos três eixos da aeronave e informações

da direção com relação ao norte magnético da terra. No entanto, [6] afirma que RPAs pequenas têm limitações quanto ao uso de soluções da aeronáutica tradicional para resolver problemas de estimação de atitude (orientação angular nos três eixos) e posição. Por exemplo, os giroscópios de navegação precisos são grandes demais para caber nessas aeronaves. Como solução às restrições de tamanho e ruído de sensores, técnicas de fusão de sensores e filtragem de sinais são empregadas em conjunto com sensores de tamanho reduzido para estimar a posição e atitude de RPAs de pequeno porte [5]. Existe também, a redução da densidade de força nos motores e queda de eficiência na transmissão devido ao aumento da dominância do atrito em engrenagens e rolamentos [6]. Nos RPAs com sistema de propulsão do tipo quadricóptero, a eficiência das hélices também é reduzida [5]. Para atenuar essas restrições que impactam diretamente na autonomia das aeronaves, os fabricantes de RPAs reduzem ao máximo o peso dos componentes do veículo. Os algoritmos de controle das RPAs que são processados pelo microcontrolador principal da aeronave são de extrema importância para manter a estabilidade de voo [9]. As RPAs, por usarem controladores digitais têm o desempenho influenciado pelo tempo de amostragem do controlador. Mudanças na taxa de amostragem não somente alteram a natureza da resposta de superamortecida para subamortecida, mas também podem transformar um sistema estável

em um sistema instável [8]. Isto significa que existe um tempo de amostragem mínimo onde o controlador digital é capaz de manter o sistema estável. Esta propriedade determina a restrição do período de amostragem que o sistema de controle da aeronave precisa ter. Essa característica de limite de tempo para as operações do sistema o define como um sistema em tempo real “do inglês *hard real time*”, impõe um limite de processos a serem realizados. Para reduzir este impacto, são utilizadas técnicas de programação paralela ou múltiplos processadores e microcontroladores dedicados [14].

Estimadores de estados de atitude e direção usualmente trabalham com matrizes contendo cossenoides ou senoides parametrizadas por ângulos de Euler [4]. Como sensores nos RPAs, são utilizados: acelerômetros de três eixos, giroscópio de três eixos e magnetômetro de três eixos. A abordagem clássica de matrizes de rotação utilizada para determinar os ângulos de rotação da aeronave apresentam singularidades conhecidas como *Gimbal Lock* [16]. Esta singularidade essencialmente é o momento onde durante um grau de liberdade de um giroscópio de três eixos some. Matematicamente isto pode ser observado quando em uma sequência de rotações com matrizes de rotação, uma das matrizes resulta na matriz identidade. O uso de quatérnios apresenta-se como uma alternativa para solucionar esse problema. Quatérnios ao invés de ângulos Euler é uma abordagem interessante na representação da orientação, porque permitem trabalhar no domínio  $\mathbb{R}^4$  [20]. Os quatérnios evitam "Gimbal Lock" e em geral, são mais eficientes que ângulos Euler de um ponto de vista computacional [2].

Outro desafio encontrado na abordagem clássica na estimação da orientação de RPAs está relacionado ao uso de magnetômetros. Estes sensores são capazes de serem usados como bússolas digitais. Independentemente do método de estimação, é necessário a calibragem do sensor magnético para remover o campo gerado pela plataforma [15]. Um método apresentado por [17] sugere a remoção de todas as fontes de campo magnético artificiais que perturbam o campo magnético da terra. Porém, este método envolve detectores que geralmente falham em diferenciar perturbações do campo magnético terrestre [16]. Madgwick [10] apresentou em seu trabalho um eficiente algoritmo de captura de movimento humano capaz de reduzir o efeito de perturbação magnética. Um método de detecção do campo magnético local e calibração das medições a partir do viés medido é capaz de corrigir satisfatoriamente a interferência magnética provocada por metais próximos a aeronave [21].

Nas últimas décadas, muitos algoritmos de estimação de orientação que fazem a fusão de sensores inerciais e magnéticos têm sido desenvolvidos [4]. As principais abordagens usadas são filtros complementares e filtros de Kalman estendido (do inglês Extended Kalman Filter - EKF).

O Filtro de Kalman e sua versão estendida é geralmente utilizado na estimação da orientação de RPAs fundindo as saídas do giroscópio e do acelerômetro [18].

O projeto de um AHRS com filtro de Kalman estendido desenvolvido por [20] é um bom exemplo de uso do filtro. No trabalho, o autor afirma que o filtro de Kalman Estendido evita o processo de linearização do modelo dinâmico e reduz a carga computacional.

Os algoritmos eficientes na estimação de orientações propostos

neste trabalho são: algoritmo de Madgwick, Mahony e filtro de Kalman Estendido. O algoritmo Madgwick, apresentado em 2010 [10], faz uso de sensores inerciais e magnetômetro para estimar a orientação de um objeto com quatérnios, e uma constante única para definir a contribuição de cada sensor [7]. Um dos benefícios deste algoritmo é o baixo custo computacional, exigindo de 109 a 277 operações aritméticas a cada atualização [10]. O algoritmo de Mahony é um filtro complementar capaz de estimar a orientação baseando-se na junção das informações dos sensores acelerômetro, giroscópio e magnetômetro [11]. O filtro inicialmente proposto utilizando matrizes de rotação possui estrutura simples e é adequado para implementação em sistemas embarcados. Ao utilizar rotações com quatérnios, o custo computacional é ainda mais reduzido [11].

A proposta deste trabalho é implementar o melhor dentre os algoritmos de Madgwick, Mahony e Filtro de Kalman Estendido em um Arduino Uno para posteriormente viabilizar o uso em RPAs.

## II. CALIBRAÇÃO E COLETA DE DADOS

Inicialmente é programado no Arduino um código simples capaz de calibrar os sensores acelerômetro, giroscópio e magnetômetro. Para o acelerômetro, o valor de aceleração é calibrado baseado na aceleração gravitacional  $G$ . A equação (1) determina o valor calibrado do sensor no eixo  $x$ , fazendo uma interpolação linear. Desta maneira, os valores medidos pelo sensor ficam limitados ao intervalo -1 a 1. Essa equação é repetida para todos os demais eixos.

$$a_c^x = -1 + 2 * \frac{a_i^x - a_0^x}{a_1^x - a_0^x} \quad (1)$$

onde  $a_c$  é o valor calibrado e  $a_i$  o valor medido. Os valores  $a_1$  e  $a_0$  são respectivamente os valores máximo e mínimo detectados pelo sensor.

Para calibrar o giroscópio, é determinado o viés estático do sensor. Para isso, o sensor deve ficar completamente imóvel em uma superfície. Em seguida, 100 medições de cada eixo são coletadas. A média das medições de um único eixo determina o viés de medição do sensor para aquele eixo. A equação (2) determina o valor calibrado  $g_c$  e o valor medido  $g_i$  usando o viés de medição  $g_v$ .

$$g_c = g_i - g_v \quad (2)$$

Finalmente, a calibração do sensor magnético é muito mais complexa. Um algoritmo apresentado por [21] exige que sejam detectadas os valores mínimos e máximos de cada eixo do sensor. A equação (3) determina a média local de cada eixo.

$$\bar{m} = \frac{m_{max} + m_{min}}{2} \quad (3)$$

Subtraindo o valor da média do valor medido, é possível retirar o efeito dos campos magnéticos provocados por elementos que produzem campo magnético como fios conduzindo eletricidade e ímãs permanentes. Porém, apenas este valor não é capaz de corrigir o efeito provocado por materiais ferromagnéticos como aço. Portanto um segundo cálculo é realizado para retirar o efeito dos

campos magnéticos "macios". Usando o valor das médias já calculadas, uma constante  $M$  é obtida para normalizar os valores corrigidos pelo campo magnético inicial.

$$Avg = \frac{\bar{m}_x + \bar{m}_y + \bar{m}_z}{3} \quad (4)$$

$$M_x = \frac{\bar{m}_x}{Avg}; M_y = \frac{\bar{m}_y}{Avg}; M_z = \frac{\bar{m}_z}{Avg} \quad (5)$$

Os valores normalizados das medições do magnetômetro no eixo  $x$ ,  $m_c^x$  podem ser obtidos pelo valor medido  $m_i^x$  e a constante determinada em (5). O mesmo se aplica para os eixos  $y$  e  $z$ .

$$m_c^x = (m_i^x - \bar{m}_x) * M_x \quad (6)$$

Um programa em Arduino foi desenvolvido para calibrar e coletar os dados dos sensores giroscópio, acelerômetro e magnetômetro. Foi utilizado o Arduino Uno, que é um microcontrolador programável de 8 Bits. Este dispositivo de pequeno porte possui uma placa processadora ATmega328 com 32KB de memória flash e 2KB de memória RAM. O módulo de sensores GY-80 é um conjunto de sensores ao todo com 10 graus de liberdade. Composto por um grupo de três acelerômetros, três giroscópios, três magnetômetros e um barômetro. O acelerômetro ADXL345 possui resolução capaz de captar acelerações de até 16G. O sensor giroscópio L3G4200D é capaz de medir velocidade angular com escala de até 2000 graus por segundo. Por fim, o sensor magnetômetro HMC5883L é sensível ao fluxo magnético. Esses sensores comunicam através do protocolo I2C e permitem a fácil conexão com o Arduino Uno.

Juntamente com os dados dos sensores, foi coletado o tempo de amostragem para o ciclo de coleta e construção dos dados calibrados dos sensores. A média do tempo de amostragem para coleta de dados dos três sensores é de 5 milissegundos, o que limita a frequência máxima de ciclo para 200Hz. Após as rotinas de calibração que devem ser executadas antes da coleta de dados, os valores máximos e mínimos do sensor acelerômetro e magnetômetro, e os valores de viés do giroscópio são salvos na memória EEPROM interna do microcontrolador para serem utilizadas pelos demais programas.

### III. ESTABELECEENDO REFERENCIAL E QUATERNIÕES

O quaternião  $q$  é um vetor  $\mathbb{R}^4$  pode ser usado para definir a orientação do referencial  $b$  em relação ao referencial global  $n$ . Seja determinado um referencial inercial global  $\{n\}$ , que adota o padrão *ENU* (do inglês *East-North-Up*). Seja o referencial  $\{b\}$  fixo ao corpo da RPA adota o padrão *NED* (do inglês *North-East-Down*) que tem como origem o centro de massa da aeronave. A partir desses dois referenciais, é então determinado uma matriz de rotação  $R$  que faz a transformação dos ângulos do referencial  $n$  para  $b$ .

$$A^b = RA^b \quad (7)$$

Seja  $r$  um vetor tridimensional tal que, quando o referencial  $n$  é rotacionado  $\alpha$  graus em torno do eixo definido por  $r$ , obtemos o referencial  $o$ .

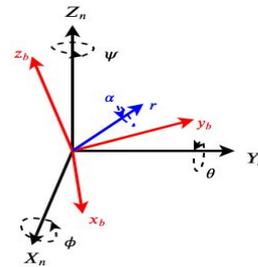


Fig. 1. Representação dos referenciais globais e locais em relação ao vetor tridimensional  $r$ .

É definido um quaternião  $q$ , que representa uma rotação de  $\alpha$  em torno de  $r$  como,

$$q = [q_0 \ q_1 \ q_2 \ q_3] = \left[ \cos \frac{\alpha}{2} \quad r_x \sin \frac{\alpha}{2} \quad r_y \sin \frac{\alpha}{2} \quad r_z \sin \frac{\alpha}{2} \right] \quad (8)$$

[21] definem em seu trabalho a relação de transformação entre os ângulos Euler e o quaternião  $q$  usando a expressão da matriz de cossenos direta.

$$\begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} atan2(2q_2q_3 + 2q_0q_1, q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ -asin(2q_1q_3 - 2q_0q_2) \\ atan2(2q_1q_2 + 2q_0q_3, q_1^2 + q_0^2 - q_3^2 - q_2^2) \end{bmatrix} \quad (9)$$

Substituindo assim a necessidade da matriz de rotação  $R$ . Deve se atentar que, pela matriz de transformação ser composta por funções trigonométricas, a relação da sua derivada não é linear.

Portanto a derivada do quaternião  $q$  e a velocidade angular  $w$  é descrita pela equação.

$$\dot{q} = \frac{1}{2} \times q \otimes w \quad (10)$$

#### A. Algoritmo de Filtro de Kalman Estendido

A maioria dos modelos do processo do filtro de Kalman Estendido para esse tipo de aplicação são baseados na modelagem de [2]. O modelo do processo que é usado neste trabalho será o mesmo proposto por [20] e está apresentado na figura 2.

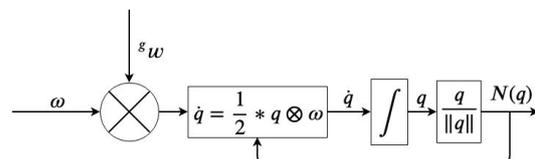


Fig. 2. Diagrama de blocos do sistema dinâmico de rotação de um corpo.

No diagrama,  $g_w$  é o ruído do sensor giroscópio,  $w$  a velocidade angular,  $q$  é o quaternião de orientação e  $\dot{q}$  sua derivada respectivamente, e  $N(q)$  é o quaternião normalizado. Em [20], é apresentado um filtro de Kalman Estendido que utiliza um algoritmo de otimização de gradiente descendente. Inicialmente, é utilizada a abordagem de gradiente descendente com passo adaptável (do inglês



estimações de orientação e erro do viés do giroscópio em tempo real.

A representação em quaternião do filtro complementar explícito é descrito pelas equações (19) a (21).

$$\omega_{mes} = -vex \left( \sum_{i=1}^n \frac{k_i}{2} (v_i \hat{v}_i^T - \hat{v}_i v_i^T) \right) \quad (19)$$

$$\dot{\hat{q}} = \frac{1}{2} \hat{q} \otimes p(\Omega_k - \hat{b} + k_p \omega_{mes}) \quad (20)$$

$$\dot{\hat{b}} = -k_i \omega_{mes} \quad (21)$$

A implementação de Madgwick do filtro de Mahony estima a orientação dos campos gravitacional e magnético com o quaternião inicial. Em seguida, o erro entre o campo identificado pelo sensor e o estimado pelo quaternião e inserido como entrada em um observador. Os ganhos  $K_p$  e  $K_i$  do observador dão uma estimação da velocidade angular do corpo. Finalmente essa estimação da velocidade é adicionada as medições do giroscópio e usadas para calcular um novo quaternião estimado. Deste modo, obtendo uma estimação rápida e precisa da orientação do corpo. Na figura 6 está representado o diagrama de blocos do filtro de Mahony.

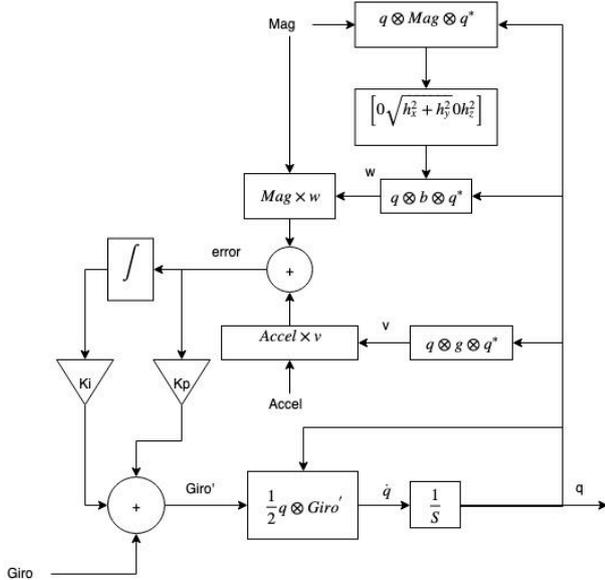


Fig. 6. Diagrama de blocos do filtro de Mahony.

#### IV. RESULTADOS E DISCUSSÃO

O sensor é fixado em um RPA de teste para facilitar a parametrização das rotações. A Figura 7 apresenta a montagem do módulo de sensores GY-80 e o Arduino Uno utilizados neste experimento. Duas etapas foram realizadas para obter os resultados deste trabalho. A primeira, foi a coleta de dados do modulo GY-80 com uma sequencia de operação padronizada, e em seguida a avaliação do desempenho dos três filtros propostos utilizando a ferramenta MATLAB para realizar a filtragem do sinal e execução dos algoritmos. Posteriormente, foram implementados no Arduino Uno os filtros propostos para avaliar a taxa de atualização de cada um deles.

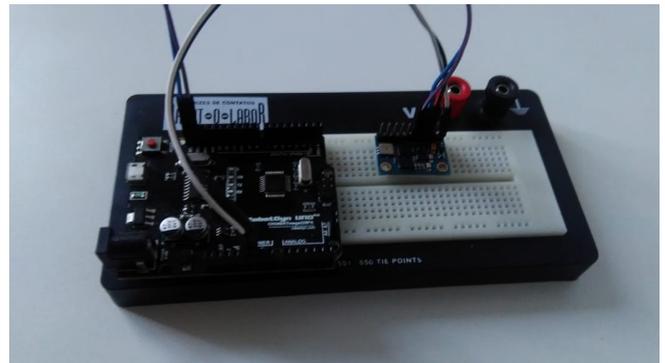


Fig. 7. Montagem do sensor junto ao Arduino em corpo de testes.

Uma sequência de operações de rotação foi realizada para obter um vetor de testes dos sensores. A sequência é composta por uma rotação de 90 graus em X, seguida de uma rotação de -180 graus em X, e uma rotação de 90 graus em X novamente. Deste modo o corpo de teste fica plano a superfície da mesa, em seguida apoiado na sua lateral direita, a seguir apoiado na sua lateral esquerda, e por fim plano a superfície novamente. Este mesmo procedimento de rotação é realizado para os eixos Y e Z do corpo de teste. A figura 8 apresenta os dados coletados pelos sensores giroscópio, acelerômetro e magnetômetro quando a sequencia de testes descrita acima é executada.

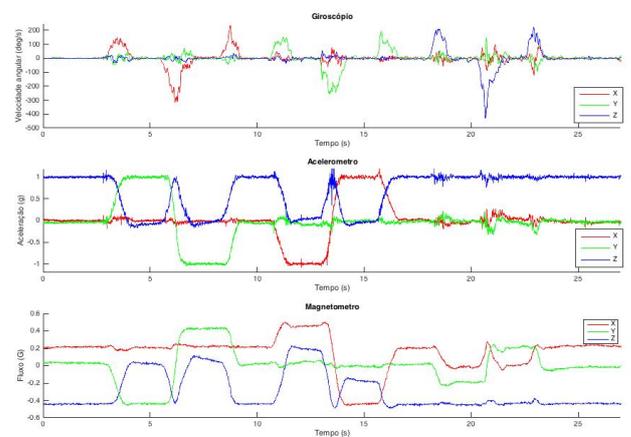


Fig. 8. Vetor de testes dos sensores

A orientação do sensor pode ser observada avaliando a relação entre os dados do sensor giroscópio e acelerômetro. Inicialmente com o corpo de prova em repouso em uma superfície plana, observa-se que o eixo Z do acelerômetro indica 1G. O que significa que a orientação do eixo Z está apontada para baixo. Em seguida ao rotacionar o corpo 90° no eixo X, observado pelo salto na curva X do acelerômetro, há a troca da aceleração de 1G de Z para Y no giroscópio. Isto significa que com essa rotação o eixo Y está apontado para baixo. Com essas duas informações e segundo a regra da mão direita, deduzimos que no estado de repouso inicial, Z aponta para baixo, X para frente do corpo e Y para o lado direito. Isto indica que o conjunto de sensores GY-80 utiliza a orientação North-East-Down (NED). Em seguida, as implementações dos

algoritmos de Madgwick, Mahony e EKF foram usadas para processar esses vetores. Os algoritmos sempre trabalham com rotações e ângulos representados como quaterniões, no entanto a visualização dessas rotações não é intuitiva. Portanto esses quaterniões estimados pelos sensores são convertidos em ângulos Euler para facilitar sua interpretação. Deve atentar-se que durante a conversão de quaterniões para Euler, o fenômeno de *Gimbal Lock* surge. Ao realizar as três rotações descritas pelo quaterniões em sucessão, na tentativa de formar os ângulos Euler, durante as duas primeiras rotações,  $\psi$  se alinha com um dos demais eixos. Quando isto acontece, o sistema perde efetivamente um grau de liberdade. O trabalho de [3] elabora o problema de *Gimbal Lock*. As Figuras de 9 a 11 apresentam o resultado dos quaterniões estimados pelos algoritmos de Madgwick, Mahony e EKF respectivamente ao utilizar como entrada o vetor de testes descrito no início desta seção e apresentados na Figura 8 quando executados utilizando a ferramenta MATLAB.

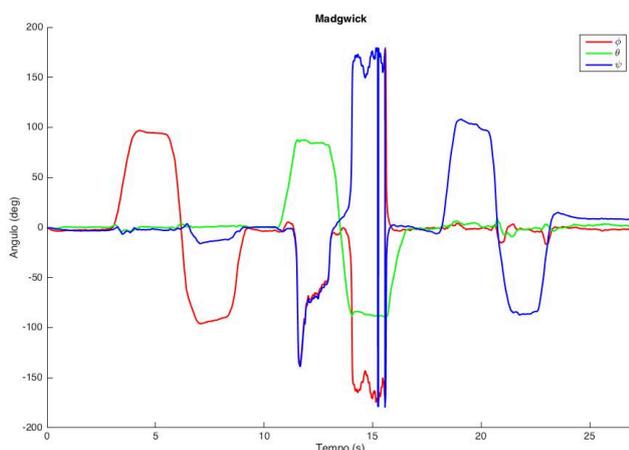


Fig. 9 Ângulos Euler convertidos pelo quaterniões calculado pelo algoritmo de Madgwick.

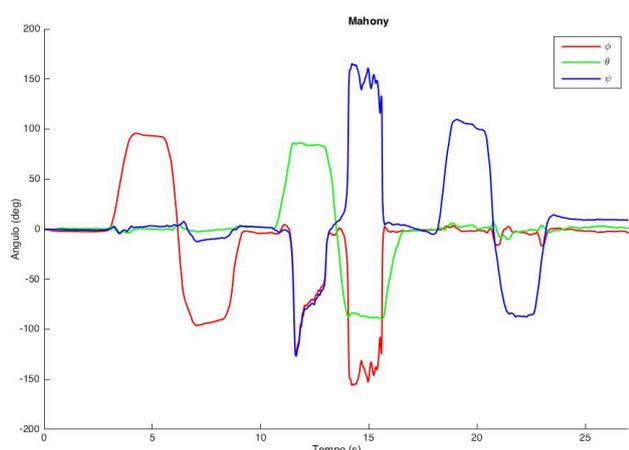


Fig. 10. Ângulos Euler convertidos pelo quaterniões calculado pelo algoritmo de Mahony.

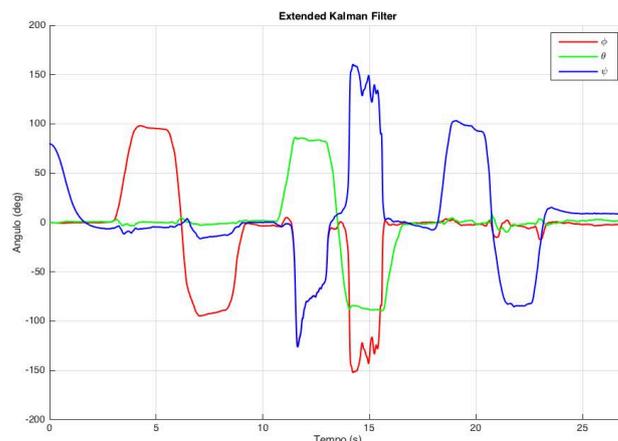


Fig. 11. Ângulos Euler convertidos pelo quaterniões calculado pelo algoritmo de EKF.

Observando as figuras de 9 a 11, percebe-se o efeito do *Gimbal Lock* quando ocorre a rotação em  $\theta$ . Percebe-se que após o instante 10, a curva verde transaciona de  $0^\circ$  a  $+90^\circ$ , ambas as curvas vermelha e azul se comportam de maneira errática. Este comportamento é o fenômeno de *Gimbal Lock*. Este evento continua até a curva verde retornar à posição  $0^\circ$ . No entanto, o fenômeno não afeta a orientação calculada no quaterniões, apenas a representação em Euler do gráfico. Esta característica pode ser identificada observando o comportamento do filtro após alcançar o suposto estado de *gimbal lock*. Caso o quaterniões tivesse entrado neste estado, após o instante 15 segundos, o filtro não seria capaz de se recuperar e voltar a medir a orientação do objeto. Observa-se no entanto, mesmo que a representação em Euler tenha entrado em *gimbal lock*, o filtro em si não sofre esse efeito, pois a representação em Euler retorna ao normal assim que o corpo deixa a posição angular  $\theta = -90$ .

Somente analisando visualmente as figuras 9 a 11, não é possível determinar qual dos três algoritmos obteve o melhor desempenho. Para avaliar a qualidade de cada observador, é usado uma função de erro quadrático médio nos instantes onde não há movimento do sensor, e o corpo de testes está no plano da superfície da mesa. Quatro trechos são identificados nos quais, o corpo de teste está nivelado com relação a superfície da mesa. Os trechos são separados em T1, T2, T3 e T4 como apresentado na tabela 1.

Tabela 1: Trechos para análise de erro quadrático médio.

Trecho	Início (s)	Fim (s)
T1	0	2,5
T2	9,3086	10,3789
T3	16,8203	17,8203
T4	23,8633	27,1641

Tabela 2: Erro quadrático médio

Método	T1	T2	T3	T4	Soma	Média
Madgwick	2,677	2,881	1,013	1,960	8,533	2,133
Mahony	2,089	3,501	1,094	2,171	8,856	2,214
EKF	0,264	2,750	0,427	2,220	5,662	1,415

A tabela 2 mostra o erro quadrático de cada algoritmo em cada

um dos trechos, e avalia a média geral do erro. O erro quadrático é calculado utilizando apenas o eixo  $\phi$ . No entanto, o cálculo poderia usar o eixo  $\psi$  e  $\theta$  desde que o corpo sempre voltasse a posição inicial. Como o corpo de teste utilizado possui uma maior dimensão ao longo do eixo X, há maior garantia que este eixo sempre ficará nivelado ao repousar o corpo sobre uma superfície plana. Avaliado as informações da tabela 2, é possível determinar qual dos algoritmos possui o menor erro quadrático. Quanto menor for o erro, mais preciso é o algoritmo. O erro médio permite visualizar de forma geral qual dos algoritmos obteve a melhor precisão. Fica evidente que o algoritmo de Kalman apresenta os melhores resultados na estimação da orientação do corpo de testes, seguido pelo algoritmo de Madgwick e por fim o algoritmo de Mahony.

A seguir é feita a implementação e análise da frequência de atualização para cada algoritmo implementado no microcontrolador Arduino Uno. Os algoritmos são programados em C e compilados no microcontrolador utilizando a IDE Arduino. A comunicação entre os sensores e o Arduino é feita utilizando o protocolo I2C. Ao compilar o programa utilizando a IDE, a mesma mostra o espaço gasto pelo programa na memória de armazenamento Flash do Arduino. Também é mostrado pela IDE a quantidade de memória dinâmica que é reservada para variáveis de programa, estas são alocadas na memória SRAM do Arduino. O tempo de ciclo de cada interação dos algoritmos é mostrado no monitor serial da plataforma de desenvolvimento do Arduino. A tabela 3 apresenta as informações de implementação de cada algoritmo quando gravado.

Tabela 3: Implementação dos algoritmos

Método	Armazenamento (%)	Memória dinâmica (%)	Frequência de atualização (Hz)
Madgwick	59	26	114.4296
Mahony	59	27	131.6145
EKF	115	103	-

A coluna do armazenamento informa a porcentagem de espaço usada pelo programa dos 32KB de memória Flash disponíveis no Arduino Uno. A coluna de Memória dinâmica informa a porcentagem de memória usadas pelas variáveis do programa, dos 2KB de memória SRAM disponíveis no Arduino Uno. Pode ser observado na tabela 3 que o algoritmo de Kalman exige mais memória dinâmica e armazenamento que o disponível no microcontrolador. Exigindo efetivamente 15% de espaço de armazenamento e 3% a mais de memória SRAM. Seria necessário portanto a aquisição de um microcontrolador com maior espaço de armazenamento e memória. Isso se dá devido a necessidade de declarar matrizes 4x4. Cada matriz declarada exige um vetor de 16 variáveis do tipo float. Uma única variável float ocupa 4 bytes. Uma única matriz exige então 64 bytes da memória de armazenamento. Durante a execução do código, boa parte da memória dinâmica é reservada para operações aritméticas, que são executadas com dupla precisão. Isto significa que durante a execução, as variáveis momentaneamente duplicam a memória ocupada. A constante necessidade de recursão para operações matriciais também aumenta o uso dessa memória dinâmica rapidamente. Essa característica de algoritmos que envolvem cálculos matriciais impede sua implementação em dispositivos com armazenamento e memória limitada. Porém, a utilização de microcontroladores mais potentes

não é prevista neste trabalho. Logo, o algoritmo de Kalman é desconsiderado das demais análises.

Analisando o desempenho de ambos os algoritmos foi observado um fenômeno quanto ao algoritmo de Mahony. As figuras 12 e 13 apresentam as saídas convertidas para ângulos Euler dos algoritmos de Mahony e Madgwick respectivamente capturadas da IDE do Arduino. Essas saídas foram extraídas durante a execução do algoritmo dentro do próprio Arduino. As linhas azul, vermelha e laranja representam respectivamente os ângulos de arfagem, rolagem e guinada respectivamente. A linha laranja é apenas uma referência do ângulo  $0^\circ$ . Também foi observado que ambos os algoritmos de Madgwick e Mahony apresentaram frequência de atualização muito próximas uma da outra.

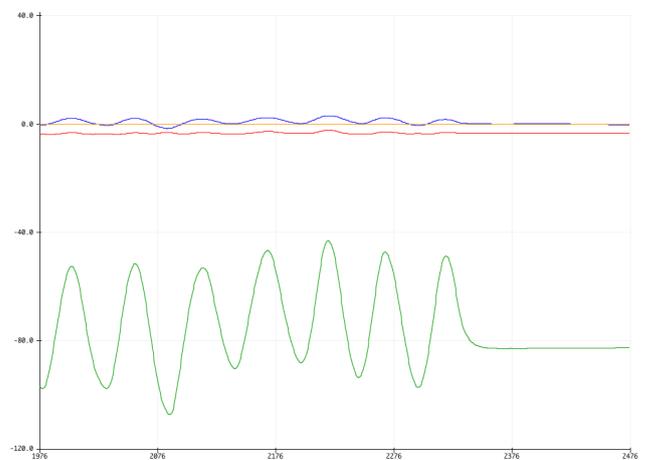


Fig. 12. Oscilações em Z - Algoritmo de Mahony implementado no Arduino Uno.

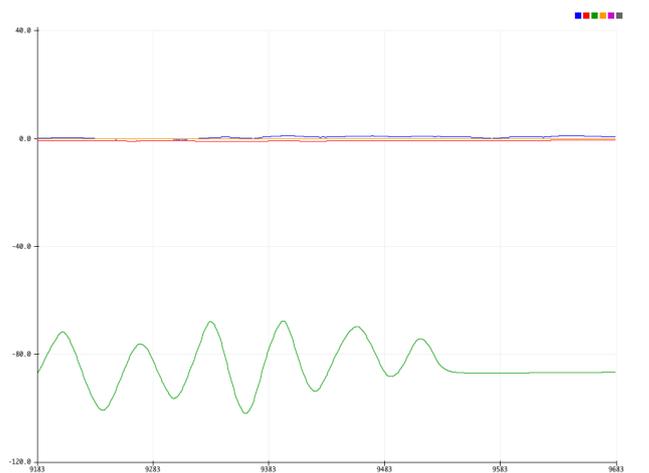


Fig. 13. Oscilações em Z - Algoritmo de Madgwick implementado no Arduino Uno.

Para obter as figuras 12 e 13, o corpo de prova foi posicionado em uma superfície plana. Com o plano inferior do corpo tocando a superfície da mesa, o corpo de teste foi rotacionado sob o seu eixo vertical. Observou-se que quando acontecem oscilações no eixo de guinada (curva verde), os demais eixos também sofrem uma

influência dessa oscilação em ambos os algoritmos de Mahony e Madgwick. No entanto, o algoritmo de Mahony apresenta significativa maior influência dessa oscilação. Como a estimativa do ângulo no eixo de guinada está diretamente ligado com o sensoriamento do campo magnético, esta característica sugere que o algoritmo de Mahony é mais sensível a distúrbios do campo magnético. Esse distúrbio provoca oscilações na estimativa dos ângulos de arfagem e rolagem. Tais oscilações são indesejadas pois podem desestabilizar o voo da aeronave. Deve-se lembrar que o resultado apresentado nas figuras 12 e 13 são as conversões para ângulos Euler das mesmas rotações em quatérnios. Finalmente, ao analisar todos os resultados obtidos, observa-se que o algoritmo de Madgwick apresenta-se como o mais indicado para o desenvolvimento de um sistema AHRS em um microprocessador de 8 bits. O algoritmo apresentou uma taxa de atualização de aproximadamente 100Hz, boa representação das rotações e relativa resistência a distúrbios do campo magnético.

## V. CONCLUSÃO

Com a implementação dos algoritmos para estimação da rotação de um drone a partir da leitura de sensores ruidosos concluiu-se os seguintes aspectos. Não é possível implementar o filtro de Kalman Estendido em uma Arduino Uno com 2KB de memória SRAM e 32KB de armazenamento devido a grande demanda de memória e armazenamento exigida por este algoritmo. As implementações dos algoritmos de Madgwick e Mahony nas suas versões utilizando quatérnios apresentam bons resultados pois são capazes de estimar a orientação com uma taxa de atualização de 100 Hz. Usando o algoritmo de Madgwick, é possível evitar a interferência do campo magnético na estimação da arfagem e rolagem. A representação em quatérnios possibilita a implementação desses dois algoritmos em um sistema embarcado 8 bits com 16MHz de clock atingindo uma frequência de atualização acima de 100Hz. Apesar dos sensores usados no experimento serem de baixa qualidade e produzirem ruído, os filtros são capazes de estimar a atitude e orientação de maneira suficientemente aceitável para permitir o uso em aeronaves remotamente pilotadas. As características da fusão de sensores evita o efeito de deslizamento provocado pela integração da velocidade angular ruidosa do sensor giroscópio. No entanto, o algoritmo de Mahony se mostrou mais sensível a distúrbios no campo magnético interferindo na estimativa dos ângulos de arfagem e rolagem do corpo. Como a estimativa desses ângulos é extremamente importante para garantir a operação da aeronave, o algoritmo de Madgwick apresenta-se como a melhor opção neste quesito. Este estimador implementado que utiliza o filtro de Madgwick possui uma frequência de atualização de aproximadamente 100 Hz, que é suficientemente rápida para garantir a estabilização do sistema. Portanto, o uso do filtro de Madgwick para estimar a rotação e atitude de uma aeronave remotamente pilotada apresentou-se como a opção mais viável dentre os três algoritmos propostos neste trabalho. Vale ressaltar que atualmente existem microcontroladores de 32 bits de baixo custo e baixo consumo de energia que podem ser usados para implementar algoritmos com melhor desempenho como o filtro de Kalman estendido. O microcontrolador STM32 F103C8T6 é uma opção de custo, dimensões e consumo energético similar às do

Arduino utilizado, porém com um poder de processamento superior. Desta maneira, sugere-se em trabalhos futuros a implementação de EKF no mencionado microcontrolador e podem ser adicionados sensores como GPS e barômetro para melhorar o desempenho. A adição de um filtro com GPS e barômetro acoplados permitiria estimar a posição tridimensional do veículo. Com isso seria possível “travar” o veículo em uma posição no espaço, se assimilando com o modo de pilotagem comum a RPAs comerciais do mesmo tipo.

## REFERENCIAS

- [1] Agência Nacional de Aviação Civil (2017), Regulamento Brasileiro de Aviação Civil Especial nº 94. Resolução nº 419.
- [2] Alaimo A, Artale V, Milazzo C, Ricciardello A. (2013). Comparison between Euler and Quaternion Parametrization in UAV Dynamics. AIP Conference Proceedings.
- [3] Brezov D, Mladenova C, Mladenov I. (2013). New Perspective on the Gimbal Lock Problem. AIP Conference Proceedings. 1570. 367-374.
- [4] Feng, K, Li J, Zhang X, Shen C, Bi Y, Zheng T, Liu J. (2017). A new quaternion-based Kalman filter for real-time attitude estimation using the two-step geometrically-intuitive correction algorithm. *Sensors*, 17(9), 2146.
- [5] Floreano D, Wood R J. (2015). Science, technology and the future of small autonomous drones. *Nature*, 521(7553), 460.
- [6] Kingston D, Beard R. (2004) Real-time attitude and position estimation for small UAVs using low-cost sensors. In: AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit.
- [7] Madgwick S O, Harrison A J, Vaidyanathan R. (2011) Estimation of IMU and MARG orientation using a gradient descent algorithm. In Proceedings of the 2011 IEEE International Conference on Rehabilitation Robotics (ICORR), Zurich, Switzerland.
- [8] Nise N S. (2013) Engenharia de sistemas de controle; Tradução e revisão técnica Jackson Paul Matsuura. - 6. ed. Rio de Janeiro: LTC.
- [9] Nonami K. (2007) Prospect and recent research & development for civil use autonomous unmanned aircraft as UAV and MAV. *Journal of System Design and Dynamics*, v. 1, n. 2, p. 120-128.
- [10] Madgwick S. (2010) An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK), v. 25, p. 113-118.
- [11] Mahony R, Tarek H, Michel J P. (2008) Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*, Institute of Electrical and Electronics Engineers.
- [12] Qin Y, et al. (2012) Kalman filter and integrated navigation principle. 2nd ed. Xi-an, Shan Xi, China: Northwest Industrial University Press.
- [13] Qin T, et al. (2013) Error Compensation Method for Electronic Compass Based on Best Ellipse-Matching Error Compensation Algorithm. *Chinese Journal of Sensors and Actuators*.
- [14] Ramamritham, K., & Stankovic, J. A. (1994). Scheduling algorithms and operating systems support for real-time systems. *Proceedings of the IEEE*, 82(1), 55-67.
- [15] Renaudin V, Afzal M H, Lachapelle G. (2010) Complete Triaxis Magnetometer Calibration in the Magnetic Domain. *J. Sens.*
- [16] Renaudin V, Combettes C. (2014) Magnetic, Acceleration Fields and Gyroscope Quaternion (MAGYQ)-Based Attitude Estimation with Smartphone Sensors for Indoor Pedestrian Navigation. *Sensor*.
- [17] Roetenberg D, Luinge H J, Baten C T, Veltink P H. (2005). Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.
- [18] Sabatini A M. (2006) Quaternion-based extended Kalman Filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering*.
- [19] Roetenberg D, Luinge H J, Baten C T, Veltink P. (2005) Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE Trans. Neural Syst. Rehabil. Eng.*
- [20] Wang L, Zhang Z, Sun P. (2015). Quaternion-Based Kalman Filter for AHRS Using an Adaptive-Step Gradient Descent Algorithm. *International Journal of Advanced Robotic Systems*.
- [21] Winer K. (2017) Simple and Effective Magnetometer Calibration. Github, Ed. 14 de Agosto. Acessado em 25 de maio de 2019. Disponível em

---

<https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration>.