

LUCAS PAULO BARBOSA DA SILVA



Escola Politécnica

Estimativa de orientação de veículos aéreos não tripulados a partir de IMU's de baixo custo utilizando matriz de cossenos diretores, filtros complementares e filtro de Kalman

São Paulo

2013

LUCAS PAULO BARBOSA DA SILVA

Estimativa de orientação de veículos aéreos não tripulados a partir de IMU's de baixo custo utilizando matriz de cossenos diretores, filtros complementares e filtro de Kalman

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Ciências.

Orientador:  
Prof. Dr. Eduardo L. Cabral

São Paulo  
2013

## **FICHA CATALOGRÁFICA**

**Silva, Lucas Paulo Barbosa da**

**Estimativa de orientação de veículos aéreos não tripulados a partir de IMU's de baixo custo, utilizando matriz de cossenos diretores, filtros complementares e filtro de Kalman / L.P.B. da Silvia. -- São Paulo, 2013.**

**65 p.**

**Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.**

**1.Filtros de Kalman 2.Processamento de sinais 3.Aeronaves I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.**

*Dedico este trabalho a todos que estiveram ao meu lado possibilitando que eu  
chegasse tão longe.*

## **AGRADECIMENTOS**

Agradeço à minha família e à minha companheira, que estiveram comigo nestes anos de graduação, apoiando escolhas, celebrando vitórias e contornando derrotas. Aos professores que se dispuseram a nos fornecer a melhor formação possível e se dedicaram a isso. Ao meu orientador que permitiu meu desenvolvimento e o deste trabalho. E ao Criador e Engenheiro dos Mundos que permitiu não apenas que contemplemos a sua criação, mas que tivéssemos ferramentas para compreendê-la.

## RESUMO

Este trabalho tem por objetivo estudar, comparar e implementar métodos de estimativa de orientação (atitude) de veículos aéreos não tripulados (drones), baseados em unidades de medições inercias (IMU) de 9D (Giroscópio 3D, Acelerômetro 3D e Magnetômetro 3D) de baixo custo.

Estimativa de orientação de drones tem como requisito o fato de esses veículos não terem uma direção de movimento preferencial entre a direção x e y. Desta forma sensores como medidores de velocidade do ar devem ser substituídos por magnetômetros para estimativa do ângulo de yaw.

Os métodos estudados neste trabalho são matriz de cossenos diretores, filtros complementares baseados em observadores de estados e filtro discreto de Kalman. Fatores como custo computacional, complexidade matemática, performance e consistência dos resultados são comparados. Para tal, são realizadas simulações em software gráfico matemático (MatLab), utilizando um modelo matemático em quaternions ou em ângulos de Euler. Implementações foram realizadas utilizando o método de filtro complementar e a combinação deste filtro com o filtro discreto de Kalman , utilizando um IMU de baixo custo, com o auxílio de um Arduino Pro-Mini.

**Palavras-chave:** filtro, Kalman, cossenos, veículos aéreos não tripulados, orientação

## ABSTRACT

This work has as objective the study, comparison and implementation of attitude estimation methods of unmanned flying vehicles (drones), based on the measurements of a low cost 9D (3D gyroscope, 3D accelerometer, 3D magnetometer) Inertial Measurements Unit (IMU).

Estimation methods for drones' attitude have as requirement the fact that these vehicles do not have a preferential movement direction, on the xy plane. Regarding these characteristics, air speed sensors must be replaced by magnetometers in order to better estimate the yaw angle.

The methods are direction cosine matrices, complementary filters based on state observers, and discrete Kalman filters. Computational cost, convergence rate, difficulty of implementation, both mathematical and physical implementation, and coherence of the results will be taken into account in the comparisons. In order to do so, simulations are implemented in mathematical software (MatLab), using a mathematical model, expressed in quaternions or in Euler's angles. Implementations were made using complementary filters and the combination of complimentary filters and Kalman filter, using as communication platform Arduino Pro-Mini.

**Keywords:** filter, Kalman, cosines, unmanned flying vehicle, attitude.

## **LISTA DE ABREVIATURAS, SIGLAS E NOTAÇÕES**

Neste trabalho, as seguintes notações foram adotadas:

**A** ou **a** (em negrito): denotam matrizes.

*A* ou *a* (em itálico): denotam vetores ou matrizas nx1.

A ou a (normal): denotam grandezas escalares.

e1: versor ortonormal de um eixo de coordenadas, na direção x.

e2: versor ortonormal de um eixo de coordenadas, na direção y.

e3: versor ortonormal de um eixo de coordenadas, na direção z.

{A}: sistemas de coordenadas inercial, fixo ao solo.

{B}: sistemas de coordenadas solidário ao corpo rígido estudado.

$A^B$ : vetor ou matriz A descrito no sistemas de coordenadas B.

$A_k$ : vetor ou matriz A calculado no passo, ou instante de tempo k.

$A_x$ : matriz antissimétrica  $\in \mathbb{R}^{3x3}$  composta pelas componentes do vetor A  $\in \mathbb{R}^3$ .

$\hat{A}$  ou  $\hat{a}$ : matriz ou vetor estimado. Observador de estados de A ou a.

$s\alpha$ :  $\sin(\alpha)$ .

$c\alpha$ :  $\cos(\alpha)$ .

$t\alpha$ :  $\tan(\alpha)$ .

$\Omega^q$ : Matriz assimétrica de velocidades angulares, denotada em quaternions.

# SUMÁRIO

1.	INTRODUÇÃO.....	10
2.	Estado da arte.....	11
3.	Fundamentos teóricos .....	12
3.1.	Orientação de corpos rígidos no espaço .....	12
3.1.1.	Matriz de rotação.....	12
3.1.2.	Ângulos de Euler de <i>Roll</i> , <i>Pitch</i> e <i>Yaw</i> .....	12
3.1.3.	Quaternions .....	13
3.2.	Modelagem cinemática de veículos aéreos não tripulados.....	15
3.3.	Unidades de medições inerciais .....	16
3.4.	Estimativa de posicionamento por Matriz de Cossenos Diretores.....	18
3.5.	Análise em frequência .....	19
3.6.	Filtro de Kalman .....	24
3.6.1.	Filtro de Kalman extendido.....	24
3.6.2.	Filtro de Kalman de estágio duplo .....	25
4.	Simulações.....	30
4.1.	Sensoriamento.....	30
4.2.	Matriz de cossenos diretores .....	32
4.3.	Filtro Complementar .....	32
4.3.1.	Observador não linear .....	33
4.4.	Filtro de Kalman de estágio duplo .....	34
4.5.	Resultados das simulações.....	36
5.	Implementação.....	53
5.1.	Comunicação serial.....	54

6.	CONCLUSÕES .....	65
7.	Referências Bibliográficas.....	67
	APÊNDICE A: Código em linguagem C do método implementado.....	69

# 1. INTRODUÇÃO

O desafio da estimativa discreta de propriedades e grandezas é um assunto de grande interesse e sempre é fonte de motivação para inúmeros trabalhos. A estimativa de orientação (atitude) e posicionamento de drones é com certeza um dos tópicos que tem recebido atenção especial pela sua extensa gama de aplicações. Em aeronaves tripuladas, a orientação é estimada com o auxílio da interação do piloto com a máquina. O piloto é a interface entre as leituras dos sensores e o sistema de controle da aeronave. No caso de uma aeronave não tripulada, esta interface deve ser direta, o que requer uma estimativa confiável e acurada da orientação do veículo. No caso especial de drones quadrotores, que não possuem uma direção preferencial de movimento, a orientação não pode ser estimada por sensores de velocidade do ar. Para isso utiliza-se uma unidade de medição inercial, IMU, com 9 graus de liberdade, ou seja, acelerômetro com 3gdl, giroscópio com 3gdl e magnetômetro com 3gdl, embora nem todos os graus de liberdade de todos os sensores sejam utilizados nas estimativas de orientação.

Este trabalho tem por objetivo o estudo, simulação e comparação e implementação de alguns métodos de estimativa de orientação desses veículos, apresentando os resultados comparativos de performance levando em consideração complexidade matemática, performance e custo computacional.

No capítulo 3, apresenta-se os conceitos teóricos necessários para o desenvolvimento deste trabalho, como sistemas de representação de orientação de corpos rígidos, unidades de medições iniciais, análise de sinais, conceito de filtros complementares e filtro de kalman.

No capítulo 4, mostra-se como foram realizadas as simulações destes métodos, utilizando dados de diferentes sensores, em diferentes frequências de aquisição além de apresentar os resultados destas simulações e as discussões pertinentes sobre os mesmos.

No capítulo 5, apresenta-se as implementações em tempo real dos métodos simulados e as adapatações necessárias em cada caso, além de uma discussão quanto os resultados obtidos nos experimentos.

A conclusão, no capítulo 6, discute os resultados obtidos nas simulações e implementações dentro do contexto dos requisitos do projeto.

## **2. ESTADO DA ARTE**

Nas últimas décadas, o estado da arte da estimativa da orientação de drones tem recebido enormes contribuições. A isso pode ser atribuído à vasta aplicação a qual este tipo de aeronaves tem sido submetida, desde vigilância, passando por operações militares, até atividades agrícolas. A popularização deste tipo de equipamento está levando as pesquisas a procurarem tecnologias mais eficazes e economicamente viáveis, e uma das frentes de pesquisa é a estimativa de orientação do veículo.

As pesquisas mais recentes estão focando em métodos híbridos de estimativa de orientação, sendo as combinações de filtros complementares, matrizes de cossenos diretores e filtros de Kalman as mais comuns. Os trabalhos listados na bibliografia deste trabalho com a correção da leitura de um dos sensores com a leitura de outro, passando esta última pelo algoritmo proposto. Desta forma encaixam-se os métodos de filtros complementares e filtros de Kalman estendidos de estágio duplo.

Este tópico é importante para a evolução na tecnologia dos drones, pois é a partir da estimativa da orientação que pode-se desenvolver sistemas de controle mais robustos e eficazes. Desta forma, a contribuição deste trabalho poderá ser a determinação dos pontos fortes e fracos de cada método aqui abordado, sendo assim possível a decisão da utilização deles para uma dada aplicação em drones.

## 3. FUNDAMENTOS TEÓRICOS

A seguir, vê-se alguns fundamentos teóricos que embasam este trabalho.

### 3.1. Orientação de corpos rígidos no espaço

Existem inúmeras formas de representar matematicamente a orientação e a posição de corpos rígidos no espaço. Considerando que existem ao menos dois sistemas de coordenadas num estudo dessa natureza, um solidário ao um referencial fixo,  $\{A\}$ , o solo por exemplo, e outro solidário ao corpo rígido em estudo,  $\{B\}$ , calcula-se as matrizes de transformação de coordenadas, o que permite descrever o posicionamento do corpo em relação a ambos os sistemas de coordenadas.

#### 3.1.1. Matriz de rotação

Enquanto a posição pode ser descrita pelo vetor de 3 componentes cartesianos, a orientação pode ser representada pelos versores de  $\{B\}$  descritos em relação a  $\{A\}$ . Para tal, define-se a matriz de rotação, que descreve a orientação de um sistemas de coordenadas em relação a outro. Como esta matriz descreve apenas a orientação, e não a posição, as linhas e colunas devem ter módulo unitário.

Para a composição desta matriz, deve-se ter inicialmente um padrão de representação da orientação de um corpo rígido. Neste trabalho, as formas de representação abordadas serão ângulos de Euler e quaternions.

#### 3.1.2. Ângulos de Euler de *Roll, Pitch e Yaw*

Uma das formas de representação de orientação é o conjunto de ângulos de roll, pitch e yaw. Sendo a forma mais utilizada em robótica e modelagem de veículos, representa a orientação apartir de ângulos de rotação ao redor dos eixos cartesianos do sistema de  $\{B\}$ , observando o sentido de rotação (positivo ou negativo).

As rotações são representadas da forma:

- Roll: rotação de um ângulo  $\phi$  ao redor do eixo x.
- Pitch: rotação de um ângulo  $\theta$  ao redor do eixo y
- Yaw: rotação de um ângulo  $\psi$  ao redor do eixo z.

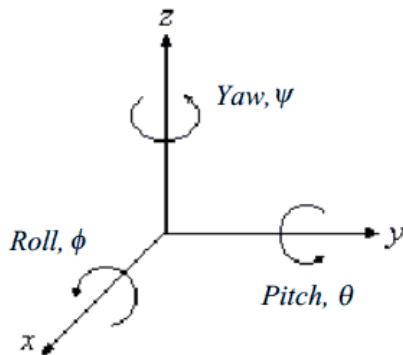


Figura 3-1 - Representação de orientação em Roll, Pitch e Yaw, de {B}. Fonte: Cabral, E.L.L. (2013)

A partir da definição deste padrão de representação, pode-se definir a matriz de rotação de {B} em relação a {A}, considerando que o primeiro estava paralelo a {A}. Assim a matriz de rotação é escrita da forma matricial:

$$\begin{aligned}
 \mathbf{R}_{\varphi,\theta,\psi} = \mathbf{R}_{z,\psi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\varphi} &= \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{bmatrix} \\
 &= \begin{bmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\varphi s\psi \\ s\psi c\theta & s\psi s\theta s\varphi + c\varphi c\psi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} \quad (3.1)
 \end{aligned}$$

Esta matriz é também conhecida por Matriz de Cossenos diretores, e é um dos métodos utilizados para estimativa da orientação de {B} em relação à {A}. Observe que a ordem das transformações interfere no resultado final, sendo a primeira transformação a de rotação de yaw, seguida de pitch e roll.

### 3.1.3. Quaternions

Quaternion é um vetor de 4 componentes que descreve o posicionamento em função do ângulo de rotação,  $\alpha$ , ao redor de um eixo  $n$ , que acompanha o corpo a ser estudado.

$$q_0 = \cos\left(\frac{\alpha}{2}\right) \quad (3.2.a)$$

$$q_1 = n_x * \sin\left(\frac{\alpha}{2}\right) \quad (3.2.b)$$

$$q_2 = n_y * \sin\left(\frac{\alpha}{2}\right) \quad (3.2.c)$$

$$q_3 = n_z * \sin\left(\frac{\alpha}{2}\right) \quad (3.2.d)$$

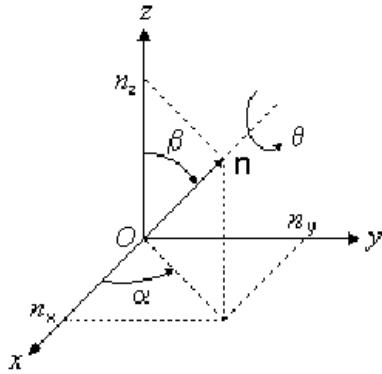


Figura 3-2 - Representação das grandezas que definem um quaternion no espaço cartesiano. Fonte: Cabral, E.L.L. (2013)

A matriz de cossenos diretores, utilizando a notação de quaternions, ficará então:

$$\mathbf{R}_{n,\alpha} = \begin{bmatrix} 2(q_1^2 + q_0^2) - 1 & 2(q_1q_2 + q_3q_0) & 2(q_1q_3 - q_2q_0) \\ 2(q_1q_2 - q_3q_0) & 2(q_2^2 + q_0^2) - 1 & 2(q_2q_3 + q_1q_0) \\ 2(q_1q_3 + q_2q_0) & 2(q_2q_3 - q_1q_0) & 2(q_3^2 + q_0^2) - 1 \end{bmatrix} \quad (3.3)$$

Nas recursões dos métodos mostrados a seguir, a cada iteração, os ângulos de orientação são descritos em ângulos de Euler. Considerando que os métodos serão computados fazendo uso da notação em quaternions, uma conversão se faz necessária a cada início de interação.

$$\text{Roll: } \varphi = \text{atan2}(2(q_2q_3 + q_0q_1), (q_0^2 - q_1^2 - q_2^2 + q_3^2))$$

$$\text{Pitch: } \theta = \text{asin}\left(\frac{2(q_0q_2 + q_1q_3)}{(q_0^2 + q_1^2 + q_2^2 + q_3^2)}\right) \quad (3.4)$$

$$\text{Yaw: } \psi = \text{atan2}(2(q_1q_2 + q_0q_3), (q_0^2 + q_1^2 - q_2^2 - q_3^2))$$

Normalmente, as malhas de controle utilizadas em VANT (veículos aéreos não tripulados) também utilizam os ângulos de Euler no modelo da planta. Isso pede que uma conversão ao fim de cada iteração ocorra a operação inversa a da equação mostrada acima, para retornar de quaternion para ângulos de Euler. Para tal, a simples operação inversa a equação (2), é o suficiente.

### 3.1.4. Velocidades ângulares

Para descrever a variação da orientação do corpo rígido no tempo, o vetor de velocidades angulares é definido da forma:

$$\omega = [\omega_x \ \omega_y \ \omega_z]^T \quad (3.5)$$

A variação da matriz de rotação  $R$  no tempo, será a derivada da mesma no tempo,  $\dot{R}$ . Desta forma, obtemos:

$$\dot{R} = \Omega_B R \quad (3.6)$$

sendo

$$\Omega_B = \omega \times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (3.7)$$

a matriz antissimétrica de velocidades. Esta transformação será de grande importância para o cálculo dos métodos a serem estudados por este trabalho.

## 3.2. Modelagem cinemática de veículos aéreos não tripulados

Formaliza-se as equações de movimento angular do corpo rígido em relação ao referência fixo.

Como visto acima, podemos representar a orientação em angulos de Euler ou em quaternions.

Realizando a modelagem utilizando ângulos de Euler, temos que a taxa de variação de roll, pitch e yaw no tempo é descrita por:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{d(O_B - O_A)}{dt} + \omega \quad (3.8)$$

A equação 3.8 descreve a variação temporal dos ângulos de euler como a variação da orientação do sistemas de coordenadas somada a variação ao redor de cada eixo, sendo  $\omega$  o vetor de velocidades ângulares medidas pelo giroscópio. Assim, a equação 2.7 pode ser reescrita na forma

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{d(O_B - O_A)}{dt} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.9)$$

Para obtermos a posição final de  $\{B\}$  após um espaço de tempo  $dt$ , fazemos a integração da equação 3.8

$$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_k = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_{k-1} + dt * \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.10)$$

Já a modelagem em quaternion, nos fornece a taxa temporal de variação angular na forma

$$\dot{q} = \frac{1}{2} \boldsymbol{\Omega}_b^q q \quad (3.11)$$

Onde  $\boldsymbol{\Omega}_b^q = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$  é a matriz anti-simétrica de velocidades adaptada para

a modelagem em quaternion.

Da mesma forma que na modelagem utilizando ângulos de Euler, podemos obter o quaternion final depois de dt integrando a equação diferencial obtida em 2.10

$$q_k = q_{k-1} + dt * \frac{1}{2} \boldsymbol{\Omega}_b^q * q_{k-1} \quad (3.12)$$

A frequência de movimentação do sistema, ou seja, a velocidade com que o corpo rígido completará um ciclo de oscilação ao redor de qualquer um dos eixos, foi estimada, de forma conservadora, em 1Hz. Isto garante com que movimentos mais bruscos do que a movimentação normal do corpo rígido ainda considerados nas estimativas de orientação a seguir.

### 3.3. Unidades de medições iniciais

Uma unidade de medições iniciais é um dispositivo capaz de medir as velocidades, acelerações e forças gravitacionais que agem sobre um corpo no qual o IMU (sigla em inglês para Inertial measurement unit) estiver embarcado. O IMU é uma combinação de sensores iniciais (acelerômetro e giroscópio) e magnético (magnetômetro) e é amplamente utilizado em aeronaves não tripuladas, já que a estimativa de orientação deve ser realizada em tempo real sem a interação de um piloto.

Neste trabalho, são utilizados dois diferentes IMU's: sensores de movimento de um IPHONE® e o Razor IMU 9D.

O IPHONE® conta com um compasso eletrônico da série AKM que baseia-se em microcircuito com sensores de efeito Hall; com um acelerômetro STMicroelectronics de três eixos e com um giroscópio STMicroelectronics de três eixos. Todos esses dispositivos são MEM's, sistemas microeletro-mecânicos.

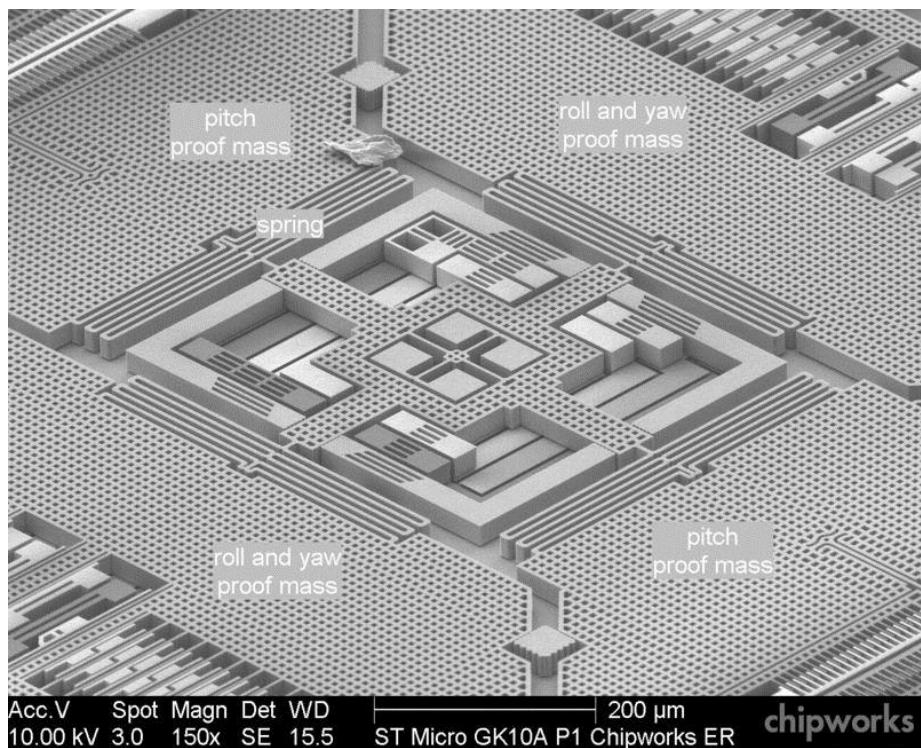


Figura 3-3 Imagem aumentada de um giroscópio STM de um IPHONE. Fonte: <http://www.memsjournal.com/>. Acesso em 25/06/2013

A direção dos eixos de coordenadas sobre o qual a orientação será estimada no IPHONE® é a mostrada na figura a seguir.

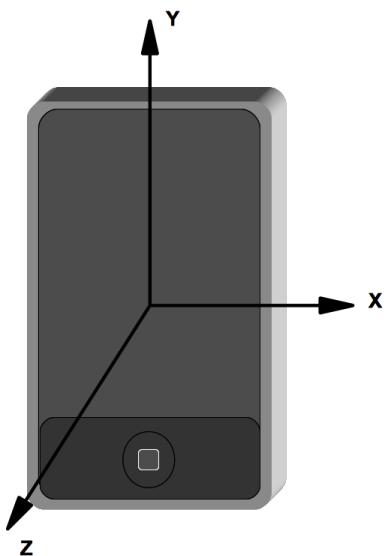


Figura 3-4 Sistemas de coordenadas dos sensores de movimento do IPHONE®. Fonte: <http://rouel-projects.blogspot.com.br/p/calibration-of-iphone-accelerometers.html>. Acesso em 15/06/2013

O outro IMU, Razor 9DOF AHRS vem equipado com o giroscópio ITG-3200 de três eixos, com acelerômetro ADXL345, de três eixos e resolução de 13 bits, e com o magnetômetro digital

HMC5883L, também de três eixos. A orientação do IMU pode ser vista na figura 2-5. Assim como o IPHONE®, o Razor avalia sua dinâmica e cinética ao redor dos eixos x, y e z, o que induz o usuário a implementar métodos de estimativa de orientação em ângulos de Euler ou Quaternions.

Este dispositivo já vem preparado para comunicação com Arduíno, motivo pelo qual escolhido para este trabalho. Desta forma, a aquisição de dados e o embarque em qualquer veículo deste dispositivo ficam facilitados.



*Figura 3-5 IMU Razor 9DOF .Detalhe do sistema de coordenadas considerado para as medições do sensor . Fonte: <https://www.sparkfun.com/products/10736>. Acesso em 15/06/2013*

### 3.4. Estimativa de posicionamento por Matriz de Cossenos Diretores

A maneira mais simples de estimativa de orientação de um drone utilizando IMU's é através da matriz de cossenos diretores,  $\mathbf{R}_{\phi,\theta,\psi}$ .

A medição da direção da aceleração da gravidade, realizada pelo acelerômetro, é descrita em relação ao sistema de coordenadas solidário ao corpo, {B}. As grandezas angulares que nos interessam são as de {B} em relação a {A}. Desta forma, a direção da gravidade em {A} pode ser calculada na forma:

$$A_A = \mathbf{R}_{\phi,\theta,\psi}^B * A_B \quad (3.13).$$

$A_B$  é a direção da gravidade expressa no sistema de coordenadas {B}, ou seja, a medição da direção da gravidade realizada pelo acelerômetro. As incógnitas do método são os ângulos de roll, pitch e yaw, que estão representados dentro da matriz de cossenos diretores  $\mathbf{R}_{\phi,\theta,\psi}^B$ .

Considerando que a direção medida pelo acelerômetro é um vetor normalizado, assim como o

vetor  $A_A$  da direção da gravidade em  $\{A\}$ , pode-se então estimar a orientação pela relação inversa da equação 3.13, usando-se da propriedade da matriz de cossenos diretores de ter a sua matriz inversa igual à transposta.

$$A_B = \mathbf{R}_{\varphi, \theta, \psi}^{B^{-1}} * A_A = \mathbf{R}_{\varphi, \theta, \psi}^{B^T} * A_A \quad (3.14).$$

Explicitando os componentes de cada vetor e matriz da equação acima, temos:

$$\begin{bmatrix} A_{B_x} \\ A_{B_y} \\ A_{B_z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ c\psi s\theta s\varphi - s\psi c\varphi & s\psi s\theta s\varphi + c\varphi c\psi & c\theta s\varphi \\ c\psi s\theta c\varphi + s\varphi s\psi & s\psi s\theta c\varphi - c\psi s\varphi & c\theta c\varphi \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s\theta \\ c\theta s\varphi \\ c\theta c\varphi \end{bmatrix} \quad (3.15)$$

Da equação 3.15 podemos retirar os valores dos ângulos de roll e pitch, resolvendo o sistema:

$$\theta = -\text{asen}(A_{B_x}) \quad (3.16.a)$$

$$\varphi = \text{asen}\left(\frac{A_{B_y}}{c\theta}\right) \quad (3.16.b)$$

Para estimar o ângulo de Yaw, podemos aplicar o mesmo princípio nas medições da direção do campo magnético terrestre pelo magnetômetro. Considerando que o eixo y de  $\{A\}$  é paralelo à direção norte, as medições realizadas pelo magnetômetro são escritas, em função da direção do campo magnético descrito em  $\{A\}$  é

$$\begin{bmatrix} M_{B_x} \\ M_{B_y} \\ M_{B_z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ c\psi s\theta s\varphi - s\psi c\varphi & s\psi s\theta s\varphi + c\varphi c\psi & c\theta s\varphi \\ c\psi s\theta c\varphi + s\varphi s\psi & s\psi s\theta c\varphi - c\psi s\varphi & c\theta c\varphi \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} s\psi c\theta \\ s\psi s\theta s\varphi + c\varphi c\psi \\ s\psi s\theta c\varphi - c\psi s\varphi \end{bmatrix} \quad (3.17)$$

Da equação 3.17, estima-se o ângulo de yaw, satisfazendo a equação

$$\psi = \text{asen}\left(\frac{M_{B_x}}{c\theta}\right) \quad (3.18)$$

### 3.5. Análise em frequência

As leituras dos sensores estão sujeitas a ruídos de medição e bias que interferem na qualidade da medição da posição angular do corpo ao qual o IMU está acoplado. O acelerômetro, que fornece a medição direção da aceleração da gravidade no sistema de coordenadas  $\{B\}$  é especialmente afetado pelas altas frequências, já que os ângulos roll e pitch serão estimados por relações trigonométricas. Isto é, assumindo ruído branco de média zero, as variações em alta frequência podem fazer com que a estimativa exploda.

Já o giroscópio fornece as velocidades angulares. Desta forma, a posição angular é estimada pela integração, descrita na equação 2.11. Frequências menores do que a de movimento do corpo rígido podem fazer com que a integração exploda, e a estimativa também fique inválida.

Alguns dos métodos propostos se utilizam de filtros para tratar o sinal destes sensores, de forma a diminuir a influência dos ruídos nas estimativas da orientação. Neste trabalho, serão desenvolvidos filtros de primeira ordem.

### 3.5.1. Filtros de Primeira ordem

Filtros de primeira ordem são descritos por funções de transferência que relacionam o sinal de entrada com o sinal de saída filtrado. Como o nome diz, esta função de transferência é de primeira ordem, o que significa que o sinal filtrado na saída será o sinal de entrada dividido por uma equação de Laplace de primeira ordem, no qual, o termo de ordem um é dividido pela frequência de corte. Existem várias formas de projetar este filtro, e neste trabalho, o filtro utilizado foi o proposto por S.Butterworth, já projetado no domínio discreto.

#### 3.5.1.1. Filtro passa-baixas

O filtro passa-baixas de Butterworth depende da frequência de oscilação do sistema, assumida como 1Hz como descrito na seção 3.2 e da frequência de corte desejada. O filtro passa-baixas atuará no sinal do acelerômetro, de forma a reduzir o ruído de frequências maiores do que a do movimento do corpo. Desta forma, podemos projetar a frequência de corte como 1Hz. No filtro de Butterworth no domínio discreto, o valor considerado é a razão entre a frequência de corte e a frequência de aquisição do sensor. Para as simulações foram implementadas uma interrupção para aquisição de dados a cada  $dt=0.02s$ , ou seja, a frequência de aquisição foi de 50Hz, e outra a cada 0.002s, ou seja, frequência de aquisição de 500Hz, o que satisfaz o critério de Nyquist.

Isso nos fornece um filtro passa-baixas com as seguintes funções de transferência

$$\frac{Y(z)}{U(z)} = \frac{0.0305z+0.0305}{z-0.9391}, \text{ para } f_a=50\text{Hz} \quad (3.19.a)$$

$$\frac{Y(z)}{U(z)} = \frac{0.0031z+0.0031}{z-0.9937}, \text{ para } f_a= 500\text{Hz} \quad (3.19.b)$$

Sendo  $Y(z)$  o sinal já filtrado e  $U(z)$  o sinal original obtido do acelerômetro.

### **3.5.1.2. Filtro passa-altas**

Da mesma forma, o filtro passa-altas de Butterworth depende das frequências do sistema, de aquisição e de corte desejada. Utilizando os mesmos valores do filtro passa-baixas, obtemos a seguinte função de transferência

$$\frac{Y(z)}{U(z)} = \frac{0.9695z+0.9695}{z-0.9391}, \text{ para } f_a=50\text{Hz} \quad (3.20.a)$$

$$\frac{Y(z)}{U(z)} = \frac{0.9969z+0.9969}{z-0.9937}, \text{ para } f_a= 500\text{Hz} \quad (3.20.b)$$

### **3.5.1.3. Frequência de Cross-Over**

O intuito de usar mais de um sensor para a estimativa da orientação de um corpo rígido é de que onde um deles é falho, ou não tem suas medições confiáveis, os outros serão utilizados para o cálculo da orientação. Ou seja, os dados do acelerômetro serão utilizados quando as variações angulares em baixa frequência forem mais importantes, e os do giroscópio, quando as de alta frequência o forem, de forma que um complemente o outro.

Isso pode ser observado ao analisar o espectro de frequências resultante das filtragens. No gráfico, vemos que na frequência de corte, onde o ganho é de -3dB, há o cruzamento do espectro de frequências do filtro passa-baixas e do passa-altas, de forma que a soma dos ganhos se mantém aproximadamente constante para todas as frequências do espectro. No ponto onde os espectros se cruzam, dá-se o nome de frequência de cross-over, ou frequência de cruzamento. Ou seja, onde o sinal filtrado por passa-baixas é ausente, o sinal de passa alta o complementa, e vice-versa.

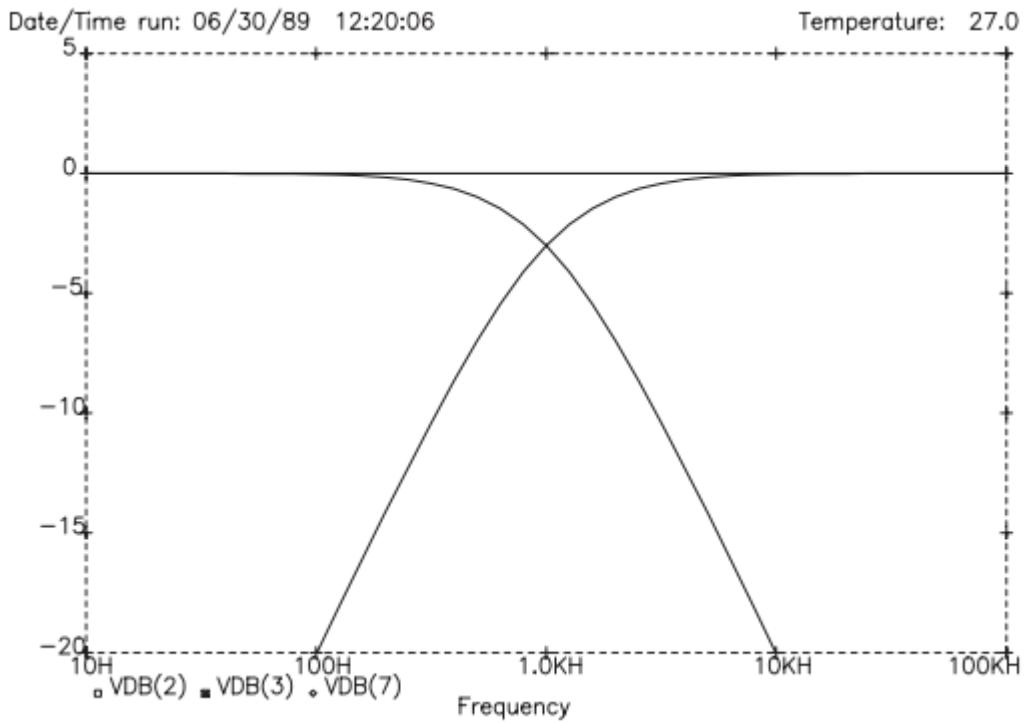


Figura 3-6 Crossover de respostas de filtros de primeira ordem. Fonte: RaneNote 2005.

### 3.5.2. Filtros Complementares

O filtro complementar é um método de estimativa da orientação do corpo rígido solidário a {B} em relação à {A} que utiliza os sinais do acelerômetro e do giroscópio filtrados, respectivamente por passa-baixa e passa-altas. Apartir realimentações e iterações entre estes sinais, eles se corrigem a cada iteração para fornecer uma melhor estimativa da orientação do corpo rígido.

Para este trabalho, utilizou-se uma adaptação do método proposto por Mahony em *A complementary filter for attitude estimation of fixed-wings UAV's*, 2006. O autor propõe um observador para estimativa do estado  $q$ , mas como o nome do trabalho diz, isso é feito considerando um veículo aéreo não tripulado de asa fixa, ou, seja um avião. Por isso, o autor se utiliza de um barômetro posicionado na direção do movimento, para medir a velocidade do ar, e assim estimar o ângulo de yaw. A adaptação que fez-se para este trabalho, foi combinar a medição do acelerômetro de roll e pitch com a do magnetômetro para yaw. Tendo em mãos a estimativa da orientação pelo acelerômetro e pelo magnetômetro,  $\bar{q}$ , faz-se uma comparação com o estado estimado na recursão anterior,  $\hat{q}$ , obtendo-se assim o erro em regime.

$$e_k = \bar{q}_k - \hat{q}_{k-1} \quad (3.21)$$

O controle do observador será utilizado para zerar o erro em regime. Outra adaptação feita ao trabalho de Mahony é que a saída do controlador será utilizada para corrigir o estado estimado pelo observador, e não o vetor de velocidades angulares medidas pelo giroscópio.

O autor optou por utilizar um controlador proporcional integrativo, PI, no observador, de forma que o termo proporcional governa o cross-over de frequências, para garantir a complementaridade das frequências abrangidas pelos sensores, e o termo integrativo corrige o viés da medição do giroscópio.

$$L(s) = K_p + \frac{K_i}{s} \quad (3.22)$$

Os dados do giroscópio serão utilizados pelo observador para compor a matriz  $\Omega_b^q$  e calcular o estado  $\hat{q}^-$ .

$$\hat{q}^-_k = \hat{q}^-_{k-1} + dt * \frac{1}{2} \Omega_b^q * \hat{q}_{k-1} \quad (3.23)$$

A estimativa final do estado é feita somando o erro em regime, controlado pelo controle do observador  $L(s)$

$$\hat{q}_k = \hat{q}^-_k + L(s) * e_k \quad (3.24)$$

O diagrama de blocos do observador que utiliza o filtro complementar fica então:

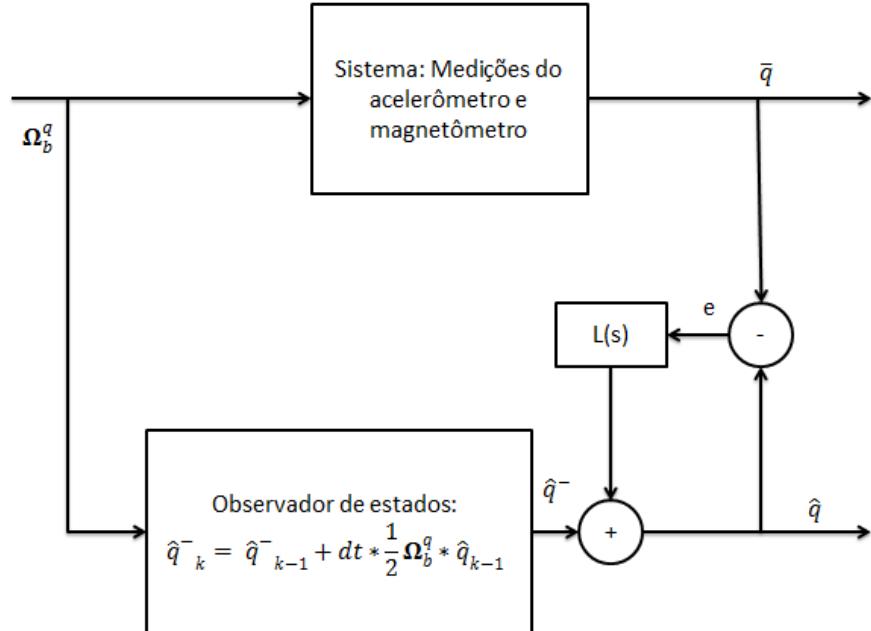


Figura 3-7 Diagrama de blocos do Observador não linear proposto.

### 3.6. Filtro de Kalman

Este método de filtragem de dados foi proposto por R.E. Kalman, em 1960 e sugere um tratamento recursivo de dados discretos de forma linear. É amplamente aplicado em navegação e seu uso tem sido cada vez mais requerido devido ao avanço na tecnologia computacional. Se trata de um conjunto de equações que estimam o estado de um dado sistema dinâmico, apartir de ganhos aplicados sobre a comparação entre os dados medidos e os dados estimados pelas equações.

Considerando a relação entre as medições realizadas pelo IMU e a dinâmica do veículo em questão, o filtro de Kalman deve ser aplicado observando a não linearidade da dinâmica do corpo rígido estudado aqui. Para tal, uma ferramenta variante do filtro de Kalman a ser utilizada é o filtro de Kalman extendido.

#### 3.6.1. Filtro de Kalman extendido

Considere um sistema cujo estado possa ser escrito, no tempo discreto, da forma:

$$x_k = Ax_{k-1} + Bu_{k-1} - w_{k-1} \quad (3.25)$$

e a medição realizada por um sensor:

$$z_k = Dx_k + v_k \quad (3.26)$$

onde

- $x$ : vetor de coordenadas
- $u$ : vetor de entradas e excitações externas
- $A$ : matriz de estado,  $n \times n$ , que relaciona o estado anterior,  $k-1$ , com o estado atual  $k$
- $B$ : matriz,  $n \times 1$ , que relaciona o input de controle  $u$  com  $x$
- $D$ : matriz,  $m \times n$ , que relaciona a medição  $z$  ao estado  $x$ .
- $w_k$ : ruído de processo, independente, branco e normalmente distribuído
- $v_k$ : ruído de medição, independente, branco e normalmente distribuído,

Define-se então  $\hat{x}_k^- \in \mathbb{R}^N$  como estado estimado a priori, dado conhecimento do estado do sistema no instante anterior a  $k$ , e  $\hat{x}_k \in \mathbb{R}^N$  como estado definido a posteriori, no passo  $k$ , a partir da medição no instante  $k$ . A partir disto pode-se definir os erros:

$$A \text{ priori: } e_k^- = x_k - \hat{x}_k^- \quad (3.27)$$

$$A \text{ posteriori: } e_k = x_k - \hat{x}_k \quad (3.28)$$

E também as covariâncias :

$$A \text{ priori: } \mathbf{P}_k^- = \mathbf{E}(e_k^- e_k^{T-}) \quad (3.29)$$

$$A posteriori: \mathbf{P}_k = \mathbf{E}(\mathbf{e}_k \mathbf{e}_k^T) \quad (3.30)$$

A partir daí pode-se definir um ganho de Kalman que pondere a inovação, isto é, a diferença entre medição real  $z_k$  e a estimativa da medição  $\mathbf{J}x_k$ , na forma:

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}(z_k - \mathbf{J}\hat{x}_k^-) \quad (3.31)$$

Mais adiante será explicado que as estimativas dos vetores de aceleração da gravidade, medido pelo acelerômetro, e do campo magnético, medido pelo magnetômetro, são não lineares (são definidos apartir do produto das componentes do quaternion que define a posição angular do drone). Desta forma faz-se necessário utilizar o filtro Kalman extendido, que define o ganho K como:

$$\mathbf{K}_k = \mathbf{P}_k^{-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \quad (3.32)$$

Sendo  $\mathbf{R}_k$  a matriz das covariâncias dos erros de medição, levando em consideração os ruídos do acelerômetro e do magnetômetro, e  $\mathbf{V}_k$  e  $\mathbf{H}_k$  as matrizes jacobianas de derivadas parciais das estimativas da gravidade e do campo magnético em relação ao ruído e aos quaternions, respectivamente.

### 3.6.2. Filtro de Kalman de estágio duplo

De forma a obter uma melhor estimativa da posição angular do veículo, um método proposto é o filtro de Kalman de estágio duplo, por Sabatelli et al. (2013), que primeiramente utiliza o acelerômetro para corrigir a medição realizada pelo giroscópio e a seguir o magnetômetro para corrigir a direção yaw, baseado no campo magnético da Terra.

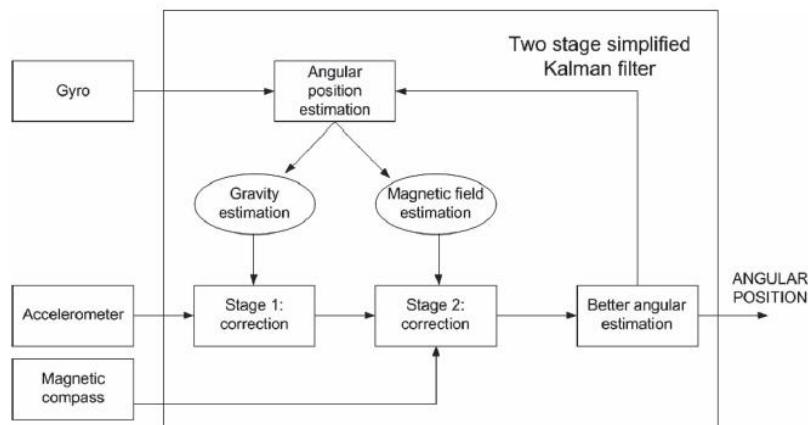


Figure 3-8 Algoritmo de fusão de sinais de sensores. Fonte: Sabatelli, et al. (2013)

Tendo a equação de estado no tempo contínuo, em quaternions, no referencial do corpo:

$$\dot{q}^b = \frac{1}{2} \boldsymbol{\Omega}_b q^b \quad (3.33)$$

E sendo

$$\boldsymbol{\Omega}_b^q = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \quad (3.34)$$

a matriz rotacional do corpo em relação ao eixo n, e  $\omega_x$ ,  $\omega_y$ , e  $\omega_z$  as velocidades angulares do corpo medidas inicialmente pelo giroscópio, temos no domínio discreto do tempo:

$$\frac{q^b(t+dt) - q^b(t)}{dt} = \mathbf{A}_{TC} q^b(t) \quad (3.35)$$

Sendo

$$\mathbf{A}_{TC} = \frac{1}{2} \boldsymbol{\Omega}_b \quad (3.36)$$

A partir da equação de estados no domínio discreto, calcula-se a matriz das covariâncias dos erros a priori:

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q}_{k-1} \quad (3.37)$$

sendo  $\mathbf{Q}_{k-1}$  a matriz de covariância dos erros do processo e  $\mathbf{A}_k = \mathbf{I} + \mathbf{A}_{TC} * dt$ , e  $dt = 0.002s$ , considerando que o filtro realiza iterações a uma frequência de 500Hz, segundo. Esta matriz é calculada apartir da média dos erros,

$$\mu = \frac{1}{N} \sum_{j=1}^N e_j \quad (3.38)$$

sendo que  $e_k$  se refere ao erro no estágio k. Fazendo-se as variâncias parciais como

$$\sigma_{ii}^2 = \frac{1}{(N-1)} \sum_{j=1}^N (e_{i,j} - \mu_i)^2 \quad (3.39)$$

e

$$\sigma_{ij}^2 = \frac{1}{(N-1)} \sum_{k=1}^N (e_{i,k} - \mu_i) (e_{j,k} - \mu_j) \quad (3.40)$$

define-se, então, a matriz de covariância dos erros definida por

$$\mathbf{P} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1M}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{M1}^2 & \sigma_{M2}^2 & \dots & \sigma_{MM}^2 \end{bmatrix} \quad (3.41)$$

O primeiro estágio deste método utiliza os dados do giroscópio para estimar o vetor da gravidade em relação ao referencial do corpo. A partir disso, subtrai-se este valor da medição realizada pelo acelerômetro, sendo esta diferença ponderada pelo fator K do filtro de Kalman extendido, definindo-se o primeiro valor residual  $q_{E1}$ .

Para definir o ganho K, primeiro deve-se definir  $H_k$  e  $V_k$ . Considera-se que os ruídos do acelerômetro e do magnetômetro são independentes entre si. Desta forma  $V_k$  pode ser definida como simplesmente a matriz identidade. Já para  $H_k$ , é necessária a estimativa do vetor de gravidade. Para estimar a gravidade faz-se

$$h_1(q_k) = \hat{g} = \mathbf{R}^B \begin{bmatrix} 0 \\ 0 \\ |g| \end{bmatrix} = |g| \begin{bmatrix} 2(q_1 q_3 - q_2 q_0) \\ 2(q_2 q_3 + q_1 q_0) \\ 2(q_3^2 + q_0^2) - 1 \end{bmatrix} \quad (3.42)$$

onde  $\mathbf{R}_n^b$  é a matriz de cossenos diretores, demonstrada na equação 2.3.

A estimativa  $h_1$  da gravidade carrega consigo erros acumulados, já que o acelerômetro não mede apenas a gravidade quando o sistema está em estado dinâmico.

Desta forma, defini-se  $H_k$  como a matriz jacobiana das derivadas parciais de  $h_1$  em relação aos quaternions.

$$H_{k1} = \frac{\partial h_1[i]}{\partial q[j]} = \begin{bmatrix} -2q_2 & 2q_3 & -2q_0 & 2q_1 \\ 2q_1 & 2q_0 & 2q_3 & 2q_2 \\ 2q_0 & -2q_1 & -2q_2 & 2q_3 \end{bmatrix} \quad (3.43)$$

Assim, teremos o ganho  $K_{k1}$

$$K_{k1} = P_k^- H_{k1}^T (H_{k1} P_k^- H_{k1}^T + V_{k1} R_{k1} V_{k1}^T)^{-1} \quad (3.44)$$

para o cálculo do resíduo,

$$q_{E1} = K_{k1} (z_{k1} - h_1(\hat{q}_k^-)) \quad (3.45)$$

sendo  $z_{k1} = (a_x, a_y, a_z)$ , a medição realizada pelo acelerômetro .

Para que esta primeira etapa não afete a estimativa de yaw, que só pode ser corrigida pelo magnetômetro, faz-se:

$$q_{E1} = [1 \ 1 \ 1 \ 0] \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3.46)$$

Então, calcula-se a primeira estimativa a posteriori do estado:

$$\hat{q}_{k1} = \hat{q}_{k1}^- + q_{E1} \quad (3.47)$$

Antes de prosseguir, deve-se calcular novamente a matriz das covariâncias dos erros deste processo, para inicialização do proximo passo, k+1. Esta matriz é dada por

$$\mathbf{P}_{k1} = (\mathbf{I} - \mathbf{K}_{k1} \mathbf{H}_{k1}) \mathbf{P}_k^- \quad (3.48)$$

O segundo estágio é semelhante ao primeiro, mas trata-se agora da leitura feita pelo magnetômetro.

Tendo  $z_{k2}$ , leitura dos magnetômetros, e fazendo  $h_2$ , estimativa do campo magnético, na forma:

$$h_2(q_k) = \hat{\mathbf{m}} = \mathbf{R}_n^b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2(q_1 q_2 + q_3 q_0) \\ 2(q_3^2 + q_0^2) - 1 \\ 2(q_3 q_2 - q_1 q_0) \end{bmatrix} \quad (3.49)$$

Note que o campo magnético tem seu norte considerado paralelo ao eixo y, e a componente vertical desconsiderada, ou seja, a curvatura da Terra não influencia significativamente o modelo.

Tendo  $h_2$  pode-se calcular a matriz jacobiana  $\mathbf{H}_2$

$$\mathbf{H}_{k2} = \frac{\partial h_2[i]}{\partial q[j]} = \begin{bmatrix} 2q_3 & 2q_2 & 2q_1 & 2q_0 \\ 2q_1 & -2q_1 & -2q_2 & -2q_3 \\ -2q_1 & -2q_0 & 2q_3 & 2q_2 \end{bmatrix} \quad (3.50)$$

podendo-se assim definir o ganho  $\mathbf{K}_{k2}$  como

$$\mathbf{K}_{k2} = \mathbf{P}_k^- \mathbf{H}_{k2}^T (\mathbf{H}_{k2} \mathbf{P}_k^- \mathbf{H}_{k2}^T + \mathbf{V}_{k2} \mathbf{R}_{k2} \mathbf{V}_{k2}^T)^{-1} \quad (3.51)$$

Para este fator de correção, novamente faz-se a diferença entre o medido pelo magnetômetro e a estimativa do campo magnético, ponderado pelo ganho de Kalman calculado acima.

$$q_{E2} = \mathbf{K}_{k2} (z_{k2} - h_2(\hat{q}_k)) \quad (3.52)$$

Como os ângulos de roll e pitch já foram corrigidos no primeiro estágio deste método, faz-se

$$q_{E2} = [1 \ 0 \ 0 \ 1] \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3.53)$$

Agora soma-se este segundo resíduo ao calculado no primeiro e temos uma posição estimada com melhor acurácia.

$$\hat{q}_k = \hat{q}_{k1} + q_{E2} \quad (3.54)$$

Por fim, para que o próximo passo, k+1, possa ser inicializado com a matriz de covariâncias de erros preenchida, calcula-se, usando a matriz  $\mathbf{P}_{k1}$ , definida anteriormente:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_{k2} \mathbf{H}_{k2}) \mathbf{P}_{k1}^- \quad (3.55)$$

Este autor não levou em consideração a bias dos sensores. A forma de torná-la irrelevante aos resultados deste método, foi a normalização dos vetores de quaternion a cada início e fim de iteração, e da leitura do magnetômetro a cada aquisição, para torná-lo coerente.

## 4. SIMULAÇÕES

Aqui seguem as adaptações dos métodos apresentados para a implementação

### 4.1. Sensoriamento

#### 4.1.1. IMU Razor 9DOF

O Razor IMU incorpora 3 sensores – o giroscópio ITG-3200, o acelerômetro ADXL345 e o magnetômetro HMC5883L – para obter a medição de 9 dimensões inerciais.

As saídas dos sensores podem ser obtidas através de comunicação serial com Arduino. Para tal, a placa conta com saídas TX e RX para transmissão e recebimento de dados, além dos pinos de Vcc (3.3V) e terra. Estes quatro pinos são conectados com as respectivas portas do arduíno, e desta forma, através de uma recursão armazenada no processador ATMega328, os dados experimentais foram tomados.

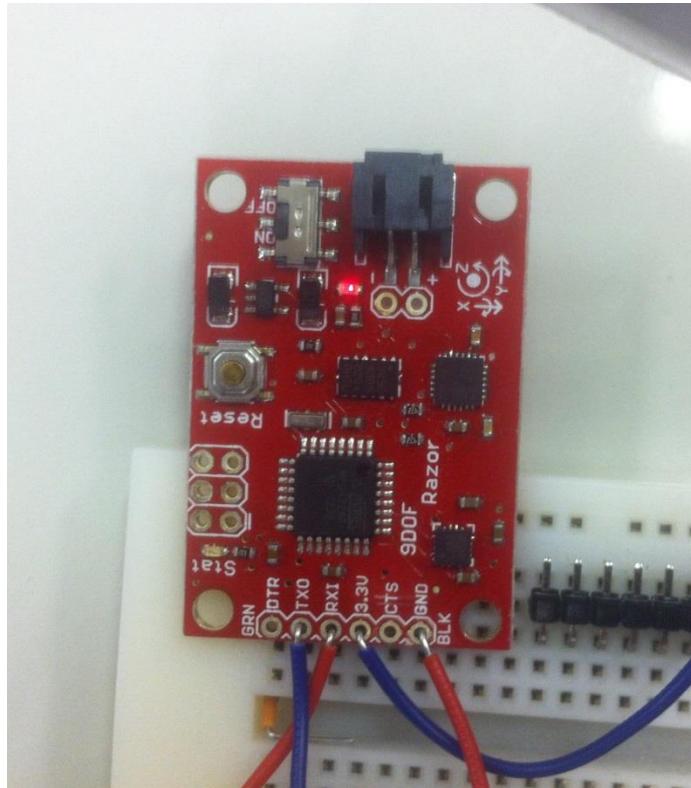
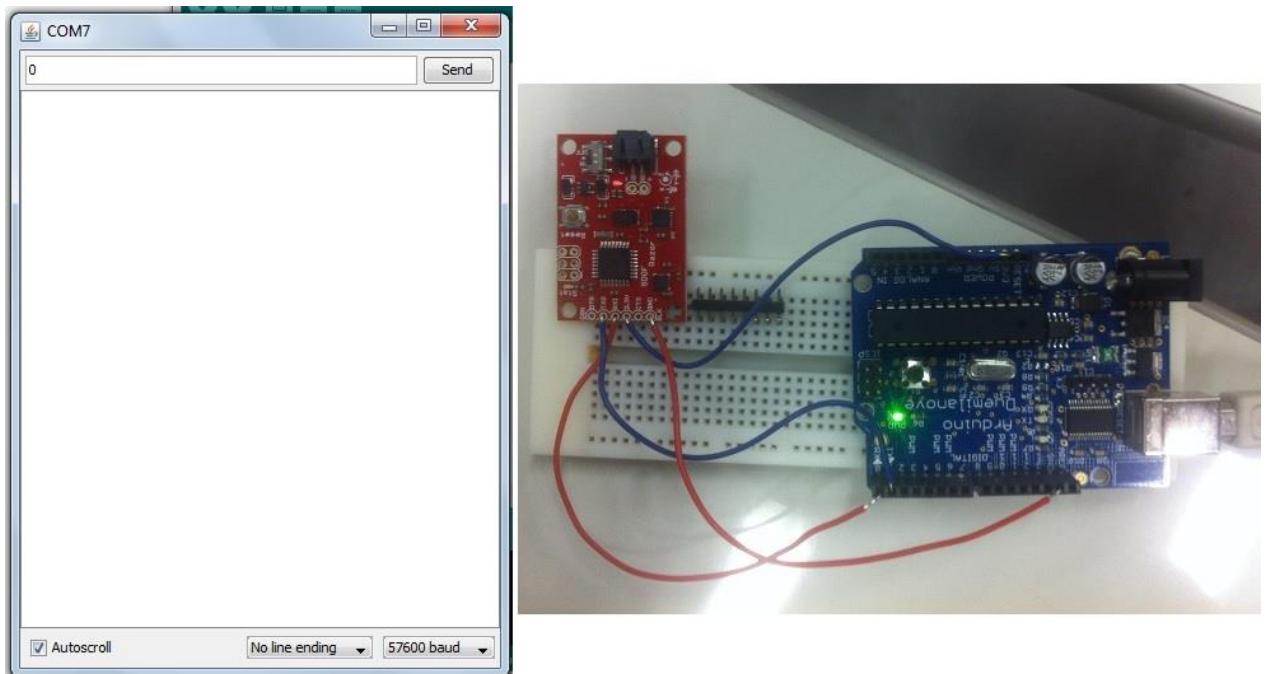


Figura 4-1- Detalhe: Sensor Razor 9DOF com as portas TX, RX, GND e 3V3 conectadas ao Arduino Duemillanove.

#### **4.1.2. Arduino Duemillanove**

A comunicação foi realizada através da plataforma Arduino Duemillanove. As portas de comunicação serial e alimentação do sensor se conectaram com as respectivas portas do Arduino, que por sua vez se comunica com o computador através da porta USB, por onde transmite os dados para o monitor de porta serial do programa Arduino. Neste monitor, pode-se monitorar as leituras cruas de cada eixo de cada sensor, totalizando 9 dados por aquisição.



*Figura 4-2 - Direita: Serial Monitor do Arduino. Esquerda: Razr 9DOF conectado ao Arduino pela porta serial, e Arduino conectado ao computador via USB.*

#### **4.1.3. Aquisição de dados**

A aquisição dos dados se deu através da comunicação serial entre Arduino e computador, via porta USB. Para garantir que os dados fossem adquiridos com frequência constante, uma interrupção foi implementada no Arduino, utilizando o Timer1, com períodos de 2000ms e 200ms, ou seja, a aquisição ocorreu com frequências de 50Hz e 500Hz, e dentro destas interrupções, faz-se a chamada da rotina de aquisição de dados da porta serial. Estas frequências de aquisição foram escolhidas de forma a maximizar a performance dos algoritmos de estimativa de posicionamento. A frequência de oscilação de posicionamento do sistema foi estimada em 1Hz. Desta forma, pelo critério de Nyquist, a frequência de aquisição necessária para garantir a

qualidade da reconstrução do sinal seria de 2Hz. Isto é, a frequência de aquisição escolhida atende o critério.

As aquisições de dados dos sensores, em geral duraram por volta de 10s, e movimentos sistemáticos foram realizados de forma a avaliar a eficácia dos métodos aplicados. Os movimentos foram de 90° no sentido positivo e 90° no sentido negativo de cada um dos eixos. Desta forma, a leitura esperada é uma variação de aproximadamente 90° nas duas direções do roll, aproximadamente 90° nas duas direções do pitch e aproximadamente 90° nas duas direções do yaw.

## **4.2. Matriz de cossenos diretores**

Uma forma de obter o sinal puro, sem tratamento, para comparar com os algoritmos estudados, foi a aplicação da matriz de cossenos diretores como método de estimativa da orientação do corpo rígido.

Para obter-se a direção da gravidade, sem grandeza escalar agragada, assim como a direção do campo magnética terrestre, fez-se antes a normalização das leituras do acelerômetro e do magnetômetro. O algorítmo utilizado foi o explicado na seção 3.4. O método foi simulado com os dados adquiridos a frequências de 50Hz e 500Hz.

Para fins de comparação, também realizou-se a integração simples do sinal do giroscópio, como forma de estimativa da variação dos ângulos de Euler.

## **4.3. Filtro Complementar**

Como explicado na seção 3.4, filtros complementares se utilizam da medição do acelerômetro filtrada de altas frequências e do giroscópio filtradas de baixas frequências, tendo a estimativa complementada, neste trabalho, pela estimativa do ângulo de yaw medido pelo magnetômetro.

O algorítmo implementado neste trabalho foi uma adaptação do proposto por Mahony et al, 2006. Como explicado acima, um observador de estados se utiliza das medições do giroscópio para estimar o estado, utilizando o acelerômetro e o magnetômetro para corrigir o erro do sistema.

As simulações utilizaram dados do Razor 9DOF, adquiridos em duas frequências: 50Hz e 500Hz. Escolheu-se essas frequências pois o arduíno tem limitado poder de processamento.

### 4.3.1. Observador não linear

A implementação em MatLab, demandou a discretização de todo o equacionamento.

A integração do sinal do giroscópio foi feita componente por componente do quaternion, para evitar erros acumulados das operações com matrizes. O método utilizado foi o trapezoidal, fazendo uso da medição atual e da anterior:

$$q_k^n = \frac{(q_k^n + q_{k-1}^n) * dt}{2} \quad (4.1)$$

As funções de transferências dos filtros (passa-baixas e passa-altas) e do controlador também tiveram de ser discretizadas, sendo escritas em equações de diferenças.

As equações 3.19 do filtro passa-baixas, aplicada sobre a medição realizada pelo acelerômetro, se apresenta em equações de diferenças como segue:

$$A_k^f = 0.0305 * (A_k^r + A_{k-1}^r) + 0.9391 * A_{k-1}^f, \text{ para } f_a=50\text{Hz} \quad (4.2.a)$$

$$A_k^f = 0.0031 * (A_k^r + A_{k-1}^r) + 0.9937 * A_{k-1}^f, \text{ para } f_a=500\text{Hz} \quad (4.2.b)$$

sendo  $A_k^f$  o sinal do acelerômetro filtrado no passo k e  $A_k^r$  e  $A_{k-1}^r$  os sinais não tratados do acelerômetro nos passos k e k-1, respectivamente.

As equações 3.20 do filtro passa-altas, aplicada sobre a medição das velocidades angulares realizada pelo giroscópio, é escrita em equações de diferenças na forma:

$$G_k^f = 0.9695 * (G_k^r - G_{k-1}^r) + 0.9391 * G_{k-1}^f, \text{ para } f_a=50\text{Hz} \quad (4.3.a)$$

$$G_k^f = 0.9969 * (G_k^r - G_{k-1}^r) + 0.9937 * G_{k-1}^f, \text{ para } f_a=500\text{Hz} \quad (4.3.b)$$

sendo  $G_k^f$  o sinal filtrado do giroscópio no passo k e  $G_k^r$  e  $G_{k-1}^r$  os sinais não tratados do giroscópio nos passos k e k-1, respectivamente.

O controle PI, da equação 3.16, que anula o erro em regime, é escrito em equações de diferenças na forma:

$$L_k = (K_p + K_i) * e_k - K_p * e_{k-1} + L_{k-1} \quad (4.4)$$

sendo  $L_k$  a saída do controlador no passo k,  $e_k$  e  $e_{k-1}$  os erros calculados nos passos k e k-1, respectivamente.  $K_p$  e  $K_i$  são os ganhos proporcional e integrativo e a eles foram atribuídos os valores 0.04 e 0.002, respectivamente, assim como em Mahony et al, 2006.

#### 4.4. Filtro de Kalman de estágio duplo

Duas implementações do método proposto por Sabatelli (2013) foram realizadas no Matlab, a primeira utilizando como dados as leituras dos sensores de movimento de um iphone e a segunda, utilizando os dados adquiridos da IMU Razor 9DOF.

Alguns passos do algoritmo foram observados com maior atenção na implementação. O primeiro foi a atualização da matriz de covariância dos erros,  $P$ . Essa matriz tem que se manter antissimétrica em todas as iterações. Ruídos e arredondamentos durante as iterações comprometem esta propriedade, o que põe em xeque a qualidade dos resultados do método. Outro ítem que também mereceu atenção foi a normalização dos vetores de quaternions de descreve a orientação do veículo. O método consiste em dois estágios, sendo que em cada um destes estágios o quaternion é recalculado a partir das correções geradas pelo filtro, e em cada cálculo, o quaternion deve ser renormalizado, com exceção dos quaternions  $q_{E1}$  e  $q_{E2}$  que fornecem os gradientes de correção.

Algumas simplificações foram adotadas, da mesma forma que o autor fez, para estimativa das matrizes  $\mathbf{Q}$ ,  $\mathbf{R}_1$  e  $\mathbf{R}_2$ .

$$\mathbf{Q} = 10^{-6} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$$\mathbf{R}_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (4.6)$$

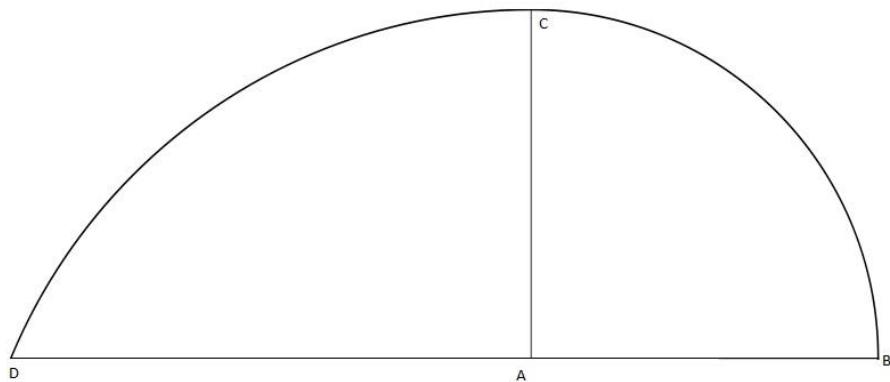
$$\mathbf{R}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

Na primeira iteração, foi necessário inicializar a matriz de covariância de erros. A inicialização que o autor propõe, e que também foi adotado neste trabalho foi:

$$\mathbf{P}_0 = \begin{bmatrix} 0.125 & 0.0003 & 0.0003 & 0.0003 \\ 0.0003 & 0.125 & 0.0003 & 0.0003 \\ 0.0003 & 0.0003 & 0.125 & 0.0003 \\ 0.0003 & 0.0003 & 0.0003 & 0.125 \end{bmatrix} \quad (4.8)$$

Em Sabatelli(2013), propõe-se que esta inicialização seja assumida nas 128 primeiras iterações, e após isso, a matriz de covariância dos erros fosse atualizada a cada iteração. Nas simulações aqui realizadas, esta matriz só foi utilizada na primeira iteração, sendo atualizada com os valores calculados apartir da segunda iteração.

Os dados obtidos pelos sensores do iPhone para implementação do método foram as medições de um movimento controlado. O movimento consistiu no percurso demonstrado na figura 4-3. Ou seja, o ângulo de yaw variou de acordo com o percurso ABCD. O dispositivo se manteve com ângulos de rotação de Euler ( $\text{yaw}, \text{pitch}, \text{roll}$ ) = ( $\text{yaw}, 0, 0$ ) de A até B, em seguida girou-se o mesmo até ( $\text{yaw}, -90^\circ, 0$ ) de B até C; de C até D os ângulos foram alterados para ( $\text{yaw}, 0, 90^\circ$ ), e de D até A voltou-se a ( $0, 0, 0$ ).



*Figura 4-3- Trajetória do iPhone durante a aquisição das medições dos sensores inerciais.*

Vale salientar que o período de amostragem dos sensores do iPhone não pode ser controlado, mas os valores das medições do tempo vem agregados ao log das medições inerciais. Desta forma, o  $dt$  foi estimado como a diferença média entre cada valor do tempo medido. Este já é uma grande fonte de erro de estimativa pelo método.

As simulações seguintes foram realizadas com os dados obtidos do Razor 9DOF, realizando um movimento de aproximadamente  $90^\circ$  no sentido positivo e negativo de roll, pitch e yaw, nessa ordem.

A primeira simulação, foi realizada com uma frequência de aquisição de 50Hz, e a segunda com uma frequência de aquisição de 500Hz.

Todas as simulações realizadas adotaram os valores de  $Q$ ,  $R_1$  e  $R_2$  mostrados acima.

## 4.5. Resultados das simulações

Apresenta-se, nesta sub-seção, os resultados obtidos nas simulações.

### 4.5.1. Matriz de cossenos diretores

Seguem os resultados da simulação aplicando a matriz de cossenos diretores sobre os sinais do acelerômetro e do magnetômetro como forma de estimativa dos ângulos de roll, pitch e yaw.

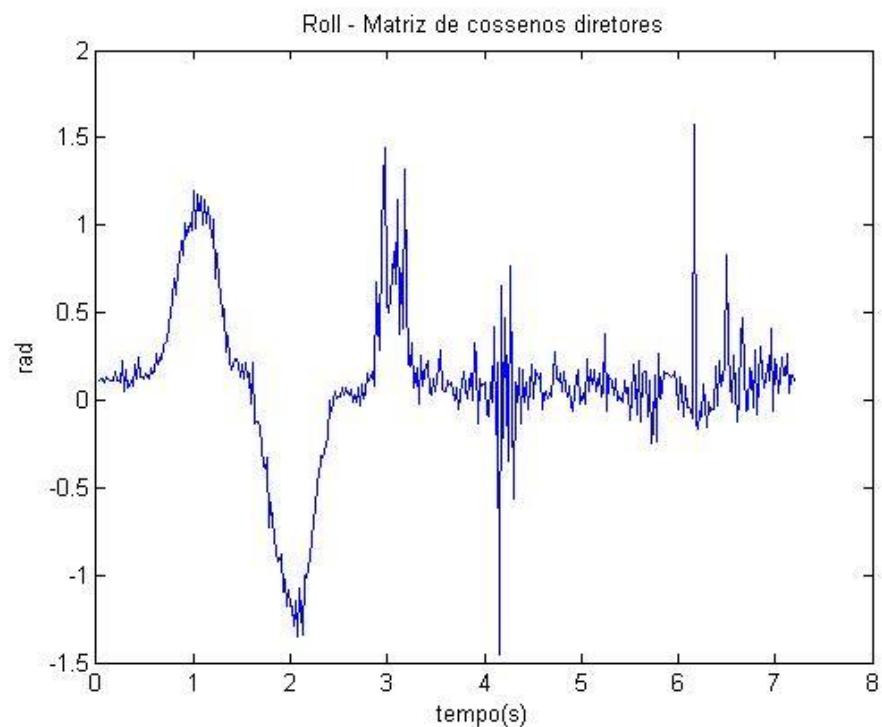


Figura 4-4- Variação do ângulo de Roll por matriz de cossenos diretores. Frequência de aquisição = 50Hz s.

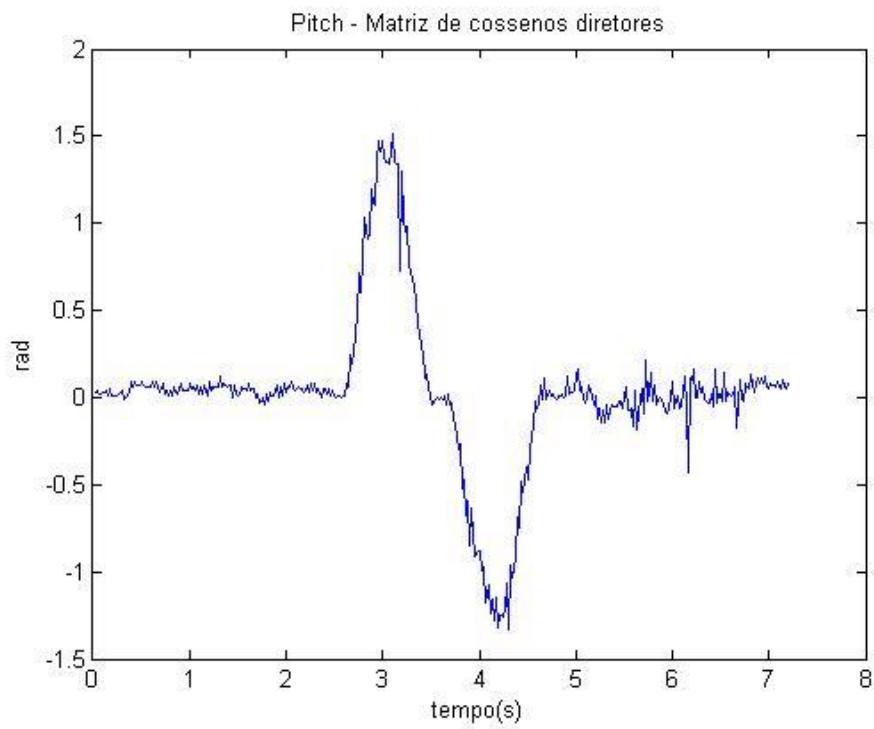


Figura 4-5-Variação do ângulo de Pitch por matriz de cossenos diretores. Frequência de aquisição = 50Hz

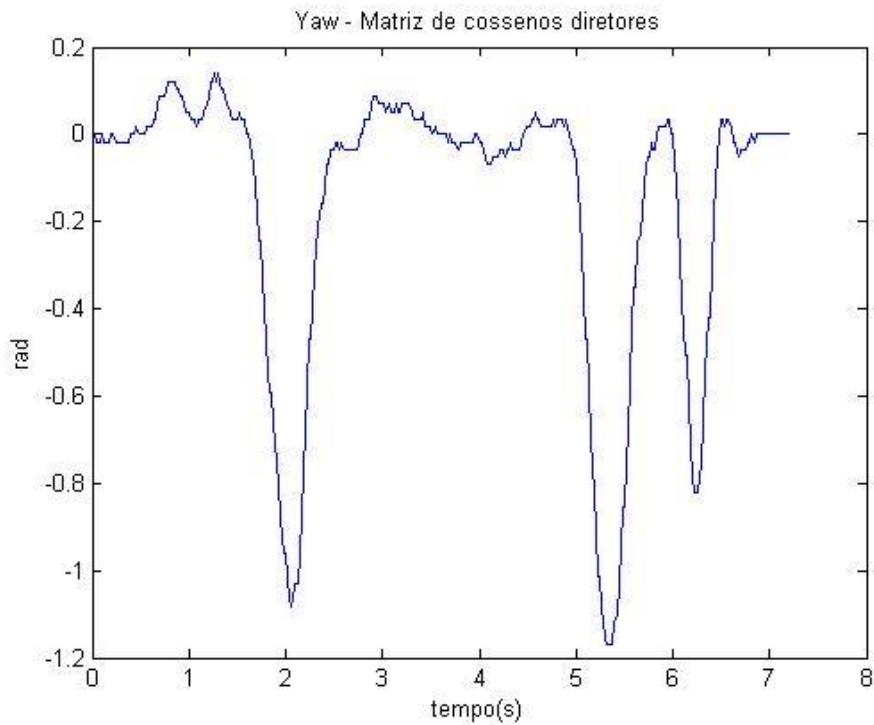


Figura 4-6-Variação do ângulo de Yaw por matriz de cossenos diretores. Frequência de aquisição = 50Hz

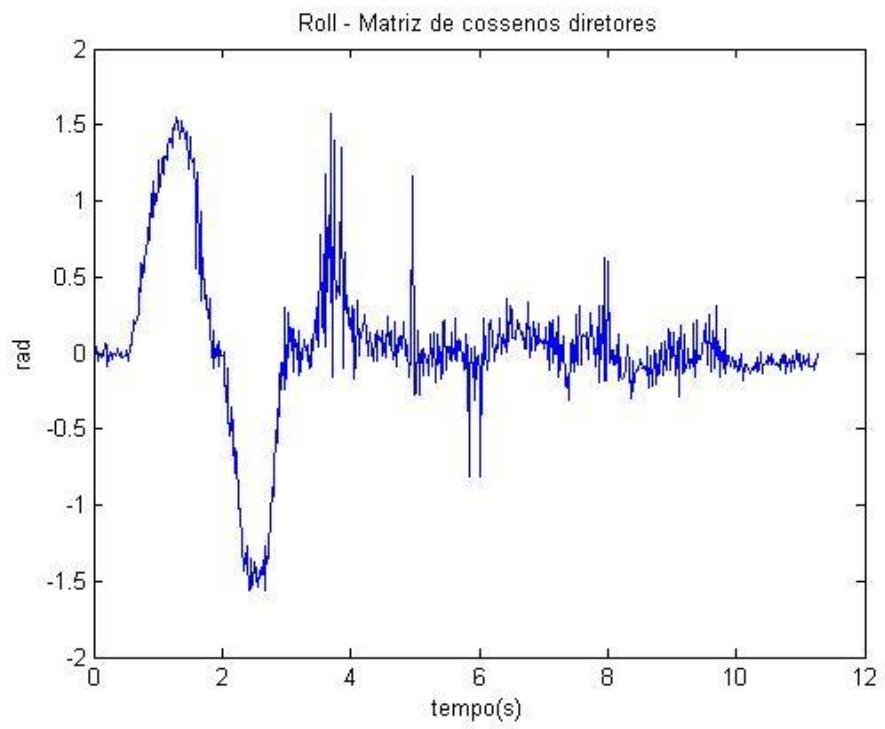


Figura 4-7 - Variação do ângulo de Roll por matriz de cossenos diretores. Frequência de aquisição = 500Hz

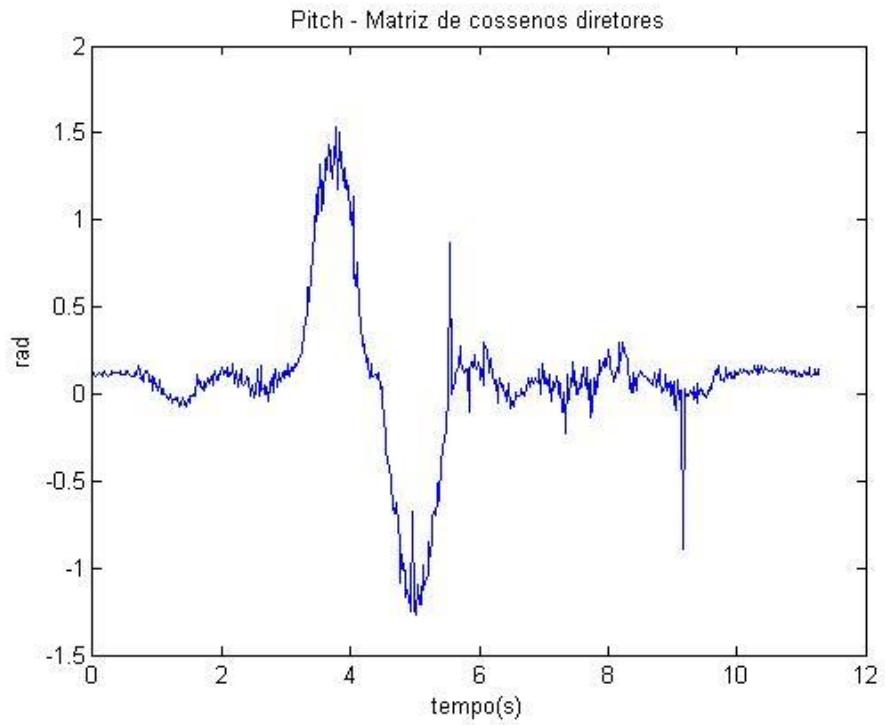


Figura 4-8 - Variação do ângulo de Pitch por matriz de cossenos diretores. Frequência de aquisição = 500Hz

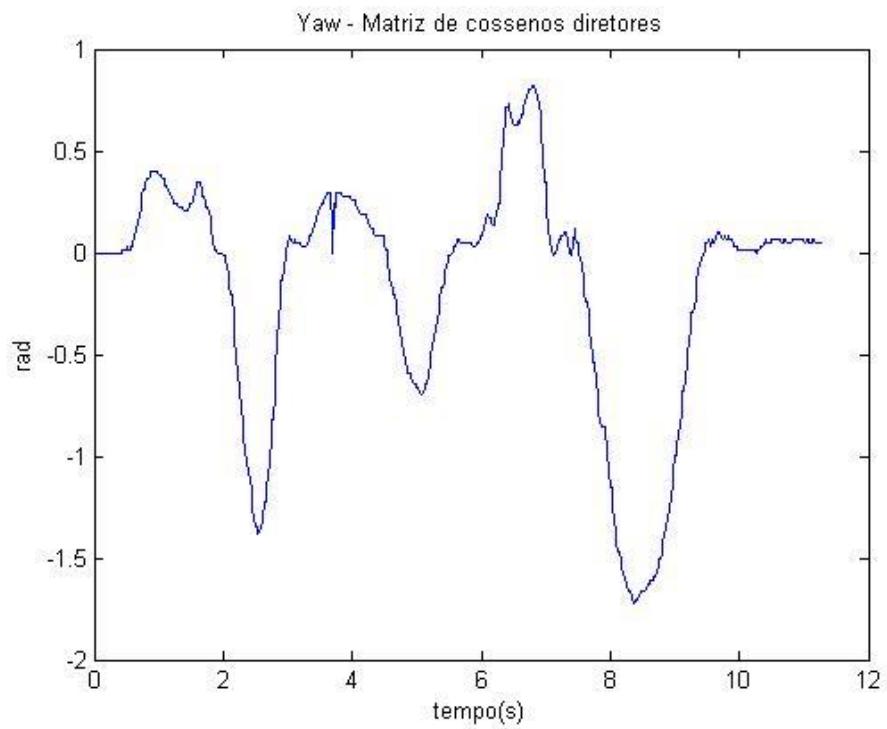


Figura 4-8 - Variação do ângulo de Yaw por matriz de cossenos diretores. Frequência de aquisição = 500Hz

Para fins comparativos, realizou-se também a integração do sinal do griscópio para estimativa dos ângulos de Euler. Seguem os resultados.

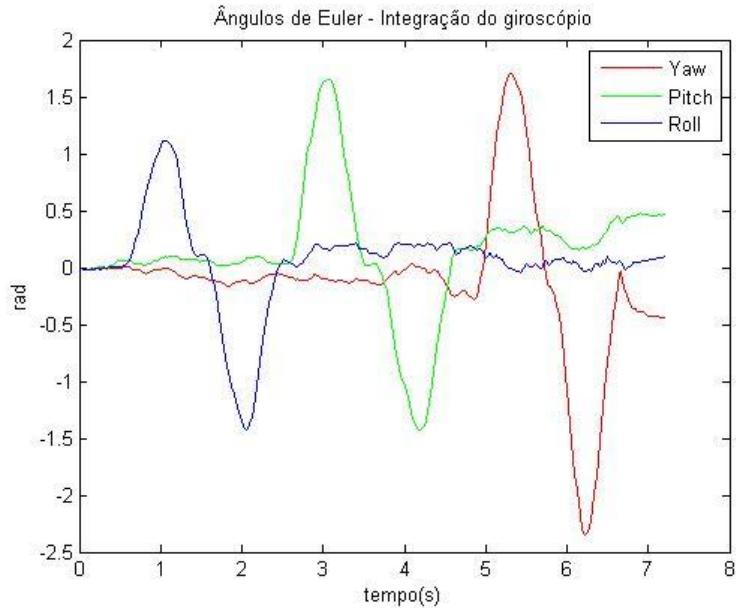
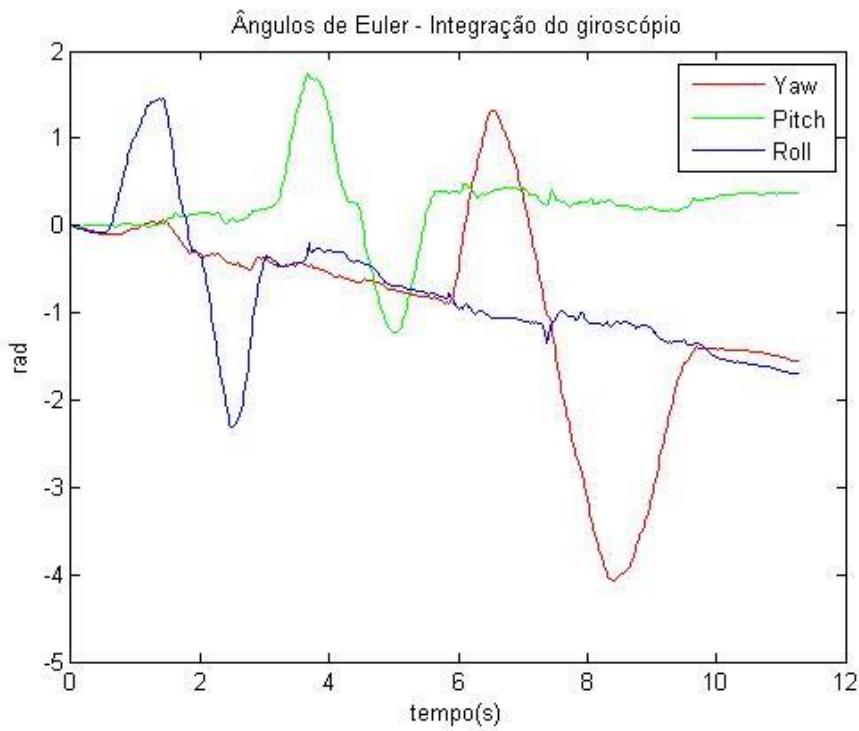


Figura 4-9- Variação de Roll, Pitch e Yaw, apartir da integração do sinal do giroscópio. Frequência de aquisição =50Hz.



*Figura 4-10- Variação de Roll, Pitch e Yaw, apartir da integração do sinal do giroscópio. Frequência de aquisição = 500Hz.*

É válido observar que, mesmo sem tratamento algum, o método dos cossenos diretores e a integração dos sinais do giroscópio fornecem uma estimativa razoável para os ângulos de roll, pitch e yaw.

#### 4.5.2. Observador de estados

Neste método aplicou-se um filtro passa-baixas no sinal do acelerômetro, que corrigiu o resultado da integração das medições do giroscópio, que teve seu sinal filtrado por um filtro passa-altas. Abaixo, seguem os espectros de frequência dos sinais do acelerômetro e do giroscópio, crus e filtrados, demonstrando que as frequências indesejadas foram realmente removidas.

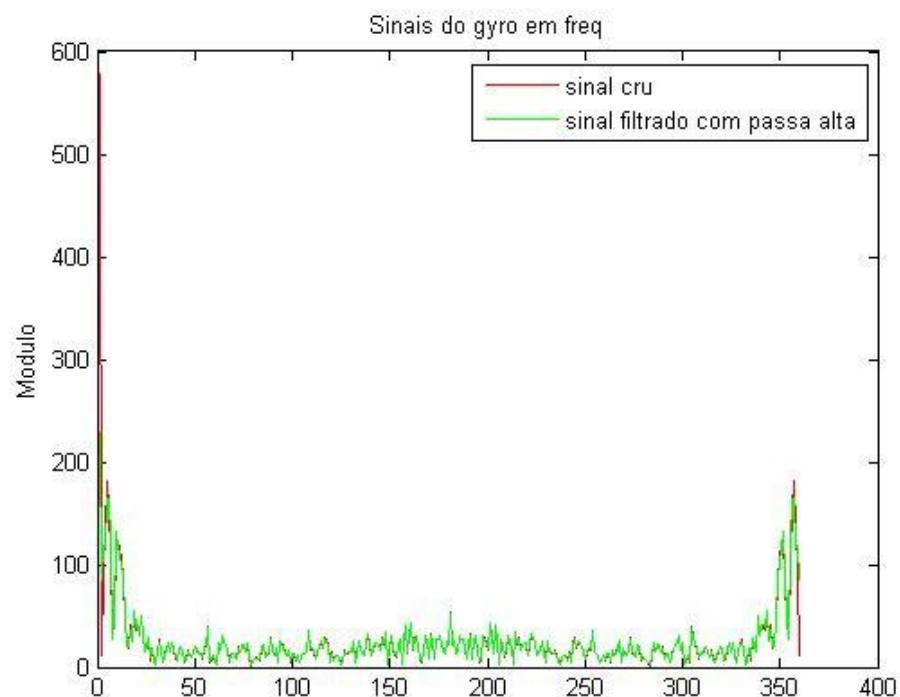


Figura 4-11 - Sinal cru e filtrado, após passar pelo passa-altas, do giroscópio.

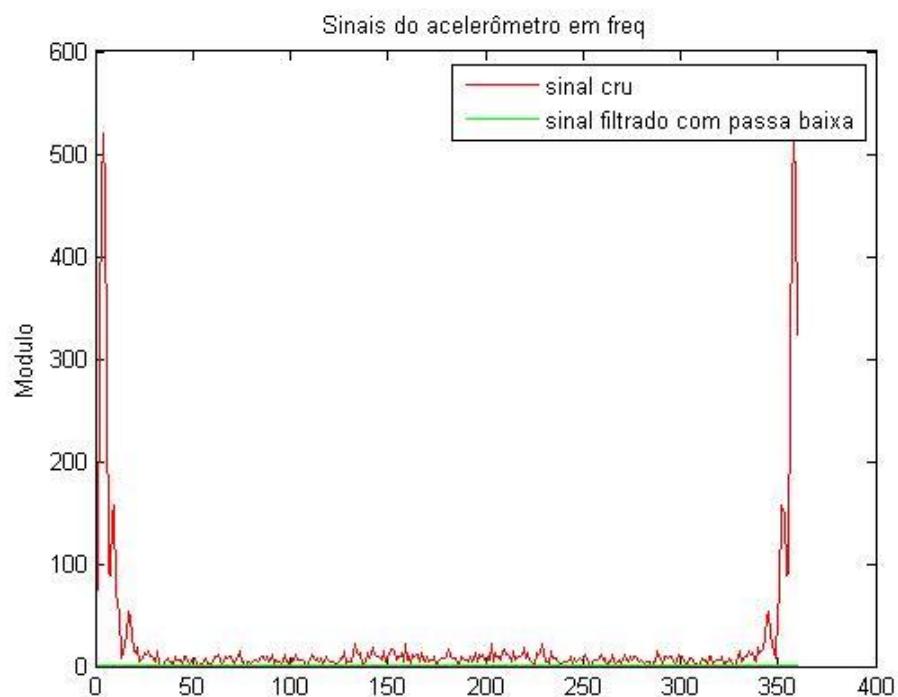


Figura 4-12 - Figura 5 1 - Sinal cru e filtrado, após passar pelo passa-baixas, do acelerômetro.

Os resultados das simulações do método de filtros complementares com observadores de estados segue abaixo.

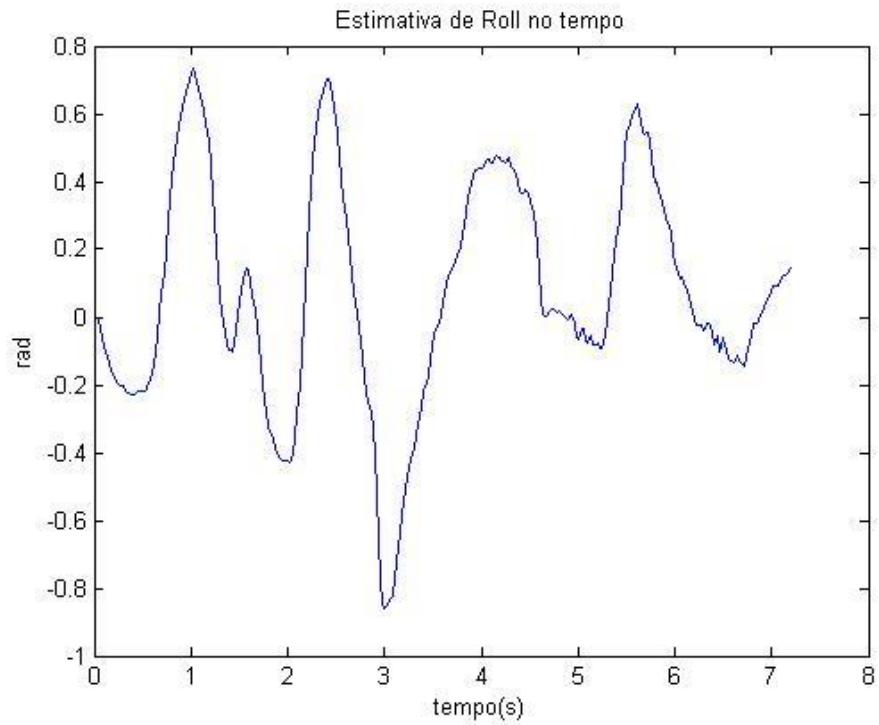


Figura 4-13 - Variação de Roll no tempo. Frequência de aquisição 50Hz.

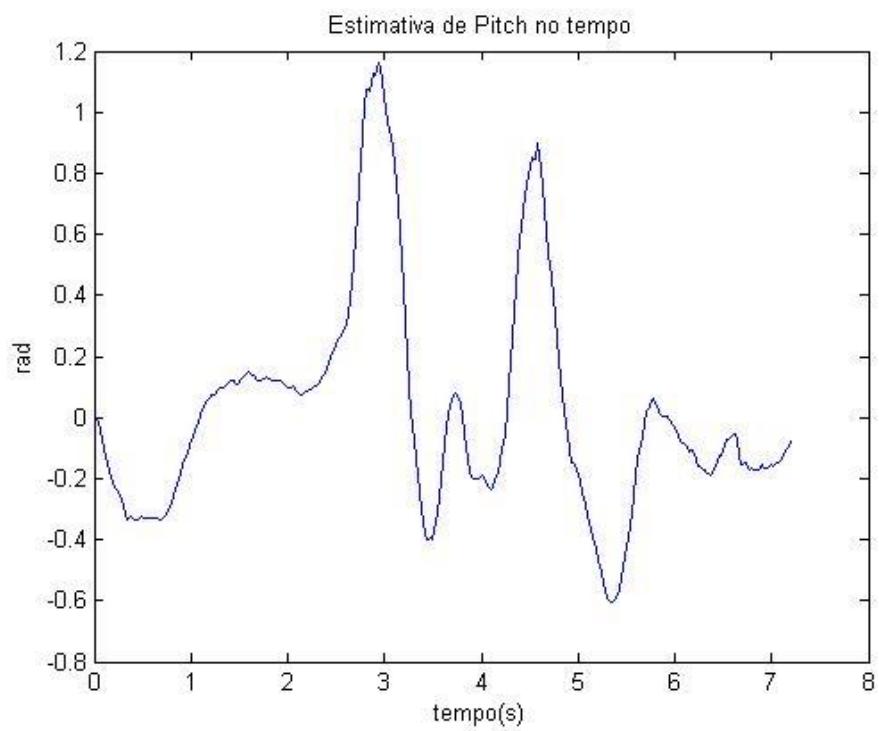


Figura 4-14 - Variação de Pitch no tempo. Frequência de aquisição 50Hz.

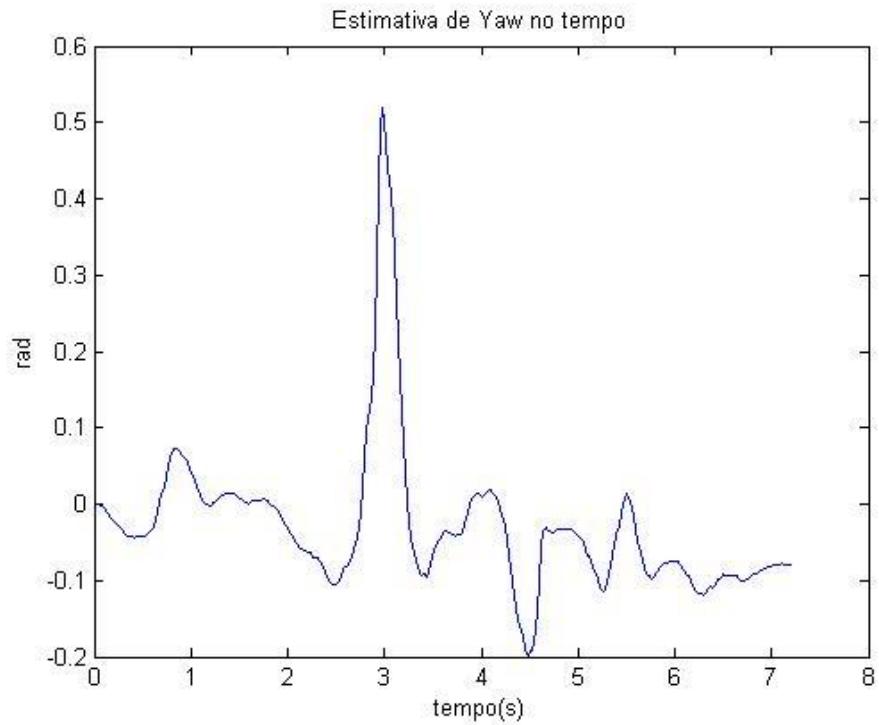


Figura 4-15- Variação de Yaw no tempo. Frequência de aquisição 50Hz.

Os resultados da simulação utilizando os dados adquiridos a 500Hz foram:

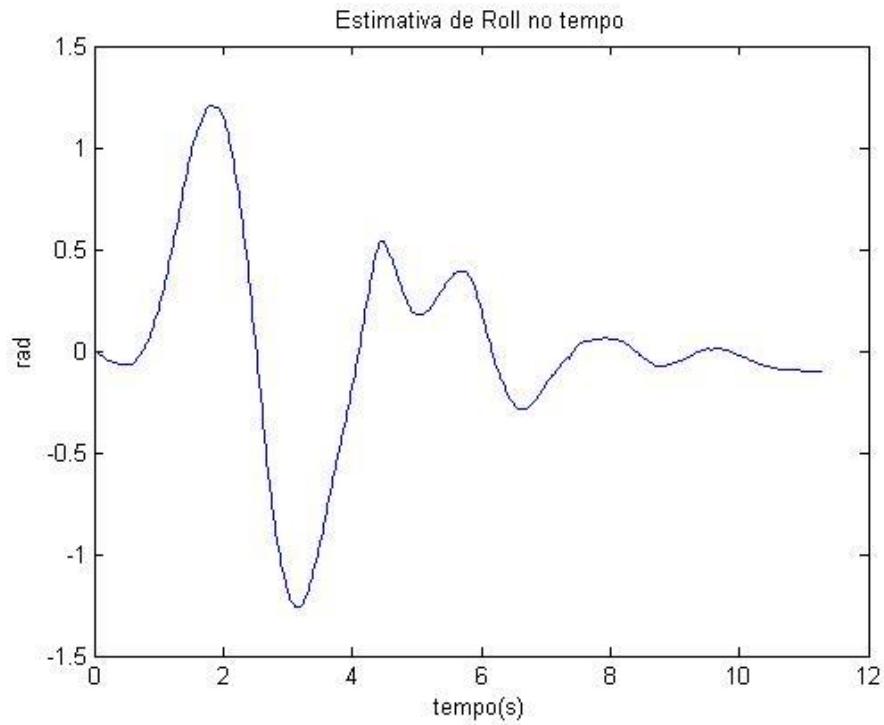


Figura 4-16- Variação de Roll no tempo. Frequência de aquisição 500Hz.

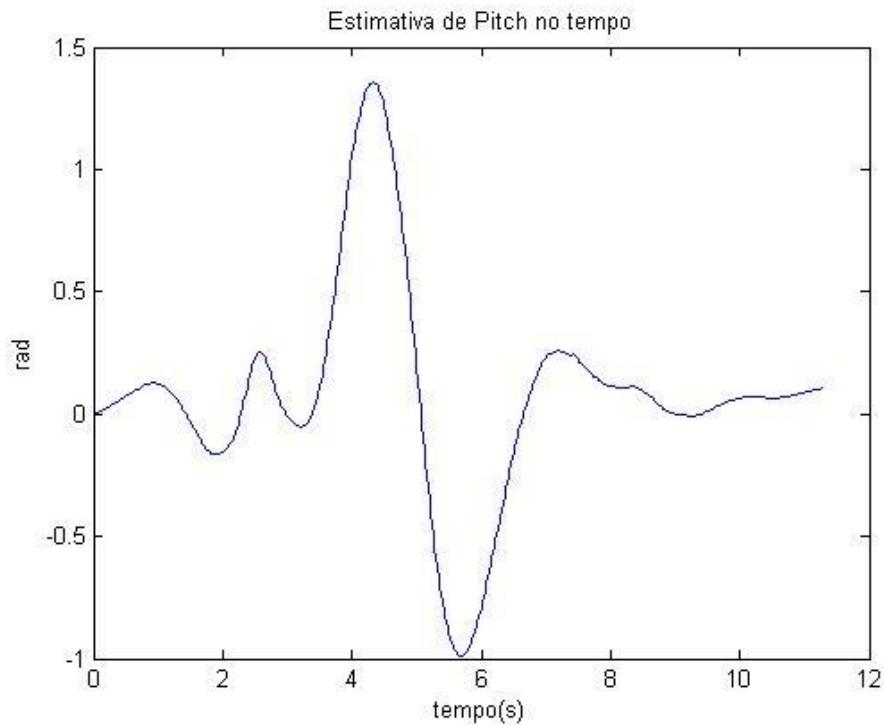


Figura 4-17- Variação de Pitch no tempo. Frequência de aquisição 500Hz.

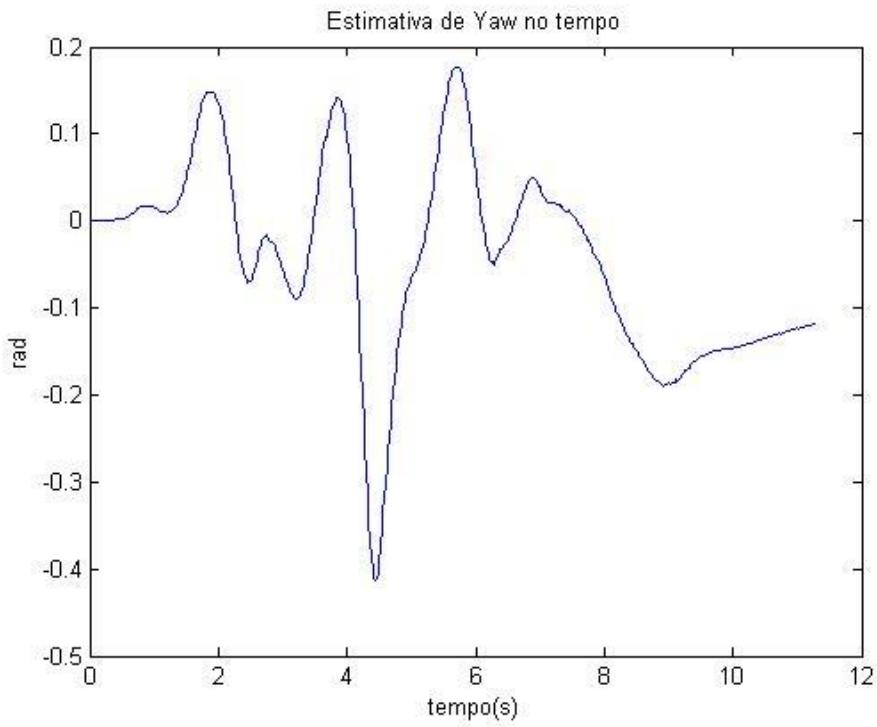


Figura 4-18 - Variação de Yaw no tempo. Frequência de aquisição 500Hz.

Dos resultados podemos fazer algumas observações. A primeira é de que, mesmo utilizando o magnetômetro, a estimativa do ângulo de Yaw ainda não é satisfatória para o uso em sistemas de controle. Mesmo com uma frequência de aquisição maior, a estimativa ainda foi imprecisa. Outra observação é que a estimativa dos ângulos de Roll e Pitch tiveram uma melhora considerável com o aumento da frequência de aquisição. A filtragem do sinal do giroscópio foi eficiente em anular as baixas freqüências que poderiam modular o valor integrado das velocidades angulares.

Por fim, outra observação que se tira dos resultados, é que mesmo com uma alta freqüência de aquisição, a precisão oferecida pelo método não é confiável em sistemas de controle que necessitem de precisão maior que 0,5 rad na estimativa da orientação.

#### 4.5.3. Filtro de Kalman de estágio duplo

A primeira simulação do filtro de Kalman de estágio duplo foi realizada utilizando os dados dos sensores iniciais de um iPhone. Os resultados seguem a baixo.

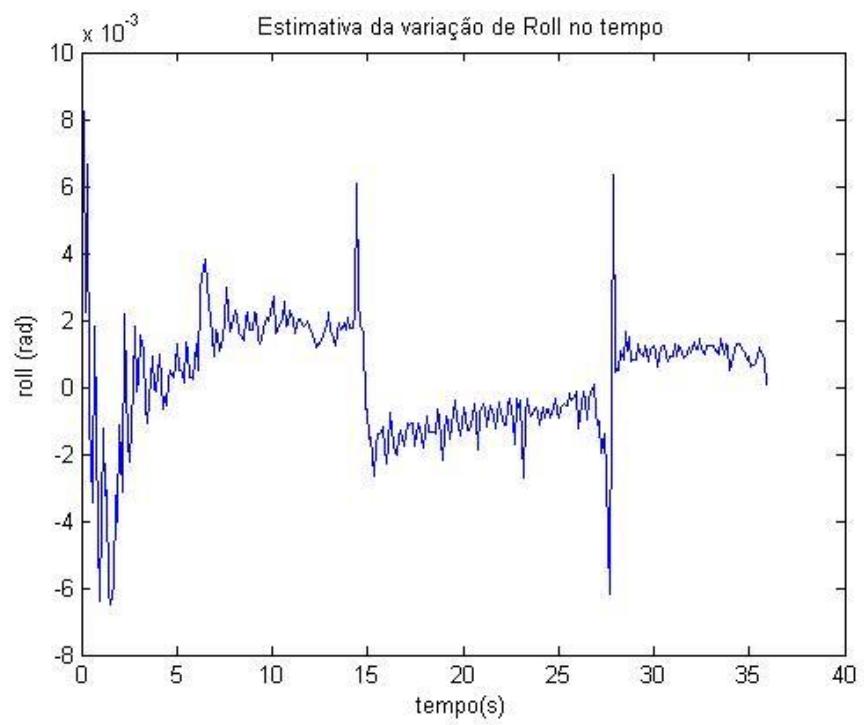


Figura 4-19 Resultado da estimativa de roll, usando o filtro de Kalman de estágio duplo.

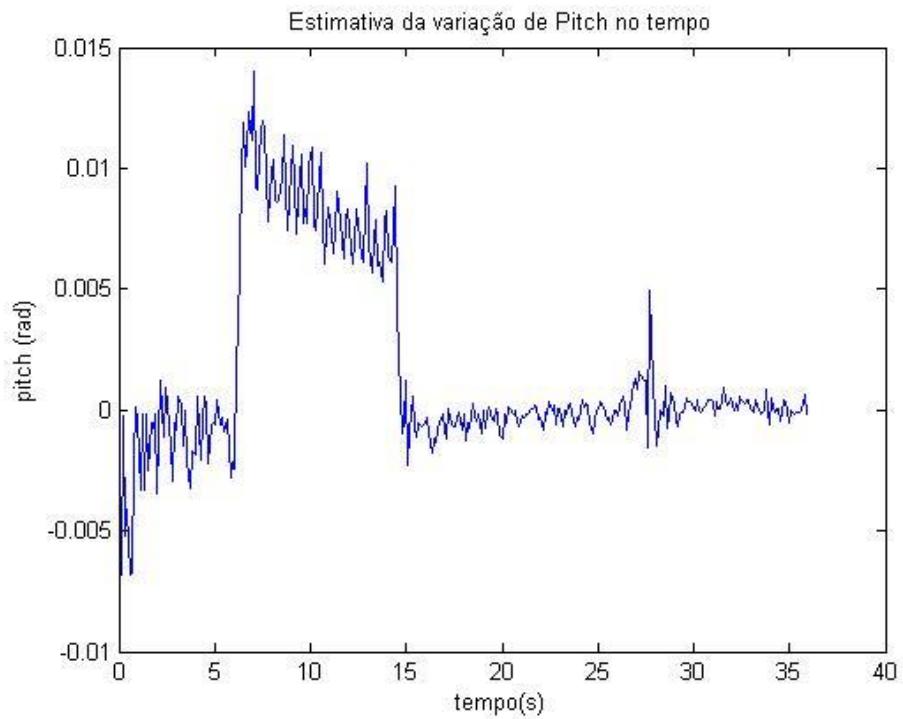


Figura 4-20 Resultado da estimativa de pitch, utilizando o filtro de Kalman de estágio duplo.

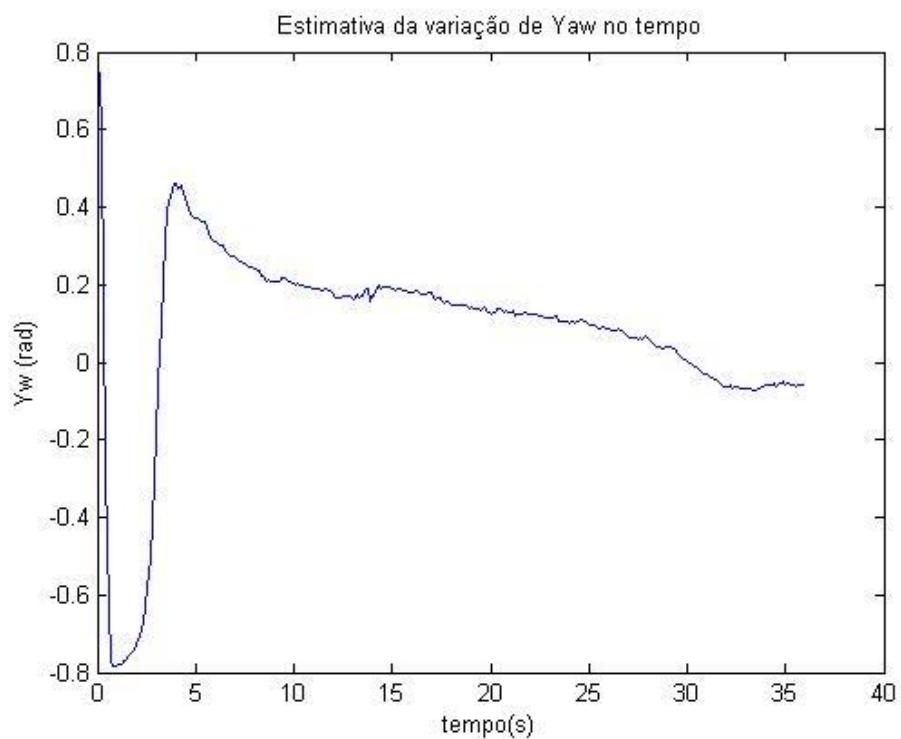


Figura 4-21 Resultado da estimativa de yaw, utilizando o filtro de Kalman de estágio duplo.

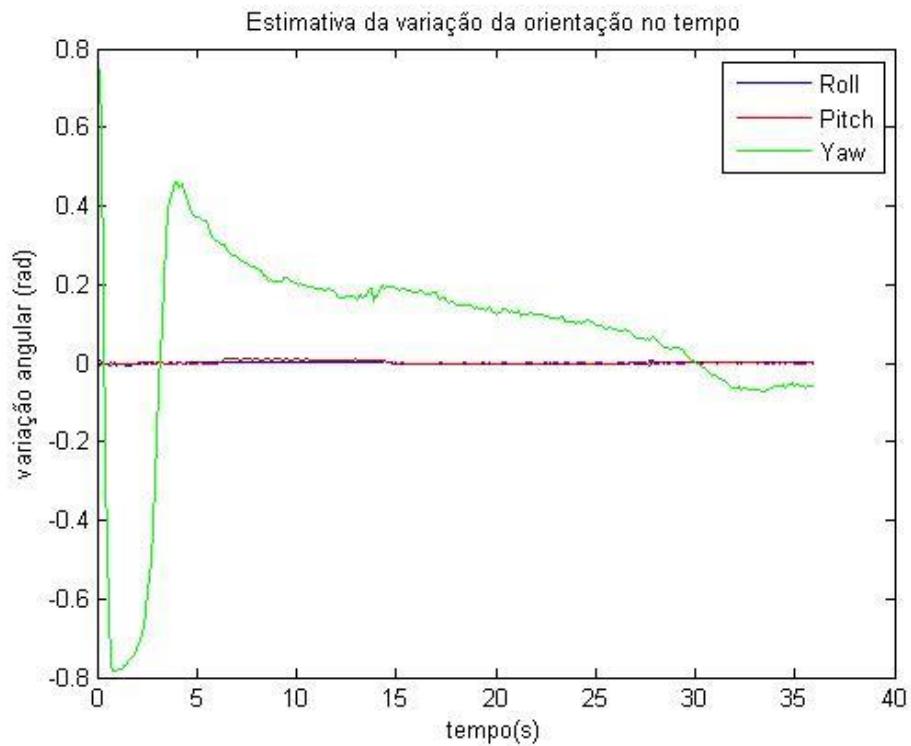
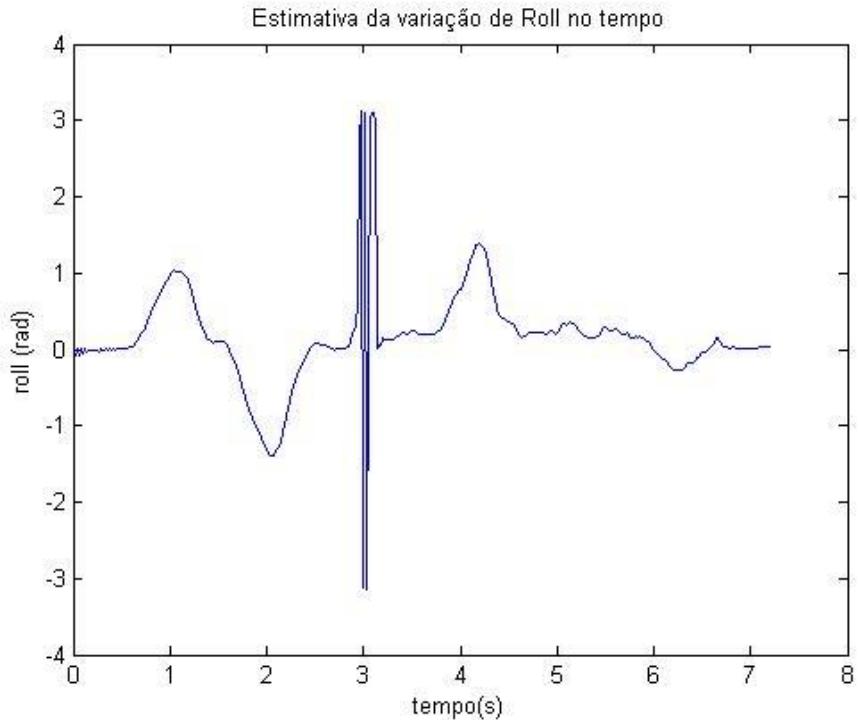


Figura 4-22 Resultado combinado dos angulos de roll, pitch e yaw, utilizando o filtro de Kalman de estágio duplo.

Observa-se que os ângulos de roll e pitch, mesmo com ruídos consideráveis, ainda fornecem uma boa estimativa do movimento do dispositivo, apesar de o valor absoluto de saída não indicar os 90° de giro. Já o ângulo de yaw começou apontando corretamente os 90° em relação ao norte magnético, mas em seguida pode-se observar que a medição se perde, em relação às direções esperadas, o que descredencia estas medições para o uso confiável no controle de navegação desejado para um drone. A variação angular que a estimativa de yaw permitira traçar seria um giro lento de 90° no sentido anti-horário, e depois a continuação deste giro por mais 90° mas em tempo maior. No início da trajetória, o dispositivo estava virado para leste, mas o magnetômetro demorou cerca de 5 segundos para conseguir identificar uma direção, e foi logo quando o mesmo virou-se para o norte, direção a qual o magnetômetro tem maior influência no algoritmo, como explicado nas seções anteriores, o que pode explicar as leitura longe do esperado.

As simulações seguintes foram realizadas utilizando dados do IMU Razor 9DOF. No total foram duas simulações, a primeira utilizando dados adquiridos a 50Hz, e a segunda utilizando dados adquiridos a 500Hz. Os resultados seguem abaixo.



*Figura 4-23 - Variação de Roll no tempo. Frequência de aquisição 50Hz.*

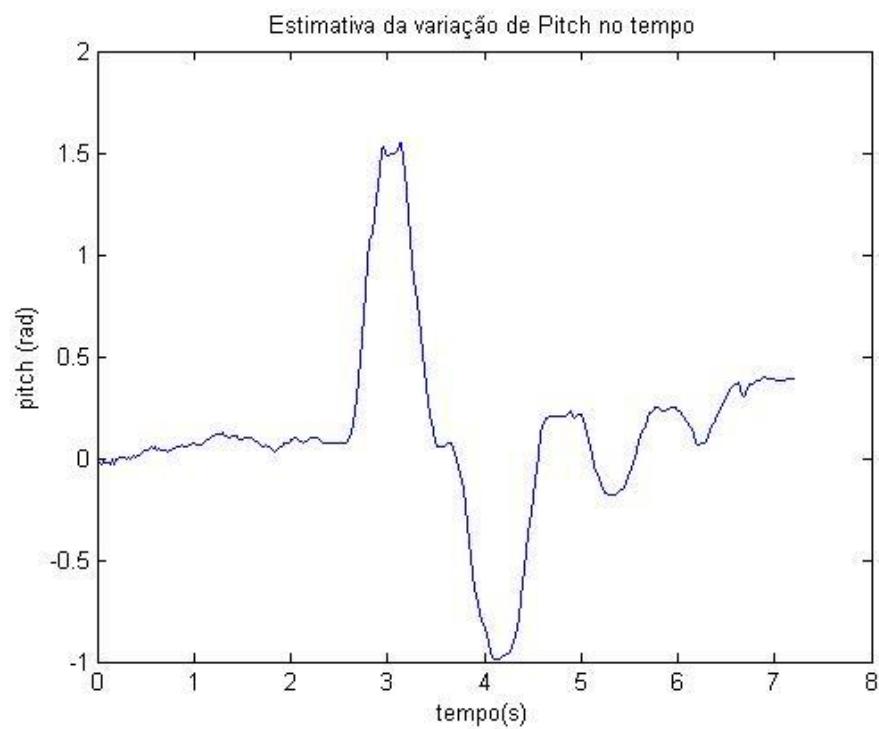


Figura 4-24 - Variação de Pitch no tempo. Frequência de aquisição 50Hz.

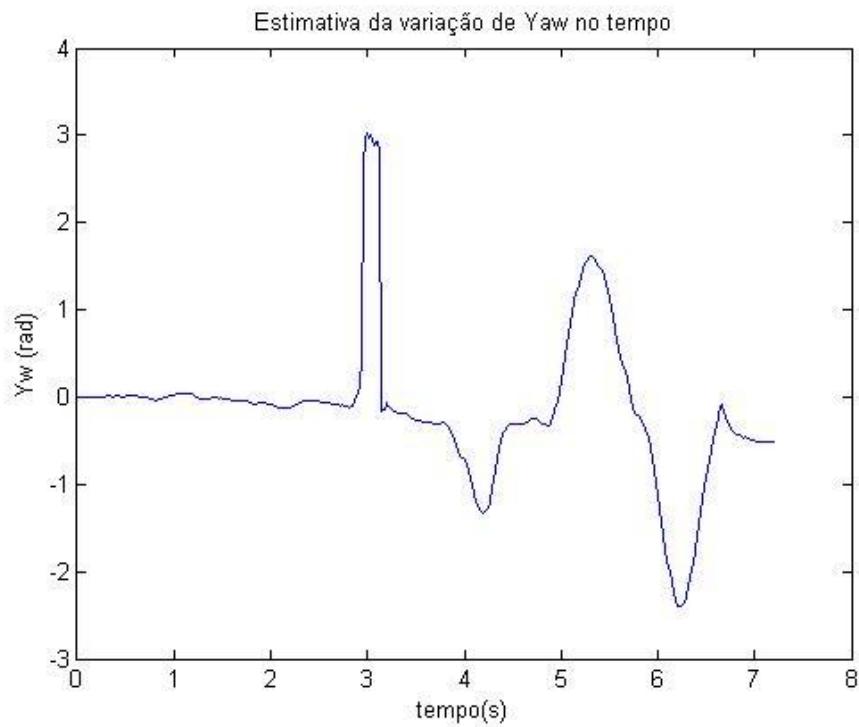


Figura 4-25 - Variação de Yaw no tempo. Frequência de aquisição 50Hz.

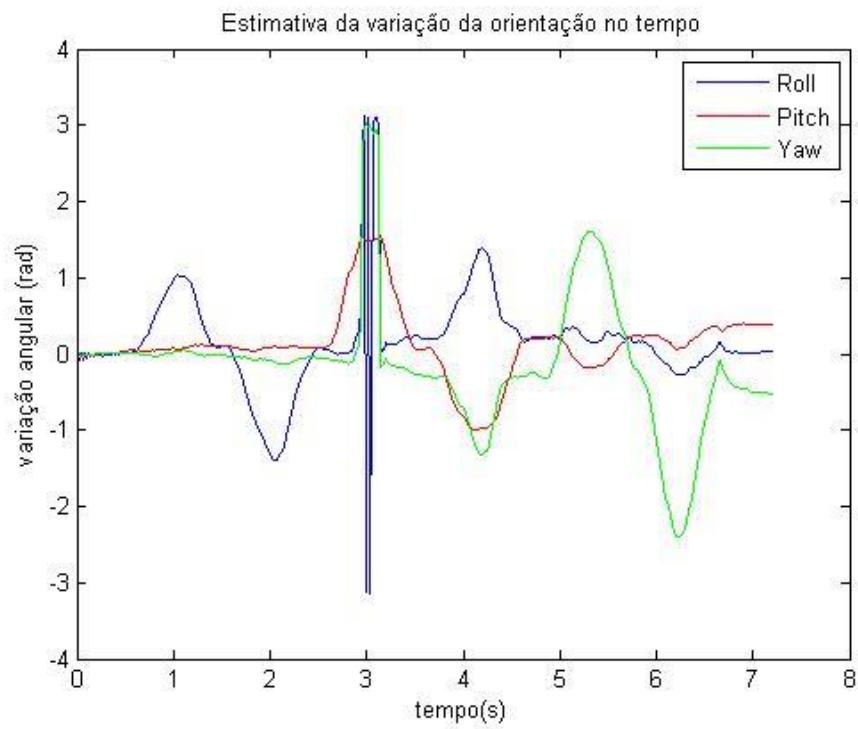


Figura 4-26 - Variação de Roll, Pitch e Yaw no tempo. Frequência de aquisição 50Hz.

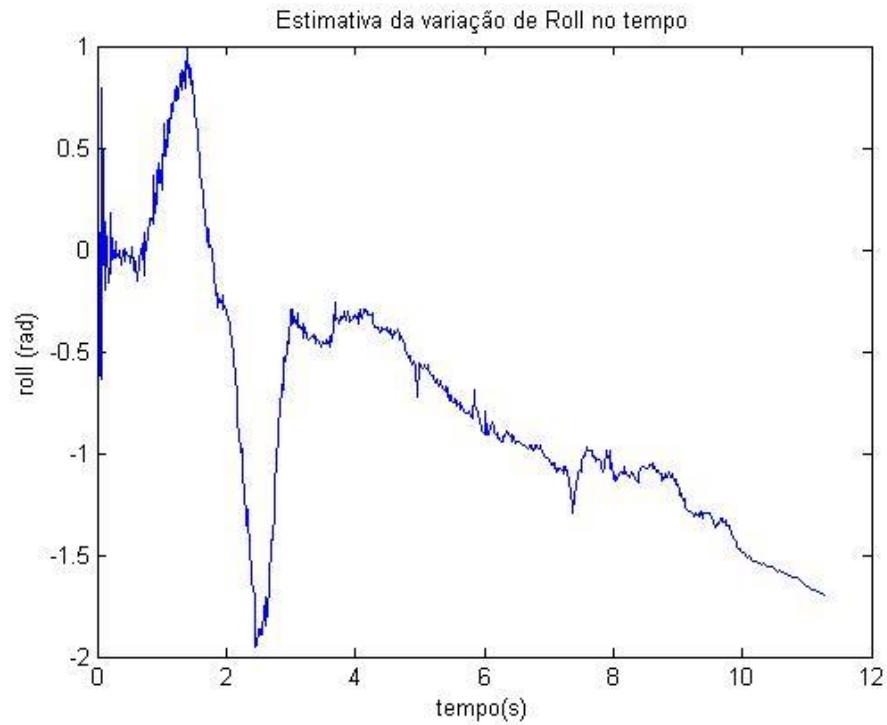


Figura 4-27 - Variação de Roll no tempo. Frequência de aquisição 500Hz.

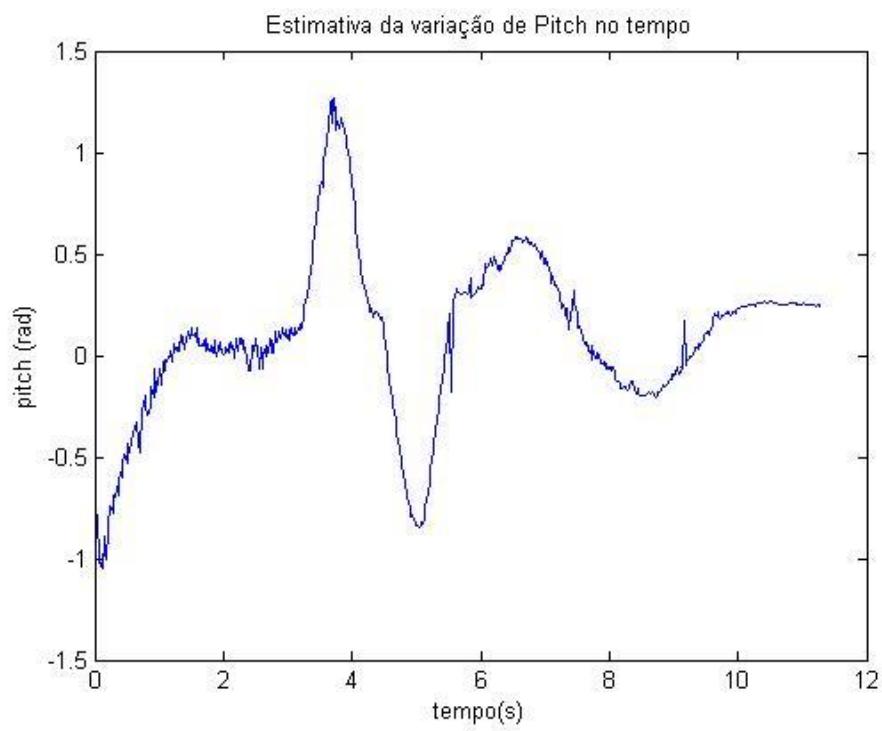


Figura 4-28 - Variação de Pitch no tempo. Frequência de aquisição 500Hz.

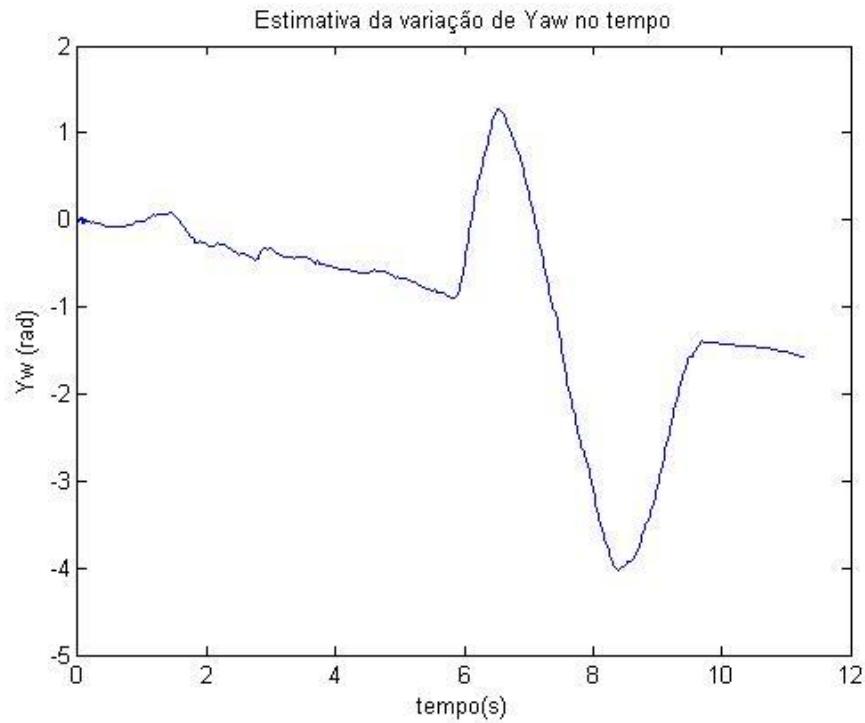


Figura 4-29 - Variação de Yaw no tempo. Frequência de aquisição 500Hz.

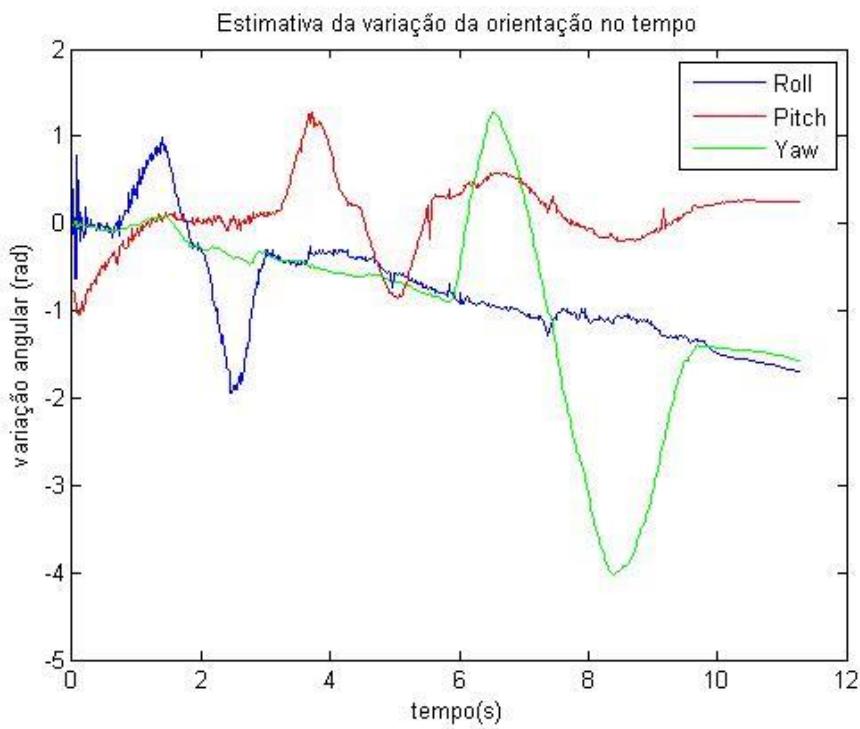


Figura 4-20 - Variação de Roll, Pitch e Yaw no tempo. Frequência de aquisição 500Hz.

A primeira observação ser feita é quanto à estimativa do ângulo de Yaw, que apresenta a mesma precisão da estimativa dos ângulos de Roll e Pitch, destacando o método de Kalman em relação ao de filtros complementares. A simulação a 50Hz se mostrou eficiente e confiável. O comportamento da orientação do corpo rígido é estimado satisfatoriamente, e, para controles nos quais a frequência de realimentação é da ordem dos 50Hz da frequência de aquisição, o método já é o suficiente.

Caso o sistema de controle requira uma maior frequência de aquisição, na ordem dos 500Hz, o método não apresenta o mesmo desempenho, mas ainda assim, consegue fornecer boa estimativa da orientação do corpo.

## 5. IMPLEMENTAÇÃO

Em posse dos resultados colhidos nas simulações, parte-se para a implementação dos métodos em tempo real. Os algorítmos devem agora ser escritos em linguagem C, e então embarcados no processador do arduíno ou do próprio sensor (ambos são ATmega328). A implementação pode ser realizada utilizando duas arquiteturas:

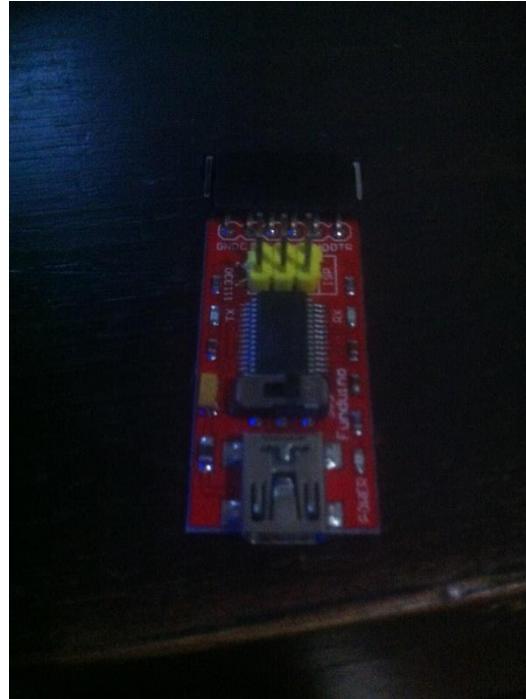
- Arquitetura com intermédio do Arduino: sensor transmite dados crus para o Arduino, que processa os dados utilizando um dos métodos utilizados, e então transmite os resultados ao computador, onde pode-se observar o comportamento dos ângulos de euler em um gráfico ângulo versus tempo.
- Arquitetura direta: sensor processa e transmite dados para o computador, utilizando o Arduino como simples plataforma de comunicação serial.

Toda comunicação entre sensor, arduíno e computador se realiza pelas respectivas portas seriais. Em ambas as arquiteturas acima, existe a comunicação sensor-arduino e arduino-computador. Essas interfaces tem um custo computacional, o que limita a capacidade do Atmega328 tanto do sensor como do Arduino de realizar as recursos necessárias para os métodos de filtragem e estimativa de posicionamento. Como será exposto a seguir, os tempos de duração de cada recursão não são constantes, para permitir que o processador consiga realizar todos os cálculos necessários sem comprometer a qualidade dos resultados. Qualidade essa que também pode ser comprometida pela precisão do ponto flutuante adotado no processador, em relação às simulações. As simulações ocorreram em MatLab que processa os dados na capacidade que o computador onde está instalado permite, 64 bits no caso. O arduíno possui ponto flutuante em 16 bits. Em primeiras análises, constatou-se que tal limitação não afeta de forma significante os resultados, mas deve se levar em consideração que o custo de cálculos mais complexos, como multiplicação e inversão de matrizes, faz com que as iterações não sejam tão rápidas como nas simulações.

## **5.1. Comunicação serial**

Da mesma forma que nas simulações, a comunicação entre computador, arduino e sensor é feita pelas respectivas portas seriais. A uma taxa de Baud de 57600 bits por segundo, estabelece-se a transmissão de dados entre os 3 terminais do sistema.

Na implementação da arquitetura direta é necessário o embarque do código em C no processador Atmega328 do sensor. Para isso, estabelece-se uma comunicação direta entre o computador e o Razor 9DOF, utilizando o Arduíno Pro-Mini 3.3V como conversor do porta serial para porta USB, conectando as portas DTR e CTS do sensor nas respectivas portas do Pro-Mini 3.3V.



*Figure 5-1 - Arduino Pro-Mini 3.3V ATmega328*

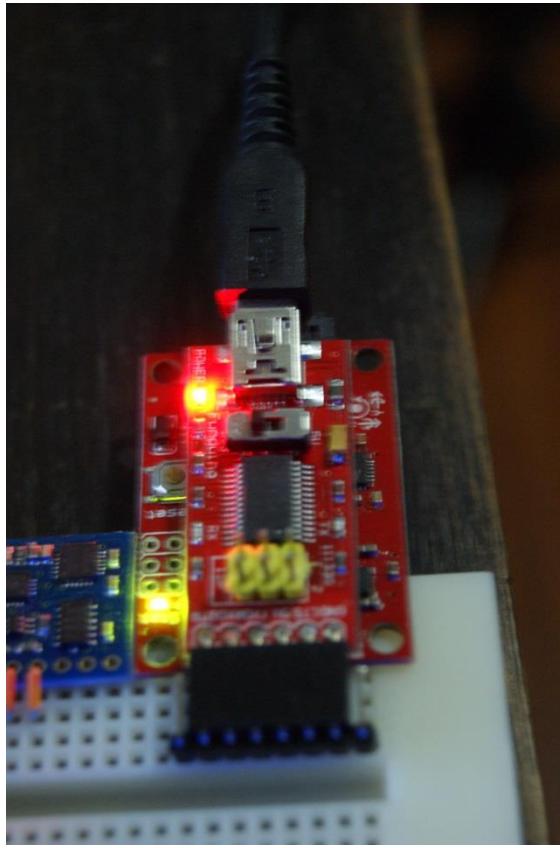


Figure 5-2 - Arduino Pro-Mini 3.3V ATmega328 conectado e transmitindo para o sensor Razor 9DOF.

Já na implementação com intermédio do Arduino Duemilanove, a comunicação serial é feita da mesma forma como na simulação, conectando-se simplesmente a porta serial do sensor à porta serial do Arduino, que conecta-se por USB ao computador. O Arduino Duemilanove pode simular o funcionamento do Arduino Pro-Mini, conectando-se os pinos das portas seriais, TX e RX, de alimentação, 3.3V e GND, além das portas DTR no reset do Arduino e CTS no GND do Arduino. A vantagem do Arduino Pro-Mini é o espaço ganho, pelo fato da montagem do Arduino no sensor sem compacta e direta, como mostrado na figura 5-2.

## 5.2. Interface Arduino – Matlab

Uma forma de observar qualitativamente o resultado das implementações dos métodos em tempo real é através da exibição do comportamento dos ângulos de euler em função do tempo transcorrido em um gráfico. O MatLab permite que as portas seriais do computador adquiram

dados vindos do sensor, e no software, através de recursões para manipulações destes dados, exibir o gráfico atitude versus tempo, enquanto o movimento acontece. Como o objetivo deste trabalho é que os algorítmos estejam embarcados, a única recursão a ser escrita no MatLab é a de captura e plotagem dos dados vindos do Atmega328. O programa embarcado no sensor faz com que os dados sejam transmitidos na ordem roll, pitch, yaw e intervalo de tempo entre a ultima transmissão e a transmissão em questão. Desta forma, a recursão no MatLab reconhece facilmente os dados e os envia para a janela do gráfico.

### **5.3.        Implemetação do filtro complementar**

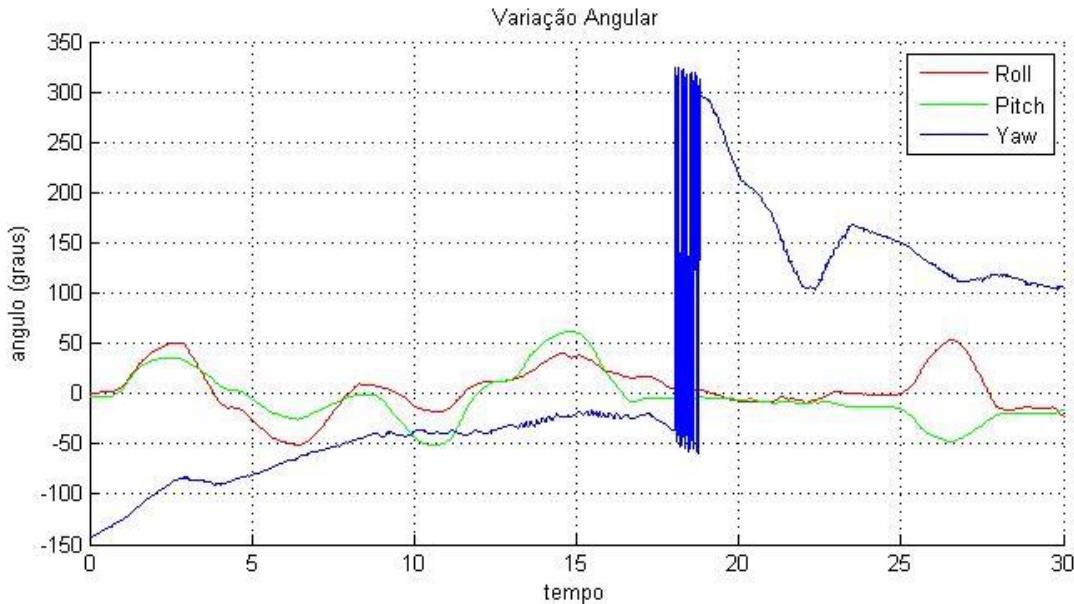
A comunicação representa um custo computacional para o Atmega328 que não pode ser desprezado. Como discutido acima, a comunicação entre três dispositivos, sensor, Arduino e computador, não permite que a memória dos processadores dos dois primeiros tenha espaço o suficiente para processar um método mais sofisticado como o filtro de Kalman de estágio duplo. Isso por que o Arduino teria de usar a porta serial para receber os dados do sensor, ao invés das portas analógicas, e, através da mesma porta serial, enviar os dados já processados ao computador. Como o sensor Razor 9DOF possui o mesmo processador Atmega328 que o Arduino Duemilanove, a melhor opção nesta aplicação é o embarque do algorítmico de tratamento no próprio sensor.

Mesmo utilizando o Atmega328 do sensor, que obtém as medições dos sensores por suas portas analógicas, a complexidade dos algorítmos tem de ser observado. O método de melhor desempenho nas simulações, o filtro de Kalman de estágio duplo, proposto por Sabatelli et al, 2013, não pode ser embarcado por completo no processador, por seu tamanho e complexidade de cálculos. Por este motivo, a primeira implementação a ser realizada é a da adaptação do filtro complementar de Mahony et al, 2006, que utiliza os dados do magnetômetro para estimativa do ângulo de yaw, como proposto na seção 4.5.2.

Este método tem a desvantagem de não tratar o sinal do magnetômetro, que entre os sensores, é o que é mais sensível a interferências externas, pois mede a direção do campo magnético da Terra. Quaisquer equipamentos elétricos que estiverem ligados nas proximidades do sensor, incluindo o arduino, ou motores do drone, podem afetar a leitura.

Esta adaptação é de baixíssimo custo computacional, mas peca em precisão, especialmente no ângulo de yaw.

A implementação do método em tempo real, realizando os mesmos movimentos das simulações, fornece o resultado como segue.



*Figure 5-3 - Resultado da implementação dos filtros complementares em tempo real.*

Considerando movimentos de rotação ao redor dos eixos a uma baixa frequência, nota-se que os ângulos de Roll e Pitch obtém bom resultado, apesar de a variação de um afetar levemente a estimativa do outro. É possível observar que as variações na ordem de 50 graus nos sentidos positivos e negativos dos eixos x e y são estimadas de forma eficiente.

Já a variação de yaw é notavelmente mal estimada, já que os dados do magnetômetro não recebem nenhum tratamento. Um fato que pode ser observado em seguidas aquisições de dados é que por mais que tente-se corrigir pelo software, a direção inicial tem um offset na ordem de 100°, o que acarreta em um tempo de estabilização ao redor de 0, o que na figura 5-3 acontece por volta de 18 segundos, que é quando a variação de 90° é aplicada ao redor de yaw no experimento. O início do movimento é mal interpretado pelo magnetômetro e só depois de 1 segundo, que a direção volta a ser indicada com certa acurácia, mas com um offset gerado pelos intensos ruídos gerados no início do movimento. Esse comportamento se observou em todos os experimentos, o que mostra que este método não é eficaz na estimativa do ângulo de yaw. Uma

forma de corrigir isto, é o processamento dos dados do magnetômetro pelo método discreto de filtro de Kalman.

#### 5.4. Estimativa por filtros de frequência, observadores de estados e filtro de Kalman

Uma forma de contornar o problema da implementação da adaptação dos filtros complementares de Mahony et al, 2006, quanto ao tratamento dos dados do magnetômetro, é a implementação do segundo estágio do filtro de Kalman de estágio duplo, proposto por Sabatelli et al, 2013.

Foi constatado que o filtro de Kalman de estágio duplo só é suportado pelo processador Atmega328 no primeiro estágio, apresentando graves problemas de desempenho, pelo grande volume acumulado de cálculos com matrizes no segundo estágio. Por esta razão, propõe-se aqui uma fusão entre os métodos de Sabatelli et al, 2013, e a adaptação do de Mahony et al, 2006.

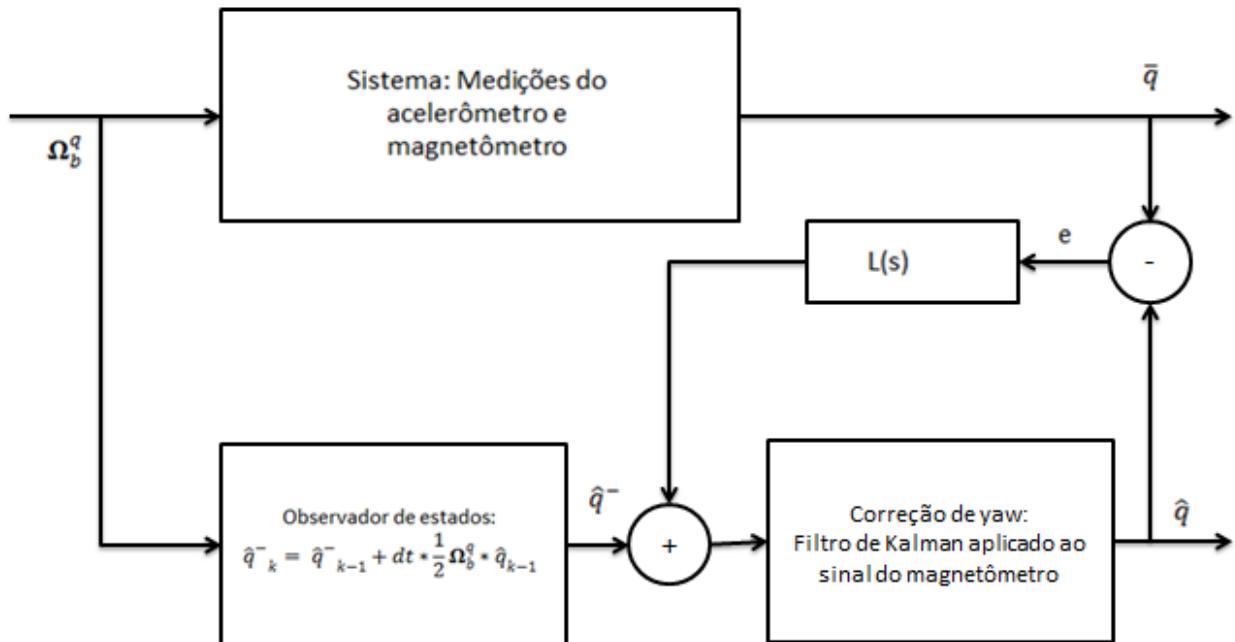


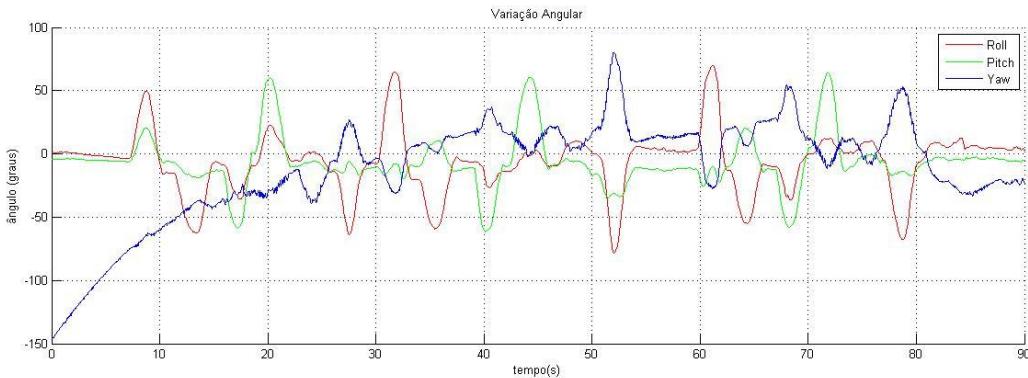
Figure 5-4 - Diagrama de blocos do algoritmo proposto neste trabalho.

O método proposto então estima a atitude pelos sinais do acelerômetro, após passarem pelo filtro passa-baixas da equação 3.19a, utilizando o método de matriz de cossenos diretores.

Aqui vale ressaltar que embora os intervalos de tempo não sejam determinados por rotinas de interrupções, eles se mantém na média de 0.02 segundos, o que permite a utilização de filtros de Butterworth dimensionados para frequência de aquisição de 50Hz.

A atitude estimada é comparada com a atitude anterior para definição do vetor de erros. O controlador do observador se mantém com os mesmos ganhos aplicados nas simulações da seção 4.5.2. O sinal integrado do giroscópio é então corrigido pelos ganhos do observador. Neste ponto, então corrige-se o ângulo de yaw, aplicando o filtro de Kalman discreto sobre os dados do magnetômetro. A figura 5-4 mostra o diagrama de blocos da malha fechada para a estimativa da atitude do corpo rígido.

Embora, a idéia seja um mais complexa, este método tem grande vantagem sobre o filtro de Kalman de estágio duplo de Sabatelli et al, 2013, considerando o custo computacional.



*Figure 5-5 - Resultado da implementação da adaptação de filtros complementares adicionando filtro de Kalman aplicado aos dados do magnetômetro.*

O primeiro experimento mostra a resposta à movimentos em altas frequências. Para este experimento, um tempo maior de aquisição de dados, para que a estabilização do ângulo de yaw pudesse ser completamente observada. Os resultados mostraram a mesma performance em relação ao método anterior para as estimativas de Roll e Pitch. Para o ângulo de Yaw porém houve significativa melhora. Como o experimento foi realizado a mão livre, não é possível a manutenção de todos os ângulos fixos em zero, assim como fazer com que o movimento seja exclusivamente em relação a um único eixo, o que explica a observável variação de Pitch quando Roll é testado, e vice versa. Ainda assim, é possível observar que as variações são identificáveis ao longo do experimento. A estabilização de yaw acontece por volta dos 20 segundos após o início da aquisição dos dados, e com este método, pelo fato do filtro de Kalman ponderar os ganhos para privilegiar a direção melhor estimada entre a feita pelo magnetômetro e a feita pela

integração das velocidades angulares, medidas pelo giroscópio, não se apresentam ruídos como no método anterior.

O segundo experimento mostra a precisão do método. Para tal, é necessária a realização de movimentos controlados e com ângulos definidos e medidos de outra forma além do sensor. Pode-se então utilizar-se um manipulador robótico com os três graus de liberdade de rotação ao redor de XYZ.

O departamento de engenharia mecatrônica da Escola Politécnica da Universidade de São Paulo dispõe do KUKA, uma manipulador robótico com 6 graus de liberdade, composto por 6 articulações.

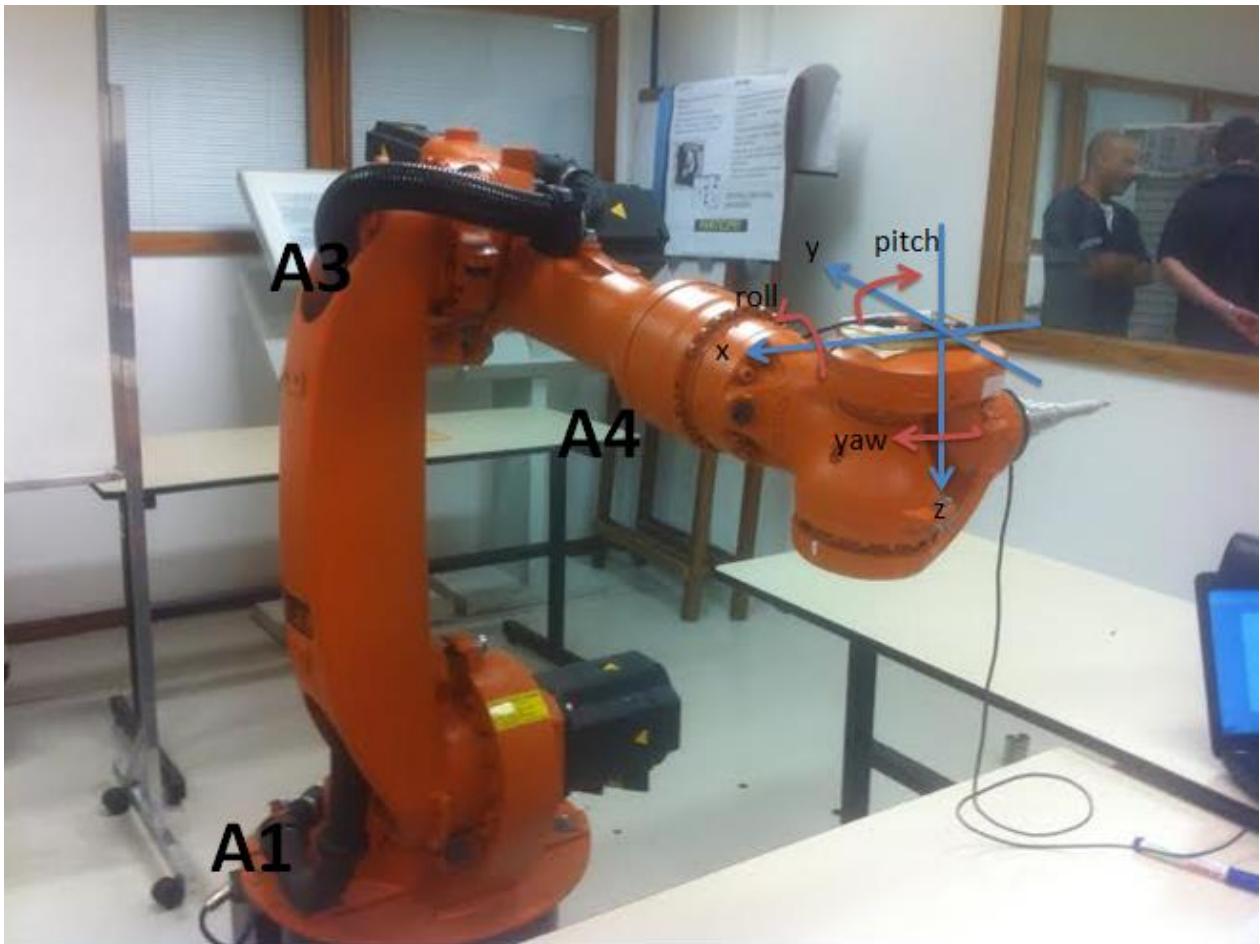


Figure 5-6 - Manipulador KUKA com sensor acoplado. Destacados estão as articulações cujos movimentos correspondem aos ângulos de roll, pitch e yaw.

Como mostrado na figura, 3 das 6 articulações são utilizadas para gerar as rotações ao redor de roll (A3), pitch (A4) e yaw (A1). O KUKA dispõe de um controle manual, onde pode-se

movimentar cada articulação, monitorando através do display do controle os respectivos ângulos de rotação.

Desta forma, pode-se realizar movimentos controlados, e comparar o ângulo mostrado no display com o apontado pelo método. De forma a comprovar a acurácia, os movimentos foram realizados em magnitudes levemente diferentes entre si. O primeiro movimento, foi de  $30.44^\circ$  no sentido positivo de roll. Em seguida, movimentou-se o manipulador de volta para zero e em seguida para  $-30.56^\circ$ . O último movimento em roll foi voltar para  $-4.19^\circ$ , para verificar se o método conseguiria distinguir esta precisão em relação ao zero. A mesma lógica foi realizada para pitch e yaw. Seguem os valores dos ângulos dos movimentos realizados, lidos no display do KUKA.

**Tabela 1 - Primeira movimentação do KUKA.**

	Roll( $^\circ$ )	Pitch( $^\circ$ )	Yaw( $^\circ$ )
Posição Inicial	0	0	0
Máx.	30,04	34,56	32,59
Mín.	-30,56	-31,71	-31,24
Posição Final	-4,19	2,14	1,32

Ao fim destes movimentos, a partir dos ângulos finais, realizou-se movimentos de menor magnitude, para verificar o quanto preciso pode ser a estimativa deste método.

**Tabela 2 - Segunda movimentação do KUKA**

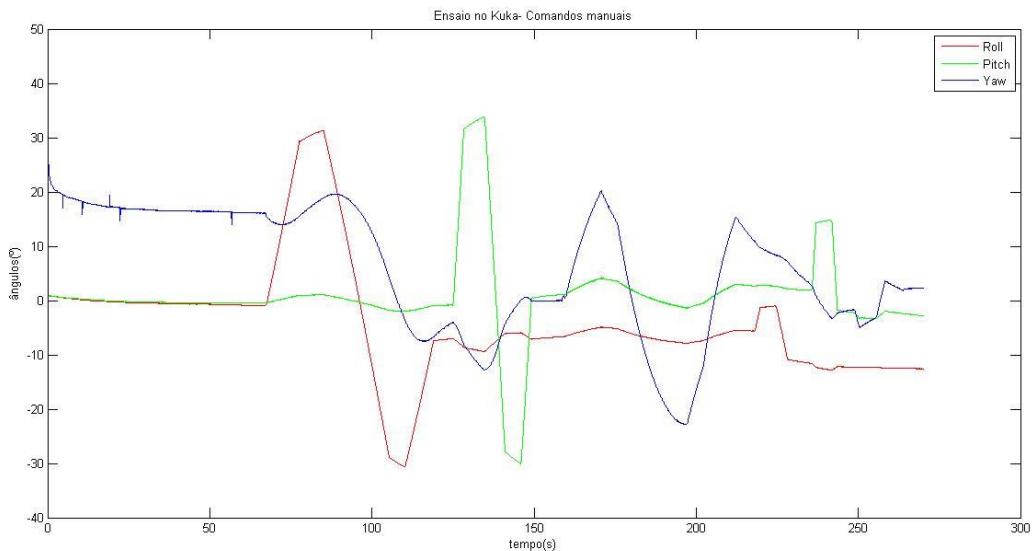
	Roll( $^\circ$ )	Pitch( $^\circ$ )	Yaw( $^\circ$ )
Posição Inicial	-4,19	2,14	1,32
Máx.	0,47	15,64	5,77
Mín.	9,99	-2,47	-2,17
Posição Final	9,99	-2,47	-2,17

Comparando-se as posições do KUKA com as medições realizadas pelo método, pode-se verificar a precisão da estimativa. Tanto pela comparação numérica como pela qualitativa, analisando a figura 5-7, pode-se notar que o método apresenta precisões distintas para Roll/Pitch e Yaw. Isso deve-se ao fato de o magnetômetro ser extremamente sensível à variações do campo magnético ao redor do sensor. Como o sensor está acoplado a uma máquina com motores e rodeado de computadores, em certos momentos o campo magnético da terra fica quase imperceptível ao sensor.

**Tabela 3 - Comparação de ângulos de movimento do KUKA e os ângulos estimados pelo método.**

	Roll( $^{\circ}$ ) - KUKA	Roll( $^{\circ}$ ) - Est.	Pitch( $^{\circ}$ ) - KUKA	Pitch( $^{\circ}$ ) - Est.	Yaw( $^{\circ}$ ) - KUKA	Yaw( $^{\circ}$ ) - Est.
Posição Inicial	0	de 0,2 a -0,9	0	de 0,9 a -0,4	0	de 0,05 a -0,15
Máx.	30,04	de 29 a 30	34,56	de 31,4 a 34	32,59	20
Mín.	-30,56	de -28,7 a -30,5	-31,71	de -28 a -30	-31,24	-23
Posição Final	-4,19	de -7 a -7,4	2,14	de 0,2 a 1,1	1,32	de -1,7 a -2,3
Posição Inicial	-4,19	de -7 a -7,4	2,14	de 0,2 a 1,1	1,32	de -1,7 a -2,3
Máx.	0,47	de -1 a -1,3	15,64	de 14,4 a 14,8	5,77	de -1,8 a -2,3
Mín.	-9,99	de -10,7 a -11,4	-2,47	de -1,8 a -2,2	-2,17	de -5 a -3,4

Vale salientar que o tempo de estabilização do magnetômetro foi respeitado antes de qualquer movimento fosse realizado em yaw. Outro ponto também a se salientar é que o sensor não estava perfeitamente paralelo aos eixos das articulações. Isso acarreta que durante o movimento de um dos ângulos, pequenas variações ao redor dos outros dois serão percebidas.



*Figura 5-7- Movimento de Roll, Pitch e Yaw em função do tempo. Experimento com sensor acoplado ao KUKA.*

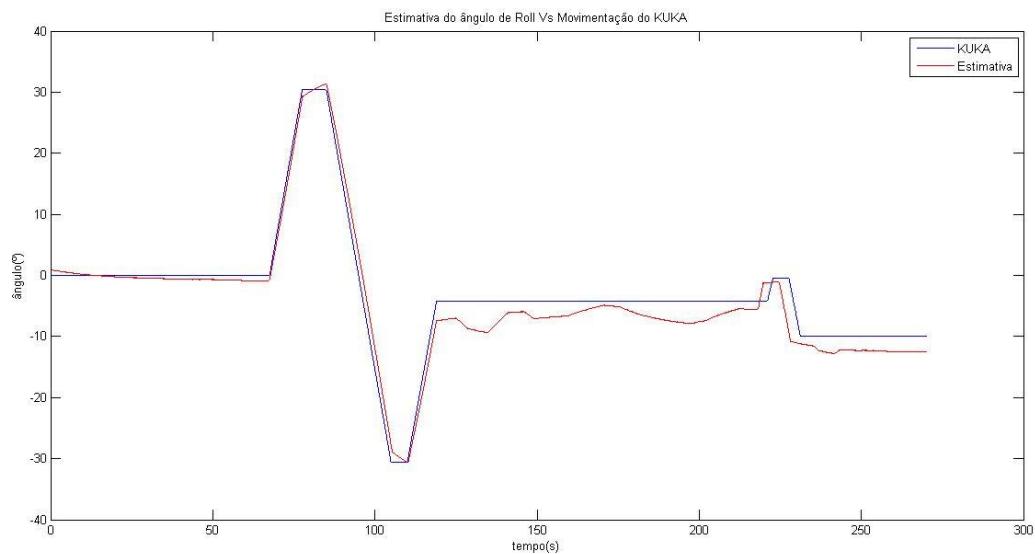


Figure 5-8 - Comparação entre o movimento realizado pelo manipulador KUKA e a estimativa do ângulo de Roll.

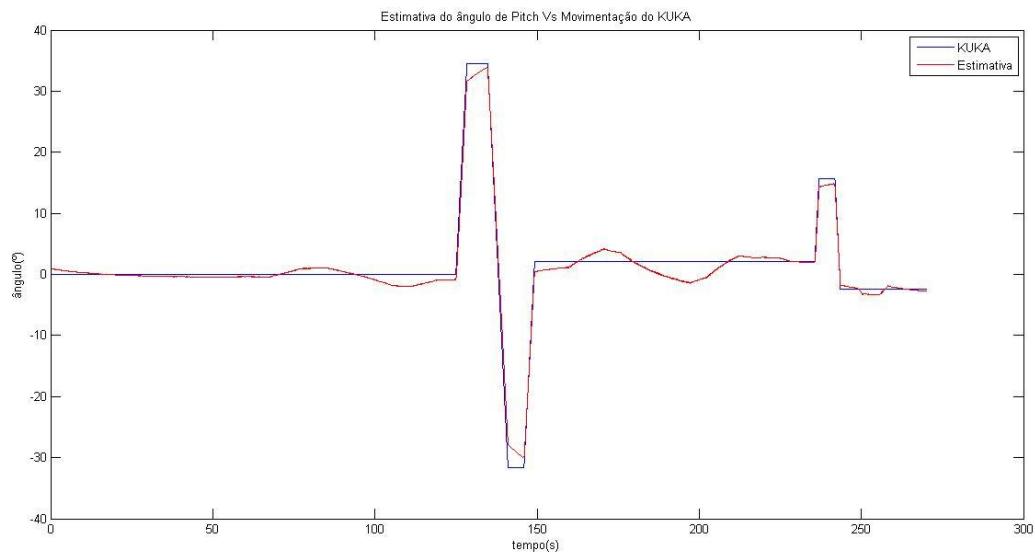


Figure 5-9- Comparação entre o movimento realizado pelo manipulador KUKA e a estimativa do ângulo de Pitch.

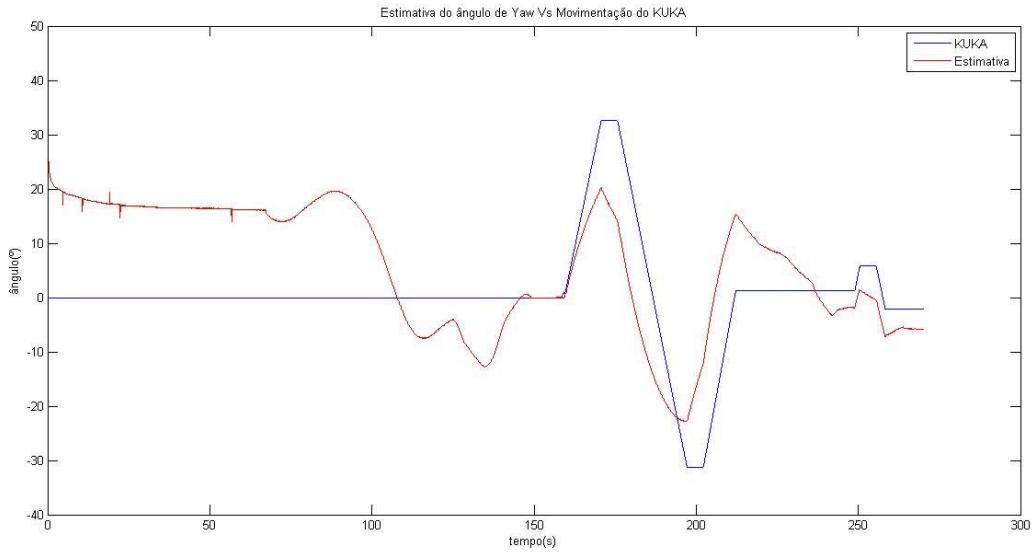


Figure 5-10 - Comparação entre o movimento realizado pelo manipulador KUKA e a estimativa do ângulo de Yaw.

Analizando as tabelas 1, 2 e 3 nota-se que para Roll e Pitch, a maior diferença entre ângulo gerado no KUKA e ângulo estimado pelo método é de aproximadamente  $3^\circ$ . Já para Yaw, esta precisão varia em função dos campos magnéticos presentes ao redor do sensor, mas em condições ideais, como na figura 5-5, após a estabilização da estimativa fornecida pelo filtro de Kalman, também percebe-se a eficácia do método.

As figuras 5.7, 5.8, 5.9 e 5.10 mostram a comparação entre a trajetória realizada pelo manipulador KUKA e a estimativa dos ângulos de Roll, Pitch e Yaw, respectivamente. De forma qualitativa, agora, pode-se novamente comprovar a eficácia do método de estimativa, dentro das condições acima discutidas. Agora de forma visual, pode-se notar a acurácia do método tanto para ângulos da ordem de  $30^\circ$  como para ângulos da ordem de  $5^\circ$ .

## 6. CONCLUSÕES

Os métodos de estimativa de orientação são importantes ferramentas para o fornecimento de dados precisos para os sistemas de controle de veículos aéreos não tripulados. Atualmente, o estado da arte neste assunto nos fornece variadas opções para realizar estas estimativas, mas elas devem ser compreendidas e adaptadas para a aplicação desejada. No caso de veículos aéreos não tripulados, a orientação deve vir de sensores que não dependam da velocidade do ar. Desta forma, os IMU's encontram um forte mercado, onde se apresentam vários modelos destes sensores. Com o desenvolvimento destes métodos matemáticos, a aplicação de IMU's na aeronáutica não tripulada se torna ainda mais ampla.

Os métodos aqui estudados se mostram eficazes tanto separadamente, quanto quando combinados. As simulações mostram que os trabalhos de autores como Mahony et al, 2006 e Sabatelli et al, 2013, são extremamente eficientes dentro das aplicações que os trabalhos estavam inseridos. Utilizando os mesmos dados para todos os métodos estudados, pôde-se identificar as vantagens e desvantagens de cada um em relação aos outros e às leituras cruas dos dados dos sensores.

Mas também é necessário observar aspectos, além da qualidade do sensor e da matemática dos métodos de estimativa. O custo do sistema de sensoriamento de orientação depende dos componentes, e quando se trabalha com processadores e sensores de baixo custo, a capacidade de processamento e de armazenamento de dados destes equipamentos passa a ser uma variável limitante ao processo. Neste momento deve identificar-se qual método encontra o máximo de sua performance dentro das capacidades do processador, e a partir daí, encontrar-se melhorias e adaptações, não só para otimizar o método, mas também para que seja um método que melhor atenda às especificações do projeto.

A alternativa encontrada, neste trabalho, para trabalhar-se com sensor Razor 9DOF e com Arduino foi a fusão de métodos de filtragem Butterworth de sinais, malhas de observadores de estados para estimativa dos ângulos de Roll e Pitch, e utilização do filtro de Kalman para estimativa do ângulo de Yaw. Comparando-se os resultados desta implementação com os resultados das simulações, feitas com dados estáticos, constata-se que o método aqui proposto é igualmente eficiente e capaz de calcular os ângulos de orientação de um corpo rígido.

Além disso a implementação do método em um IMU acoplado a uma manipulador robótico mostrou que a precisão da estimativa tem uma precisão de aproximadamente 3º. Isso comprova que pode-se chegar a uma boa estimativa de orientação com IMU's de baixo custo, através de métodos de filtragem e estimativa de posicionamento devidamente aplicados.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- Baldwin, Grant; Mahony, Robert; Trumpf, Jochen; Hamel, Tarek; Chevron, Thibault.** (2006) *Complementary filter design on the special Euclidean group SE(3)*.
- Baldwin, Grant; Mahony, Robert; Trumpf, Jochen.** (2009) *A nonlinear observer for 6DOF pose estimation from inertial and bearing measurements*. IEEE International conference on robotics and automation.
- Bierman, Gerald J.** (1977). *Factorization Methods for Discrete Sequential Estimation*. Editora DOVER.
- Bresciani, Tommaso** (2008). *Modelling, Identification and Control of a Quadrotor Helicopter*. Department of Automatic Control. Lund University.
- Cabral, Eduardo L.** (2013) *Análise de Robôs*. Livro em elaboração.
- Cabral, Eduardo L.** (2013) *Sistemas dinâmicos: Posição e Orientação*. Notas de Aula. Slides.
- Euston, Mark; Coote, Paul; Mahony, Robert; Kim, Jonghyuk; Hamel, Tarek.** (2007) *A complementary filter for attitude estimation of a fixed-wing UAV*.
- Mahony, Robert; Cha, Sung-Han; Hamel, Tarek.** (2006) *Coupled estimation and control for attitude stabilisation of mini aerial vehicles*.
- Mahony, Robert; Hamel Tarek; Pflimlin, Jean-Michel.** (2008) *Nonlinear Complementary filters on the special Orthogonal group*. IEEE Transactions on automatic control.
- RaneNote.** (2005). *Linkwitz-Riley Crossovers: A Primer*. Rane Corporation.
- Sabatelli, Simone; Galgani, Marco; Fanucci, Luca.** (2012) *A double stage kalman filter for sensor fusion and orientation tracking in 9D IMU*. Università di Pisa.
- Sabatelli, Simone; Galgani, Marco; Fanucci, Luca; Rocchi, Alessandro.** (2013) *A double-stage kalman filter for orientation tracking with an integrated processor in 9-D IMU*. Università di Pisa.
- Vasconcelos, J.R.; Silvestre, C; Oliveira, P..** (2008) *A nonlinear GPS/IMU based observer for rigid body attitude and position estimation*. IEEE Conference on decision and control.

**Welch, Greg; Bishop, Gary. (2006) *An Introduction to the Kalman Filter.***  
*http://rouel-projects.blogspot.com.br/p/calibration-of-iphone-accelerometers.html.*  
**Acesso 15/06/2013.**  
*http://www.memsjournal.com/ . Acesso em 25/06/2013*

## APENDICE A

Código do método híbrido de filtro complementar e filtro de Kalman embarcado no processador.  
(Código comprehende apenas a função chamada para tratamento de dados. Variáveis globais declaradas no corpo principal do código.)

```
1. void Compl_Kalman(void)
2. { // entradas devem ser os angulos de euler anteriores, e matriz P
3. float Accel_magnitude;
4. float R1[3][3] = {{2,0,0},{0,2,0},{0,0,2}};
5. float R2[3][3] = {{1,0,0},{0,1,0},{0,0,1}};

6. angle2quat(roll,pitch,yaw,q);

7. //*****Roll and Pitch*****
8. //
9. wx = GYRO_SCALED_RAD(gyro[0])-OffSetGyroX; //gyro x roll -OffSetGyroX
10. wy = GYRO_SCALED_RAD(gyro[1])-OffSetGyroY; //gyro y pitch -OffSetGyroY
11. wz = GYRO_SCALED_RAD(gyro[2])-OffSetGyroZ; //gyro z yaw -OffSetGyroZ

12. //Filtro passa alta no giroscópio
13. wx = 0.9695*(wx-wx_prev)+0.9391*wx_prev;
14. wy = 0.9695*(wy-wy_prev)+0.9391*wy_prev;
15. wz = 0.9695*(wz-wz_prev)+0.9391*wz_prev;

16. ax = accel[0]; //accel x
17. ay = accel[1]; //accel y
18. az = accel[2]; //accel z

19. //Filtro passa baixa no acelerômetro
20. ax = 0.0305*(ax+ax_prev)+0.9391*ax_prev;
21. ay = 0.0305*(ay+ay_prev)+0.9391*ay_prev;
22. az = 0.0305*(az+az_prev)+0.9391*az_prev;

23. Accel_magnitude = sqrt(ax*ax + ay*ay + az*az);

24. ax = ax/Accel_magnitude;
25. ay = ay/Accel_magnitude;
26. az = az/Accel_magnitude;
```

27. //Obtendo medidas de Roll Pitch e Yaw a apartir das medidas do acelerômetro e magnetômetro

28. PitchBar = asin(-ax);  
29. SinPsi = ay/cos(PitchBar);  
30. RollBar = asin(SinPsi);  
31.YawBar = MAG\_Heading;

32. float Kp = 0.04;  
33. float Ki = 0.002;

34. ex = RollBar - roll;  
35. ey = PitchBar - pitch;  
36. ez = YawBar - yaw;

37. deltax = (Kp+Ki)\*ex - Kp\*ex\_prev + deltax\_prev;  
38. deltay = (Kp+Ki)\*ey - Kp\*ey\_prev + deltay\_prev;  
39. deltaz = (Kp+Ki)\*ez - Kp\*ez\_prev + deltaz\_prev;

40. // Integração da leitura do **Giroscópio**  
41. int l1[4][4];  
42. Eyes4(l1);

43. // Integração utilizando ângulos de euler.  
44. ang[0][0] = roll;  
45. ang[1][0] = pitch;  
46. ang[2][0] = yaw;

47. angles\_pt[0][0] = wx;  
48. angles\_pt[1][0] = wy;  
49. angles\_pt[2][0] = wz;

50. // Integração por método dos Trapézios  
51. Trapez[0] = (angles\_pt[0][0]+angles\_pt\_prev[0][0])\*G\_Dt/2;  
52. Trapez[1] = (angles\_pt[1][0]+angles\_pt\_prev[1][0])\*G\_Dt/2;  
53. Trapez[2] = (angles\_pt[2][0]+angles\_pt\_prev[2][0])\*G\_Dt/2;  
  
54. angles[0] = angles\_prev[0][0] + Trapez[0];  
55. angles[1] = angles\_prev[1][0] + Trapez[1];  
56. angles[2] = angles\_prev[2][0] + Trapez[2];

```

57. roll = angles[0] + deltax;
58. pitch = angles[1] + deltay;
59. yaw = angles[2] + deltaz;

60. //*****YAW*****
61. //float q1[4][1];
62. angle2quat(roll,pitch,yaw,q1);

63. //Estimando a matriz de covariâncias a priori
64. float P_prior[4][4];
65. P_prior[0][0] = P[0][0];
66. P_prior[0][1] = P[0][1];
67. P_prior[0][2] = P[0][2];
68. P_prior[0][3] = P[0][3];
69. P_prior[1][0] = P[1][0];
70. P_prior[1][1] = P[1][1];
71. P_prior[1][2] = P[1][2];
72. P_prior[1][3] = P[1][3];
73. P_prior[2][0] = P[2][0];
74. P_prior[2][1] = P[2][1];
75. P_prior[2][2] = P[2][2];
76. P_prior[2][3] = P[2][3];
77. P_prior[3][0] = P[3][0];
78. P_prior[3][1] = P[3][1];
79. P_prior[3][2] = P[3][2];
80. P_prior[3][3] = P[3][3];

81. mx = cos(MAG_Heading);
82. my = sin(MAG_Heading);
83. mz = 0;

84. //Matriz Jacobiana dos ruídos da equação não linear h2
85. H2[0][0]= 2*q[3][0];
86. H2[0][1]= 2*q[2][0];
87. H2[0][2]= 2*q[1][0];
88. H2[0][3]= 2*q[0][0];
89. H2[1][0]= 2*q[0][0];
90. H2[1][1]= -2*q[1][0];
91. H2[1][2]= -2*q[2][0];
92. H2[1][3]= -2*q[3][0];
93. H2[2][0]= -2*q[1][0];
94. H2[2][1]= -2*q[0][0];
95. H2[2][2]= 2*q[3][0];

```

```

96. H2[2][3]= 2*q[2][0];

97. // Ganho de Kalman para Yaw
98. float Faux[4][3];
99. float Gaux[3][3];
100.    float Maux[3][3];
101.    float H2t[4][3];
102.    TransposeH(H2,H2t);
103.    MultiplyPHt(P_prior,H2t,Faux);
104.    MultiplyHPHt(H2,Faux,Gaux);
105.    Add3(Gaux,R2,Maux);
106.    Invert3(Maux);
107.    MultiplyPHtFilter(Faux,Maux,K2);

108.    // Estimativa da direção do campo magnético
109.    h2[0][0] = 2*(q[1][0]*q[2][0] + q[0][0]*q[3][0]);
110.    h2[1][0] = q[0][0]*q[0][0] - q[1][0]*q[1][0] + q[2][0]*q[2][0] - q[3][0]*q[3][0];
111.    h2[2][0] = 2*(q[2][0]*q[3][0] - q[0][0]*q[1][0]);

112.    // Fator de correção do Yaw
113.    zk2[0][0] = mx;
114.    zk2[1][0] = my;
115.    zk2[2][0] = mz;
116.    Subtract3(zk2,h2,dif2);
117.    MultiplyKdif(K2,dif2,qE2_aux);
118.    qE2[0][0] = qE2_aux[0][0];
119.    qE2[1][0] = 0;
120.    qE2[2][0] = 0;
121.    qE2[3][0] = qE2_aux[3][0];

122.    //cálculo estimado a posteriori
123.    float q_atual[4][1];
124.    Addq(q1,qE2,q_atual);
125.    float Modulo;
126.    float Fator;
127.    Modulo =
        sqrt(q_atual[0][0]*q_atual[0][0]+q_atual[1][0]*q_atual[1][0]+q_atual[2][0]*q_atual[2][0]+q_atu
        al[3][0]*q_atual[3][0]);
128.    Fator = 1/Modulo;
129.    Scaleq(q_atual,Fator); //Normalizando quaternion final

130.    //transformando em ângulos de Euler

```

```
131.     quat2angle(q_atual,euler);
132.     roll = roll;
133.     pitch = pitch;
134.     yaw = euler[2];

135.     //Primeira atualização da matriz P
136.     float J1[4][4];
137.     float L1[4][4];
138.     MultiplyP(K2,H2,L1);
139.     Subtract4(I1,L1,J1);
140.     Multiply4(J1,P_prior,P);

141. }
```