

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DANIEL DOS SANTOS CORRÊA

**Realidade Virtual e Sensores Inerciais no Desenvolvimento da
Tecnologia Assistiva: Um sistema para estudo da marcha humana
baseado em fusão de sensores inerciais**

Porto Alegre

2015

DANIEL DOS SANTOS CORRÊA

**Realidade Virtual e Sensores Inerciais no Desenvolvimento da
Tecnologia Assistiva: Um sistema para estudo da marcha humana
baseado em fusão de sensores inerciais**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação: Instrumentação.

ORIENTADOR: Dr. Prof. Alexandre Balbinot

Porto Alegre

2015

Daniel dos Santos Corrêa

**Realidade Virtual e Sensores Inerciais no Desenvolvimento da
Tecnologia Assistiva: Um sistema para estudo da marcha humana
baseado em fusão de sensores inerciais**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Alexandre Balbinot, UFRGS

Doutor pela UFRGS – Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Milton Antonio Zaro, UFRGS

Doutor pela UFRGS – Porto Alegre, Brasil

Prof. Dr. Eduardo Perondi, UFRGS

Doutor pela UFSC– Blumenau, Brasil

Prof. Dr. Carlos Eduardo Pereira, UFRGS

Doutor pela Universität Stuttgart – Stuttgart, Alemanha

Coordenador do PPGEE: _____

Prof. Dr. Alexandre Sanfelice Bazanella

Porto Alegre, Abril de 2015.

AGRADECIMENTOS

A minha família pelo carinho dedicado à minha formação.

A minha esposa Giordana Vargas pelo total apoio, amor e compreensão durante mais essa etapa da minha vida - agradeço por ser minha base.

Agradeço aos meus amigos que sempre estavam prontos para me ajudar ou dar algumas palavras de apoio, mostrando o significado de uma verdadeira amizade.

Aos meus colegas que de várias formas colaboraram com a minha formação, em especial aos meus colegas do Laboratório de Instrumentação Eletro-Eletrônica (IEE).

Ao meu orientador por estar me guiando em minha vida acadêmica, por sua dedicação e por sua paciência. Agradeço por ser mais que um professor, um grande amigo.

RESUMO

A marcha humana, ou caminhada, é um padrão cíclico de movimentos corporais que se repetem a cada passo que desloca um indivíduo de um local a outro. Atualmente, avaliações biomecânicas da marcha humana tem sido utilizado no diagnóstico de alterações neuromusculares, músculo-esqueléticas e como forma de avaliação pré e pós-tratamento cirúrgico, medicamentoso e/ou fisioterapêutico. O presente trabalho apresenta o desenvolvimento de uma ferramenta acadêmica de baixo custo para o estudo da marcha humana. Esse sistema consiste no sensoriamento da marcha de um usuário através de sensores inerciais e de um modelo virtual do corpo humano para permitir a visualização do movimento gerado. Dessa maneira o usuário poderá ter suas ações corrigidas por sua percepção visual e também corrigida pelas orientações de um fisiatra ou fisioterapeuta que terá a reprodução do modelo virtual conforme a movimentação detalhada do paciente para análise. O sistema ainda efetuará os registros das variáveis cinemáticas da marcha (tais como aceleração, velocidade angular, angulações dos membros sensorizados) para estudos e acompanhamento mais detalhado da sua recuperação e/ou tratamento. Como resultado, o sistema desenvolvido obteve erros médios de X $0,52^\circ$ Y $1,20^\circ$ Z $1,80^\circ$ e erros em RMS de X $3,01^\circ$ Y $3,30^\circ$ Z $5,70^\circ$ quando comparados com um sistema comercial, sendo esse resultado próximo à literatura e aplicável em exames biomecânicos de marcha.

Palavras-chave: sensores inerciais, exames biomecânicos, marcha humana, modelo virtual.

ABSTRACT

The human gait is a cyclical pattern of body movements that are repeated every step that moves a subject from one location to another. Currently, biomechanical assessments of human gait has been used for diagnosing neuromuscular disorders, musculoskeletal and as a way of pre and post-surgical treatment, medication and/or physical therapy. This paper presents the development of a low cost academic tool for the study of human gait. This system consists of sensing the motion of a user through inertial sensors and a virtual model of the human body to allow the visualization of the generated movement. In this way, the user can have its actions corrected by his visual perception and also corrected by therapist or physiotherapist who will visualize the virtual model as the detailed movements of patient. The system will also record the kinematic gait variables (as acceleration, angular velocity, angles of the sensed members) for studies and more detailed monitoring of their recovery and/or treatment. As result, the developed system obtained average errors of X 0,52° Y 1,20° Z 1,80° and errors in RMS X 3,01° Y 3,30° Z 5,70° compared to a commercial system, and these results close to the ones seen in literature and applicable in biomechanical tests of gait

Keywords: inertial sensors, biomechanical testing, human gait, virtual model.

SUMÁRIO

1	INTRODUÇÃO	11
2	REVISÃO BIBLIOGRÁFICA	14
2.1	MARCHA HUMANA	14
2.2	ALGUNS TRABALHOS QUE UTILIZAM SENSORES INERCIAIS PARA CARACTERIZAÇÃO DE MOVIMENTOS HUMANOS	18
2.3	NOTAÇÕES BÁSICAS DE MATRIZES E VETORES TÍPICAMENTE USADAS EM SISTEMAS DE POSICIONAMENTO	23
2.3.1	Sistemas INS e AHRS	24
2.3.2	Sistemas de Referência	25
2.3.3	Attitude	27
2.3.4	Um breve relato sobre sensores inerciais	34
3	ESTUDO E DESENVOLVIMENTO DE ALGORITMOS DE FUSÃO	36
3.1	INTRODUÇÃO	36
3.2	DCM IMU	36
3.3	BÚSSOLA ELETRÔNICA <i>E</i>COMPASS	42
3.4	FILTRO DE KALMAN	46
3.4.1	Descrição do Filtro de Kalman	46
3.4.2	Desenvolvimento do algoritmo do filtro de Kalman	50
4	DESENVOLVIMENTO DO SISTEMA EXPERIMENTAL	53
4.1	APARATO EXPERIMENTAL	53
4.2	REDE DE SENSORES	57
4.2.1	Hardware	58
4.2.2	Calibração dos sensores	63
4.2.3	Programação do dispositivo	67
4.3	MODELO VIRTUAL	69
4.4	PROGRAMA PARA ENSAIO DA MARCHA	74
5	RESULTADOS E DISCUSSÕES	80
5.1	ENSAIOS PRELIMINARES DE FUSÃO DOS SENSORES	80
5.2	ENSAIO ESTÁTICO	89
5.3.1	Testes de perda de pacotes	91
5.3.2	Testes de autonomia do sistema	93
5.4	ENSAIO DINÂMICO	94
5.5	RESULTADOS DE ENSAIO DA MARCHA	102
6	CONCLUSÃO	107
6.1	PROPOSTA DE TRABALHOS FUTUROS	108
	REFERÊNCIAS	110

LISTA DE ILUSTRAÇÕES

Figura 2.1 Ciclo de marcha humana.....	16
Figura 2.2 Planos de delimitação e secção do corpo humano.....	19
Figura 2.3 Sistema de estimativa de angulação baseada em acelerômetros.....	20
Figura 2.4 Dois sensores reais de cada membro geram um sensor virtual cuja orientações são comparadas para gerar a rotação no algoritmo <i>virtual-sensor-difference-based</i>	21
Figura 2.5 Arquitetura do sistema de unidade inercial com <i>Bluetooth</i>	22
Figura 2.6 Sistema videogrametria e modelo virtual.....	23
Figura 2.7 Sistemas de coordenadas.....	27
Figura 2.8 Rotação matricial.....	28
Figura 2.9 Ângulos ZYX de Euler.....	30
Figura 3.1 Diagrama de bloco do algoritmo DCM IMU.....	37
Figura 3.2 Esquema do filtro de Kalman.....	50
Figura 3.3 Desenvolvimento do Filtro de Kalman.....	52
Figura 4.1 Módulos do sistema.....	54
Figura 4.2 Transceptor Xbee.....	55
Figura 4.3 ArduIMU V3.....	55
Figura 4.4 Fluxo do sistema.....	57
Figura 4.5 XBEE conectado a placa CON-USBBEE.....	61
Figura 4.6 ArduIMU e Xbee acoplados (esquerda) e encapsulamento final (direita).....	62
Figura 4.7 Módulos no corpo do usuário. Esquerda – parte inferior. Direita – parte superior sem baterias.....	63
Figura 4.8 Tela de calibração do magnetômetro – MagMaster (MATSELENAK, 2014).....	66
Figura 4.9 Fluxograma do algoritmo implementado na IMU.....	68
Figura 4.10 Interface <i>MakeHuman</i>	70
Figura 4.11 Interface <i>MakeHuman</i> de exportação de formato.....	71
Figura 4.12 Interface <i>Fuse</i> da <i>Mixamo</i>	71
Figura 4.13 Interface <i>Fuse</i> da <i>Mixamo</i> – detalhe modelagem.....	72
Figura 4.14 Interface <i>Blender Game Engine</i>	73
Figura 4.15 Interface <i>Blender</i> vista pelo usuário.....	74
Figura 4.16 Diagrama de uma <i>thread</i> genérica do programa.....	76
Figura 4.17 Comunicação entre <i>thread</i> do programa.....	76
Figura 4.18 Diagrama do sistema de marcha.....	77
Figura 4.19 Pacote API.....	78
Figura 5.1 Ferramenta gráfica desenvolvida para apresentar o comportamento da métrica <i>attitude</i>	81
Figura 5.2 Gráficos representando a métrica <i>attitude</i> em Euler XYZ com os três métodos: DCM, <i>eCOMPASS</i> e Kalman.....	82
Figura 5.3 Goniômetro.....	83
Figura 5.4 Módulo do sistema fixado em um goniômetro.....	83
Figura 5.5 Gráficos representando a métrica <i>attitude</i> em Euler XYZ com os três métodos utilizados (DCM, <i>eCOMPASS</i> e Kalman) contendo divergência entre eles.....	85
Figura 5.6 Algoritmo para identificação e recuperação da <i>attitude</i> corrente.....	87

Figura 5.7 Ensaio dinâmico - comparação entre o sistema desenvolvido (DCM) e um sistema comercial – <i>Row</i>	96
Figura 5.8 Ensaio dinâmico - comparação entre o sistema desenvolvido (DCM) e um sistema comercial – <i>Pitch</i>	96
Figura 5.9 Ensaio dinâmico - comparação entre o sistema desenvolvido (DCM) e um sistema comercial – <i>Yaw</i>	97
Figura 5.10 Posicionamento dos sensores para ensaio da marcha.	102
Figura 5.11 Modelo virtual acompanhando a marcha.	103
Figura 5.12 Angulação da flexão do joelho direito obtido pelo sistema.	103
Figura 5.13 Angulação da flexão do joelho através do sistema Dvideow.....	104
Figura 5.14 Angulações da flexão do joelho obtido pelo sistema.	105
Figura 5.15 Angulação de um sensor no topo da cabeça do usuário.....	105
Figura 5.16 Deslocamento do centro de massa.	106

LISTA DE TABELAS

Tabela 4.1 Especificações Xbee IEEE 802.15.4 (<i>Digi International</i> 2006).....	60
Tabela 5.1 Resultado para ensaio estático com goniômetro e sistema desenvolvido (eixo x).	90
Tabela 5.2 Resultado para ensaio estático com goniômetro e sistema desenvolvido (eixo y).	90
Tabela 5.3 Resultado para ensaio estático com goniômetro e sistema desenvolvido (eixo z).	90
Tabela 5.4 Resultado dos ensaios de perda de pacotes.....	92
Tabela 5.5 Resultado para ensaio de autonomia do sistema.....	94
Tabela 5.6 Dados da orientação e posição do sensor MTi-G-700 XSens (XSens, 2014).	95
Tabela 5.7 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (DCM - <i>Row</i>).....	97
Tabela 5.8 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (DCM - <i>Pitch</i>).	97
Tabela 5.9 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (DCM - <i>Yaw</i>).....	98
Tabela 5.10 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (Kalman - <i>Row</i>).....	98
Tabela 5.11 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (Kalman - <i>Pitch</i>).	98
Tabela 5.12 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (Kalman - <i>Yaw</i>).....	98
Tabela 5.13 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (<i>eCompass</i> - <i>Row</i>).....	99
Tabela 5.14 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (<i>eCompass</i> - <i>Pitch</i>).	99
Tabela 5.15 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (<i>eCompass</i> - <i>Yaw</i>).....	99
Tabela 5.16 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido - resumo.....	99
Tabela 5.17 Resultado para ensaio dinâmico com algoritmo Kalman tendo como base o algoritmo DCM - <i>Yaw</i>	100
Tabela 5.18 Resultado para ensaio dinâmico com algoritmo eCompass tendo como base o algoritmo DCM - <i>Yaw</i>	101

LISTA DE ABREVIATURAS

ADC: *Analog-to-Digital*

AHRS: *Attitude and Heading Reference System*

CC: *Creative Commons*

DCM: *Direction Cosine Matrix*

ECEF: *Earth-Centered, Earth-Fixed*

GPS: *Global Positioning System*

IMU: *Inertial Measurement Unit*

INS: *Inertial Navigation System*

LSB: *Least Significant Bit*

NED: *North-East-Down*

1 INTRODUÇÃO

A marcha humana, ou caminhada, é uma conduta motora caracterizada por eventos cíclicos que transcorrem durante o percurso de um indivíduo de um lugar a outro. Esses eventos são descritos como o deslocamento rítmico das partes do corpo que mantém o indivíduo em progresso para diante (PERRY, 2005).

Atualmente, avaliações biomecânicas da marcha humana têm sido utilizadas no diagnóstico de alterações neuromusculares, músculo-esqueléticas e como forma de avaliação pré e pós-tratamento cirúrgico, medicamentoso e/ou fisioterapêutico. Esse tipo de exame consiste na captura do movimento (através de sensores e/ou câmeras) realizado durante o ato da marcha humana, onde é registrado o movimento do paciente através de dispositivos e a partir dos dados registrados, calcula-se as variáveis cinemáticas do movimento. Contudo, o elevado custo do equipamento e programas computacionais necessários para esse fim têm evitado sua ampla utilização (CORREA; 2010).

Segundo as pesquisas de Bohannon *et al.* (1997), a capacidade de caminhar é a pressuposição do funcionamento independente de um indivíduo. Uma vez privada ou parcialmente privada a caminhada, grande parte da sua independência está comprometida. Então o esforço para retomada desse ato torna-se justificável, uma vez que estar-se-á devolvendo a liberdade de um homem.

Baseado nas justificativas apresentadas anteriormente, o objetivo principal deste estudo é o desenvolvimento de uma ferramenta acadêmica de baixo custo para o estudo da marcha humana. Esse sistema consiste no sensoriamento da marcha de um paciente através de sensores inerciais e de um modelo virtual do corpo humano para permitir a visualização do movimento gerado. Dessa maneira, o usuário poderá ter suas ações corrigidas por sua percepção visual e também pelas orientações de um fisiatra ou fisioterapeuta que terá a

reprodução do modelo virtual conforme a movimentação detalhada do paciente para análise. O sistema ainda efetuará os registros das variáveis cinemáticas da marcha (tais como aceleração, velocidade angular, angulações dos membros sensorizados e outras variáveis derivadas desses) para estudos e acompanhamento mais detalhado da sua recuperação e/ou tratamento.

Os objetivos secundários deste estudo são:

- Desenvolvimento de uma rede de sensores inerciais sem fio;
- Determinar a angulação do membro do usuário em relação ao meio através da fusão das informações dos sensores inerciais;
- Avaliação de algoritmos de fusão de sensores;
- Desenvolvimento de uma de interface entre o sistema e o modelo virtual;
- Otimização do sistema para reduzir perda de pacotes;
- Comparação do sistema desenvolvido com um sistema comercial;

Para que esse desenvolvimento fosse possível, foi criada uma rede de sensores sem fio, onde cada módulo é posicionado em um segmento do corpo do paciente. Os módulos são compostos por um transceptor, um microcontrolador, sensores (acelerômetro, giroscópio e magnetômetro) e uma bateria. O transceptor utilizado é o modelo Xbee da *Digi International* que utiliza o protocolo IEEE 802.15.4 para comunicação. O microcontrolador e sensores são integrados em uma única placa denominada ArduIMU V3 desenvolvida pela comunidade mundial sobre licença CC (*Creative Commons*). O ArduIMU V3 foi utilizado para calcular a angulação do membro do paciente em relação ao corpo e ao meio – métrica denominada de *attitude*. A *attitude* corresponde à orientação relativa ente dois sistemas de referência e será apresentada com maiores detalhes na Seção 2.3.3.

Além disso, o sistema desenvolvido tem como complemento um programa para receber as informações de todos os sensores, gravar as variáveis dos mesmos (tais como aceleração,

velocidade angular, *attitude* dos membros sensoriados e tempo da leitura) em arquivos para estudos mais detalhados da cinemática do usuário e reproduzir as angulações lidas pelos sensores no paciente em um modelo virtual. O programa foi desenvolvido na linguagem de programação *Python* e o modelo virtual foi desenvolvido sobre o *Blender Game Engine*, um software de animação gráfica e jogos virtuais, onde as informações do comportamento do paciente são recebidas através de uma interface desenvolvida entre os programas e então reproduzida no modelo.

Para melhor apresentação deste trabalho, o mesmo está dividido em quatro etapas principais:

1. Revisão Bibliográfica: onde é apresentado o estudo que foi realizado durante o desenvolvimento do projeto;
2. Estudo e desenvolvimento de algoritmos de fusão: onde são demonstrados e trabalhados os algoritmos utilizados para o desenvolvimento do projeto;
3. Desenvolvimento do Sistema Experimental: onde são definidos os equipamentos, tecnologias e procedimentos para criação do sistema experimental;
4. Resultados e Discussões: onde será apresentado os resultados e discussões relacionadas ao projeto;
5. Conclusões: onde é realizado o fechamento e apresentação das conclusões do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são abordados os principais temas que contextualizam o sistema desenvolvido. Inicialmente é apresentada uma breve descrição de conceitos relacionados a marcha humana, focos da utilização do sistema desenvolvido no presente trabalho. Logo após, serão apresentados trabalhos similares nessa linha de desenvolvimento, bem como seus resultados para futuras comparações.

Para um melhor entendimento dos algoritmos de fusão utilizados e para trabalhar com a orientação dos membros lidos em relação ao sensores, foram estudados sistema de navegação e orientações. Assim, neste capítulo também serão apresentados os principais sistemas de navegação e serão feitas descrições matemáticas das rotações e orientações acerca das unidades inerciais. Para isso, são definidos os sistemas de referência comumente utilizados em sistemas de navegações, bem como, a conversão das grandezas entre os diferentes sistemas. Por fim, são introduzidos os sensores utilizados em estimação de *attitude* utilizados nesse sistema.

2.1 MARCHA HUMANA

A marcha humana, ou caminhada, é um padrão cíclico de movimentos corporais que se repetem indefinidamente a cada passo que desloca um indivíduo de um local a outro. Esses movimentos corporais mantém o indivíduo em progresso para diante (PERRY, 2005).

Duas fases distintas caracterizam os eventos cíclicos que compõem a marcha aos quais compreendem o apoio e o balanço. O primeiro período compreende o apoio, no qual o pé está em contato com a superfície e constitui aproximadamente 60% do ciclo da marcha. Os restantes 40% do ciclo constituem o período de balanço, que é iniciado quando os dedos se desprendem da superfície. Assim, cada passada inclui dois períodos de duplo apoio (ROSE e GAMBLE, 1998; PERRY, 1992; VAUGHAN *et al.*, 1992).

Um estudo mais detalhado da marcha propõe uma divisão de períodos, sendo a fase de apoio dividida em: contato inicial, resposta de carga, médio apoio, apoio terminal e pré-balanço. Por sua vez, a fase de balanço inclui o balanço inicial, o médio balanço e o balanço terminal (PERRY, 1992) (VAUGHAN *et al.*, 1992).

De acordo com os autores Perry (1992) e Vaughan *et al.* (1992), a fase de apoio pode ser dividida nas seguintes etapas:

- contato inicial (0 a 2% do ciclo): período que inicia o ciclo, determinado pelo instante em que o pé toca o solo (em um marcha normal, o calcanhar tem o primeiro contato). Esse período se encerra quando todo o pé apóia-se sobre a superfície;
- resposta à carga (0 a 10% do ciclo): período em que o centro de gravidade está na posição mais baixa. Esta fase representa o início do período de apoio e começa com o contato inicial no solo e continua até o outro pé começar o balanço. Este período corresponde ao primeiro duplo apoio, pois os dois pés estão em contato com o solo;
- médio apoio (10 a 30% do ciclo): começa com a retirada dos dedos da perna contralateral e termina quando o centro de gravidade alinha-se acima do pé, atingindo sua posição mais alta. O peso corporal está completamente na perna de apoio, sendo que o pé permanece parado enquanto o corpo se transfere para frente;
- apoio terminal (30% a 50% do ciclo): inicia quando o centro de gravidade está sobre o pé de suporte e termina quando o outro pé toca o solo;
- pré-balanço (50% a 60% do ciclo): inicia com o contato do membro contralateral e termina com a retirada dos dedos ipsilaterais. Representa o final do apoio e corresponde ao segundo duplo apoio.

A fase de balanço ainda é subdividida em:

- balanço inicial (60% a 73% do ciclo): começa com a saída do pé do solo e termina quando o membro em balanço está oposto ao apoio do pé contralateral. Representa o período de aceleração;
- balanço intermediário (73% a 87% do ciclo): é a segunda fase do balanço, que termina quando o membro está à frente do outro membro e a tíbia assume orientação vertical. Esta fase está sob a ação das forças inerciais que movimentam o membro inferior para frente;
- balanço terminal (87 a 100%): fase final do ciclo que termina quando o pé toca o solo. Representa o período de desaceleração.

É importante ressaltar que a caminhada não está associada apenas ao movimento das pernas, mas à resposta e o equilíbrio de todo o corpo. Por exemplo, o impacto do passo é sentido pela coluna e o fator de equilíbrio da passada também é dado pelo movimento dos braços. A Figura 2.1 apresenta o ciclo do marcha humana em desenhos.

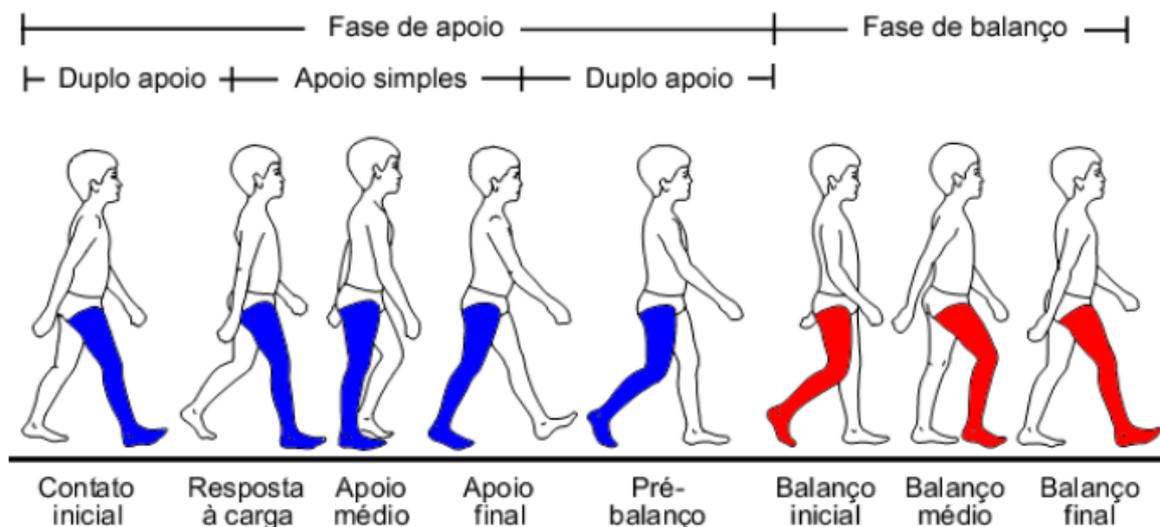


Figura 2.1 Ciclo de marcha humana.

Fonte: VAUGHAN; DAVIS; O'CONNOR, 1992 (livre adaptação)

Avaliações biomecânicas da marcha humana têm sido utilizadas no diagnóstico de alterações neuromusculares, músculo-esqueléticas e como forma de avaliação pré e pós-tratamento cirúrgico, medicamentoso e/ou fisioterapêutico. Esse tipo de avaliação consiste na captura do movimento realizado durante o ato da marcha humana onde é registrado o movimento do paciente através de um ou mais dispositivos (como por exemplo, câmera de vídeo, sensores eletromagnéticos e acelerômetros) e a partir dos dados registrados, calcula-se as variáveis do movimento, tais como, força, potência muscular, torque, entre outras (CORREA, 2010).

Outra vertente das avaliações biomecânicas tem sido a realimentação visual, onde através da imagem do movimento, o paciente acompanha um movimento padrão e tenta reproduzi-lo; ou ainda acompanha o próprio movimento para corrigir-se. Dessa maneira, tem-se o exercício do paciente na busca da movimentação correta e a geração de resultados gravados por sistemas para o acompanhamento da recuperação, além do estudo detalhado do movimento do paciente (BARTON *et al*, 2014; SHEMA *et al*, 2013). Em 2013, houve uma importante conferência de reabilitação visual realizada pela IEEE na Pensilvânia, onde trabalhos sobre realimentação visual foram apresentados, sendo muitos com o foco na área de instrumentação biomecânica (ICVR IEEE 2014).

Segundo as pesquisas de Bohannon *et al*. (1997), na opinião da maioria dos indivíduos, o funcionamento independente na sociedade pressupõe a capacidade de caminhar. Tanto que a maioria dos pacientes e familiares que participam de tratamentos de reabilitação, consideram a caminhada como uma prioridade alta (BOHANNON *et al*, 1988). Uma vez privada ou parcialmente privada a caminhada, grande parte da independência do indivíduo está comprometida. Então o esforço para retomada desse ato, torna-se justificável. O Laboratório de Instrumentação Eletroeletrônica da Universidade Federal do Rio Grande do Sul (IEE UFRGS) tem voltado parte de seus trabalhos a estudos acerca da marcha humana para fins de

desenvolver ferramentas que auxiliam na sua reabilitação. Em trabalhos anteriores (CORRÊA; BALBINOT, 2013), tem-se focado na redução do custo de ferramentas para especialistas acompanharem a marcha e efetuarem análise através de videogrametria e acelerometria. Outra vertente de trabalhos analisa os impactos da caminhada ao solo (BALBINOT; BRUXEL; ZARO, 2013) e a distribuição do peso no corpo durante a caminhada (MACEDO, 2013). Essa linha de pesquisa tem acompanhado tendências mundiais de ferramentas para análise de marcha (Martínez-Solís *et al.*, 2014) e recuperação/acompanhamento do movimento da caminhada (MAZILU *et al.*, 2014).

2.2 ALGUNS TRABALHOS QUE UTILIZAM SENSORES INERCIAIS PARA CARACTERIZAÇÃO DE MOVIMENTOS HUMANOS

O presente trabalho tem por objetivo o desenvolvimento de um sistema para análise da marcha humana, sendo que a avaliação e constatações dos resultados sobre os ensaios do paciente são efetuadas apenas por um profissional da área da saúde. Assim, para caracterização do sistema, os resultados dos ensaios serão submetidos à análise de o quão próximo as leituras das variáveis, nesse caso angulações das articulações, são próximas dos valores esperados e o quão próximo dos resultados de trabalho da mesma natureza ele se aproxima.

Trabalhos com sensores inerciais voltados para análise da marcha humana têm obtido bons resultados, dentre os quais pode-se destacar trabalhos como Martínez-Solís *et al.* (2014), onde foi desenvolvido um sistema portátil para avaliação das angulações das articulações dos membros inferiores no plano sagital. O plano sagital representa um plano vertical que passa longitudinalmente através do corpo, dividindo-o em metades iguais direita e esquerda (Figura 2.2). O trabalho é baseado em acelerometria com uma placa de aquisição munida de um FPGA (*Field-programmable gate array*) e ligada a um computador – onde os dados são

gravados. O algoritmo se baseia em trigonometria para estimar a angulação através da gravidade como referência (Equação (2.1)). A taxa de aquisição e cálculo da angulação é de 250Hz e o erro médio do sistema (comparado com um braço robótico com um *encoder*) é de 0,75° e um erro máximo de 3°.



Figura 2.2 Planos de delimitação e secção do corpo humano.

Figura direita - Planos frontais/coronal. Figura central - Plano mediano/sagital. Figura esquerda - Planos transversos/axial. Fonte: MOORE, 2011 (livre adaptação)

$$\theta = \tan^{-1} \left(\frac{a_x}{a_z} \right) \quad (2.1)$$

As variáveis a_x e a_z representam a gravidade nos eixos x e z respectivamente e θ o ângulo lido.

Outro trabalho utilizando a estimativa de angulações de membros inferiores por acelerômetros foi desenvolvido por Djuric *et al.* (2011). Nesse trabalho, utilizou-se dois acelerômetros, microcontrolador e transceptor sem fio em cada módulo do sistema. Através de quatro módulos é efetuado a leitura das angulações dos membros inferiores do usuário para leitura no plano sagital. Os módulos transmitiam para um coordenador (formando uma rede estrela), o qual envia as informações para um computador com uma taxa de aquisição de 100Hz - Figura 2.3. No computador há um algoritmo implementado em CVI (LabWindows, National Instruments, USA) que, através da diferença de aceleração entre os acelerômetros,

calcula a velocidade angular e por sua vez a angulação do membro do usuário. O erro dinâmico em RMS (*root-mean-square*) foi de 6° e a correlação foi de 97% quando o sistema foi comparado em ensaio com um goniômetro flexível (*biometrics flexible Penny & Giles sensors*).

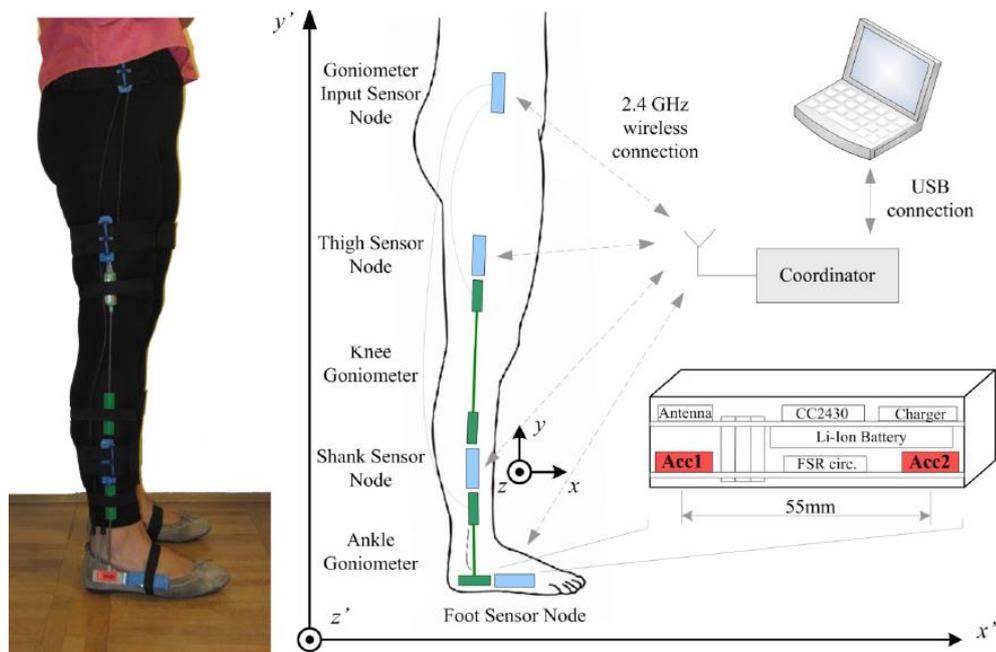


Figura 2.3 Sistema de estimativa de angulação baseada em acelerômetros.

Fonte: DJURIC *et al.*, 2011.

Kun *et al.* (2011) desenvolveram um sistema utilizando acelerômetros e magnetômetros para análise da marcha em três dimensões para os membros inferiores. O sistema continha um acelerômetro triaxial e uma unidade inercial composta de acelerômetro triaxial, giroscópio triaxial e magnetômetro triaxial em cada segmento da perna onde as informações eram adquiridas com uma taxa de aquisição de 100Hz por um microcontrolador que gravava os dados em um cartão de memória para análise posterior. O sistema utilizava um algoritmo denominado *virtual-sensor-difference-based* que consiste na geração de sensores virtuais através dos dados dos acelerômetros e magnetômetro e na diferença da orientação dos mesmos para aferir a rotação dos joelhos - Figura 2.4. Em ensaios seu erro dinâmico em RMS

foi de $2,52^\circ$, seu erro máximo de $4,13^\circ$ e sua correlação de 0,95 quando comparado com um sistema comercial de videogametria (NAC Hi-Dcam II digital high-speed camera systems - NAC Image Technology. Japan).

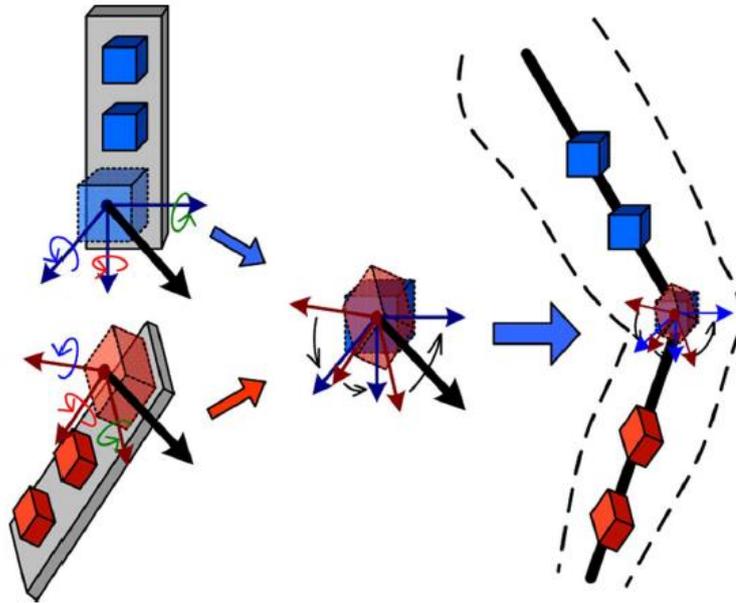


Figura 2.4 Dois sensores reais de cada membro geram um sensor virtual cuja orientações são comparadas para gerar a rotação no algoritmo *virtual-sensor-difference-based*.

Fonte: KUN *et al.*, 2011.

Mohamed *et al.* (2012) trabalhou com duas unidades inerciais com acelerômetros, giroscópio e magnetômetro triaxiais para medição da angulação tridimensional de uma das pernas. Um microcontrolador adquiria as informações dos sensores e enviava para um computador através de um transceptor *Bluetooth* com uma frequência de aquisição de 100Hz - Figura 2.5. O autor desenvolveu um filtro de Kalman estendido para fusão dos sensores e aquisição das angulações. O maior erro dos eixos em ensaio estático foi de $0,91^\circ$ RMS (X $0,91^\circ$, Y $0,29^\circ$ e Z $1,3^\circ$), já em ensaio dinâmico o maior erro foi de $4,51^\circ$ RMS (X $3,86^\circ$, Y $4,51^\circ$ e Z $4,09^\circ$).

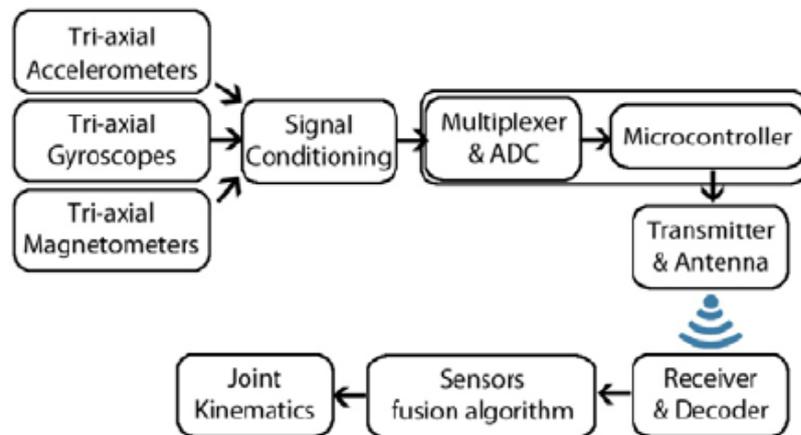


Figura 2.5 Arquitetura do sistema de unidade inercial com *Bluetooth*.

Fonte: MOHAMED, 2011.

Corrêa (2011), apresentaram um sistema com foco na redução do custo de ferramentas para especialistas acompanharem a marcha e efetuarem análise através de videogrametria, acelerometria e também com o acompanhamento de um modelo virtual. O trabalho foi baseado no plano sagital. Com acelerômetros ADXL 330 e com o transceptores Xbee foram desenvolvidas unidades de leitura sem fio com uma taxa de amostragem de 50Hz, onde as angulações eram calculadas através da Equação (2.1). O sistema de videogrametria era composto de uma câmera de baixo custo e marcadores coloridos aos quais através dessas marcações eram calculadas as angulações dos membros marcados com uma taxa de 40Hz. Para ensaios estáticos obteve-se um erro médio de 3,1° para a videogrametria e um erro médio de 1,4° para acelerometria quando comparados a um goniômetro. O erro médio entre eles durante um ensaio dinâmico foi de 5°. A Figura 2.6 apresenta uma fotografia do sistema desenvolvido ressaltando o posicionamento dos marcadores e o modelo virtual replicando o movimento executado pelo usuário do sistema.

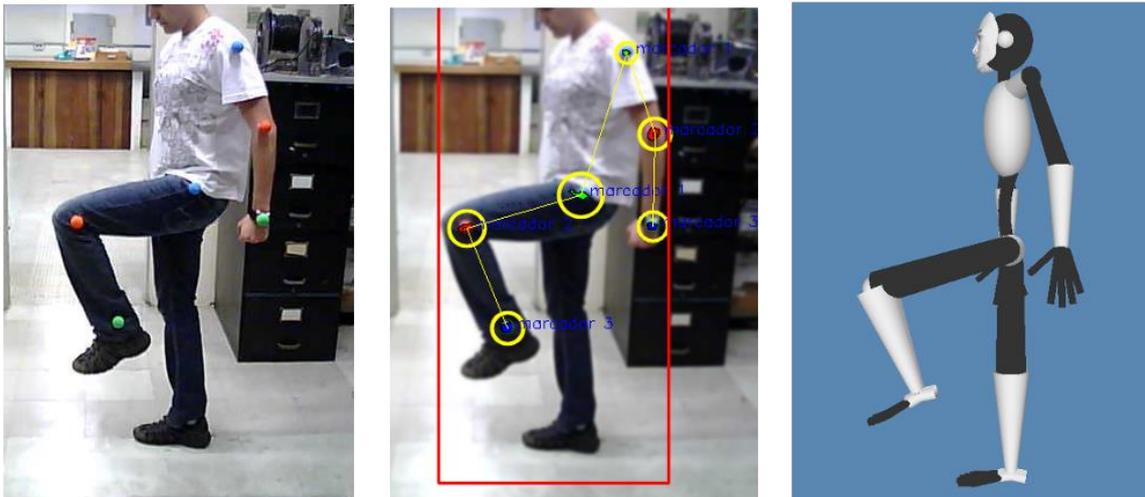


Figura 2.6 Sistema videogrametria e modelo virtual.

Fonte: CORREA, 2010.

2.3 NOTAÇÕES BÁSICAS DE MATRIZES E VETORES TÍPICAMENTE USADAS EM SISTEMAS DE POSICIONAMENTO

Nos estudos de movimentos, geralmente são usados vetores, uma referência ou plano de referência, assim como, um sistema de coordenadas de duas ou três dimensões escolhidos convenientemente para referenciar o objeto de estudo no espaço. Para que seja possível essa descrição, são definidas notações que devem ser seguidas para expressar de forma clara a representação do movimento e orientação. As definições usadas no presente trabalho adotam as referências baseadas nas literaturas utilizadas, principalmente em: GREWAL *et al*, 2001; PREMERLANI; BIZARD, 2009; CABRAL, 2009.

- Matrizes são representadas por letras maiúsculas;
- Vetores são representados por letras minúsculas;
- Os referenciais envolvidos em transformações de coordenadas são representados em sobrescrito e subscrito. Uma matriz de transformação de coordenadas do referencial do corpo para o referencial de navegação será descrito por C_b^n .

- Os símbolos utilizados pelos operadores são definidos como:

- - derivada temporal;
- $^{-1}$ - matriz inversa;
- T - matriz transposta;
- $\hat{\cdot}$ - vetor unitário;
- \times - produto vetorial;
- \cdot - produto escalar;
- δ - erro entre o valor estimado e o valor esperado.

2.3.1 Sistemas INS e AHRS

O sistema INS (*Inertial Navigation System*) ou Sistema de Navegação Inercial é um sistema que usa um computador e sensores inerciais, tais como acelerômetros e giroscópios, para calcular continuamente a posição, a orientação e a velocidade de um objeto. O sistema não tem a necessidade de referências externas como, por exemplo, o GPS (*Global Positioning System*). Esta vantagem torna o INS atraente para aplicações de navegação, tais como: aeronaves, mísseis, submarinos, torpedos, navios, plataformas flutuantes, carros, etc (FRANÇA e GOIS, 2009).

Uma grande desvantagem do sistema inercial é que pode haver um acúmulo de erros com o decorrer do tempo, causando assim imprecisões na trajetória ou na orientação. Esses erros são consequências das propagações dos erros de integração numérica e de erros de medição dos sensores. Esta desvantagem pode ser atenuada se for usado o INS integrado com outros sensores não inerciais, como o baroaltímetro, o magnetômetro, etc (PARK, 2014).

Um conjunto de sensores inerciais em um único encapsulamento é tipicamente denominado de IMU (*Inertial Measurement Unit*) ou Unidade de Medida Inercial. A união de sensores IMU com outros sensores e com um sistema de processamento para fins de

determinar a *attitude* de um objeto é denominado de AHRS (*Attitude and Heading Reference System*). AHRS são projetados para substituir sistemas de instrumentos mecânicos de navegação, principalmente na área de aeronaves, e proporcionar maior confiabilidade e precisão da *attitude*. Como usa outros sensores além dos inerciais, o AHRS minimiza as imprecisões associadas ao mesmo e em conjunto com filtros especializados (tais como filtro de Kalman e suas extensões) aumentam sua confiabilidade (GEBRE-EGZIABHER, 2004).

2.3.2 Sistemas de Referência

Segundo Grewal, Weill e Andrews (2001), os principais sistemas de referências utilizados em sistemas de navegação são demonstrados na Figura 2.7 e descritos abaixo:

- Referencial ECI (*Earth-Centered Inertial*): o referencial ECI é um sistema de coordenadas na qual a origem localiza-se no centro de massa da Terra. Os eixos x e y estão contidos no plano equatorial, onde o eixo x é apontado para o equinócio de primavera no hemisfério norte; o eixo z é direcionado para direção norte, assim como, o eixo de rotação terrestre; e, por fim, o eixo y completa a regra da mão direita. A representação das coordenadas é dada por x_i , y_i e z_i ;
- Referencial ECEF (*Earth-Centered, Earth-Fixed*): é um sistema de coordenadas cuja origem é o centro da Terra e os eixos fixos à superfície terrestre. O eixo z está direcionado ao eixo de rotação da Terra e apontando para o norte. O eixo x está direcionado ao ponto de interseção da Terra que ocorre a latitude e longitude 0° . A representação das coordenadas é dada por x_e , y_e e z_e ;
- Referencial NED (*North-East-Down*): representa o sistema de coordenadas geográficas cujos eixos estão alinhados com as direções latitude e longitude. O eixo x está alinhado com o norte geográfico, o eixo z aponta para o centro da Terra e o eixo

y completa a regra da mão direita apontando para o leste. A representação das coordenadas é dada por x_n, y_n e z_n ;

- Referencial do Objeto: o referencial do objeto é o sistema cujos eixos coordenados estão fixos em relação ao centro de massa do objeto onde o eixo x está alinhado com o eixo longitudinal do objeto para frente; o eixo z é normal ao plano do objeto e aponta para baixo; e o eixo y completa a regra da mão direita. O movimento de rotação pode ser registrado por este referencial através dos ângulos de Euler conhecidos como *roll* (ϕ), *pitch* (θ) e *yaw* (ψ). A representação das coordenadas é dada por x_b, y_b e z_b .

O Sistema de referência para localização do objeto utilizado neste trabalho para uma referência global será um sistema, baseado no Referencial NED, tendo como base o eixo z apontado para o centro da Terra e os eixo x e eixo y ortogonais entre si. Todavia, os eixos x e y serão criados de acordo com a posição inicial do corpo mensurado, uma vez que não é necessária sua referência à Terra, e sim ao cenário no qual o corpo se encontra. Assim, a rotação sobre o eixo z seguirá como referência o estado inicial do objeto no ensaio. As mesmas transformações e sistemas de rotações terão o sistema de Referência NED como base. Apenas a rotação sobre o eixo-z seguirá a referência inicial do ensaio. As componentes deste sistema são denotadas por x_g, y_g e z_g .

referencial de corpo para o referencial de navegação e compensar a aceleração da gravidade (GRANDO, 2010).

Conforme Titterton e Weston (2004), a *attitude* pode ser representada matematicamente através da Matriz de Cossenos Diretores (DCM), dos Ângulos de Euler ou de Quaternios. Para um maior entendimento desses conceitos, amplamente usados neste trabalho, as próximas seções apresentam discussões sobre os mesmos.

2.3.3.1 Matriz de Rotação

Uma matriz de rotação é um operador que ao ser aplicado sobre a representação matemática de um vetor ou de outra matriz de rotação, tem o efeito de mudar sua direção e/ou sentido, não alterando porém a sua magnitude. O vetor ou a matriz são rotacionados com os valores angulares em torno de eixos de rotações definidos pelos elementos da matriz de rotação aplicada (GOLDSTEIN, 1922).

Como o vetor p é único, ele está localizado em um ponto fixo do espaço, sendo somente representado de diferentes formas nos dois sistemas (original p_0 e após a rotação p_1), representação da Figura 2.8. Pode-se apresentar as relações entre os eixos por meio da Equação (2.2) (CABRAL, 2009).

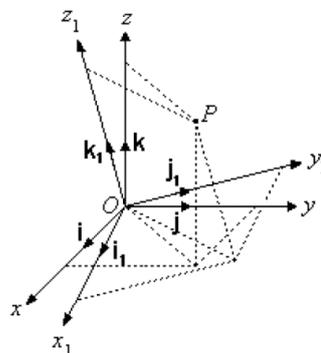


Figura 2.8 Rotação matricial.

Fonte : CABRAL, 2009.

$$\begin{aligned}
p_{x0} &= p_0 \cdot i = p_{x1}i_1 \cdot i + p_{y1}j_1 \cdot i + p_{z1}k_1 \cdot i \\
p_{y0} &= p_0 \cdot j = p_{x1}i_1 \cdot j + p_{y1}j_1 \cdot j + p_{z1}k_1 \cdot j \\
p_{z0} &= p_0 \cdot z = p_{x1}i_1 \cdot k + p_{y1}j_1 \cdot k + p_{z1}k_1 \cdot k
\end{aligned} \tag{2.2}$$

sendo p_{x0} , p_{y0} e p_{z0} as componentes do vetor p_0 ; e i , j e k vetores unitários nas direções dos eixos x , y e z , respectivamente. Ainda, pode-se representar a Equação (2.2) de forma matricial conforme a Equação (2.3).

$$\begin{bmatrix} p_{x0} \\ p_{y0} \\ p_{z0} \end{bmatrix} = \begin{bmatrix} i_1 \cdot i & j_1 \cdot i & k_1 \cdot i \\ i_1 \cdot j & j_1 \cdot j & k_1 \cdot j \\ i_1 \cdot k & j_1 \cdot k & k_1 \cdot k \end{bmatrix} \begin{bmatrix} p_{x1} \\ p_{y1} \\ p_{z1} \end{bmatrix} \tag{2.3}$$

que, resumidamente pode ser escrita como,

$$p_0 = R_1^0 p_1 \tag{2.4}$$

sendo R_1^0 a matrix de rotação que altera o referencial de p_1 para p_0 .

Dessa forma, a matriz de rotação que representa a transformação do sistema $O_1-x_1y_1z_1$ para o sistema $O-xyz$, R_1^0 é dada pela Equação (2.5).

$$R_1^0 = \begin{bmatrix} i_1 \cdot i & j_1 \cdot i & k_1 \cdot i \\ i_1 \cdot j & j_1 \cdot j & k_1 \cdot j \\ i_1 \cdot k & j_1 \cdot k & k_1 \cdot k \end{bmatrix} \tag{2.5}$$

Analogamente, pode-se representar o vetor p_1 em função de p_0 da seguinte forma:

$$p_1 = R_0^1 p_0 \tag{2.6}$$

sendo R_0^1 a matrix de rotação que altera o referencial de p_0 para p_1 .

A matriz de rotação que representa a transformação do sistema para $O-xyz$ o sistema $O_1-x_1y_1z_1$, R_0^1 é dada por:

$$R_0^1 = \begin{bmatrix} i \cdot i_1 & i \cdot j_1 & i \cdot k_1 \\ j \cdot i_1 & j \cdot j_1 & j \cdot k_1 \\ k \cdot i_1 & k \cdot j_1 & k \cdot k_1 \end{bmatrix} \tag{2.7}$$

Assim, pode-se analisar as Equações (2.5) e (2.7). Sabendo que a matriz de rotação é uma matriz ortonormal, pode-se escrever:

$$R_0^1 = (R_1^0)^{-1} = (R_1^0)^t \quad (2.8)$$

2.3.3.2 Ângulos de Euler

Os ângulos de Euler são ângulos que descrevem a orientação de um corpo rígido que gira em um espaço tridimensional. Esta formulação foi proposta por Leonard Euler em 1776. Do ponto de vista de rotações, os ângulos de Euler representam os ângulos de rotações sucessivas de um sistema de coordenadas cartesiano de três eixos. Sendo as rotações executadas nos três planos não comutativas, ou seja, a ordem da sequência de rotações quando alterada gera diferentes resultados, a sequência deve ser arbitrada e mantida durante a resolução do sistema (CRAIG, 2006).

Para que o referencial do objeto coincida com o Referencial NED, deve haver uma sequência de rotação ordenada dos eixos – por exemplo Z, Y e X.

Essa rotação é apresentada na Figura 2.9 e é descrita pela Equação (2.9) (CRAIG, 2006).

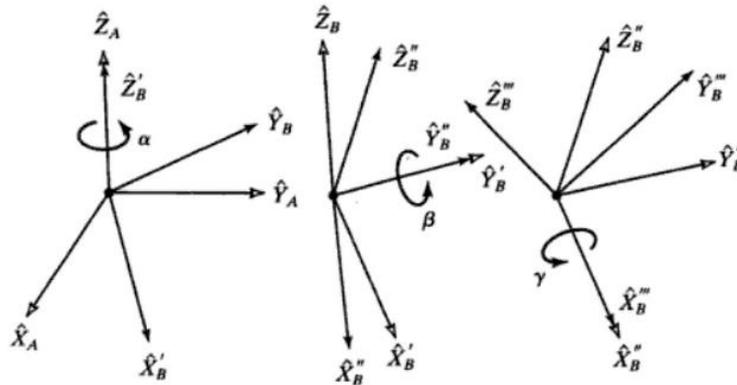


Figura 2.9 Ângulos ZYX de Euler.

$$\begin{aligned}
& R_z(\psi)R_y(\theta)R_x(\phi) \\
&= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \\
&= \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi - \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi + \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \\
&= C_b^n
\end{aligned} \tag{2.9}$$

Seguindo essa ordem, define-se os três ângulos de Euler, por meio da junção do referencial do objeto com o NED, usando a nomenclatura definida como *roll* (ϕ), *pitch* (θ) e *yaw* ou *heading* (ψ).

2.3.3.3 DCM (*Direction Cosine Matrix*)

A *Direction Cosine Matrix* ou Matriz de Cosseno Diretores (DCM) recebe este nome porque são usadas funções cossenos para formação da matriz e representação da orientação do objeto no espaço. Sejam dois sistemas de coordenadas tridimensionais $\hat{n} = [\hat{n}_1 \hat{n}_2 \hat{n}_3]$ e $\hat{b} = [\hat{b}_1 \hat{b}_2 \hat{b}_3]$, onde \hat{n}_k são vetores ortonormais (ortogonal e unitário) que representam um referencial qualquer; e \hat{b}_k um sistema de coordenadas no referencial do objeto (também ortonormais); os ângulos $\alpha_{11}, \alpha_{12}, \alpha_{13}$ formados entre o vetor \hat{n}_1 e os vetores $\hat{b}_1 \hat{b}_2 \hat{b}_3$ do referencial \hat{b} são conhecidos como cossenos diretores de \hat{n}_1 em relação ao referencial do sistema \hat{b} e sua decomposição pode ser expressa por meio da Equação (2.10) (LIMA, 2013).

$$\hat{n}_1 = \cos \alpha_{11} \hat{b}_1 + \cos \alpha_{12} \hat{b}_2 + \cos \alpha_{13} \hat{b}_3 \tag{2.10}$$

Seguindo com a projeção dos outros dois vetores \hat{n}_2 e \hat{n}_3 em \hat{b} e agrupando os termos em notação matricial, obtém-se:

$$\begin{bmatrix} \hat{n}_1 \\ \hat{n}_2 \\ \hat{n}_3 \end{bmatrix} = C_b^n \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} \quad (2.11)$$

em que

$$C_b^n = \begin{bmatrix} \cos \alpha_{11} & \cos \alpha_{12} & \cos \alpha_{13} \\ \cos \alpha_{21} & \cos \alpha_{22} & \cos \alpha_{23} \\ \cos \alpha_{31} & \cos \alpha_{32} & \cos \alpha_{33} \end{bmatrix} \quad (2.12)$$

2.3.3.4 Quatérnio

Quatérnio é uma ramificação da matemática que generaliza o cálculo vetorial e os números complexos. A teoria dos quatérnios foi desenvolvida pelo irlandês William Rowan Hamilton em 1843. O vetor q que representa um quatérnio pode ser expresso na forma de quatro parâmetros de números complexos, desta forma:

$$q = q_1 + q_2 \vec{i} + q_3 \vec{j} + q_4 \vec{k} \quad (2.13)$$

onde \vec{i}, \vec{j} e \vec{k} são vetores unitários. O conjugado é definido como:

$$q = q_1 - q_2 \vec{i} - q_3 \vec{j} - q_4 \vec{k} \quad (2.14)$$

Com base em Titterton e Weston (2004), sua rotação pode ser definida da seguinte maneira:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos(\mu/2) \\ (\mu_x/2) \sin(\mu/2) \\ (\mu_y/2) \sin(\mu/2) \\ (\mu_z/2) \sin(\mu/2) \end{bmatrix} \quad (2.15)$$

onde μ_x, μ_y, μ_z são componentes do vetor de rotação μ e $\mu = \sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2}$.

Segundo Walchko (2002), seja q e p dois vetores quatérnio, a multiplicação de ambos pode ser realizada da seguinte forma:

$$q \otimes p = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad (2.16)$$

$$p \otimes q = \begin{bmatrix} p_1 & -p_2 & -p_3 & -p_4 \\ p_2 & p_1 & p_4 & -p_3 \\ p_3 & -p_4 & p_1 & p_2 \\ p_4 & p_3 & -p_2 & p_1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (2.17)$$

onde o operador \otimes denota a multiplicação de quaténio, sendo que a o resultado depende da ordem da multiplicação.

2.3.3.5 Transformações

Para o mapeamento da rotação de um corpo de um referencial para outro, usa-se matrizes de rotação. Dessa forma, para a transformação do objeto no referencial do objeto para o referencial NED, aplica-se matrizes de transformação conforme as orientações do espaço utilizado. Para a transformação com base nos ângulos de Euler, usando a sequência de rotação ZYX, tem-se (PREMERLANI *et al.*, 2009) (NASA, 1977):

$$C_b^n = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi - \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi + \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (2.18)$$

onde os operadores ϕ , θ e ψ representam os ângulos de Euler *roll*, *pitch* e *yaw* respectivamente e C_b^n é a matriz de rotação de coordenadas do referencial do corpo para o referencial de navegação.

Essa sequência também é compatível com a matriz DCM (ambas notações descrevem a mesma matriz), ou seja, pode-se obter os ângulos de Euler através da matriz dos cossenos diretores com base na Equação (2.18) e através das Equações (2.19), (2.20) e (2.21).

$$\theta = \sin^{-1}(DCM_{3,1}) = \sin^{-1}(-\sin \theta) \quad (2.19)$$

$$\psi = \tan^{-1}\left(\frac{DCM_{2,1}}{DCM_{1,1}}\right) = \tan^{-1}\left(\frac{\cos \theta \sin \psi}{\cos \theta \cos \psi}\right) \quad (2.20)$$

$$\phi = \tan^{-1}\left(\frac{DCM_{3,2}}{DCM_{3,3}}\right) = \tan^{-1}\left(\frac{\sin \phi \cos \theta}{\cos \phi \cos \theta}\right) \quad (2.21)$$

Se forem usado quatérnios, a matriz de rotação é definida como (NASA 1977):

$$C_b^n = \begin{bmatrix} (q_1^2 + q_2^2 - q_3^2 - q_4^2) & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & (q_1^2 - q_2^2 + q_3^2 - q_4^2) & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{bmatrix} \quad (2.22)$$

2.3.4 Um breve relato sobre sensores inerciais

Movimentos humanos são passíveis de serem caracterizados através de sensores inerciais, os quais medem variáveis como a aceleração (acelerômetros) e velocidade angular (giroscópios). Esses sensores são tipicamente leves e de pequeno porte o que os tornam adequados para caracterização do movimento de membros humanos (EMIL *et al*, 2005; ZHOU e HU, 2007). Acelerômetros já são utilizados para monitorar o movimento humano desde 1950, mas o custo elevado dos sensores tornava restritivo a ampla utilização desse

sensor para esse fim. Com o avanço tecnológico, que seguiu com a baixa no preço desses sensores, ensaios com sensores inerciais voltaram a ser realizados com mais intensidade na década de 70 para mensuração do movimento humano. A partir de então, o uso desses sensores para esse fim tem aumentado e se tornado popular (GODFREY *et al.*, 2008).

Acelerômetros são sensores inerciais capazes de medir a aceleração resultante em um dado instante de tempo. O princípio de funcionamento da maioria dos acelerômetros é baseado no sistema massa-mola, ou seja, a deformação da mola é proporcional à aceleração resultante naquele eixo (BALBINOT; BRUSAMARELLO, 2007). Um dos modelos de acelerômetros de baixo custo existentes no mercado é o acelerômetro capacitivo. Esse tipo de acelerômetro funciona através da alteração da capacitância de um capacitor com a aceleração. Dessa forma, uma placa paralela do capacitor é colocada em uma superfície que se desloca proporcionalmente à aceleração exercida e a outra placa do capacitor é fixa. Quando a aceleração sobre as placas varia, a distância entre as mesmas varia também e, por consequência, ocorre uma alteração na sua capacitância. Através dessa variação, é gerado o sinal de saída do sensor.

O giroscópio é um sensor inercial capaz de medir a velocidade angular em certo instante de tempo. Giroscópios micromecânicos (MEMS) funcionam de acordo com a teoria de que quando uma velocidade angular é aplicada a um corpo, uma força de Coriolis é gerada, proporcional à velocidade angular. A força de Coriolis então é transformada em sinal elétrico e é gerado o sinal de saída do sensor (FORHAN *et al.*, 2010).

Apesar de não fazer parte dos sensores inerciais, o magnetômetro é muito utilizado em conjunto com os mesmos para determinação da *attitude*, conforme descrito em capítulos anteriores. Magnetômetros são sensores capazes de detectar a direção do campo magnético da Terra. O uso dos mesmos é comum em fusões de sensores em conjunto com sensores inerciais com o objetivo de determinar a orientação de um objeto no ambiente estudado.

3 ESTUDO E DESENVOLVIMENTO DE ALGORITMOS DE FUSÃO

3.1 INTRODUÇÃO

A principal métrica do sistema desenvolvido é a determinação da *attitude* do membro do usuário através de sensores. Assim, os estudos aqui abordados são sobre os algoritmos de fusão de sensores utilizados (e suas modificações) para determinação dessa métrica chamada de *attitude*.

Salienta-se que o termo *fusão de sensores*, utilizado neste trabalho, é a combinação de dados de sensores (ou de dados derivados de dados de sensores) de diferentes fontes de tal forma que a informação resultante é melhor (resultados mais precisos, mais completos ou mais confiáveis) do que seria possível obter se essas fontes fossem utilizadas individualmente. O sentido desse termo aqui utilizado tem como base a coletânea de definições selecionadas pelo grupo de estudos *Sensor Data and Information of Institute of Computer Science of Universitat Bonn* (2012).

Os algoritmos usados para gerar a fusão de sensores com intuito de determinação da *attitude* a serem descritos são: DCM IMU (PREMERLANI e BIZARD, 2009), eCompass (MANSUCLAL e SEABRA, 2010; OZYAGCILAR, 2012) e filtro de Kalman (SABATELLI *et al.*, 2011; KIM, 2011).

3.2 DCM IMU

Consiste em um algoritmo descrito em William Premerlani e Paul Bizard (2009), tendo como base o trabalho de Robert Mahony (2009), onde é utilizado um giroscópio tri-axial para gerar uma matriz DCM e usado os dados do acelerômetro e do GPS ou magnetômetro para corrigir os erros numéricos causados pelas funções de integração na geração da matriz DCM.

O algoritmo segue a seguinte ordem:

1. O giroscópio é usado como principal fonte de informação sobre a orientação. É efetuada a integração da velocidade angular para determinar o deslocamento angular e é então atualizado a orientação atual com essa informação;
2. tendo em vista que os erros numéricos na integração gradualmente violam as restrições de ortogonalidade que o DCM deve satisfazer, é efetuada uma correção de ortogonalidade nos elementos da matriz para satisfazer essas restrições;
3. sabendo que existem erros numéricos, fatores inerciais (*drift*) e *offset* do giroscópio que gradualmente vão acumulando erros nos elementos da matriz DCM, são usadas as referências vetoriais para detectar esses erros e um controlador proporcional integral (PI) para gerar uma realimentação entre o erro detectado e a entrada dos giroscópios. A referência vetorial de *roll* (ϕ) e *pitch* (θ) é oriunda da leitura dos acelerômetros com relação à gravidade, enquanto que a referência vetorial de *yaw* (ψ) é oriunda da leitura do magnetômetro com relação ao eixo magnético terrestre.

A Figura 3.1 apresenta um diagrama de blocos representando as etapas do algoritmo.

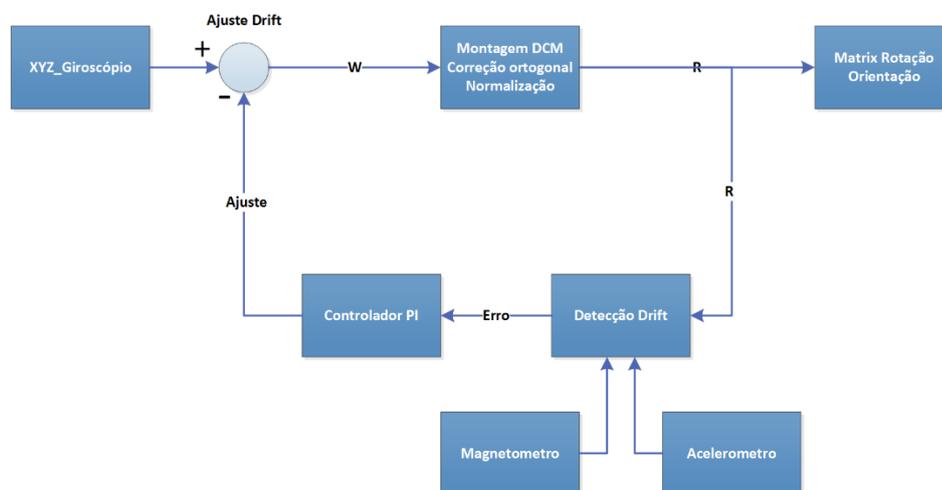


Figura 3.1 Diagrama de bloco do algoritmo DCM IMU.

A atualização da matriz DCM, através do giroscópio, pode ser expressa por meio da Equação

(3.1):

$$R(t) = R(t - dt) + R(t - dt) \begin{bmatrix} 0 & -\omega_z dt & \omega_y dt \\ \omega_z dt & 0 & -\omega_x dt \\ -\omega_y dt & \omega_x dt & 0 \end{bmatrix} \quad (3.1)$$

onde R é a matrix de rotação (DCM), dt é a variação do tempo entre amostras e ω_x , ω_y e ω_z são as componentes da velocidade angular sobre os eixos.

Como trata-se de integração das informações do giroscópio, os erros citados anteriormente podem causar não ortogonalidade no eixo da matriz DCM. Para tratar os erros de não ortogonalidade dos eixos é usado um algoritmo de normalização conforme Equações (3.3) à (3.6) e seguindo as nomenclaturas da Equação (3.2).

$$R(t) = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \quad (3.2)$$

$$X = \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} ; Y = \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} ; erro = X \cdot Y = X^T Y = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \end{bmatrix} \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} \quad (3.3)$$

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{ortogonal} = X_{ortogonal} = X - \frac{erro}{2} Y \quad (3.4)$$

$$\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}_{ortogonal} = Y_{ortogonal} = Y - \frac{erro}{2} X \quad (3.5)$$

$$\begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix}_{ortogonal} = Z_{ortogonal} = X_{ortogonal} \times Y_{ortogonal} \quad (3.6)$$

Para que a matriz de rotação permaneça ortonormal como a DCM, usa-se uma expansão da série de Taylor conforme Equações (3.7) a (3.9) (PREMERLANI e BIZARD, 2009).

$$X_{normalizado} = \frac{1}{2} (3 - X_{ortogonal} \cdot X_{ortogonal}) X_{ortogonal} \quad (3.7)$$

$$Y_{normalizado} = \frac{1}{2} (3 - Y_{ortogonal} \cdot Y_{ortogonal}) Y_{ortogonal} \quad (3.8)$$

$$Z_{normalizado} = \frac{1}{2} (3 - Z_{ortogonal} \cdot Z_{ortogonal}) Z_{ortogonal} \quad (3.9)$$

Após a normalização, a matriz é reconstruída de modo que a mesma esteja ortonormal com relação aos seus eixos formadores, eliminando assim os erros de ortogonalidade gerados pelos erros de integração do giroscópio.

Apesar dos procedimentos de correções adotados anteriormente é possível que a estimativa da *attitude* esteja incorreta. Para que ainda seja possível a correção da *attitude*, usa-se as referências vetoriais da gravidade e do eixo magnético terrestre para correção através da realimentação do sistema. A Equação (3.10) representa a detecção do erro com o vetor gravidade.

$$RollPitchCorreção = \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \times \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (3.10)$$

onde a_x, a_y, a_z correspondem aos eixos do acelerômetro representando a gravidade terrestre. Nota-se que o produto vetorial entre esses vetores da Equação (3.10) representa o quão defasados o eixo Z da *attitude* atual está da gravidade. Caso esses eixos não estejam defasados a resultante será um vetor de zeros.

No artigo base (PREMERLANI e BIZARD, 2009) foi usado a referência magnética do GPS, mas no presente trabalho é usado a referência magnética do magnetômetro contida no dispositivo usado (ArduIMU), sendo o equacionamento para detecção do erro expresso de acordo com as Equações (3.11) e (3.12).

$$YawCorreçãoGround = r_{xx}B_{fy} - r_{yx}B_{fx} \quad (3.11)$$

onde B_{fx} e B_{fy} representam os elementos dos eixos x e y do vetor resultante do magnetômetro respectivamente - esses eixos serão detalhados na próxima sessão. Note que a variável *YawCorreçãoGround* representa o valor do eixo Z do produto vetorial entre o eixo do magnetômetro com os eixos da atual *attitude*. Caso os eixos não estejam defasados, a resultante é zero. Para que essa detecção também represente um vetor do erro é usada a Equação (3.12).

$$YawCorreção = YawCorreçãoGround \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \quad (3.12)$$

Com os vetores de detecção usa-se um controlador proporcional e integral (PI) para correção dos valores fornecidos pelo giroscópios, conforme as Equações (3.13) a (3.15).

$$\omega_{P_Correção} = K_{P_RollPitch} \cdot RollPitchCorreção + K_{P_Yaw} \cdot YawCorreção \quad (3.13)$$

$$\begin{aligned} \omega_{I_Correção} = & \omega_{I_Correção} + K_{I_RollPitch} \cdot dt \cdot RollPitchCorreção \\ & + K_{I_Yaw} \cdot dt \cdot YawCorreção \end{aligned} \quad (3.14)$$

$$\omega_{Correção} = \omega_{P_Correção} + \omega_{I_Correção} \quad (3.15)$$

onde a variável $\omega_{Correção}$ trata-se de um vetor para correção da entrada do giroscópio x, y e z. Os valores típicos são apresentados a seguir e são usados com base na literatura (PREMERLANI e BIZARD, 2009), sendo resultados de ensaios .

- $K_{P_RollPitch} = 1,515/\text{gravidade}$
- $K_{P_Yaw} = 0,000012$
- $K_{I_RollPitch} = 0,00101/\text{gravidade}$
- $K_{I_Yaw} = 0,000005$

Sendo a Equação (3.1) alimentada com a velocidade angular corrigida ($\omega_{corrigida} = \omega_{sensor} - \omega_{Correção}$) tem-se a *attitude* do dispositivo no referencial NED.

Como o presente trabalho leva a componente *Yaw* com o referencial do posicionamento inicial, durante os ensaios iniciais a primeira leitura era adquirida e gerada a matriz inversa da mesma. Na sequência das demais essa matriz inversa era rotacionada na matriz resultante para plotagem do resultado. Todavia notou-se que havia ainda um leve *drift* em *Yaw*. Então, optou-se por retirar as Equações (3.11) e (3.12) e manter a saída sem alterações, uma vez que a componente *Yaw* começaria pelo seu posicionamento inicial (zero). Para viabilizar a solução, outras correções foram aplicadas conforme análise apresentada na Seção 5.1.

3.3 BÚSSOLA ELETRÔNICA *E*COMPASS

Baseado no algoritmo descrito em de Jitesh Mansuclal (2010) e com o uso de algumas descrições de Talat Ozyagcilar (2012), tem-se a formação de uma bússola eletrônica (*eCompass*) usando um acelerômetro tri-axial e um magnetômetro tri-axial. Esse sistema pode informa ao final do seu processo o *roll* (ϕ), *pitch* (θ) e *yaw* (ψ) do dispositivo, podendo determinar sua *attitude* com referência NED.

Considerando a posição do acelerômetro e magnetômetro e tendo como referenciais a gravidade terrestre e o polo magnético respectivamente, tem-se como modelagem inicial as Equações (3.16) e (3.17).

$$G_r = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.16)$$

$$B_r = B \begin{bmatrix} \cos \delta \\ 0 \\ \sin \delta \end{bmatrix} \quad (3.17)$$

sendo G_r a referência do acelerômetro e B_r a referência do magnetômetro em relação ao meio. A variável g representa a gravidade ($g = 9,8 \text{ m/s}^2$) e a variável B representa o campo geomagnético com valores entre $22 \mu T$ e $67 \mu T$.

As Equações (3.16) e (3.17) com referenciais do objeto (G_p e B_p) são descritas conforme as Equações (3.18) e (3.19):

$$G_p = R_x(\phi)R_y(\theta)R_z(\psi)G_r = R_x(\phi)R_y(\theta)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.18)$$

$$B_p = R_x(\phi)R_y(\theta)R_z(\psi)B_r = R_x(\phi)R_y(\theta)R_z(\psi)B \begin{bmatrix} \cos \delta \\ 0 \\ \sin \delta \end{bmatrix} \quad (3.19)$$

sendo $R_x(\phi)$, $R_y(\theta)$ e $R_z(\psi)$ descritas conforme as Equações (3.20), (3.21) e (3.22).

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (3.20)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.21)$$

$$R_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

Nota-se que se trata de uma rotação na sequência XYZ. Reescrevendo a Equação (3.18) e considerando $\psi = 0$, então, como a $R_z(\psi)$ é uma matriz identidade, chega-se a Equação (3.23).

$$R_y(-\theta)R_x(-\phi)G_P = R_y(-\theta)R_x(-\phi) \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.23)$$

sendo o vetor $\begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix}$ a descrição dos componentes da gravidade. Através da Equação (3.23)

pode-se determinar *roll* (ϕ) e *pitch* (θ) conforme Equações (3.24) e (3.25).

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.24)$$

$$\begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.25)$$

Resultando,

$$G_{py} \cos \phi - G_{pz} \sin \phi = 0 \quad (3.26)$$

Assim,

$$\tan \phi = \frac{G_{py}}{G_{pz}} \quad (3.27)$$

$$G_{px} \cos \theta + G_{py} \sin \theta \sin \phi + G_{pz} \sin \theta \cos \phi = 0 \quad (3.28)$$

obtem-se

$$\tan \theta = \frac{-G_{px}}{G_{py} \sin \phi + G_{pz} \cos \phi} \quad (3.29)$$

sendo que G_{px} , G_{py} e G_{pz} representam as componentes do acelerômetro (a leitura do acelerômetro nos eixos).

Através das Equações (3.27) e (3.29) pode-se adquirir *roll* (ϕ) e *pitch* (θ). O *yaw* (ψ) pode ser adquirida através das Equações (3.30) e (3.31).

$$R_z(\psi) \begin{bmatrix} B \cos \delta \\ 0 \\ B \sin \delta \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B \cos \delta \\ 0 \\ B \sin \delta \end{bmatrix} = R_y(-\theta) R_x(-\phi) B_P \quad (3.30)$$

$$\begin{bmatrix} \cos \psi B \cos \delta \\ -\sin \psi B \cos \delta \\ B \sin \delta \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} B_{px} \\ B_{py} \\ B_{pz} \end{bmatrix} \quad (3.31)$$

$$= \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} B_{px} \\ B_{py} \\ B_{pz} \end{bmatrix} \quad (3.32)$$

$$= \begin{bmatrix} B_{px} \cos \theta + B_{py} \sin \theta \sin \phi + B_{pz} \sin \theta \cos \phi \\ B_{py} \cos \phi - B_{pz} \sin \phi \\ -B_{px} \sin \theta + B_{py} \cos \theta \sin \phi + B_{pz} \cos \theta \cos \phi \end{bmatrix} = \begin{bmatrix} B_{fx} \\ B_{fy} \\ B_{fz} \end{bmatrix} \quad (3.33)$$

onde o vetor $\begin{bmatrix} B_{fx} \\ B_{fy} \\ B_{fz} \end{bmatrix}$ representa as componentes do magnetômetro, considerando $\theta = \phi = 0$.

Através dessa simplificação, tem-se as Equações (3.34), (3.35) e (3.36).

$$B_{fx} = \cos \psi B \cos \delta \quad (3.34)$$

$$-B_{fy} = \sin \psi B \cos \delta \quad (3.35)$$

$$\tan \psi = \frac{-B_{fy}}{B_{fx}} = \frac{B_{pz} \sin \phi - B_{py} \cos \phi}{B_{px} \cos \theta + B_{py} \sin \theta \sin \phi + B_{pz} \sin \theta \cos \phi} \quad (3.36)$$

Através da Equação 3.36 pode-se determinar o *yaw* (ψ). Finalmente, através dos valores de *roll* (ϕ), *pitch* (θ) e *yaw* (ψ) pode-se remontar a matriz de rotação com o referencial NED através da Equação 3.37. Nota-se que a sequência de rotação é a XYZ.

C_b^n

$$= \begin{bmatrix} \cos \theta \cos \psi & -\cos \theta \sin \psi & \sin \theta \\ \cos \phi \sin \psi + \cos \psi \sin \phi \sin \theta & \cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi & -\cos \theta \sin \phi \\ \sin \phi \sin \psi - \cos \phi \cos \psi \sin \theta & \cos \psi \sin \phi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix} \quad (3.37)$$

Como o presente trabalho leva a componente *Yaw* com o referencial do posicionamento inicial, a primeira leitura é adquirida e gerada a matriz inversa da mesma. Na sequência das demais essa matriz inversa é rotacionada na matriz resultante para plotagem do resultado.

3.4 FILTRO DE KALMAN

3.4.1 Descrição do Filtro de Kalman

O filtro de Kalman é um método matemático criado por Rudolf Kalman. Sua finalidade é utilizar medições de grandezas realizadas ao longo do tempo e gerar resultados com tendência a se aproximar dos valores esperados dos valores associados de grandezas medidas (KALMAN, 1960).

O filtro de Kalman é um estimador linear recursivo que calcula uma estimativa de variância mínima para um estado a partir de observações relacionadas a este que evoluem ao decorrer do tempo (LAGES, 2008). Assim, ele estima valores esperados de grandezas medidas e valores associados predizendo um valor, estimando a incerteza do valor predito e calculando uma média ponderada entre o valor predito e o medido, utilizando um peso maior para o valor de menor incerteza. As estimativas geradas pelo método tendem a estar mais próximas ao valor real, uma vez que a média ponderada apresenta uma melhor estimativa, com incerteza menor que ambos os valores individualmente utilizados.

O estado do filtro é representado por duas variáveis:

- $\hat{x}(k|k)$ é a estimativa a posteriori do estado no tempo k ;
- $P(k|k)$ é a matriz de covariância do erro a posteriori (uma medida da qualidade estimativa do estado).

O filtro de Kalman é mais comumente descrito em duas fases distintas: Predição e Atualização. A fase de predição se baseia na estimativa do estado no passo anterior para obter uma estimativa do estado atual. Essa predição é chamada de estimativa a priori, pois não inclui a informação vinda da observação do estado atual. Na fase de atualização, a predição a priori é combinada com a observação atual para refinar a estimativa do estado atual, que é denominada estimativa a posteriori. Tipicamente, o método fica alternando entre as duas

fases, com a predição prevendo o estado até o instante da próxima observação e a atualização incorporando a informação observada (LAGES, 2008).

O procedimento de predição é descrito por meio das Equações (3.38) e (3.39).

$$\text{Predição do estado (priori)} \quad \hat{x}(k|k-1) = A(k) \hat{x}(k-1|k-1) \quad (3.38)$$

$$\text{Predição da covar. (priori)} \quad P(k|k-1) = A(k)P(k-1|k-1)A^T(k) + Q(k) \quad (3.39)$$

As formulas de predição são descritas nas Equações 3.40 e 3.44.

$$\text{Resíduo da medição} \quad \tilde{y}(k) = y(k) - H(k)\hat{x}(k|k-1) \quad (3.40)$$

$$\text{Resíduo da covariância} \quad S(k) = H(k)P(k|k-1)H^T(k) + R(k) \quad (3.41)$$

$$\text{Ganho ótimo de Kalman} \quad K(k) = P(k|k-1)H^T(k)S^{-1}(k) \quad (3.42)$$

$$\text{Estado atualizado (posteriori)} \quad \hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\tilde{y}(k) \quad (3.43)$$

$$\text{Covariância estimada (posteriori)} \quad P(k|k) = (I - K(k)H(k))P(k|k-1) \quad (3.44)$$

onde $A(x)$ é o modelo de transição de estados; $H(k)$ é o modelo de observação; $Q(k)$ é a covariância do ruído do processo; $R(k)$ é a covariância do ruído de observação.

Para chegar nas equações anteriores, parte-se das equações bases, onde é descrito um sistema dinâmico com a Equação (3.45).

$$x(k+1) = A(k)x(k) + \omega(k) \quad (3.45)$$

onde relembra-se que $x(k)$ é o estado do sistema, $A(k)$ é o modelo de transição de estados, ou seja, $A(k)$ é a matriz que relaciona $x(k)$ com $x(k+1)$; e $\omega(k)$ é um ruído branco com estrutura de covariância conhecida (denominado ruído de processo).

A medida do processo é descrita pela Equação (3.46).

$$y(k) = H(k)x(k) + v(k) \quad (3.46)$$

onde $H(k)$ é a matriz que relaciona $x(k)$ e $y(k)$; e $v(k)$ é um ruído branco com estrutura de covariância conhecida (denominado ruído de medida).

Sendo o erro da estimativa a posteriori dado pela Equação (3.47), a matriz de covariância associada é dada pelas Equações (3.48) e (3.49).

$$e(k|k) = x(k) - \hat{x}(k|k) \quad (3.47)$$

$$P(k|k) = E[e(k|k)e^T(k|k)] \quad (3.48)$$

$$P(k|k) = E[(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T] \quad (3.49)$$

onde e é o erro e E é o valor esperado.

Unindo as Equações (3.40), (3.43), (3.46) com (3.49) tem-se a formação da Equação 3.50.

$$\begin{aligned} P(k|k) = E \left\{ \right. & [(I - K(k)H(k))(x(k) - \hat{x}(k|k-1)) \\ & + K(k)v(k)][(I - K(k)H(k))(x(k) - \hat{x}(k|k-1)) \\ & \left. + K(k)v(k)]^T \right\} \end{aligned} \quad (3.50)$$

sendo I a matrix identidade.

Computando-se as esperanças, tem-se a Equação (3.51). Nota-se que o erro $(x(k) - \hat{x}(k|k-1))$ não está correlacionada com o ruído de medida $v(k)$.

$$P(k|k) = (I - K(k)H(k))P(k|k-1)(I - K(k)H(k))^T + K(k)R(k)K^T(k) \quad (3.51)$$

onde a covariância do ruído de observação $R(k)$ é dada por $E[v(k)]$ e $P(k|k-1)$ pela Equação (3.52).

$$\begin{aligned} P(k|k-1) &= E[e(k|k-1)e^T(k|k-1)] \\ &= E[(x(k) - \hat{x}(k|k-1))(x(k) - \hat{x}(k|k-1))^T] \end{aligned} \quad (3.52)$$

Deseja-se encontrar o valor de $K(k)$ na Equação (3.51) que minimiza os termos da diagonal principal de $P(k|k)$, pois estes elementos são as variâncias dos erros de estimação do estado estimado. Para essa minimização, deriva-se a Equação (3.51) com relação a $K(k)$ e iguala-se a zero, conforme Equação (3.53). Na Equação (3.54) tem-se o resultado da Equação (3.53) reescrita de modo a evidenciar $K(k)$.

$$\frac{\partial P(k|k)}{\partial K(k)} = -2(I - K(k)H(k))P(k|k-1)H^T(k) + 2K(k)R(k) = 0 \quad (3.53)$$

$$K(k) = P(k|k-1)H^T(k)(H(k)P(k|k-1)H^T(k) + R(k))^{-1} \quad (3.54)$$

Sendo a Equação (3.53) a união das Equações (3.41) e (3.42). Sendo essa a equação que minimiza o erro médio quadrático e é chamado de Ganho de Kalman.

Unido-se as Equações (3.54) e (3.53) e reorganizando (LAGES, 2008) chega-se à Equação (3.44) apresentada.

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \quad (3.44)$$

3.4.2 Desenvolvimento do algoritmo do filtro de Kalman

Neste trabalho, o desenvolvimento do algoritmo, do filtro de Kalman é baseado nos estudos de Simone Sabatelli *et al.* (2011) onde o filtro de Kalman determina a *attitude* através das medidas dos acelerômetros e da estimativa do estado gerada através do giroscópio. Todavia, esse algoritmo fica limitado apenas ao uso da gravidade como referência de entrada. Então, somou-se a ideia de Phil Kim (2010) para usar como entrada a *attitude* gerada por outros sensores, como o *eCompass* (Seção 3.3). Assim, o filtro de Kalman através da *attitude* do *eCompass* e da estimativa do estado gerado através do giroscópio utilizaria os três tipos de sensores (acelerômetro, magnetômetro e giroscópio) para determinar uma *attitude* melhor. O esquema da Figura 3.2 demonstra o funcionamento desse esquema.

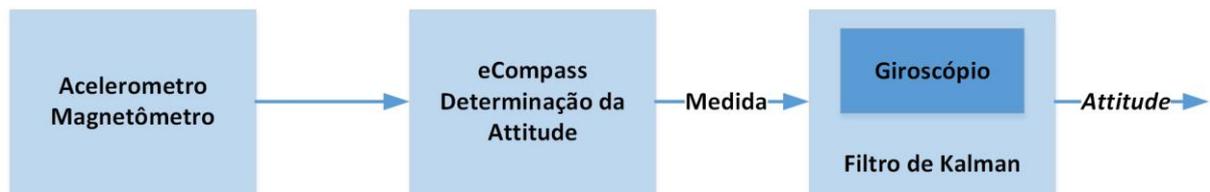


Figura 3.2 Esquema do filtro de Kalman.

A *attitude* medida (z_k) do *eCompass* é representada em quaternions com a relação de rotação ZYX pela Equação (3.38) (NASA 1977).

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \end{bmatrix} = z_k \quad (3.38)$$

Como modelagem do sistema (ou modelo de transição de estados) (A), foram utilizados as informações dos giroscópios, conforme a Equação (3.39) (SABATELLI, P; 2010)(TRAWNY e ROUMELIOTIS, 2005):

$$A = I + \Delta t \cdot \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & -p \\ r & q & -p & 0 \end{bmatrix} \quad (3.39)$$

onde p, q, r representam os eixos x, y e z do giroscópio.

O estado inicial do modelo de observação (H) é dado pela Equação (3.40).

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.40)$$

A covariância do ruído do processo (Q) e a covariância do ruído da observação (R) são dadas conforme a literatura pela Equação (3.41) (KIM; 2010):

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad (3.40)$$

O estado inicial (x) e o valor inicial da matriz de covariância do erro (P) são dados através da Equação (3.42).

$$x = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.41)$$

A Figura 3.3 apresenta um esquema do algoritmo desenvolvido neste trabalho baseado nos estudos anteriormente apresentados.

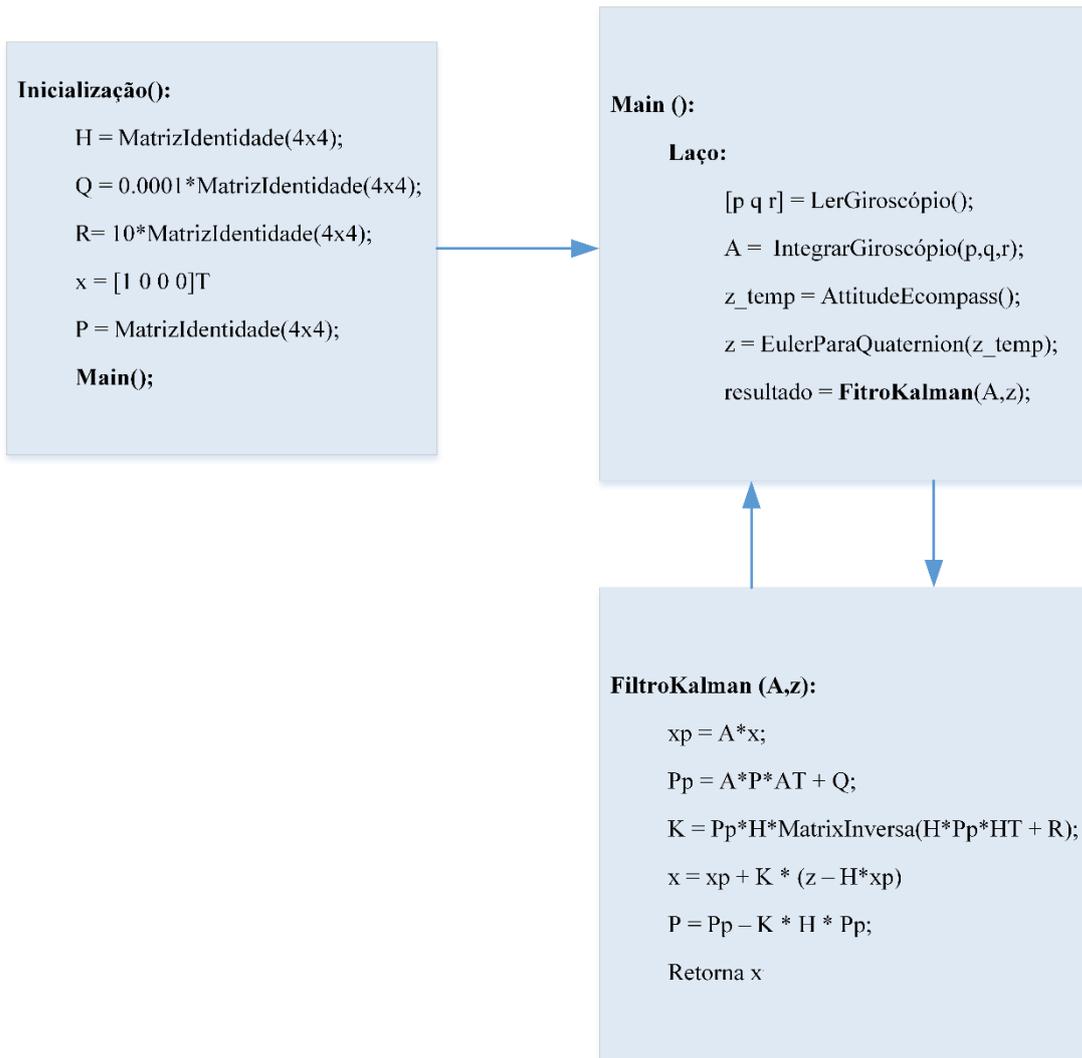


Figura 3.3 Desenvolvimento do Filtro de Kalman.

Como retorno do estado x tem-se a *attitude* descrita em quatérnio, sendo sua transcrição para uma matriz de rotação descrita por meio da Equação 2.20 e replicada Equação 3.42.

$$C_b^n = \begin{bmatrix} (q_1^2 + q_2^2 - q_3^2 - q_4^2) & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & (q_1^2 - q_2^2 + q_3^2 - q_4^2) & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{bmatrix} \quad (3.42)$$

Como toda a modelagem partiu do algoritmo *eCompass* já com o sistema de referência adaptado ao presente trabalho, não foi necessário tratar os resultados de maneira diferenciada.

4 DESENVOLVIMENTO DO SISTEMA EXPERIMENTAL

Este capítulo é dividido nas seguintes partes:

- a) aparato experimental: onde são apresentados os principais itens que estão envolvidos no trabalho, assim como suas configurações;
- b) rede de sensores: descrição do hardware e software desenvolvidos para determinar a *attitude* dos sensores baseados nas discussões e algoritmos apresentados no Capítulo 3;
- c) modelo virtual: desenvolvimento do modelo virtual biomecânico do corpo humano;
- d) programa para ensaio da marcha: funções criadas para fins de armazenamento de informações durante o ensaio e reprodução dos mesmos para ensaios acompanhados, bem como métricas de realimentação para o usuário corrigir a postura durante o ensaio.

4.1 APARATO EXPERIMENTAL

O principal objetivo do presente estudo é o desenvolvimento de uma ferramenta acadêmica de baixo custo para o estudo da marcha humana. Esse sistema consiste no sensoramento da marcha de um paciente que terá seu movimento ser reproduzido por um modelo virtual. Dessa maneira ele poderá corrigir sua ação através do modelo e ter os registros de variáveis cinemáticas de sua marcha para estudos e acompanhamento mais detalhado da sua recuperação e/ou tratamento. Como o objetivo é sensoriar o movimento do paciente para sua reprodução, optou-se por determinar as angulações dos membros de interesse e, através dessas angulações reproduzir o comportamento do membro no modelo virtual. Assim, são instalados sensores nos membros do usuário que determinam a *attitude* daquele membro em relação ao meio. Esses sensores formam uma rede que transmite as informações para um computador, onde um programa desenvolvido estará lendo as informações recebidas. Essas informações são tratadas e enviadas para reprodução em um

modelo virtual. Nessa reprodução, o modelo deve converter a *attitude* obtida em angulações dos membros para geração do movimento semelhante ao do usuário.

Para tornar o sistema flexível para o estudo da marcha humana, onde o usuário pode trabalhar apenas com as partes do corpo que escolher ou com partes do sistema que for conveniente, optou-se por desenvolver um sistema modular, que consiste em um grupo de ferramentas que podem trabalhar em conjunto com configurações diferentes para execução dos ensaios. Os módulos são formados pelo modelo virtual, rede de sensores e o programa para ensaios, conforme a Figura 4.

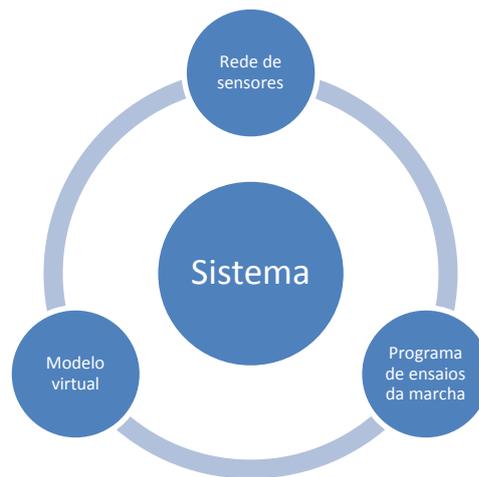


Figura 4.1 Módulos do sistema.

A rede de sensores é constituída por pequenos dispositivos sem fio, onde cada módulo é posicionado em um segmento do corpo do voluntário. Os módulos são compostos por um transceptor, um microcontrolador, sensores (acelerômetro, giroscópio e magnetômetro) e uma bateria. O transceptor utilizado é o modelo Xbee da *Digi International* (Figura 4.2) que utiliza o protocolo IEEE 802.15.4 para comunicação. O microcontrolador e sensores são integrados em uma única placa denominada ArduIMU V3 (Figura 4.3). Para a comunicação, os módulos geram uma rede estrela com um coordenador que captura as informações que os módulos

enviam (leitura dos sensores) e repassa para um computador. Os ArduIMU são programáveis então, após definir o algoritmo mais adequado para fusão de sensores, foi gravado um código que trabalhava a leitura dos sensores e envia esses dados juntamente com as *attitude* calculadas para o dispositivo coordenador.



Figura 4.2 Transceptor Xbee.



Figura 4.3 ArduIMU V3.

O sistema desenvolvido tem como complemento um código denominado, Programa de Ensaio da Marcha, que recebe as informações de todos os sensores através da comunicação com o coordenador. Esse programa desencapsula as informações dos pacotes recebidos, envia as informações para o modelo virtual e grava as variáveis de interesse (angulações dos membros, aceleração, etc) em arquivos para fins de estudos mais detalhados da cinemática do usuário e para que seja possível reproduzir posteriormente o ensaio caso for desejado. O

programa foi desenvolvido na linguagem de programação Python e permite através de um arquivo de configuração, organizar o sistema internamente de modo que seja possível ler as informações do coordenador, de uma porta serial ou ainda de arquivos; Além disso, ele permite selecionar quais sensores representam qual articulação do modelo virtual e também permite configurar qual modelo virtual será usado. Foi através desse programa que foram inicialmente implementados os algoritmos de fusão citados no Capítulo 3 e então escolhido qual algoritmo seria implementado diretamente no ArduIMU.

O modelo virtual consiste em um modelo computacional do corpo humano formado por pequenos segmentos representativos das principais estruturas ósseas do corpo e organizadas como tal. Esse modelo tem por finalidade uma representatividade mecânica do corpo humano e serve para o acompanhamento do movimento do paciente durante o exame. Isso permite que o usuário tenha uma realimentação visual do movimento executado por ele, onde o mesmo pode ater-se a corrigir o movimento conforme exercício proposto. Além disso, permite a observação pelo profissional da área da saúde do movimento executado, assim como, o histórico dos movimentos executados em determinados ensaios. O modelo virtual foi criado usando os softwares *MakeHuman* e *Fuse*, e depois foi importado para o *Blender Game Engine*, um software de animação gráfica e jogos, onde foram manipulados ambientes e desenvolvidos rotinas de interação. Então, após adquirir e tratar as informações, o programa de ensaio envia as informações para o *Blender*, que, por sua vez, também trata as informações e movimenta o modelo conforme os sensores. O desenvolvimento das rotinas internas do *Blender* foram desenvolvidas na linguagem *Python*.

O fluxo de informações do sistema é apresentado na Figura 4.4.

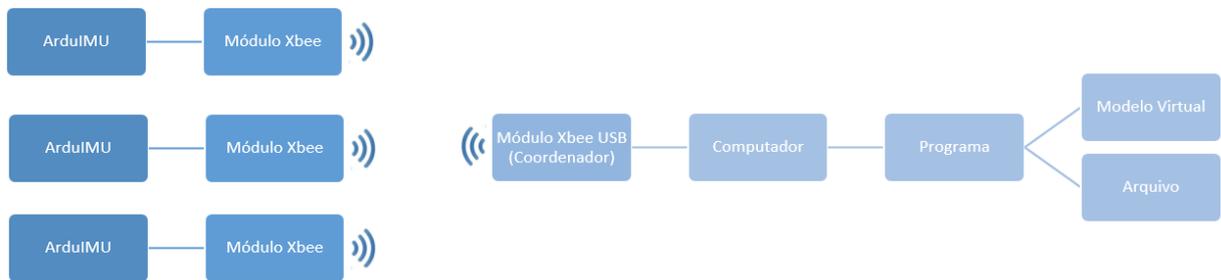


Figura 4.4 Fluxo do sistema.

4.2 REDE DE SENSORES

Com o objetivo de desenvolver um dispositivo de medição da angulação corporal que fosse compacta e que permitisse a livre movimentação do usuário durante o exame de marcha, trabalhou-se com o transceptor sem fio *Xbee* IEEE 802.15.4 em conjunto com uma placa comercial denominada ArduIMU v3. Com isso, pôde-se utilizar diversos dispositivos ao mesmo tempo, fazendo a leitura de vários segmentos do corpo, sem interferir na movimentação e deslocamento do usuário devido a fios e limitantes.

Para que todas as angulações fossem lidas e prevenindo perdas de pacotes conforme ocorreu em trabalhos anteriores (CORREA e BALBINOT, 2010), foi implementado um método de fusão de sensores para que as informações fossem validadas e a *attitude* determinada ainda no dispositivo (ArduIMU). Assim algoritmos e filtros que fazem o uso de integrações se tornam viáveis nessa aplicação, uma vez que não há perda de leitura dos sensores na transmissão, pois a informação é processada localmente. Os algoritmos foram implementados inicialmente no programa desenvolvido (no computador) para testes e o algoritmo com melhor resposta implementado no ArduIMU (algoritmo DCM – vide Capítulo 5). As descrições dos algoritmos de fusão utilizados foram apresentadas no Capítulo 3.

4.2.1 Hardware

4.2.1.1 ArduIMU V3

O ArduIMU V3 é uma unidade de medição inercial fabricada pela *3D-Robotics*. Faz parte do projeto ArduIMU, que é um projeto de código aberto (sobre licença CC) promovido pela *DIY Drones*, com o objetivo de fornecer um sistema inercial que pudesse servir como base para o desenvolvimento de sistemas de navegação em veículos aéreos não tripulados. O ArduIMU V3 possui um microcontrolador programável, Arduino Atmega 328 (16MHz), que faz a interface com as unidades de sensoriamento.

O ArduIMU V3 possui sensores do tipo MPU-6000 (Acelerômetro e Giroscópio) e do tipo HMC5883L (Magnetômetro). O HMC5883L possui as seguintes características (HONEYWELL, 2013):

- Sensor Magnetoresistivo de 3 eixos;
- ADC (Analog-to-Digital) 12-Bits;
- Protocolo I2C;
- Configuração digital de sensibilidade e resolução via protocolo I2C;
- Sensibilidade 1090 LSb/gauss (default);
- Resolução 0,92 mili-gauss (default);
- *Full Scale* -1.3/+1.3 gauss (default);
- Taxa de amostragem 75Hz.

Por sua vez, o MPU-6000 apresenta as seguintes características (INVENSENSE, 2013):

- Sensores MEM giroscópio de 3 eixos e acelerômetro de 3 eixos;
- ADC (Analog-to-Digital) 16-Bits;
- Protocolo I2C;
- Configuração digital de sensibilidade e resolução via protocolo I2C;

- Giroscópio:
 - Sensibilidade 16,4 LSb/(graus/s) (usado);
 - *Full Scale* -2000/+2000 graus/s (usado).
- Acelerometro:
 - Sensibilidade 4096 LSb/g (usado);
 - *Full Scale* -8/+8 g (usado).
- Taxa de amostragem 200Hz.

Como o microcontrolador Arduino Atmega 328 possui interface I2C, então toda a leitura e configuração dos sensores se dá através desse barramento. O microcontrolador ainda conta com uma interface serial facilitando assim a comunicação com o módulo Xbee. Essa mesma interface serial é usada para programação do microcontrolador.

O código de leitura dos sensores e de determinação da *attitude* pelo ArduIMU foi desenvolvido em linguagem C e será discutido na Seção 4.2.3.

4.2.1.2 Xbee

O módulo Xbee IEEE 802.15.4 da *Digi International*, como o próprio nome faz referência, é um dispositivo sem fio que trabalha sobre o protocolo IEEE 802.15.4, sendo capaz de criar uma rede WPAN (*Wireless Personal Area Network*). Esse dispositivo possui como principais características o baixo consumo, suporte a rede estrela, ponto-a-ponto ou *mesh*, comunicação serial UART, configurações para gerenciamento de energia, entradas/saídas digitais, conversores analógicos-digitais de 10 bits e PWM. Suas configurações podem ser visualizadas na Tabela 4.1.

Tabela 4.1 Especificações Xbee IEEE 802.15.4 (Digi International 2006).

Especificações Xbee	
Desempenho	
Alcance Urbano	30 metros
Alcance em Área Aberta	100 metros
Potência de Transmissão	1mW
Velocidade de Transmissão RF	250 000 bps
Velocidade de Transmissão Serial	1200 - 115 200 bps
Sensibilidade de Recepção	-92 dBm (1% taxa de erro de pacote)
Requisitos de Alimentação	
Tensão de alimentação	2.8 - 3.4 V
Consumo Típico	50 mA
Gerais	
Frequência de Operação	ISM 2.4 Ghz
Dimensões	2,438 cm x 2,761 cm
Temperatura de Operação	- 40 to 85° C
Rede e Segurança	
Topologia Suportada	Ponto-a-ponto, estrela e mesh
Numero de Canais	12 Canais
Opções de Endereçamento	PAN ID, Canal e Endereço
Criptografia	AES 128-bit

Outra característica relevante do módulo Xbee é a possibilidade de operar no modo API. O modo API baseia-se em quadros (*frames*), estendendo assim o nível para qual a aplicação *host* pode interagir com as capacidades de rede do módulo. Neste modo de operação, os dados transmitidos e recebidos estão contidos em quadros, os quais definem operações ou eventos dentro do módulo. Através desse modo de operação, é possível um determinado módulo enviar endereço fonte, endereço destino, nome de um determinado nó, sinal RSSI, estado, etc.

Os Xbees são configurados através da interface USB-Serial que é a mesma placa que faz a conexão do coordenador com o computador. Trata-se de um placa desenvolvida pela Rogercom denominada CON-USBBEE e pode ser visualizada na Figura 4.5. Este dispositivo, ao ser conectado na porta USB do computador, cria uma porta serial virtual, como se fosse

uma porta serial padrão RS232, possibilitando receber os dados do módulo. Esta interface permite a comunicação e a configuração dos parâmetros dos módulos Xbee através do software XCTU, distribuído gratuitamente pela *Digi International*.

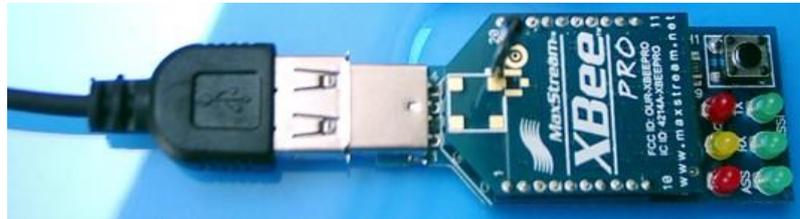


Figura 4.5 XBEE conectado a placa CON-USBBEE.

Fonte: ROGERCOM, 2009.

Os Xbee dos sensores, são também posicionados no corpo do voluntário junto à unidade de sensores. Esses módulos foram programados para enviar as informações para o módulo coordenador. O módulo coordenador, que também é um módulo Xbee, foi programado para receber as informações de todos os sensores e enviar para a porta serial os pacotes API recebidos.

4.2.1.3 Montagem dos dispositivos

Conforme referido, os módulos desenvolvidos que compõem a rede de sensores são compostos por um módulo Xbee, o ArduIMU e uma bateria de 9V. É um circuito simples para fins de otimização de hardware e para a miniaturização do mesmo, conforme mostrado na Figura 4.6. A tensão Vcc é composta pela bateria de 9V ligada na entrada do regulador de tensão presente na placa ArduIMU. A saída desse regulador gera tensões de 5V e 3,3V e alimenta os próprios componentes do ArduIMU, A saída de 3,3V alimenta o Xbee. Utilizou-se à interface serial (saída) do ArduIMU ligada a interface serial (entrada) do Xbee. Desse

modo, o transceptor encapsula os dados recebidos (leitura dos sensores e *Attitude*) em pacotes API e envia as informações para o coordenado.

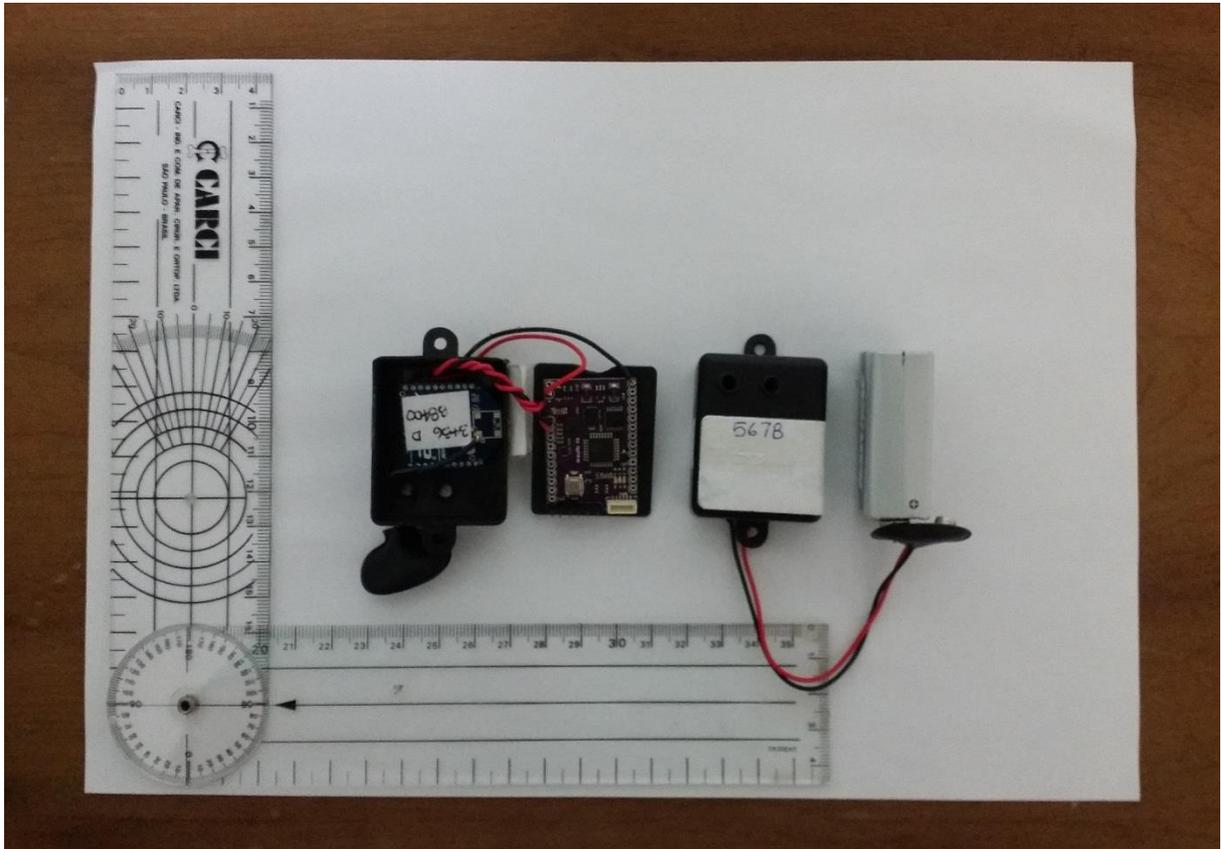


Figura 4.6 ArduIMU e XBee acoplados (esquerda) e encapsulamento final (direita).

Com o encapsulamento de plástico utilizado a dimensão do dispositivo ficou em 40x32x20mm. A Figura 4.7 apresenta os dispositivos posicionados no corpo do usuário.

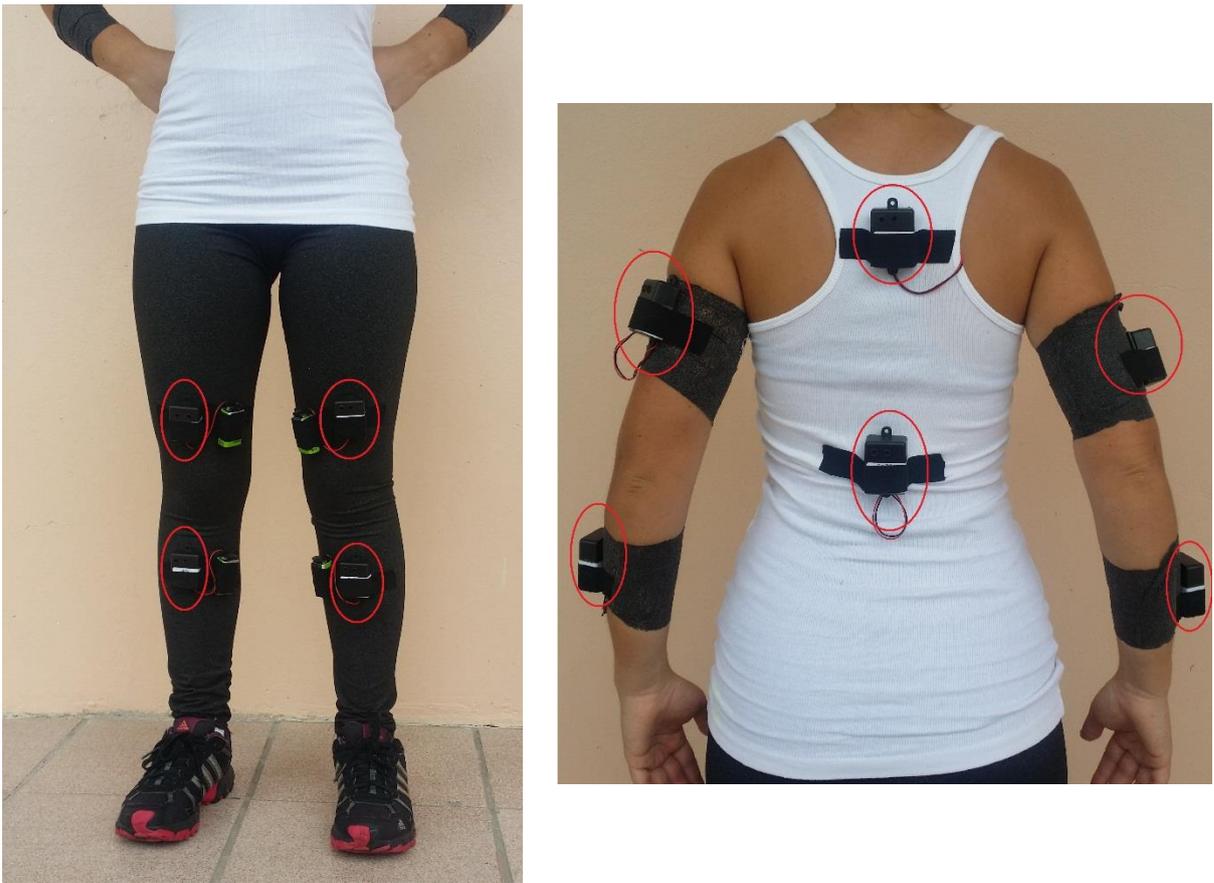


Figura 4.7 Módulos no corpo do usuário. Esquerda – parte inferior. Direita – parte superior sem baterias.

4.2.2 Calibração dos sensores

Como a determinação da *attitude* é derivada das informações dos sensores, através da fusão dos mesmos, o procedimento de calibração, assim como na grande maioria das aplicações de sistemas de instrumentação, deve ser bem realizado. O processo de calibração dos sensores da IMU foi baseado em um artigo de um fabricante de IMU comercial (MARCIAS *et al.*, 2012), onde é descrita a calibração do acelerômetro, do giroscópio e do magnetômetro.

Para calibração do acelerômetro, há o aproveitamento da leitura da aceleração da gravidade para correção de escala e do deslocamento do ponto zero (*offset*). O ArduIMU é posicionado de modo a cada eixo por vez estar alinhado com o eixo da gravidade. Desse

modo, o resultado deve ser 1 (uma) gravidade (9,807 m/s²) para cada eixo. Então o ajuste do *offset* e a escala são efetuados conforme as Equações (4.1) e (4.2), sendo executados separadamente para cada eixo do sensor.

$$valor_{offset} = \frac{valor_{max} + valor_{min}}{2} \quad (4.1)$$

$$valor_{escala} = \frac{valor_{esperado}}{valor_{max} - valor_{offset}} \quad (4.2)$$

onde, o $valor_{offset}$ é a correção do *offset*, $valor_{max}$ é o valor de uma gravidade no eixo no lado positivo, $valor_{min}$ é o valor de uma gravidade no eixo no lado negativo, $valor_{escala}$ é a correção a ser aplicada na escala e o $valor_{esperado}$ é o valor esperado para aquela leitura (1 gravidade).

Para execução dos ensaios e calibração do sistema foi desenvolvido um código em linguagem *Python* que recebe as leituras do acelerômetro do ArduIMU de sua interface serial e grava em um arquivo. Para isso, foi colocado um cabo que conecta a saída serial do ArduIMU para a USB do computador (o mesmo que é usado para programar o dispositivo). As leituras do acelerômetro foram gravadas em um arquivo e o ArduIMU foi posicionado de modo a aproveitar o máximo do alinhamento dos seus eixos com a gravidade. Após vários ensaios, foi determinado empiricamente separar 20% das maiores ou menores leituras e efetuar suas médias (para cada eixo). O resultado das médias representam o $valor_{max}$ ou $valor_{min}$ dos eixos nas Equações (4.1) e (4.2), conforme a orientação do acelerômetro. Com o resultado desse ensaio, o $valor_{offset}$ e o $valor_{escala}$ foram gravados no ArduIMU onde são utilizadas para correção das leituras, conforme a Equação (4.3):

$$valor_{calibrado} = valor_{escala} * (leitura_{eixo} - valor_{offset}) \quad (4.3)$$

onde o $valor_{calibrado}$ é o valor do a ser utilizado pelo sistema para o eixo lido, o $valor_{escala}$ é o valor determinado pela Equação (4.2) para o eixo lido, o $valor_{offset}$ é o valor determinado pela Equação (4.1) para o eixo lido e a $leitura_{eixo}$ é a leitura oriunda diretamente do sensor.

Para calibração do giroscópio, conforme artigo de referencia (MARCIAS *et al.*, 2012), foram adquiridas leituras da IMU parada. Desse modo a leitura obtida para cada eixo com o sistema estático representa o *offset* do sistema e foi usado na Equação (4.3) quando lido os valores do giroscópio. Ainda para a calibração do giroscópio foi usado um sensor de referência da marca XSens (XSENS, 2014). Notou-se que é aplicado um limiar mínimo para leitura desses sensores. Dessa maneira é evitado em algoritmos de determinação da *attitude* que utilizam da integração do giroscópio, que haja movimentação dos eixos por causa de ruídos. Após a realização de vários ensaios foi determinado um limiar mínimo e gravado juntamente com os valores de *offset* e escala.

Para calibração do magnetômetro, é tratada a distorção gerada pelo Soft-Iron e Hard-Iron. A calibração Hard-Iron consiste em remover a constante do efeito do campo magnético no sensor. Quando plotado, a saída do magnetômetro em um caso ideal, mostra uma esfera centrada no ponto de origem (0,0,0), mas que usualmente não ocorre. Normalmente, o centro da esfera tem outra posição deslocada da origem (0, 0, 0). Assim, através da Equação (4.1), pode-se remover esse *offset* e reposicionar o centro da esfera na origem. A calibração Soft-Iron consiste em remover os efeitos do campo magnético no sensor que causam o efeito elíptico na esfera esperada no caso ideal da saída do magnetômetro. Para correção do Soft-Iron em magnetômetros que possuem apenas 2 eixos (X e Y) utiliza-se as Equações (4.4) e (4.5).

$$x_{escala} = \frac{y_{m\acute{a}ximo} - y_{m\acute{i}nimo}}{x_{m\acute{a}ximo} - x_{m\acute{i}nimo}} \quad (4.4)$$

$$y_{escala} = \frac{x_{m\acute{a}ximo} - x_{m\acute{i}nimo}}{y_{m\acute{a}ximo} - y_{m\acute{i}nimo}} \quad (4.5)$$

Para facilitar o processo de calibraão em trs eixos (caso atual) foi desenvolvido por Yury Matselenak (2014) um software livre que consiste em determinar uma matriz de correão dos eixos do magnetmetro conforme a distorão do campo magntico. Para isso so feitas vrias leituras do magnetmetro em posiões diferentes e informado no programa que retorna a matriz de correão a ser aplicada conforme mostra a Figura 4.8 e Equaão (4.6).

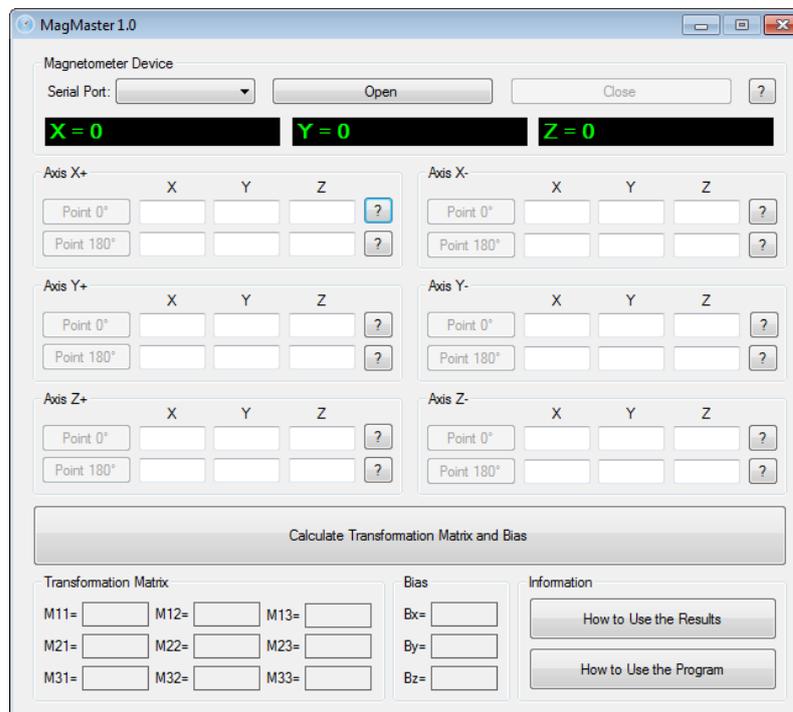


Figura 4.8 Tela de calibraão do magnetmetro – MagMaster (MATSELENAK, 2014).

$$\begin{bmatrix} x_{calibrado} \\ y_{calibrado} \\ z_{calibrado} \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} x \left(\begin{bmatrix} x_{sensor} \\ y_{sensor} \\ z_{sensor} \end{bmatrix} - \begin{bmatrix} B_{offset} \\ B_{offset} \\ B_{offset} \end{bmatrix} \right) \quad (4.6)$$

Para execução desse ensaio de calibração foi desenvolvido um código em linguagem *Python* que recebe os sinais do magnetômetro do ArduIMU em sua interface serial e grava em um arquivo. Novamente, foi colocado um cabo que adapta a saída serial do ArduIMU para uma porta USB do computador. Para o ensaio do magnetômetro, o sensor foi posicionado conforme orientação do software MagMaster com a finalidade de preencher o formulário do programa. Após o preenchimento de todos os campos, foi informada a matriz de rotação e offset para correção das leituras do magnetômetro que foram gravadas no ArduIMU.

As informações da calibração dos sensores foram gravados no ArduIMU e usadas para correção das leituras dos sensores em tempo de execução. Essa descrição será vista na Seção 4.2.3 (Programação do dispositivo).

4.2.3 Programação do dispositivo

Conforme já descrito no presente trabalho, o microcontrolador Arduino Atmega 328 (16MHz) usado no sistema é um dispositivo programável. Assim, para que todas as angulações do usuário fossem lidas e, prevendo perdas de pacotes conforme ocorrido em trabalhos anteriores (CORREA, 2010), foram implementados métodos de fusão de sensores para que as informações fossem validadas e a métrica *attitude* determinada na própria. Assim, algoritmos e filtros que fazem o uso de integrações se tornam viáveis nessa aplicação, uma vez que não há perda de leitura dos sensores pela transmissão, pois a informação é processada localmente.

Após escolher o algoritmo com melhor resposta ao sistema (algoritmo DCM – vide Capítulo 5) o mesmo foi implementado no ArduIMU. O fluxograma do código implementado no ArduIMU para o funcionamento do sistema é mostrado na Figura 4.9.

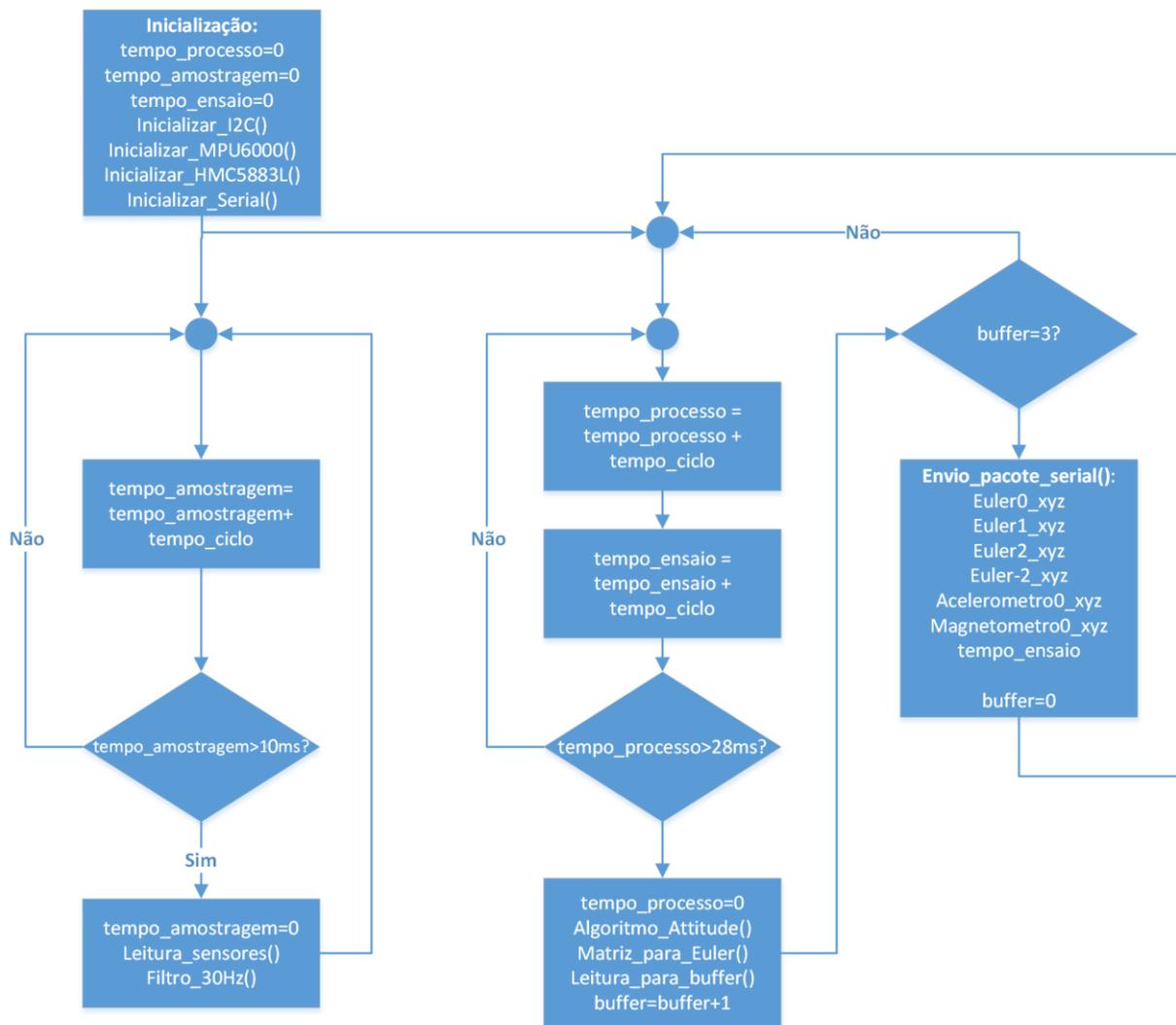


Figura 4.9 Fluxograma do algoritmo implementado na IMU.

As variáveis `tempo_amostragem`, `tempo_processo` e `tempo_ensaio` são responsáveis pelos controles temporais do sistema. O `tempo_amostragem` é responsável pelo controle das funções de leitura dos sensores e ajuste conforme Equações (4.3) e (4.6) – `Leitura_Sensores()`. Ainda nesse controle é aplicado um filtro *Butterworth* passa baixa de terceira ordem de 30Hz sobre as leituras efetuadas – `Filtro_30Hz()`. Sobre a frequência de corte, sabe-se que o tipo de IMU, a frequência de aquisição dos dados e a quantidade de eixos explorados dependem do foco do estudo, porém, geralmente são utilizados IMU triaxiais para localização espacial do membro com uma taxa de aquisição do dispositivo de 1,2 a 10 Hz para movimentos de

marcha simples (NAJAFI *et al.*, 2003). Outros estudos ainda sugerem que informações como angulações das articulações em marcha simples sejam filtradas em 6Hz (ARAÚJO *et al.*, 2004). Com base em estudos anteriores (CORRÊA; BALBINOT, 2013) e estudos mais atuais (MOHAMED, 2012; KUN, 2011) optou-se por utilizar um filtro com a frequência de corte mais elevada.

Outra variável de controle temporal é o tempo_processo. Ela é responsável pelo controle das principais funções do sistema como Algoritmo_Attitude(), Matriz_para_Euler(), Leitura_para_buffer(). Em Algoritmo_Attitude() é determinada a *attitude* através do algoritmo DCM, descrito na Seção 3.2 (DCM IMU). Essa *attitude* é convertida para variáveis de Euler XYZ para otimizar a transmissão da informação a posteriori – Matriz_para_Euler(). Após obter a *attitude* em Euler as variáveis e as leituras dos sensores são colocadas em um *buffer* para serem enviadas após o acúmulo de 3 (três) leituras. Essa quantidade foi levantada através de ensaios (Capítulo 5) e é determinada pelo tamanho do pacote em relação ao aproveitamento da rede Xbee. Após o acúmulo dessas leituras, o pacote é montado e enviado para porta Serial onde é recebido e reenviado pelo Xbee como um pacote API.

4.3 MODELO VIRTUAL

O modelo virtual consiste em um modelo computacional do corpo humano formado por pequenos segmentos conforme as principais estruturas ósseas do corpo e organizadas como tal. Esse modelo tem por finalidade uma representatividade mecânica do corpo humano e serve para o acompanhamento do movimento do paciente durante o exame. Isso permite que o usuário do sistema tenha uma realimentação visual do movimento e também na observação do profissional da área da saúde na realização do correspondente exame, fazendo-o se ater aos detalhes a serem avaliados.

O modelo é criado através dos softwares *MakeHuman* ou *Fuse*, e depois é importado para o *Blender Game Engine*, um software de animação gráfica e jogos, onde foram configurados ambientes e desenvolvidos rotinas de interação. Após adquirir e tratar as informações, o programa de ensaio envia as informações para o *Blender*, que, por sua vez, também trata as informações e movimenta o modelo conforme os sensores. O desenvolvimento das rotinas internas do *Blender* foram desenvolvidas na linguagem *Python*.

O *MakeHuman* é um programa de código aberto para criação de modelos computacionais humanos tridimensionais. Trabalha sobre a licença CC0 que dá ao usuário liberdade para criação e uso dos seus modos e modelos gerados. Foi desenvolvido pela comunidade mundial e tem por foco o seu uso para ilustrações, animações e jogos eletrônicos.

A criação do modelo se dá por uma interface simples e com muitos recursos de formatação do modelo, conforme Figura 4.10.

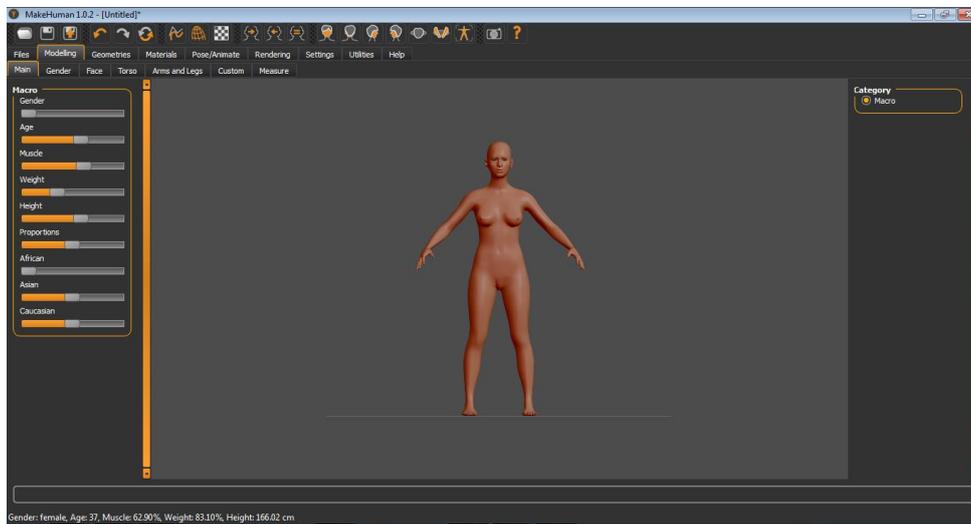


Figura 4.10 Interface *MakeHuman*.

Após o usuário completar a edição do modelo, ele deverá exportar para um formato compatível com o *Blender* (com a inclusão da estrutura do modelo). Nesse caso, o formato

Collada (.dae) – *collaborative design activity*. O exportação é feita através do menu do sistema, conforme Figura 4.11.



Figura 4.11 Interface *MakeHuman* de exportação de formato.

O *Fuse* é um produto da *Mixamo*, cuja finalidade também é a criação de modelos computacionais humanos ou humanoides tridimensionais. A modelagem é fácil e apresenta resultados bons com poucos minutos de trabalho – Figura 4.12 e Figura 4.13.

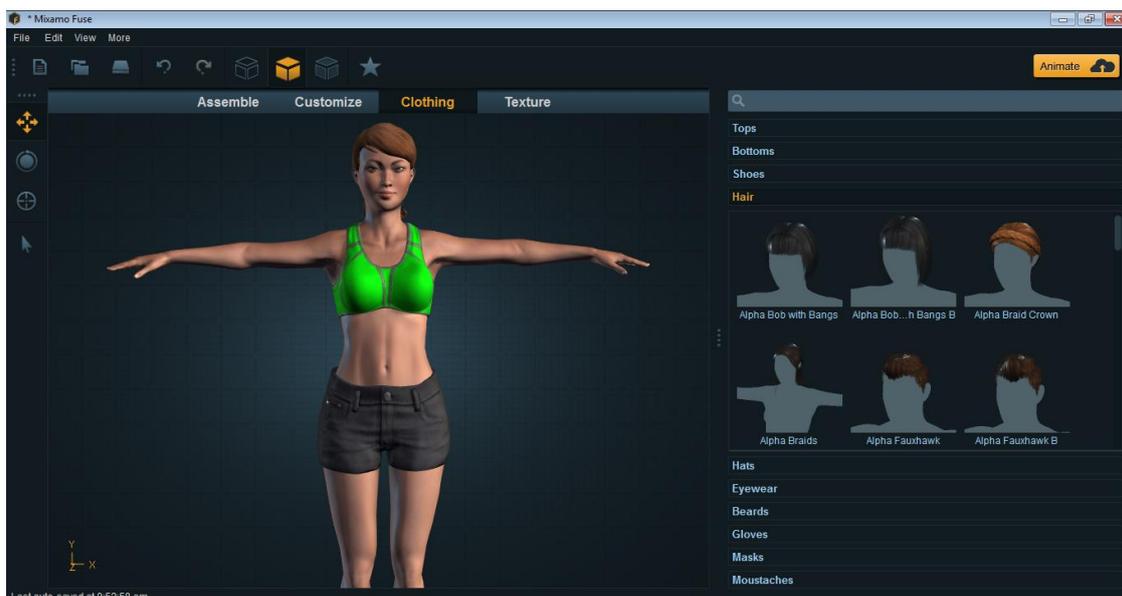


Figura 4.12 Interface *Fuse* da *Mixamo*.

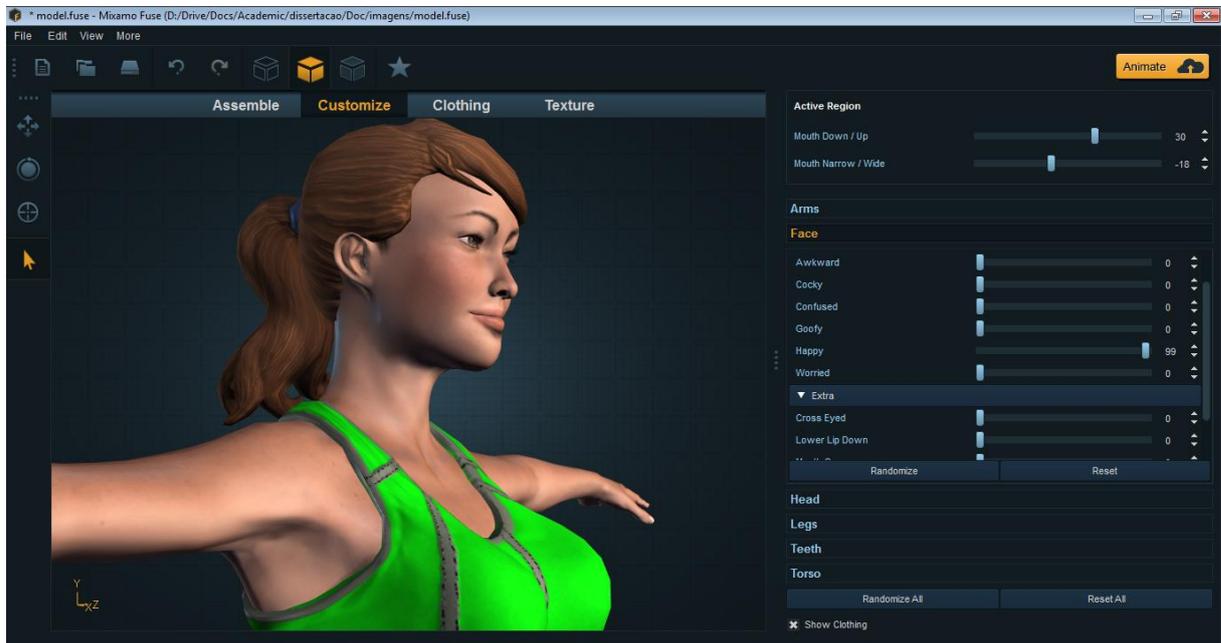


Figura 4.13 Interface *Fuse* da *Mixamo* – detalhe modelagem.

A exportação do *Fuse* (para *Collada*) se dá através do site do fabricante onde é colocada a estrutura no modelo responsável posteriormente pela movimentação e animação no *Blender*.

O *Blender* é um software de código aberto de modelagem computacional, suas aplicações vão desde criação de modelos computacionais complexos até produção de vídeos, animações e jogos. A proposta do *Blender* é ser um sistema de desenvolvimento gráfico grátis e aberto onde o usuário possa ter todos os recursos necessários para criação de animações complexas e jogos de computador. No *Blender* há uma ferramenta denominada *Blender Game Engine* onde o desenvolvedor pode criar seus códigos e interações que serão utilizadas nos jogos desenvolvidos. A linguagem de programação utilizada é o *Python* e os recursos são vastos dentro da interface do *Blender Game Engine*, como modelar a gravidade, movimentação das estruturas de modelos (no *Blender* denominado *Bones*) e modelagem de deformações por impactos.

A interface do *Blender Game Engine* é apresentada na Figura 4.14. Ele tem o recurso de importação do formato *Collada* onde o modelo já apresenta a estrutura interna (*Bones*). Cada parte dessa estrutura é responsável por uma articulação do modelo e pode ser movimentada por código em tempo real.

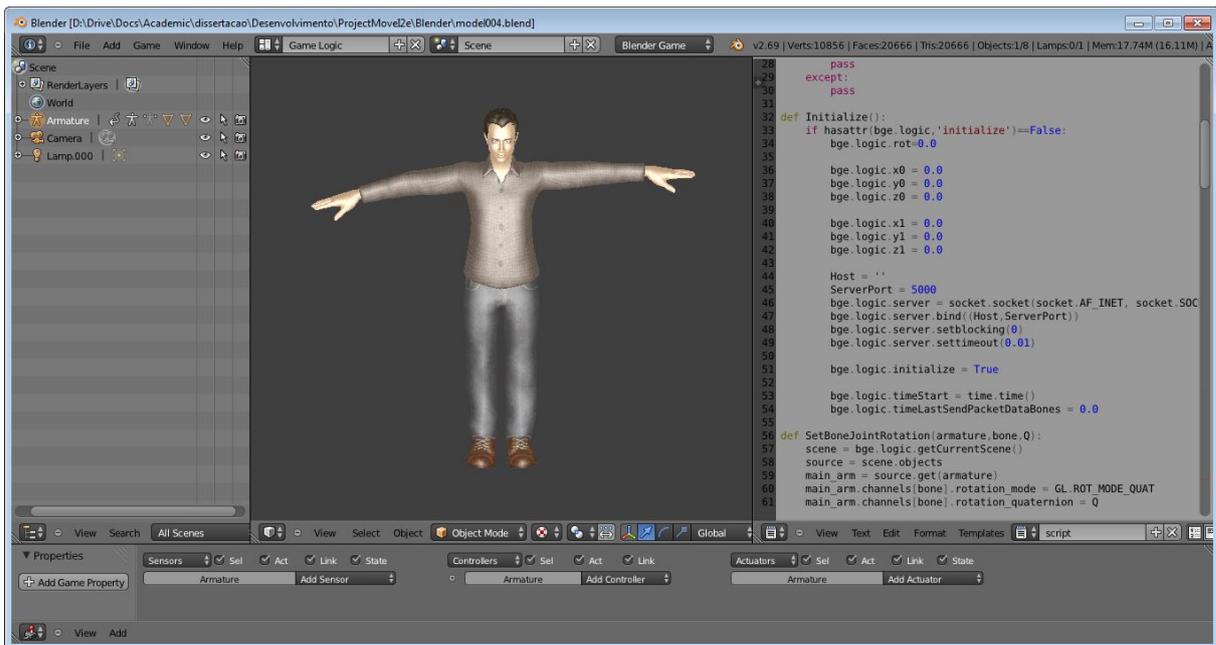


Figura 4.14 Interface *Blender Game Engine*.

Para movimentação em tempo de ensaio foi desenvolvido um código que recebe do Programa para Ensaio da Marcha (Capítulo 4.4) a *attitude* global do sensor por *Socket* (API genérica para programação sobre protocolos de comunicação). No *Blender Game Engine* é recebida a informação da *attitude* com o *Bone* alvo. O algoritmo desenvolvido então procura quem é o *Bone* pai (*Bone* que faz ligação de articulações com o *Bone* alvo) e solicita sua posição com o referencial global. Com isso é aplicada a Equação (4.7) para determinar a rotação do *Bone* alvo em relação à articulação do modelo e então é aplicada essa rotação nele para movimentação do membro no modelo virtual.

$$R_n^b = Rotação_{Bone_{Alvo}} = \left(Bone_{Pai_{global}} \right)^T \cdot Attitude_Sensor_{global} \quad (4.7)$$

Essas funções foram implementadas através do algoritmo desenvolvido em *Python* e pela API do *Blender Engine*. Por fim, tem-se o modelo virtual recebendo as informações oriundas do Programa para Ensaio da Marcha em tempo real. Como o programa de ensaio recebe as informações do sensor e transmite essas informações para o *Blender Engine*, tem-se a reprodução do modelo conforme a movimentação do usuário do sistema. A Figura 4.15 demonstra a interface do modelo vista pelo usuário final.

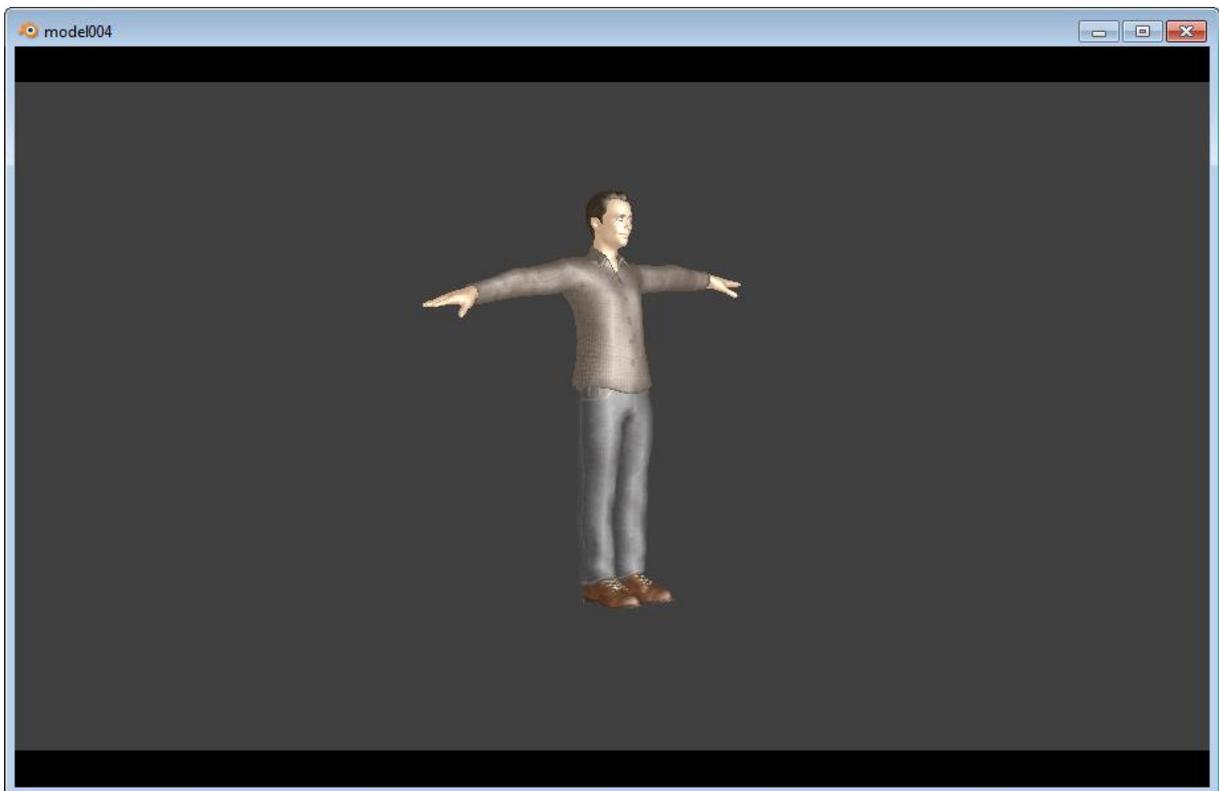


Figura 4.15 Interface *Blender* vista pelo usuário.

4.4 PROGRAMA PARA ENSAIO DA MARCHA

O sistema desenvolvido tem como complemento um programa, chamado Programa de Ensaio da Marcha, que recebe as informações de todos os sensores através da comunicação serial com o coordenador Xbee. Esse programa desencapsula as informações dos pacotes API

recebidos, envia as informações para o *Blender Game Engine* e grava as variáveis de interesse (angulações dos membros, aceleração, etc) em arquivos para fins de estudos mais detalhados da cinemática do usuário e para que seja possível reproduzir posteriormente o ensaio caso desejado.

O programa foi desenvolvido na linguagem de programação *Python* para haver uma maior proximidade entre o programa e *Blender Game Engine* – ambos em *Python*. O programa desenvolvido ainda permite através de seu arquivo de configuração, organizar o sistema internamente de modo que seja possível ler as informações do coordenador, de uma porta serial ou ainda de arquivos; além disso, ele permite selecionar quais sensores representam qual articulação do modelo virtual e também permite configurar qual modelo virtual será usado.

Para que o programa desenvolvido pudesse ser flexível e para poder tratar paralelamente todas as informações recebidas, o mesmo foi desenvolvido com várias *threads* independentes. *Thread* é uma forma de um processo ou programa dividir a si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente – a biblioteca que habilita trabalhar com *threads* em *Python* é a `Lib/threading.py`. Para comunicação entre as *threads* foram desenvolvidas filas de trabalhos de modo que todas as funções principais das *threads* tivessem um fila de entrada (tarefas) e uma fila de resultados. Desse modo, pode-se conectar umas às outras sem que fiquem presas a uma instrução ou aguardando um processo ser finalizado. O diagrama desse modo de trabalho está mostrado nas Figura 4.16 e Figura 4.17.

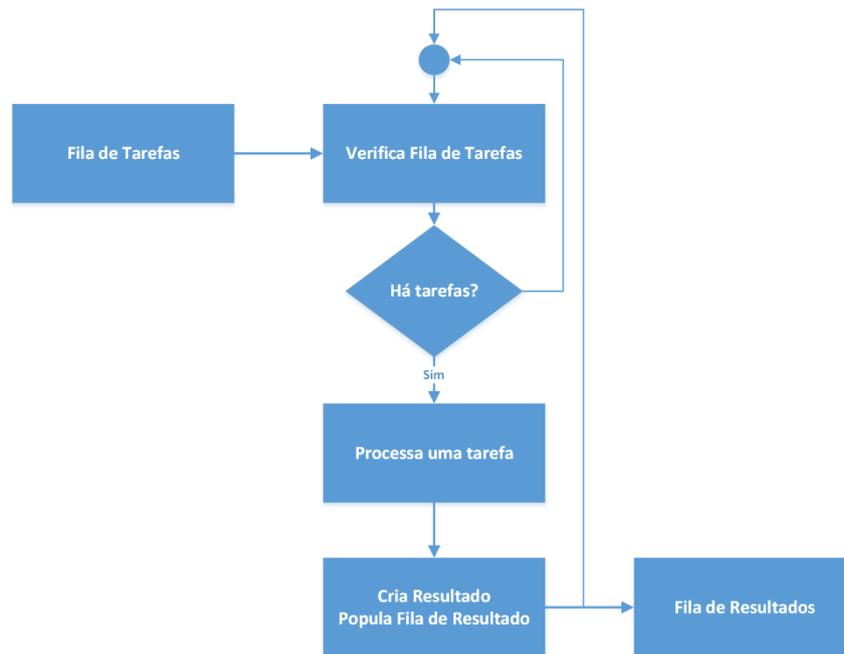


Figura 4.16 Diagrama de uma *thread* genérica do programa.



Figura 4.17 Comunicação entre *thread* do programa.

Assim em *threads* sequenciais, o resultado de um processo gera uma tarefa para a próxima ou para as próximas *threads*. Há *threads* com processos em que a fila de tarefas tem apenas uma tarefa constante, por exemplo ler porta serial, enquanto há *threads* que tem apenas um resultado na Fila de resultados; por exemplo, o registro escrito em arquivo. Assim, a fila de tarefas e a fila de resultado dependem da finalidade específica da função de processamento destinada aquela *thread*.

A Figura 4.18 apresenta as principais *threads* com suas respectivas funções no sistema.

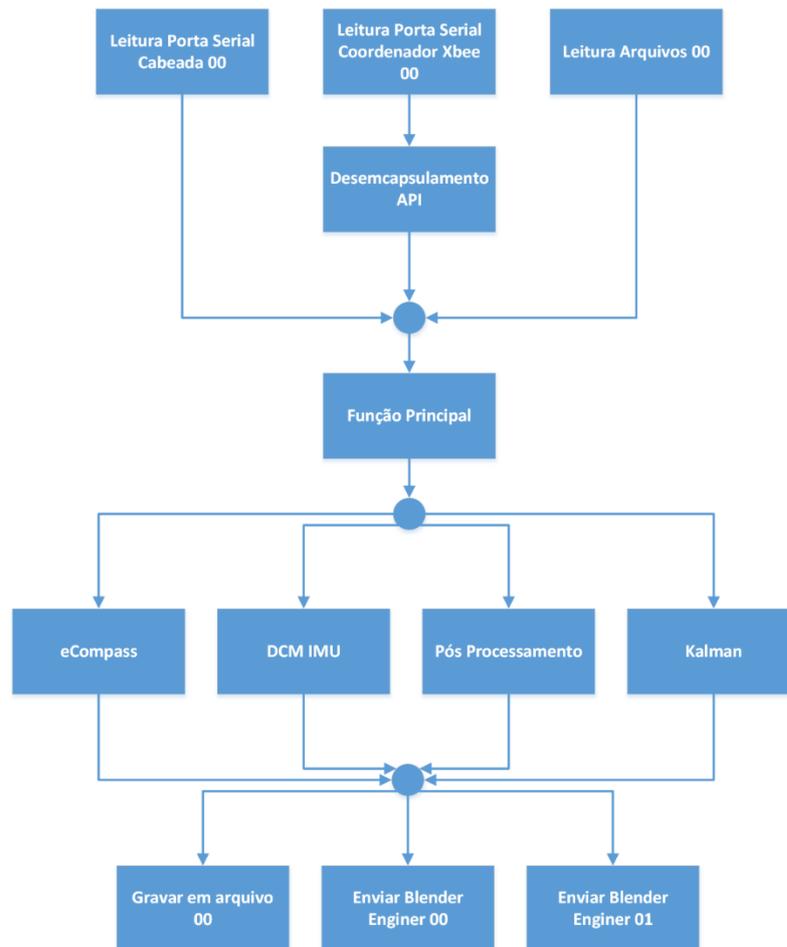


Figura 4.18 Diagrama do sistema de marcha.

Muitas das funções (cada uma com sua *thread*) demonstradas na Figura 4.18 podem ser habilitadas (ou não) conforme o arquivo de configuração que é lido na função principal, a qual carrega o arquivo de configuração elencando as possíveis entradas e saídas do sistema, sendo que é por ela que a entrada tem seu sentido determinado para saída. As possíveis entradas são a leitura direta da porta serial, a leitura através da porta serial com a comunicação com um Xbee coordenador ou então arquivos com as leituras de ensaios anteriores do sistema. As possibilidades de saída são envio de informações para o *Blender Enginer* através de *Sockets* e gravação em arquivos, sendo que ainda pode-se indicar o tipo de processamento que a leitura dos sensores recebidos pode ter, como, por exemplo, os algoritmos de fusão de sensores (Capítulo 3) ou algum tratamento de Pós Processamento

(filtro, organização das informações, etc.). Ressalva-se que foi através desse programa que também foram testados os tipos de processamentos (Capítulo 3) para então implementar o algoritmo no ArduIMU.

A leitura da porta serial serve para comunicação direta com o ArduIMU através de sua interface serial. Dessa maneira, é possível testar o funcionamento dos sensores sem o uso da rede sem fio. Essa entrada é muito útil para ensaios rápidos e para evitar perda de leituras dos sensores.

A leitura das informações do coordenador Xbee através da porta serial é o modo comum do sistema operar. É através dessa interface que o sistema recebe os pacotes API derivados dos Xbee unidos aos sensores. As informações recebidas contêm o remetente, a *attitude* e a leitura dos sensores. Após a aquisição o pacote é descapsulado e as informações são extraídas para continuação da operação do sistema. O pacote API é esquematizado na Figura 4.19.

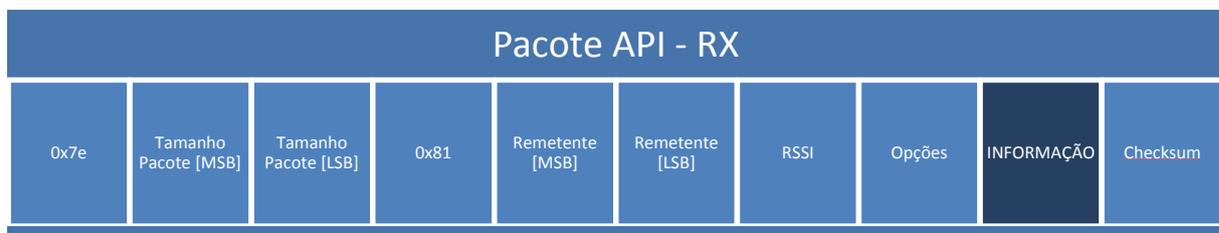


Figura 4.19 Pacote API.

Outra entrada de informação do sistema é através de arquivos. Pode-se selecionar um arquivo de um ensaio anterior para servir como entrada de dados. Dessa maneira pode-se reproduzir ensaios anteriores como se houvesse um usuário fazendo a movimentação.

Para saída há o envio de um pacote de informações por *socket* para o *Blender Engine*. Nesse pacote de informações é enviada a *attitude* do sensor e o segmento (*bone*) de destino no

modelo, conforme descrito no Capítulo 4.3. No *Blender Engine* a *attitude* é recebida, convertida para a informação local do segmento alvo do modelo e reproduzida.

Para complementar o sistema, toda informação gera um arquivo que pode ser utilizado para reproduzir novamente o ensaio sem o usuário/paciente. Através desse arquivo, é possível também estudar as leituras do sensores e *attitude* para ter os registros das variáveis cinemáticas da marcha do usuário/paciente para estudos e acompanhamento mais detalhado de seu ensaio, visando análises, recuperação e/ou tratamento.

O sistema foi desenvolvido utilizando a técnica de programação TDD (*Test Driven Development*). TDD ou Desenvolvimento orientado a testes é uma técnica de desenvolvimento de software que baseia em um ciclo curto de repetições: Primeiramente o desenvolvedor escreve um caso de teste automatizado que define uma nova funcionalidade. Então, é produzido código que possa ser validado pelo teste. O TDD foi criado por Kent Beck (2003). Como o sistema foi desenvolvido usando TDD, todas as funções podem ser testadas separadamente (teste unitário) para fins de depuração e compreensão do sistema.

5 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os resultados obtidos, bem como as discussões sobre os mesmos. Primeiramente será estudado o comportamento dos diferentes métodos de fusão de sensores utilizados, seguindo com a validação das medidas através de ensaios estático e dinâmico e comparação dos resultados com os de um sistema comercial. Após, serão expostos os ensaios preliminares realizados para levantar as características e limitações do sistema desenvolvido. Por fim, serão apresentados os resultados do exame de marcha realizados com alguns voluntários, sendo também seus resultados comparados com os da bibliografia.

5.1 ENSAIOS PRELIMINARES DE FUSÃO DOS SENSORES

Com o objetivo de escolher qual método de fusão se enquadra melhor o sistema, foram ensaiados os diferentes métodos, comparando os resultados entre si (presente Capítulo) e em com um sistema comercial (Capítulo 5.4). Durante todo o desenvolvimento do trabalho, usou-se o recurso de interface do sistema juntamente com o *Blender Engine*. Desse modo foi possível avaliar visualmente o comportamento dos algoritmos visualmente e fazer diagnósticos e melhorias durante o período de desenvolvimento. Para visualizar o comportamento da unidade inercial desenvolvida, a métrica *attitude* foi mostrado através de um cubo que deveria se comportar conforme a movimentação do sensor. A Figura 5.1 apresenta três cubos se movimentando conforme a resposta do sensor, sendo que cada cubo representa o comportamento de cada método de fusão desenvolvido (filtro de Kalman, *eCompass* e DCM IMU).

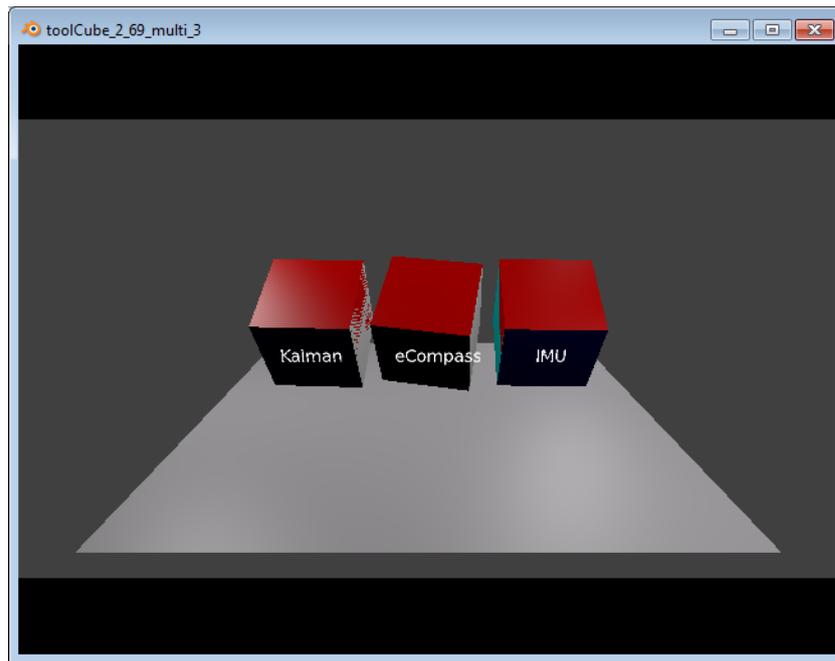


Figura 5.1 Ferramenta gráfica desenvolvida para apresentar o comportamento da métrica attitude.

Para uma análise quantitativa, plotava-se gráficos da *attitude* em Euler XYZ e através das comparações dos pontos, era possível analisar a semelhança de resultados entre algoritmos utilizados. Como exemplo, a Figura 5.2 mostra os resultados de um movimento aleatório com um tempo de execução de aproximadamente 25 segundos.

Visando avaliar o erro de medida do sistema foi usado como referência um goniômetro comercial (Goniômetro pequeno, resolução 2°, CARCI São Paulo). A palavra goniometria vem do grego e é derivada de duas palavras: Gonio que significa ângulo e Metria que quer dizer medida, sendo assim, medida do ângulo. Esse termo é amplamente aplicado em fisioterapia e fisioterapia para quantificar e avaliar o movimento de uma articulação. O instrumento utilizado para tal função é o goniômetro universal que é formado por dois braços e um eixo. Um braço acompanha o movimento enquanto o outro permanece fixo com a medida, o eixo deve ficar sobre a articulação avaliada. Seu modelo mais típico é apresentado na Figura 5.3.



Figura 5.2 Gráficos representando a métrica *attitude* em Euler XYZ com os três métodos: DCM, *eCOMPASS* e Kalman.

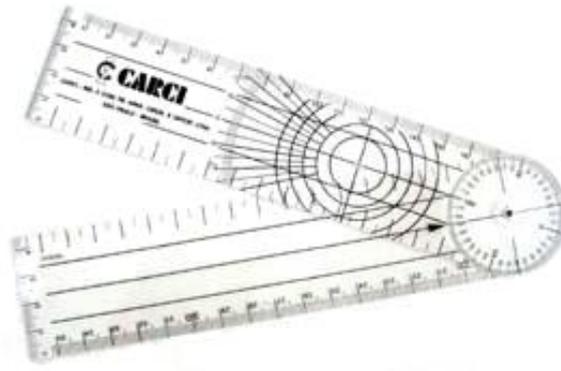


Figura 5.3 Goniômetro.

Para aferir a medida do sistema, um módulo foi fixado com uma fita dupla face no braço do goniômetro que fica livre (Figura 5.4) e o conjunto foi fixado em uma mesa para poder realizar algumas medidas do sistema. Os ensaios foram executados variando a angulação no sentido do braço, podendo assim ser medidos no goniômetro. Salienta-se que o objetivo dos ensaios apresentados neste capítulo é verificar o comportamento dos algoritmos, sendo que as avaliações dos erros de medida seguirão nos capítulos seguintes.

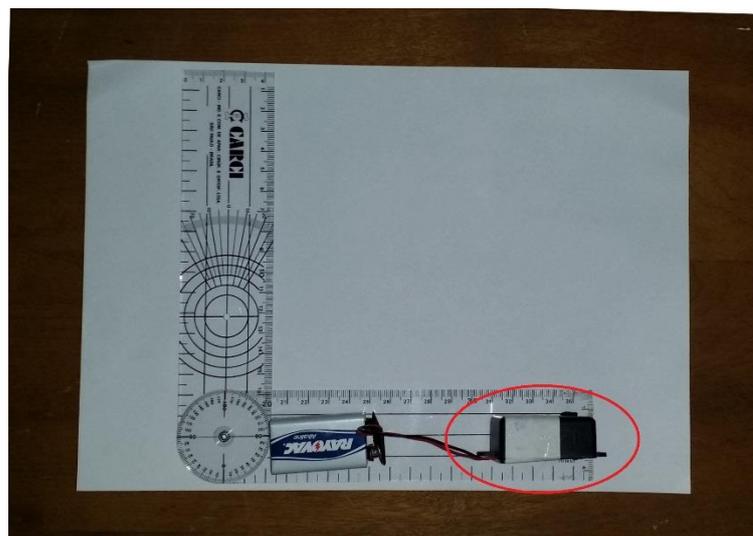


Figura 5.4 Módulo do sistema fixado em um goniômetro.

Durante os ensaios de curta duração (menores do que 5 minutos) verificou-se visualmente que o algoritmo DCM IMU comportava-se melhor que os demais (*eCompass* e Kalman), ou seja, ele seguia melhor a movimentação medida no goniômetro. Em diversos ensaios, o algoritmo de *eCompass* apresentava divergências na transição dos eixos, o que fazia com que o cubo tivesse uma rotação errônea. Como o algoritmo do filtro de Kalman usa o algoritmo *eCompass* para determinar a *attitude* medida ele também era afetado por essa transição, mas este algoritmo compensava rapidamente usando a *attitude* estimada oriunda do giroscópio.

A Figura 5.5 apresenta o resultado de um ensaio de 15 segundos. Nesse ensaio, foi executado um movimento com o goniômetro chegando a uma inclinação de 120° em Roll, sendo que tentou-se manter *Pitch* e *Yaw* constantes (próximos de 0° - fixos no goniômetro). Durante o ensaio, o cubo que estava sendo controlado pelo algoritmo *eCompass* rotacionou em *Yaw* sem que houvesse sido executado esse movimento. O cubo que estava representando o algoritmo de Kalman teve uma suave inclinação e retornou ao movimento normal, o qual o cubo com o algoritmo DCM reproduzia adequadamente. A plotagem desse ensaio representado na Figura 5.5 reproduz o que foi verificado visualmente no tempo compreendido entre 1424 e 6128ms.

O algoritmo *eCompass* trabalha com funções senocossenos, assim qualquer atraso ou não ortogonalidade dos eixos faz com que a rotação interprete a posição erroneamente. Mesmo com uma calibração cuidadosa, esse efeito foi observado, sendo mais perceptível nas rotações que havia inversão do sinal dos eixos. Todavia, esse algoritmo apresenta uma vantagem significativa ao longo do tempo, uma vez que ele não é dependente de integrações das suas variáveis, ou seja, um erro da *attitude* em uma amostra, não interfere na próxima, diferindo do método DCM IMU.

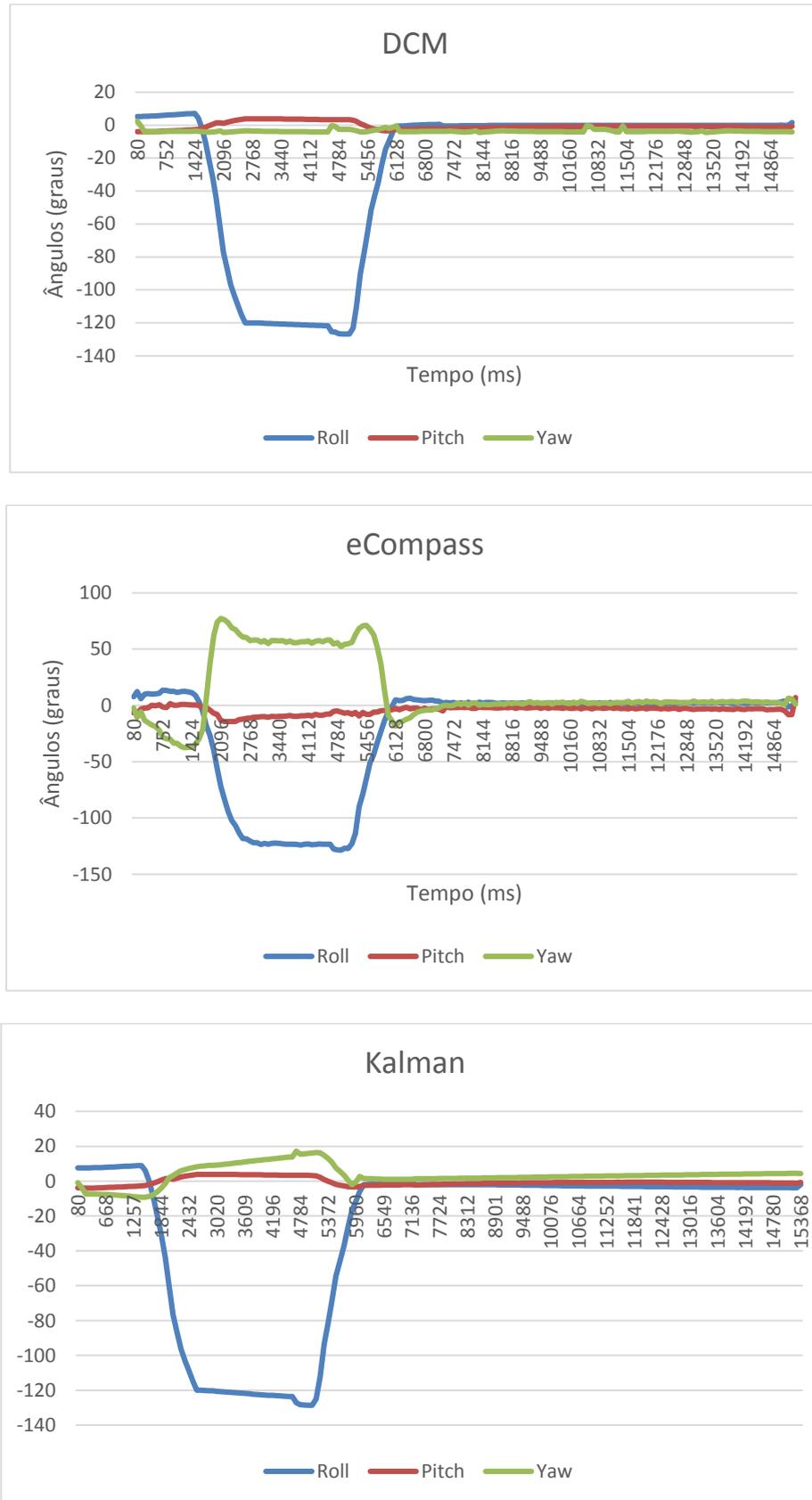


Figura 5.5 Gráficos representando a métrica *attitude* em Euler XYZ com os três métodos utilizados (DCM, *eCOMPASS* e Kalman) contendo divergência entre eles.

O algoritmo DCM comportou-se muito bem para ensaios de curta duração. Todavia, ensaios de longa duração (maiores que 5 minutos) faziam com que ele divergisse da *attitude* medida no goniômetro. A rotação era mais visível em *yaw*, que acumulava erros das leituras anteriores e divergia da movimentação corrente do sensor (medida no goniômetro). Como a base do algoritmo é a função de integração do giroscópio, os erros acabam sendo acumulados ao longo do tempo, prejudicando a resposta do sensor mesmo com as correções existentes. Para ensaios de curta duração, o resultado era mais sensível às rotações (mesmo rápidas), fazendo com que a leitura do sensor ficasse semelhante à do goniômetro, o que não necessariamente ocorria com os algoritmos de eCompass e o Kalman.

Apoiando-se na integração dos giroscópios e nas leituras do magnetômetro e do acelerômetro, o filtro de Kalman apresenta comportamento intermediário entre os dos DCM e eCompass. Sua *attitude*, em caso da transição dos eixos, não apresenta divergências tão frequentes quanto no eCompass e seu uso em ensaios por um período longo é corrigido com mais facilidade do que o DCM. Todavia, o mesmo acaba não tendo uma resposta tão próxima da leitura do goniômetro em movimentos curtos e rápidos como o DCM. Também não tem uma correção responsiva (mais rápida) quando está divergindo da *attitude* corrente do sensor como o eCompass, como mostra a Figura 5.5.

Analisando os algoritmos com base na aplicação, por ter uma grande representatividade visual (apresentação no modelo virtual) e por necessitar ter uma leitura mais próxima da real durante todo o período do ensaio (para análise posterior para estudo da marcha), verifica-se, com base nas argumentações anteriores, que para esse conjunto estudado não foi encontrado um algoritmo ideal para essa aplicação. Então, analisando cada algoritmo novamente, conclui-se que o melhor algoritmo para ser usado seria o DCM uma vez que ele possui uma resposta em curto período melhor que os demais (apresentação quantitativa nos capítulos posteriores) e

verificou-se, ainda, que há um modo de corrigir sua *attitude* um pós processamento em tempo de ensaio que será descrito a seguir.

Sabe-se que a marcha humana possui muitos movimentos repetitivos e de paradas de alguns membros (Capítulo 2.2), assim caso seja possível identificar esses momentos e sabendo qual a *attitude* deveria ser lida nessas paradas, é possível corrigir as leituras efetuadas. Dessa maneira, foi desenvolvido um algoritmo para detectar e corrigir a leitura DCM, o qual foi implementado na sessão de pós-processamento do diagrama do sistema - Figura 4.18. O diagrama do algoritmo desenvolvido é apresentado na Figura 5.6.

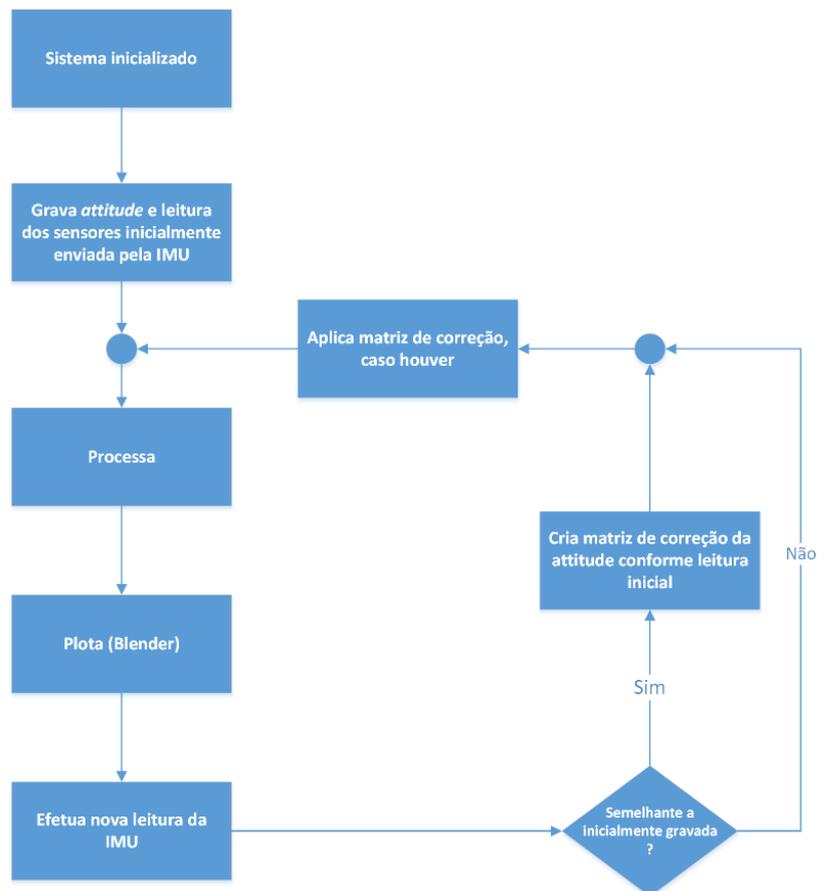


Figura 5.6 Algoritmo para identificação e recuperação da *attitude* corrente.

Nesse algoritmo são utilizados os dados iniciais da IMU (*attitude* inicial e leitura do acelerômetro e magnetômetro) para verificar se o mesmo estado se repete durante a marcha.

Caso a leitura atual seja semelhante à leitura inicial do acelerômetro e do magnetômetro, ou seja, próxima a leitura com um limiar, o sistema gera uma matriz de correção que transforma a *attitude* atual (possivelmente divergente) na *attitude* inicial. Essa correção expressa por meio da Equação (5.1) e é aplicada às futuras *attitudes* enviadas pela IMU durante o ensaio (em tempo de execução do programa). O limiar foi escolhido através de diversos ensaios e varia conforme o membro medido e da maneira como é realizado o ensaio, tipicamente seu valor não ultrapassa 4% do valor médio da leitura do sensor.

$$C_{\text{correção}} = C_{\text{attitude inicial}} \cdot (C_{\text{attitude atual}})^T \quad (5.1)$$

Desse modo, é possível corrigir e atualizar a *attitude* gerada pela DCM com respeito a eventuais erros acumulados pelo seu algoritmo.

Ainda assim, também é possível aplicar a filtragem de Kalman sobre a *attitude* calculada por DCM enviada pela IMU. Para isso, considera-se que a *attitude* gerada pelo método DCM determina o modelo de transição de estados (A) em Kalman. Inicialmente, transcreve-se a *attitude* DCM do ciclo anterior (no primeiro ciclo uma matriz identidade) e a *attitude* DCM atual para forma de matrizes quaternárias 4x4 (como a Equação (2.15) ou (2.16)). Então, utilizando essas matrizes resultantes, aplica-se a Equação (5.2), encontrando o modelo de transição de estados (A).

$$A = q_{DCM(t)} \cdot (q_{DCM(t-\Delta t)})^T \quad (5.2)$$

Dessa maneira, tem-se a matriz de rotação que informa a variação da rotação do estado anterior para o estado atual, da mesma forma que a informação do giroscópio fazia originalmente no algoritmo de Kalman. Após essa etapa, segue-se o algoritmo do filtro de

Kalman conforme descrito. Com esse método, pode-se utilizar o algoritmo DCM no Arduimu, tendo a opção de corrigir a informação utilizando Filtro de Kalman no pós-processamento.

Por fim, através da análise dos algoritmos de fusão de sensores e das adaptações apresentadas, optar-se por utilizar o algoritmo DCM IMU corrigindo em pós-processamento possíveis desvios que podem ocorrer durante o ensaio em tempo de execução. Ainda, para validar esse desenvolvimento, foram executados ensaios estáticos e dinâmicos para avaliar o erro de medida do sistema, conforme descrito nos próximos capítulos.

5.2 ENSAIO ESTÁTICO

Visando a avaliar o erro de medida estática do sistema foi usado como referência o goniômetro comercial (Goniômetro pequeno, resolução 2°, CARCI São Paulo). Trata-se de um instrumento barato, fácil de manusear e de uso ágil para fins de medida angular das articulações (MARQUES, 1997). Para que fosse possível avaliar o erro de medida do sistema para uma avaliação estática, fez-se o uso desse instrumento. O ensaio foi reproduzido para o sistema onde manteve-se o goniômetro com orientação fixa e posicionou-se a unidade inercial na parte móvel do instrumento para efetuar a medição dos ângulos. Foram realizadas vários ensaios para cada ângulo escolhido (de forma aleatório) sendo 4 ensaios para cada ângulo com uma média de 1000 leituras por ensaio. Os resultados para os eixos encontram-se na s Tabela 5.1 (eixo x), Tabela 5.2 (eixo y) e Tabela 5.3 (eixo z), sendo o algoritmo usado o DCM.

Tabela 5.1 Resultado para ensaio estático com goniômetro e sistema desenvolvido (eixo x).

Inclinação no goniômetro (graus)	Erro médio do ângulo (graus)	Desvio padrão (graus)	Erro máximo do ângulo lido (graus)
120	0,8	0,13	0,9
90	0,0	0,11	0,7
60	0,8	0,08	0,9
30	0,1	0,08	0,5
0	0,7	0,19	0,9
-30	0,4	0,04	0,6
-60	0,2	0,22	0,8
-90	0,8	0,20	1,0
-120	0,5	0,54	1,0

Tabela 5.2 Resultado para ensaio estático com goniômetro e sistema desenvolvido (eixo y).

Inclinação no goniômetro (graus)	Erro médio do ângulo (graus)	Desvio padrão (graus)	Erro máximo do ângulo lido (graus)
120	0,8	0,04	0,9
90	0,7	0,15	1,0
60	0,3	0,07	0,5
30	0,0	0,01	0,0
0	0,1	0,04	0,2
-30	0,2	0,05	0,3
-60	0,4	0,04	0,4
-90	0,6	0,03	0,8
-120	0,4	0,04	0,8

Tabela 5.3 Resultado para ensaio estático com goniômetro e sistema desenvolvido (eixo z).

Inclinação no goniômetro (graus)	Erro médio do ângulo (graus)	Desvio padrão (graus)	Erro máximo do ângulo lido (graus)
120	2,5	0,90	3,9
90	2,4	0,94	3,9
60	2,0	0,98	3,6
30	1,4	0,37	2,1
0	0,3	0,50	1,0
-30	2,2	0,90	3,7
-60	1,8	0,71	2,9
-90	2,1	0,83	3,4
-120	2,7	0,99	4,2

Através desse ensaio pode-se concluir que o presente sistema é uma boa alternativa para medição de goniometria tendo em vista que há muitas fontes de erro provenientes do instrumento utilizado – o goniômetro universal (MARQUES, 1997). Todavia a resolução do goniômetro não permite um levantamento do erro do sistema de forma mais significativa (erros menores de um grau não podem ser considerados), sendo necessário outros ensaios com outros equipamentos de melhor resolução para essa medição. Dessa maneira, para melhor caracterizar o sistema, foram executados ensaios com um sensor comercial que serão apresentados no Capítulo 5.4.

5.3 ENSAIOS PARA AVALIAÇÃO DE PERDA DE PACOTES E DA AUTONOMIA DO SISTEMA

Os ensaios preliminares foram testes do sistema para verificar as limitações do sistema. Entre os testes realizados para esse fim tem-se: testes de perdas de pacotes pelo Xbee e testes de autonomia do sistema, cujas descrições são apresentadas a seguir.

5.3.1 Testes de perda de pacotes

Os testes de perdas de pacotes servem para determinar a quantidade de pacotes API enviadas pelo Xbee que, por algum motivo, não chegam a ser recebidas pelo programa. Considerando que a taxa de amostragem é determinada no dispositivo (ArduIMU) e transmitida pelo Xbee, o sistema deve receber o mesmo número de amostras que foi enviada.

Para execução desse ensaio, foi montada uma rede de dispositivos (ArduIMU e Xbee) onde houve alterações da quantidade de dispositivos, do tamanho dos pacotes, da taxa de informação enviada e da configuração da rede. Foi realizado uma bateria de 4 testes para cada configuração, sendo o período de ensaio de 120 segundos. Os resultados são apresentados na Tabela 5.4. Cabe ressaltar que o resultado utilizado na análise foi fornecido pelo Xbee, o qual corresponde o pior desempenho por ensaio. Sendo, por exemplo, a taxa média de pacotes

perdidos tomada como a média dos resultados dos 4 testes, onde foi selecionada a informação do Xbee que teve o pior resultado por teste.

Tabela 5.4 Resultado dos ensaios de perda de pacotes.

Taxa de pacotes enviada (pacotes/segundo)	Tamanho do pacote (bytes)	Número de dispositivos (unidades)	Taxa média de pacotes perdidos (pacotes/segundo)	Desvio padrão da taxa de pacotes perdidos	Taxa máxima encontrada de pacotes perdidos (pacotes/segundo)
50	32	2	5,9	3,9	9,4
30	30	2	0,4	0,9	2,1
30	28	9	20,4	3,0	22,9
16	40	9	4,3	1,5	5,8
12	40	9	0,5	0,1	0,7
12	40	4/5*	0,6	0,2	0,9
12	40	6	0,4	0,1	0,5
10	76	2	0,1	0,2	0,4
10	76	9	1,7	0,6	2,6
10	58	9	1,0	0,4	1,4
10	58	4/5*	1,2	0,4	1,9

O tamanho do pacote foi determinado pelo o tamanho de informações de leitura dos sensores e *attitude* enviada, sendo considerado o envio de várias amostras no mesmo pacote. Em alguns caso (grifados com * na Tabela 5.4) criou-se duas redes (*ID Channels* diferentes) separadas de Xbee, cada uma com um coordenador para recebimento.

Como resultado dos ensaios foi escolhido uma taxa de envio de 12 pacotes por segundo contendo 40 bytes de informações. Uma observação interessante é que não houve ganho na separação em duas redes.

Considerando que com 12 pacotes obteve-se uma percepção visual satisfatória para plotagem do modelo virtual em tempo de ensaio (não havendo a percepção de falhas na movimentação) e que com dos 40 bytes existe a possibilidade de trabalhar com 3 amostras de *attitude* sequenciais (36 Hz), o sistema atende aos requisitos da proposta inicial de trabalho.

Os 40 bytes de informações permitem diversas possibilidades de configurações que o sistema pode contemplar. Considerando que cada leitura do sensor em 3 eixos, incluindo a *attitude* representada em ângulos de Euler em 3 eixos, tem 6 bytes de informação e que o tempo de referência interno (*tempo_ensaio*) tem 4 bytes, pode-se configurar o sistema para trabalhar com 2 amostras de cada sensor (*attitude*, acelerômetro, magnetômetro) e mais o tempo de referência. Desse modo, teria-se uma amostragem de 24Hz por sensor (*Euler0_XYZ*, *acelerometro0_XYZ*, *magnetometro0_XYZ*, *Euler1_XYZ*, *acelerometro1_XYZ*, *magnetometro1_XYZ*, *tempo_ensaio*). Nota-se que a leitura do giroscópio não precisa ser enviada, uma vez que pode ser obtida por meio da *attitude* DCM.

No presente trabalho utilizou-se a seguinte sequência de leituras: *Euler0_XYZ*, *Euler1_XYZ*, *Euler2_XYZ*, *Euler-2_XYZ*, *acelerometro0_XYZ*, *magnetometro0_XYZ*, *tempo_ensaio*, onde o dado *Euler-2_XYZ* representa a leitura da *attitude* no pacote anterior. Dessa maneira, em caso de perda de um pacote, ainda, haveria possibilidade de usar a informação do pacote passado para a leitura não ficar muito espaçada. Esse pacote é demonstrado no Capítulo 4.2.3 na Figura 4.9.

5.3.2 Testes de autonomia do sistema

Avaliou-se também a autonomia da bateria e seu consumo com o objetivo de determinar quanto tempo a bateria colocada no módulo permite o correto funcionamento do sistema. Para realização deste ensaio foram adquiridas 4 baterias novas de 9V de 520mAh (Panasonic 6LR61) e colocadas em 4 dispositivos desenvolvidos. Os módulos foram ligados e o programa de aquisição ficou processando e realizando as leituras constantemente no decorrer do ensaio.

Até o último módulo se desligar sozinho, o ensaio durou 259,53 minutos (4 horas e 19 minutos). Os resultados estão apresentados na Tabela 5.5.

Tabela 5.5 Resultado para ensaio de autonomia do sistema.

Bateria	Endereço Xbee	Tempo de duração (ms)	Leituras (unidades)
1	0x3456	15537540	179905
2	0x5678	15571896	178236
3	0x6789	14766588	169506
4	0x9012	15482456	174049

Verificou-se que a intensidade de corrente média consumida é de 88,5mA para o módulo transmitindo e 87,5mA para o módulo apenas ligado. Assim, o sistema apresentou um comportamento estável durante o período de ensaio, podendo ser usado por um período de até à 246 minutos (14766588ms) conforme condições do presente ensaio.

5.4 ENSAIO DINÂMICO

O ensaio dinâmico constituiu um conjunto de testes realizados com movimentos visando a comparação dos resultados do sistema desenvolvido com um sistema comercial XSens (MTi-G-700 XSens).

O MTi-G-700 (XSENS, 2014) é uma unidade inercial para fins de animação em três dimensões baseados em movimentos humanos, mensuração da cinemática humana, análise de movimento e controle e estabilização industrial (XSENS, 2014). Suas principais características estão apresentadas na Tabela 5.6.

O MTi-G-700 da XSens possui um frequência de saída de 2kHz (sendo de 120Hz dos sensores – Acelerômetro, Giroscópio e Magnetômetro) em uma interface USB. Suas dimensões são de 57x42x23,5mm pesando 55g. Todo seu processamento é local (no sensor), utilizando um filtro de Kalman desenvolvido pela empresa para fusão das informações e determinação da *attitude*.

Tabela 5.6 Dados da orientação e posição do sensor MTi-G-700 XSens (XSens, 2014).

	Modo de ensaio	Erro típico (graus)	Erro máximo (graus)
Orientação			
Roll/Pitch	Estático	0,2	0,25
	Dinâmico	0,3	1,0
Yaw	Em campo magnético homogêneo	1,0	-
Posição e velocidade			
Posição horizontal	1 Desvio Padrão (SBAS)	1,0 m	
Posição vertical	1 Desvio Padrão (SBAS, baro)	2,0 m	
Velocidade	1 Desvio Padrão RMS	0,1 m/s	

Os ensaios realizados consistiram em fazer movimentos aleatórios com as unidades (XSens e módulo desenvolvido) fixadas em uma superfície e comparar seus resultados. Para fixação, utilizou-se fita adesiva porosa, a qual fornece uma boa fixação sobre a superfície sem gerar muita irregularidade sobre o plano. Para os ensaios de inclinações (*Roll* e *Pitch*) foram fixados uma unidade ao lado da outra. Para ensaios de rotação (*Yaw*), uma unidade foi fixada sobre a outra. A motivação sobre a metodologia de fixação é baseada no fato das duas unidades inerciais terem um tamanho considerável e um posicionamento único poderia intervir na comparação dos resultados uma vez que o corpo onde os mesmos estão fixados é um corpo rígido e livre. Os ensaios tiveram a duração de no máximo 30 segundos.

As Figura 5.7, Figura 5.8 e Figura 5.9 apresentam, respectivamente, os resultados gráficos do Ensaio 2, Ensaio 6 e Ensaio 10 do algoritmo DCM. Enquanto as Tabela 5.7, Tabela 5.8 e Tabela 5.8 apresentam os resultados do sistema comercial com o sistema desenvolvido usando o algoritmo DCM.

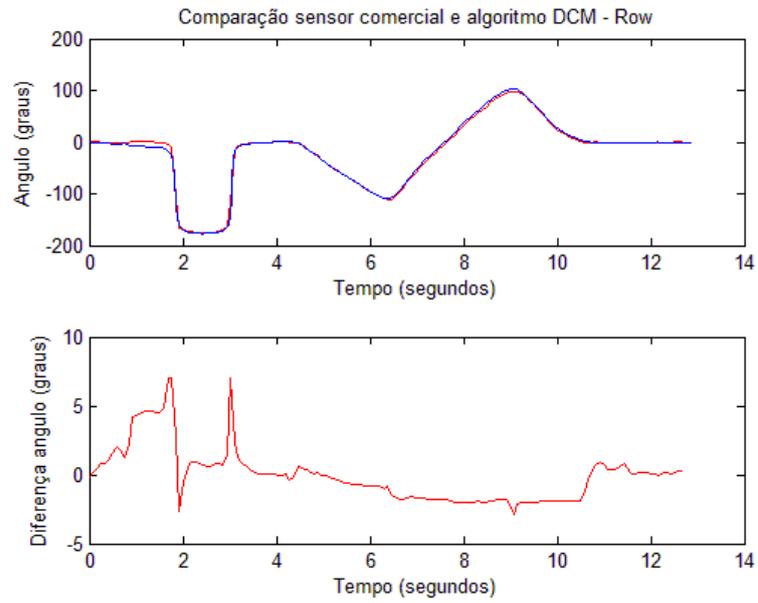


Figura 5.7 Ensaio dinâmico - comparação entre o sistema desenvolvido (DCM) e um sistema comercial – *Row*.

Linha vermelha algoritmo DCM e linha azul sensor comercial.

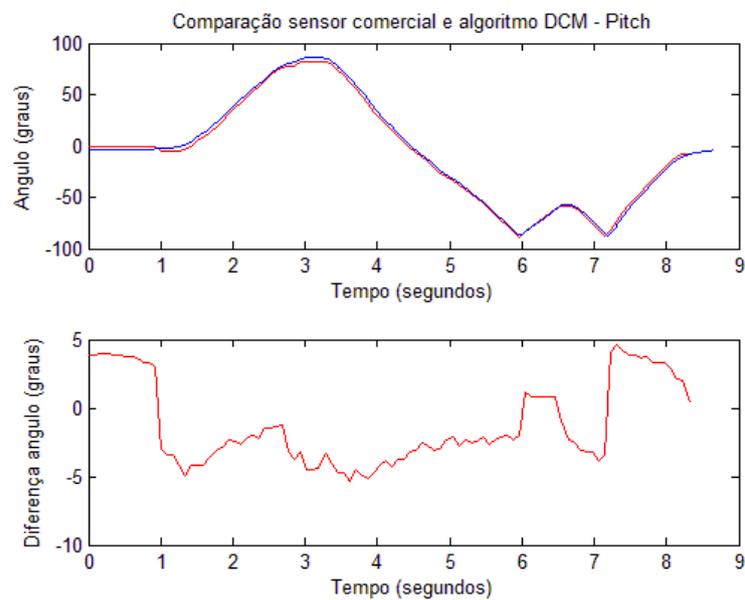


Figura 5.8 Ensaio dinâmico - comparação entre o sistema desenvolvido (DCM) e um sistema comercial – *Pitch*.

Linha vermelha algoritmo DCM e linha azul sensor comercial.

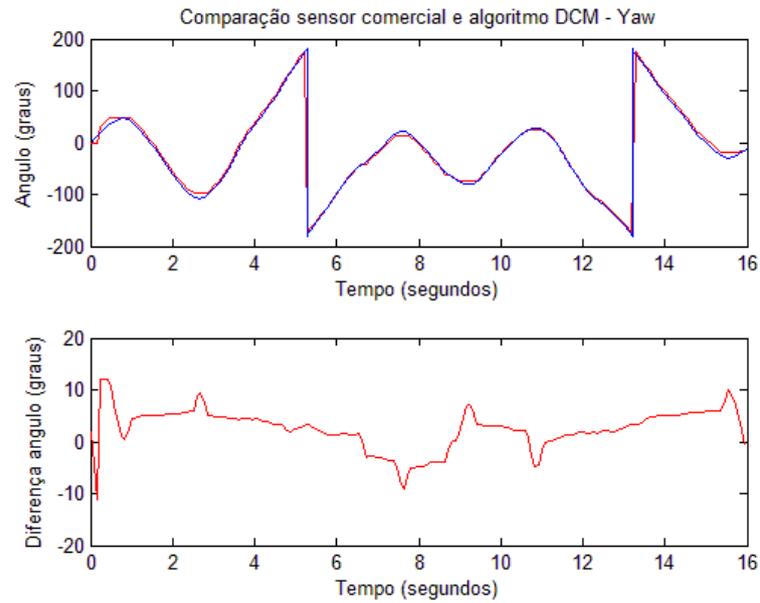


Figura 5.9 Ensaio dinâmico - comparação entre o sistema desenvolvido (DCM) e um sistema comercial –*Yaw*.

Linha vermelha algoritmo DCM e linha azul sensor comercial.

Tabela 5.7 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (DCM - *Row*).

Ensaio	Erro médio	RMS	Correlação
1	0,40	5,09	0,9799
2	0,06	4,04	0,9982
3	0,52	3,01	0,9981
4	0,40	1,60	0,9996

Tabela 5.8 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (DCM - *Pitch*).

Ensaio	Erro médio	RMS	Correlação
5	0,12	3,72	0,9848
6	1,20	3,30	0,9984
7	0,57	2,93	0,9978
8	1,67	2,88	0,9996

Tabela 5.9 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (DCM - Yaw).

Ensaio	Erro médio	RMS	Correlação
9	1,80	5,70	0,9975
10	2,53	4,58	0,9987
11	0,30	8,63	0,9943
12	5,78	7,69	0,9917

Ainda sobre o presente ensaio (os mesmos ensaios) foram aplicados os algoritmos de Kalman e *eCompass*. Os resultados para o filtro de Kalman são demonstrados nas Tabela 5.10, Tabela 5.11, Tabela 5.12. Os resultados para o *eCompass* são demonstrados nas Tabela 5.13, Tabela 5.14 e Tabela 5.15.

Tabela 5.10 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (Kalman - Row).

Ensaio	Erro médio	RMS	Correlação
1	0,62	7,91	0,9703
2	3,76	6,28	0,9957
3	4,48	8,01	0,9671
4	0,36	4,26	0,9970

Tabela 5.11 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (Kalman - Pitch).

Ensaio	Erro médio	RMS	Correlação
5	0,73	5,56	0,9835
6	5,29	8,49	0,9853
7	6,68	9,22	0,9762
8	10,96	11,46	0,9432

Tabela 5.12 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (Kalman - Yaw).

Ensaio	Erro médio	RMS	Correlação
9	1,13	37,89	0,8724
10	1,44	40,92	0,6123
11	2,03	42,93	0,6235
12	3,26	38,27	0,8648

Tabela 5.13 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (eCompass - Row).

Ensaio	Erro médio	RMS	Correlação
1	0,17	14,21	0,9241
2	3,87	12,99	0,9392
3	5,48	12,26	0,9671
4	1,39	14,49	0,9385

Tabela 5.14 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (eCompass - Pitch).

Ensaio	Erro médio	RMS	Correlação
5	3,40	12,06	0,9296
6	1,60	10,15	0,9317
7	6,68	12,38	0,9661
8	10,96	14,62	0,9432

Tabela 5.15 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido (eCompass - Yaw).

Ensaio	Erro médio	RMS	Correlação
9	7,38	63,45	0,5817
10	40,50	77,86	0,7159
11	10,43	67,91	0,6328
12	9,61	75,90	0,7047

Com os melhores resultados da diferença RMS (*root-mean-square error*) de cada bateria de testes entre o XSens e o sistema desenvolvido foi montada a Tabela 5.16.

Tabela 5.16 Resultado para ensaio dinâmico com sensor XSens e sistema desenvolvido - resumo.

Algoritmo	DCM	Kalman	eCompass
<i>Row</i>	1,60	4,26	12,26
<i>Pitch</i>	2,88	5,56	10,15
<i>Yaw</i>	4,58	37,89	63,45

O resultado que obteve melhor desempenho em ensaios dinâmicos foi o DCM. O filtro de Kalman presente nesse ensaio é o filtro que recebe a matriz de rotação diretamente do DCM, conforme descrito no Capítulo 5.1, para formação do modelo de transição de estados. Assim, ele poderia melhorar os resultados da DCM com a fusão dos dados do *eCompass*. Todavia, como os ensaios executados foram avaliados pontos de transição dos eixos, os resultados do algoritmo *eCompass* não foram bons (valores bem altos de RMS). Por consequência disso, os resultados do filtro de Kalman também foram inferiores aos resultados do algoritmo DCM.

Nos ensaios de *Yaw*, verificou-se a presença de um ruído no sinal do magnetômetro do ArduIMU ao se aproximar do XSens – podendo ser uma interferência entre os sistemas. Esse ruído foi notado apenas quando as unidades estavam uma sobre a outra, o que pode ter gerado um aumento da diferença nos ensaios de Kalman e *eCompass* que utilizam o sensor. Assim, esse fato não influencia nas conclusões do experimento, uma vez que nos demais eixos (*Roll* e *Pitch*) a diferença entre o ArduIMU e XSens eram altas e não sofreram com tal ruído, principalmente, pelo posicionamento diferente. Somando a isso, a base de cálculo desses eixos (*Roll* e *Pitch*) no ArduIMU é o acelerômetro (Equação (3.27) e Equação (3.29)), que não é suscetível a esse tipo de interferência magnética. Esse fato reforça a melhora fornecida pelo algoritmo Kalman quando exposto a uma entrada ruidosa, conforme foi apresentada nas tabelas anteriores. Para melhorar a comparação entre os algoritmos e isolar esse problema, foi feito novos ensaios em *Yaw* utilizando apenas a unidade inercial desenvolvida e o algoritmo DCM como referência. Os resultados são demonstrados na Tabela 5.17 e Tabela 5.18 e acusam ainda diferenças altas entre os algoritmos.

Tabela 5.17 Resultado para ensaio dinâmico com algoritmo Kalman tendo como base o algoritmo DCM - *Yaw*.

Ensaio	Erro médio	RMS	Correlação
13	0,41	15,81	0,9240
14	3,58	16,20	0,9359
15	1,60	15,88	0,9460

Tabela 5.18 Resultado para ensaio dinâmico com algoritmo eCompass tendo como base o algoritmo DCM - *Yaw*.

Ensaio	Erro médio	RMS	Correlação
13	11,28	47,50	0,6930
14	7,16	42,04	0,7525
15	1,20	42,21	0,7433

O algoritmo DCM IMU aplicado no sistema obteve resultados semelhantes aos dos outros sistemas utilizados na literatura para ensaio da marcha humana. Seu erro médio para o *Row* manteve-se entre os valores de $0,06^\circ$ e $0,52^\circ$ (ou *Pitch* entre $0,12^\circ$ e $1,67^\circ$), se aproximando dos estudos realizados por Martínez-Solís *et al.* (2014) onde o erro médio era de $0,75^\circ$. Por possuir sua correlação em *Pitch* e *Row* superiores a 97% quando comparado ao sistema de referência e seu erro RMS máximo de $5,09^\circ$ (*Row*), o sistema também se assemelha aos estudos de Djuric (2011), cujos erro em RMS foi de 6° e a correlação foi de 97% (quando o sistema é comparado ao sistema de referência).

Na análise aos dados, o sistema também ficou próximos de KUN (2011), onde o erro em RMS ficou entre $0,92^\circ$ e $4,13^\circ$, com correlação próxima de 0,97, sendo que o sistema desenvolvido forneceu os erros em RMS *Row* entre os valores de $1,60^\circ$ e $5,09^\circ$ e *Pitch* entre $2,88^\circ$ e $3,72^\circ$. Comparando os melhores resultados encontrados, com erros em RMS de X $1,60$, Y $2,88$ e Z $4,58$, os resultados também se assemelham a Mohamed (2012) com os resultados em erro RMS de X $3,86^\circ$, Y $4,51^\circ$ e Z $4,09^\circ$.

Com os ensaios comparativos realizados, foram executados os ensaios de marcha, os quais serão descritos no próximo Capítulo.

5.5 RESULTADOS DE ENSAIO DA MARCHA

O ensaio final consistiu em fazer exames de marcha tendo como foco o ângulo do joelho de voluntários para os ensaios (plano sagital). Para tal, posicionou-se os módulos dos sensores nos membros inferiores conforme a Figura 5.10 e um sensor no topo da cabeça do usuário.



Figura 5.10 Posicionamento dos sensores para ensaio da marcha.

O sistema foi utilizado em uma sala ampla, onde os usuários puderam executar o ensaio da caminhada para frente livre de obstáculos. Através de um notebook com configurações razoáveis para os dias de hoje (4GB RAM, Processador i5) pôde-se acompanhar a movimentação dos usuários, sendo seus resultados gráficos apresentados em tempo real no modelo virtual (Figura 5.11).

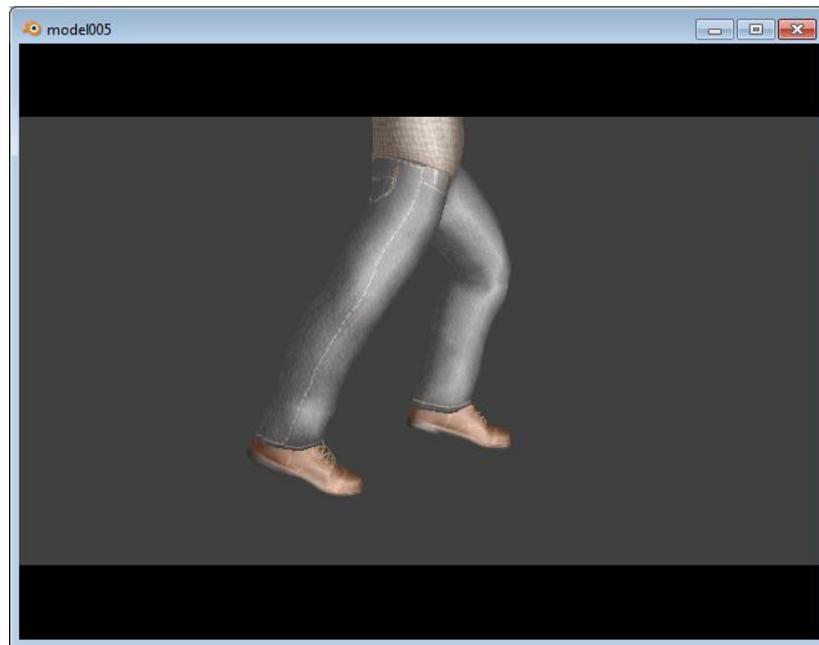


Figura 5.11 Modelo virtual acompanhando a marcha.

Após os ensaios, os dados foram analisados, subtraindo-se a angulação da tíbia (*roll*) com a angulação do fêmur (*roll*). Assim, obteve-se as curvas para análise. A Figura 5.12 apresenta o ensaio de um dos voluntários acompanhando sua perna direita.

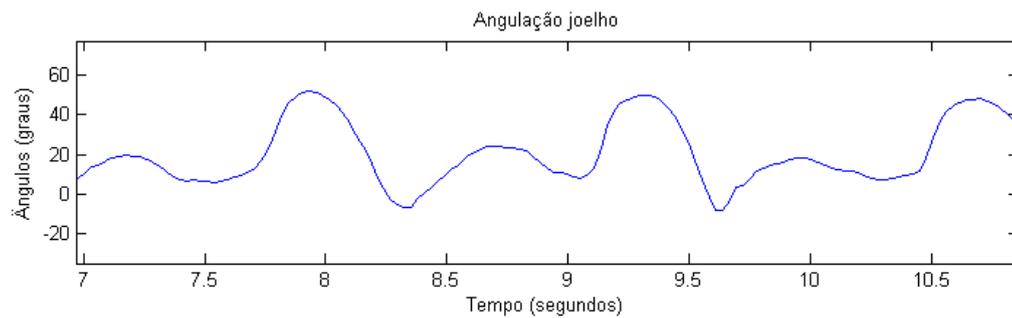


Figura 5.12 Angulação da flexão do joelho direito obtido pelo sistema.

Dessa maneira, pode-se comparar os resultados obtidos com a bibliografias de referência, como é demonstrado na Figura 5.13, gerada pelo sistema de videogrametria Dvideow (ARAÚJO *et al.*, 2004).

A Figura 5.13 demonstra a flexão do joelho esquerdo de três pessoas, com os dados adquirido pelo sistema Dvideow. Comparando a Figura 5.12 com a Figura 5.13 pode-se concluir que a mesma encontra-se de acordo com o paciente representado pela linha tracejada maior.

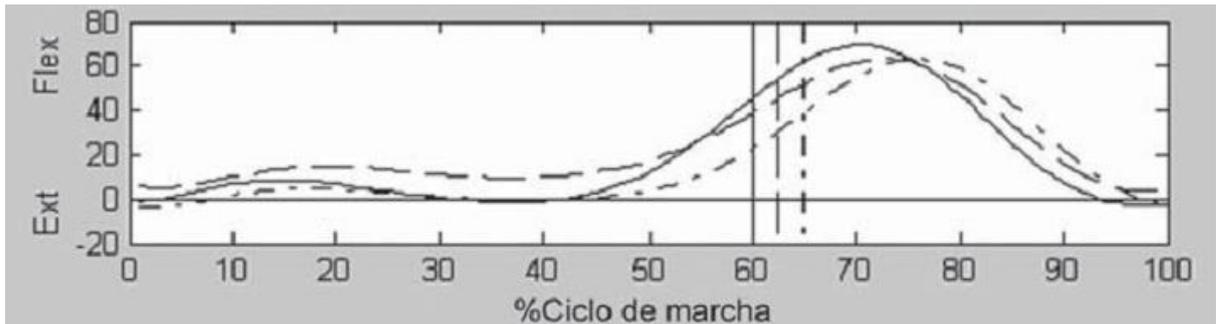


Figura 5.13 Angulação da flexão do joelho através do sistema Dvideow.

Fonte : ARAÚJO, 2004.

Como resultado, ao analisar o gráfico observa-se uma pequena onda de flexão para auxiliar na absorção de choque devido ao impacto do pé ao solo e para manutenção do peso corporal. A segunda onda de flexão atinge seu pico no início da fase de balanço e representa a liberação do pé da superfície. O presente resultado está de acordo com Rose e Gamble (1998) os quais descrevem o movimento do joelho com as duas ondas de flexão.

Ensaio de mais pacientes foram plotados na Figura 5.14 para fins de verificação das curvas resultantes. Como resultado, nota-se que as curvas estão de acordo com a análise anterior.

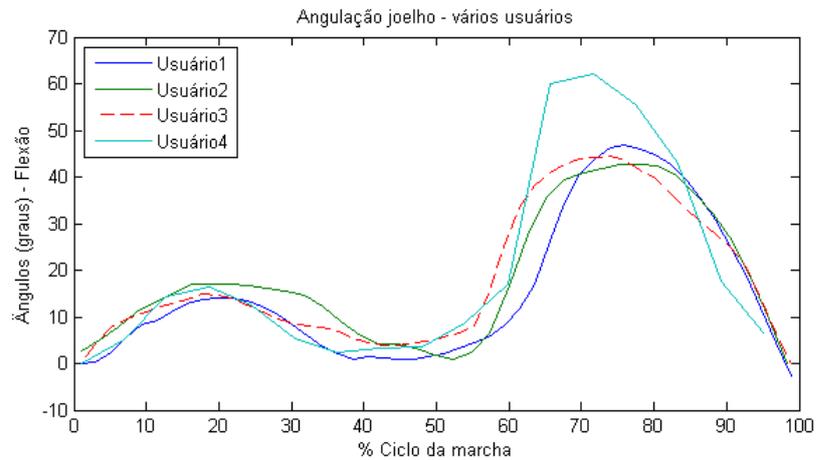


Figura 5.14 Angulações da flexão do joelho obtido pelo sistema.

Através da análise do resultado do sensor no topo da cabeça dos voluntários (*Pitch*), obteve-se uma pequena onda senoidal. A Figura 5.15 foi retirada de um dos ensaios dos voluntários e demonstra a onda gerada pelo sensor do topo da cabeça.

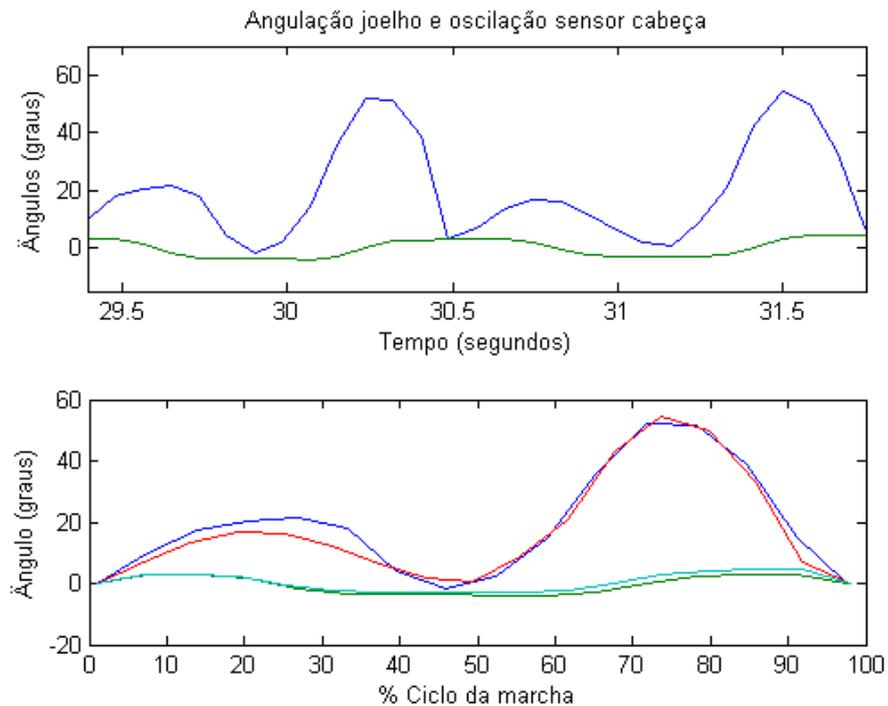


Figura 5.15 Angulação de um sensor no topo da cabeça do usuário.

Figura superior – referência tempo. Figura inferior – referência ciclo da marcha.

Linha vermelha e azul angulação do joelho e linhas verdes do sensor acima da cabeça do usuário.

A onda observada teve uma variação por voluntário com os valores entre 1° e 4° e com formas senoidais semelhantes. De acordo com a bibliografia, a mesma está associada ao equilíbrio do corpo durante a marcha cuja melhor representação está na vista superior (Figura 5.16).

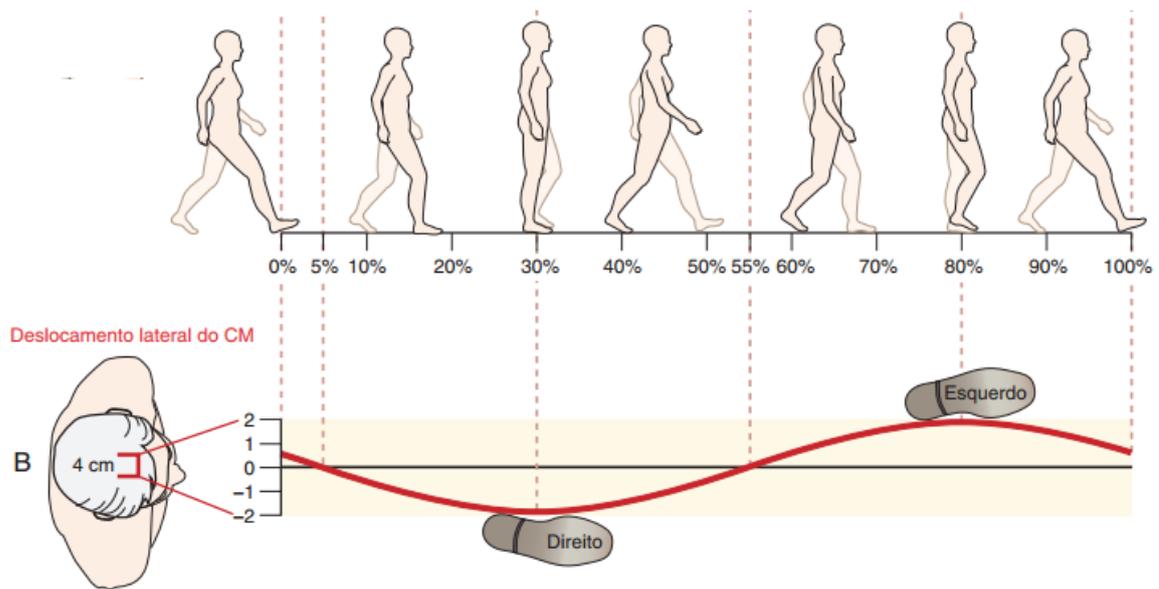


Figura 5.16 Deslocamento do centro de massa.

Fonte: Neumann, 2011.

Por fim, o ensaio revelou que o sistema é funcional para análise da marcha, podendo capturar variáveis de interesse, tais como, a angulações dos membros sensorizados para estudo, acompanhamento e marcha humana.

6 CONCLUSÃO

O objetivo traçado no início do desenvolvimento desse projeto foi alcançado com êxito. Foi desenvolvido um sistema para estudo da marcha humana com o uso de módulos sem fio desenvolvido com sensores inerciais e também um modelo virtual do corpo humano que permite a visualização do movimento gerado durante o ensaio. O sistema efetua os registros das variáveis cinemáticas da marcha tais como aceleração, velocidade angular e angulações dos membros sensoriados, para estudos detalhados da caminhada.

Os objetivos secundários elencados também foram alcançados ao passo que era desenvolvido o sistema. Foi criada uma rede de sensores sem fio com o uso dos transceptores Xbee em formato de uma rede estrela, concentrando todo o envio de informações para um coordenador ligado a um computador. Cada módulo do sistema contém uma unidade inercial denominada ArduIMU que, através dos seus sensores (acelerômetro, giroscópio, magnetometro), apresenta a *attitude* do membro do usuário ao qual está vinculada e envia essa informação para o coordenador. A *attitude* de cada membro é calculada através da fusão das informações dos sensores que foram avaliados durante o trabalho (DCM IMU, *eCompass* e Kalman) e ao qual foi escolhido o algoritmo DCM IMU para ser embarcado nos módulos sensores do sistema. Apresentação dos resultados é demonstrada em um modelo virtual desenvolvido que reproduz o movimento conforme a movimentação do usuário.

O sistema foi comparado a um sistema comercial e a artigos de outros sistemas desenvolvidos para esse fim. Como resultado, o sistema desenvolvido obteve erros médios de X 0,52° Y 1,20° Z 1,80° e erros em RMS de X 3,01° Y 3,30° Z 5,70° quando comparados com um sistema comercial, sendo esse resultado próximo à literatura (artigos) e aplicável em exames biomecânicos de marcha.

Sobre o custo total do projeto em comparação a um sistema comercial, faz-se necessário uma análise mais profunda do assunto tendo em vista que é necessário levantar muitos dados, tais como, o custo de desenvolvimento, instalações, treinamento e equipamentos necessários. Ressalta-se que o custo de uma unidade desenvolvida custa cerca de 2,3% (\$100,00 - 2014) de uma unidade comercial, determinando assim que o sistema atual representa um sistema de baixo custo quando comparado ao mercado relacionado.

6.1 PROPOSTA DE TRABALHOS FUTUROS

Sobre o ponto de vista de melhorias, o sistema poderia ter uma sincronização entre os sensores. Ele possui o registro do tempo interno de cada sensor e o tempo que o pacote entrou no sistema, mas poderia melhorar se houvesse a opção de ordenar o início do ensaio em todos os módulos. Dessa maneira, poderia sincronizar o início do ensaio com outros sistemas que trabalhassem em paralelo a ele. Para efetuar isso, apenas seria necessário ligar a porta de saída serial do Xbee na entrada do ArduIMU e modificar a programação. Haveria um ganho significativo uma vez que também permitiria a configuração de outros modos de configuração de pacote, como por exemplo, a leitura da aceleração com uma taxa de aquisição maior. Com essa alteração também haveria uma otimização do consumo de bateria, uma vez que o sistema não iniciaria o envio dos pacotes sem o sinal de sincronização.

Outra melhoria, seria a inclusão de outro algoritmo que aprimorasse a leitura do magnetômetro, possivelmente no lugar do *eCompass* para trabalhar com Kalman e o IMU DCM. Um algoritmo mais robusto contribuiria na união desses algoritmos e na determinação da variável *Yaw*.

É necessário também um estudo mais detalhado da interferência verificada entre o sensor desenvolvido e módulo comercial. Seria importante mensurar o quanto essa interferência afeta o sistema e a natureza (origem) da mesma. Após anular essa interferência, seria possível realizar ensaios com os dois sistemas trabalhando em conjunto, possibilitando novas soluções no estudo da marcha.

Ao decorrer dos ensaios o sistema também provou ser uma ótima alternativa para medição de ângulo em exames estáticos, como substituição do goniômetro para esse fim. Essa alternativa cria um novo conjunto de possibilidades para o desenvolvimento de novos exames nas áreas médicas para acompanhamento na recuperação ou tratamento de pacientes.

Por se tratar de um sistema baseado em softwares abertos e livres, as possibilidades de expansão e adaptação do sistema tornam-se ilimitadas. Não apenas focando o sistema em exames de marcha, ele pode ser aberto para novos projetos como, por exemplo, aproveitar a parametrização dos ângulos do modelo virtual para utilização dessa estrutura para movimento de próteses humanas. A partir da forma como foi estruturada a interface com o *Blender Engine* para o modelo virtual, pode-se facilmente sintetizar novas alternativas para outros projetos que focam a mecânica do corpo humano ou humanoide. O desenvolvimento da rede de sensores inerciais sem fio cria também muitas outras possibilidades na área de instrumentação biomédica, como avaliação de impactos e outros exames e exercícios na área da saúde. Assim, o presente trabalho não se torna conclusivo para esse sistema desenvolvido para exame da marcha, ele é apenas mais um passo firme em uma longa caminhada nos estudos dessa área. .

REFERÊNCIAS

- ARAÚJO, A.; ANDRADE, L.; BARROS, R. Sistema para análise cinemática da marcha humana baseado em videogrametria. **Fisioterapia e Pesquisa**, [S.l.], v. 2, p. 3-10, 2015.
- AGGARWAL, J.K.; CAI, Q. Human motion analysis: a review. In: IEEE NONRIGID AND ARTICULATED MOTION WORKSHOP, 1997, San Juan. **Proceedings...** Los Alamitos, Calif.: IEEE Computer Society, 1997. p.90-102.
- BALBINOT, A; BRUSAMARELLO, V. J. **Instrumentação e Fundamentos de Medidas**. Rio de Janeiro: LTC, 2007. v. 2. ISBN 978-852-1618-799.
- BALBINOT, Alexandre; BRUXEL, Yuri; ZARO, Milton Antônio. Evaluation of impact transmissibility on individuals with shoes and barefoot during human gait. **Measurement**, London, v. 46, p. 2547-2554, 2013.
- BARTON, G.J. *et al.* Manipulation of visual biofeedback during gait with a time delayed adaptive Virtual Mirror Box. **Journal of NeuroEngineering and Rehabilitation**, [S.l.], v. 11, 2014. Disponível em: <<http://www.jneuroengrehab.com/content/11/1/101>>. Acesso em: 02 Jan. 2015.
- BECK, K. **Test-Driven Development**. Boston: Addison Wesley, 2003. 240 p. ISBN 978-032-1146-533.
- BHATIA, S. *et al.* 3D human limb detection using space carving and multi-view eigen models. In: IEEE WORKSHOP ON ARTICULATED AND NON RIGID MOTION, 4., 2004. Washington. **Proceedings...** New York: IEEE Publishing, 2004. p. 17.
- BOHANNON, R.W.; ANDREWS, A.W.; SMITH, M.B. Rehabilitation goals of patients with hemiplegia. **International Journal of Rehabilitation Research**, [S.I.], v.11, n. 2, p. 181-184, Jun. 1988.
- BOHANNON, R.W. Comfortable and maximum walking speed of adults aged 20-79 years: reference values and determinants. **Age and Ageing**, [S.l.], v. 26, p. 15-19, 1997.
- CABRAL, E.L.L. **Preliminares Matemáticos**. São Paulo: Universidade de São Paulo, 2008. Disponível em: <<http://sites.poli.usp.br/p/eduardo.cabral/>>. Acesso em: 02 Jan. 2015.
- CHUNG, T. M. Avaliação cinética e cinemática da marcha de adultos do sexo masculino. **Acta Fisiátrica**, São Paulo, v.7, n.2, p.61-67, 2000.
- CORREA, Daniel; BALBINOT, Alexandre. Experimental Videogrammetry and Accelerometry for the 2D Analysis of Human Motion. **Sensors & Transducers**, Brussels, v. 150, p. 79-89, 2013.

CORREA, D. **Desenvolvimento de um sistema para estudo da marcha humana por videogrametria e acelerometria**. 2010. 106 p. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.

DJURIC, Milica, JOVICIC, Nenad; POPOVIC, Dejan. Kinematics of Gait: New Method for Angle Estimation Based on Accelerometers. **Sensors**, [S.l.], v. 11, n. 11, p. 10571-10585, 2011.

EMIL, J. *et al.*, A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. **Journal of Neuroengineering & rehabilitation**, [S.l.], v. 2, n. 6, p. 1-10, 2005.

FORHAN, Neisy. **Giroscópios MEMs**. São José dos Campos: INPE, 2010. Relatório técnico-científico de pesquisa supervisionado do Instituto Nacional de Pesquisas Espaciais.

FRANÇA JR. **Simulação e implementação em tempo real de sistemas de navegação inercial integrados INS/GPS**. 2009. 112 p. Dissertação (Engenharia de Defesa) – Instituto Militar de Engenharia, Rio de Janeiro, 2009.

FREITAS, J. **Sensores Inerciais e Protocolo ZigBee no Desenvolvimento de Tecnologia Assistiva para Reabilitação Motora dos Membros Superiores Humanos**. 2013. 75 p. Trabalho de Conclusão de Curso (Engenharia de Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

GEBRE-EGZIABHER, D. **Design and performance analysis of a low-cost aided dead reckoning navigator**. 2004. 285 p. Tese (Doutorado em Filosofia) - University of Stanford, California, 2004.

GOLDSTEIN, Hebert. **Classical Mechanics**. 2nd. [S.l.]: Addison-Wesley, 1922. ISBN: 978-020-1029-185.

GODFREY, A. *et al.* Direct measurement of human movement by accelerometry. **Medical Engineering & Physics**, [S.l.], v.30, p.1364–1386, 2008.

GRANDO, E. V. **Sistema de navegação inercial para veículos, terrestres usando sensores iniciais de baixo custo**. 2010. 100 p. Trabalho de Conclusão de Curso (Engenharia da Computação) - Programa de Pós-Graduação em Informática, Pontifícia Universidade Católica do Paraná, Paraná, 2010.

GREWAL, M.S; WEILL, L.R; ANDREWS, A.P. **Global Position System, Inertial Navegation and Integration**. New York: John Wiley & Sons, 2001. ISBN 978-047-0041-901.

GROUP OF SENSOR DATA AND INFORMATION. **Definitions of Sensor Data Fusion in the Literature**, Bonn: Institute of Computer Science da Universitat Bonn, 2010. Disponível em: <<http://net.cs.uni-bonn.de/de/wg/sensor-data-and-information-fusion/what-is-it/sdf-definitions/>>. Acesso em: 10 Dez. 2014

HAMILTON, W. R. On quaternions, or on a new system of imaginaries in algebra. **Philosophical Magazine**, [S.l.], p. 1844-1850.

HONEYWELL. **3-Axis Digital Compass IC HMC5883L**. [S.I.], 2013. Disponível em: <http://www51.honeywell.com/aero/common/documents/myaerospacelog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf>. Acesso em: 27 Jan. 2015.

INVENSENSE. **MPU-6000**. [S.I.], 2013. Disponível em: <<http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>>. Acesso em : 27 Jan. 2015.

KIM, Phil. **Kalman Filter for Beginners**. New York: CreateSpace, 2011. ISBN 978-146-3648-350.

KUN, Liu; *et al.* Ambulatory Estimation of Knee-Joint Kinematics in Anatomical Coordinate System Using Accelerometers and Magnetometers; **Biomedical Engineering, IEEE Transactions on**, New York, v. 58, n. 2, Feb. 2011.

LAGES, W. F. **Filtro de Kalman**. Porto Alegre : UFRGS, 2008. Disponível em: <<http://www.ece.ufrgs.br/~fetter/eng04037/kalman.pdf>>. Acesso em: 02 Jan. 2015.

LIMA, R.R. **Desenvolvimento de uma Cabeça Sensora para Veículos Aéreos Não-Tripulados**. 2013. 153 p. Dissertação (Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 2013.

MACEDO, R. **Desenvolvimento de uma plataforma para calçados para análise da força de reação do solo**. 2013. 100 p. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

MAHFOUZ, Mohamed; TO, Gary. Design of Wireless Inertial Trackers for Human Joint Motion Analysis. In: CONFERENCE ON BIOMEDICAL WIRELESS TECHNOLOGIES, NETWORKS, AND SENSING SYSTEMS, 1., 2012, Santa Clara, CA. **Proceedings...** New York: IEEE Publishing, 2012. p. 49-52.

MAHONY, R; *et al.* A Complementary Filter for Attitude Estimation of a Fixed-Wing UAV. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 1., 2008, Nice. **Proceedings...** New York: IEEE, 2008. p. 340-345.

MAHONY, R; HAMEL, T; PFLIMLIN, J.M. Complementary filter design on the special orthogonal group SO(3). In: CONFERENCE ON DECISION AND CONTROL, AND THE EUROPEAN CONTROL CONFERENCE, 44., 2005, Seville. **Proceedings...** New York: IEEE, 2005. p. 1477–1484.

MANSUCLAL, Jitesh Pramodray. **Tilt Compensated Digital Compass**. 2010. 222 p. Dissertação (Engenharia Electrotécnica e de Computadores) - Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, 2010.

MARQUES, A. P. **Manual de goniometria**. 2.ed. São Paulo: Manole, 2003. ISBN 978-852-0416-273.

MARCIAS, E.; TORRES, D.; RAVINDRAN, S. **Nine-Axis Sensor Fusion Using the Direction Corsine Matrix Algorithm on the MSP430F5xx Family**. Dallas: Texas Instruments, 2012.

MARTÍNEZ-SOLÍS, F. *et al.* A portable system with sample rate of 250 Hz for characterization of knee and hip angles in the sagittal plane during gait. **BioMedical Engineering Online**, [S.l.], v. 13, 2014. Disponível em: < <http://www.biomedical-engineering-online.com/content/13/1/34>>. Acesso em: 20 Ago. 2014.

MATSELENAK, Y. **Advanced hard and soft iron magnetometer calibration**. [S.I.], 2014. Disponível em: <<http://diydrones.com/profiles/blogs/advanced-hard-and-soft-iron-magnetometer-calibration>>. Acessado em : 27 Jan. 2015.

MAZILU, Sinziana. *et al.* GaitAssist: A Wearable Assistant for Gait Training and Rehabilitation in Parkinson's Disease. In: IEEE INTERNATIONAL CONFERENCE ON PERSVASIVE COMPUTING AND COMMUNICATIONS, 2014, Budapest. **Proceedings...** New York: IEEE, 2014. p. 135-137.

MOORE, Keith; DALLEY, Arthur. **Anatomia Orientada para a clínica**. 6. ed. São Paulo: Guanabara Koogan, 2011. ISBN 978-852-7712-576.

NAJAFI, B.; *et al.* Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. **IEEE Transactions on Biomedical Engineering**, New York, v. 50, n. 6, p.711–723, June 2003.

NASA. **Euler Angles, Quaternions and Transformation Matrices**. Texas: NASA, 1977. Disponível em: <<http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770024290.pdf>>. Acesso em: 02 Jan. 2015.

NEUMANN, D. **Cinesiologia do Aparelho Musculoesquelético**. 2.ed. Rio de Janeiro: Elsevier, 2011. ISBN 978-853-5239-669.

OZYAGCILAR, T. **Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensor**. Arizona: Freescale Semiconductor, 2012. Disponível em: < <http://www.biomedical-engineering-online.com/content/13/1/34>>. Acesso em: 20 Ago. 2014.

PARK, M. **Error analysis and stochastic modeling of mems based inertial sensors for land vehicle navigation applications**. 2004. 122 p. Tese (Mestrado em Ciências) – University of Calgary, Calgary, 2004.

PERRY, J., **Gait Analysis, normal and pathological function**. New Jersey: Slack, 1992. ISBN 978-155-6421-921.

PERRY, J. **Análise de marcha**. São Paulo: Manole, 2005. 3v. ISBN 978-852-0422-861.

PREMERLANI, W; BIZARD, P. **Direction Cosine Matrix IMU: theory**. [S.I.: s.n.], 2009. Disponível em: <<http://gentlenav.googlecode.com/files/DCMDraft2.pdf>>. Acesso em: 20 Ago. 2014.

ROSE, J; GAMBLE, J.G. **Marcha Humana**. 2. ed. São Paulo: Premier, 1998. ISBN 978-858-6067-037.

SABATELLI, Simone; *et al.* A Sensor Fusion Algorithm for an Integrated Angular Position Estimation with Inertial Measurement Units. In: DESIGN, AUTOMATION & TEST IN EUROPE CONFERENCE EXHIBITION, 1., 2011, Grenoble. **Proceedings...** New York: IEEE, 2011. p. 1-4

SHEMA, S.; *et al.* Clinical Experience Using a 5 Week Treadmill Training Program with Virtual Reality to Enhance Gait. In: INTERNATIONAL CONFERENCE ON VIRTUAL REHABILITATION, 1., 2013, Philadelphia. **Proceedings...** New York: IEEE, 2013. p. 249-253.

STEVENS, B.L.; LEWIS, F. L. **Aircraft Control and Simulation**. 2nd. ed. New York: John Wiley & Sons, 2003. ISBN 978-047-1371-458.

TITTERTON, D. H; WESTON, J.L. **Strapdown Inertial Navigation Technology**. 2nd. ed. Stevenage: Institution of Electrical Engineers, 2004. ISBN 978-086-3413-582.

TRAWNY, N; ROUMELIOTIS, S. **Indirect Kalman Filter for 3D Attitude Estimation**. Minnesota: University of Minnesota, 2005. Disponível em: <http://www-users.cs.umn.edu/~trawny/Publications/Quaternions_3D.pdf>. Acesso em: 20 Ago. 2014.

VAUGHAN, C.; DAVIS, B. L.; O'CONNOR, J. **Dynamics of Human Gait**. 2nd. ed. Cape Town, South Africa: Kiboho, 1992. ISBN 0-620-23560-8.

VAUGHAN, C; DAVIS, B. L; O'CONNOR, J. **Dynamics of Human Gait**. 2nd. ed. New York: Human Kinetics Publishers, 1992. ISBN 978-087-3223-706.

WALCHKO, K. J. **Low cost inertial navigation: learning to interning to integrate noise and find your way**. 2002. 80 p. Tese (Mestrado em Ciências) – University of Florida, Florida, 2002.

ZHOU, H. e HU, H. Inertial sensors for motion detection of human upper limbs. **Sensor Review**. [S.l.], 2007. v.27, p.151–158.