

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE CENTRAL INERCIAL
PARA SCANNER DE RECONSTRUÇÃO
TRIDIMENSIONAL (3D)**

MAURÍLIO FERNANDES MARQUES JÚNIOR

Brasília, agosto de 2006

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE CENTRAL INERCIAL
PARA SCANNER DE RECONSTRUÇÃO
TRIDIMENSIONAL (3D)**

MAURÍLIO FERNANDES MARQUES JÚNIOR

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Mecatrônica

Banca Examinadora

Prof. Geovany Araújo Borges, UnB/ENE
(Orientador)

Prof.

Prof.

Prof.

Dedicatória

"O começo é a parte mais difícil do trabalho" (Platão)

Dedico este trabalho a todos que me ajudaram na difícil caminhada para minha formação de graduação, nisto incluo minha família, professores e os amigos da faculdade e de fora dela.

MAURÍLIO FERNANDES MARQUES JÚNIOR

Agradecimentos

Agradeço primeiramente à minha família pela paciência e ajuda prestada em toda trajetória de formação de caráter e valores que possuo, um agradecimento especial também ao meu amigo Gustavo Henrique Cotta com quem comecei o desenvolvimento do primeiro projeto com navegação inercial e por motivos adversos não pôde continuar no mesmo. Pelo auxílio em diversos problemas encontrados pelo caminho agradeço ao pessoal do LCVC da Universidade de Brasília, em especial ao grupo do CARCARAH, e pela oportunidade de desenvolvimento deste sistema agradeço aos membros do projeto CNPq "MUSEU 3D: Museu Virtual de réplicas em 3D de objetos físicos com aplicação em Educação e Cultura", os profs. Geovany Araújo Borges e Ricardo de Queiroz (UnB) e profs. Olga Regina Pereira Bellon e Luciano Silva (UFPR). Ao CNPq que por meio do projeto financiou parcialmente este trabalho.

MAURÍLIO FERNANDES MARQUES JÚNIOR

RESUMO

Este trabalho é parte do projeto financiado pelo CNPq, intitulado "Museu Virtual 3D" entre a UFPR e a UnB. Neste projeto, é proposto o uso de um sistema de navegação inercial para estimação de movimentos 3D realizados num *scanner laser* de reconstrução tridimensional. Este sistema utiliza imagens 3D obtidas de diferentes posições de um determinado objeto. Para a reconstrução do modelo 3D de tal objeto pela imagens adquiridas, alguns algoritmos necessitam de uma estimação inicial da posição do laser em cada imagem. Com isto, este projeto consiste no desenvolvimento de uma central inercial para este propósito. Este sistema foi desenvolvido seguindo requerimentos específicos de acoplamento ao *scanner laser*, operação autônoma e armazenamento de dados, assim como a comunicação usando interface USB com um PC com sistema Linux. Uma experiência anterior do autor na área possibilitou a conclusão do projeto em menos de sete meses.

ABSTRACT

This work is part of a CNPq funded project titled "Museu Virtual 3D" executed by UFPR and UnB. In this project, it is proposed the use of an inertial navigating system (INS) for determining 3-D movements of a laser scanner for aiding object reconstruction from depth images. This system uses 3-D depth images gathered from different positions with respect to a target object. In order to reconstruct a 3-D model of such object from the collected images, most algorithms require initial guesses of the laser scanner positioning for each image. Thus, this project consists of the development of an INS for such purpose. The system was developed following specific requirements for coupling to laser scanner, autonomous operation and data storage, as well as communication using USB interface to a Linux-based microcomputer. Previous experience of the author in the area allowed the conclusion of this project in less than seven months.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO	3
1.4	APRESENTAÇÃO DO MANUSCRITO	4
2	FUNDAMENTOS TEÓRICOS	5
2.1	INTRODUÇÃO	5
2.2	SISTEMAS DE COORDENADAS	5
2.2.1	COORDENADAS FIXAS LOCAL	5
2.2.2	COORDENADAS DO CORPO	6
2.3	TRANSFORMAÇÃO DE COORDENADAS	6
2.4	REPRESENTAÇÃO DE TRANSLAÇÃO	7
2.5	REPRESENTAÇÕES DE ORIENTAÇÃO	7
2.5.1	REPRESENTAÇÃO DE ORIENTAÇÃO POR ÂNGULOS DE EULER	7
2.5.2	REPRESENTAÇÃO DE ORIENTAÇÃO POR QUATERNIONS	9
2.6	SENSORES INERCIAIS	12
2.6.1	GIRÔMETROS	12
2.6.2	ACELERÔMETROS	12
2.7	SISTEMAS DE NAVEGAÇÃO INERCIAL	12
2.7.1	GIMBALED INS	13
2.7.2	STRAPDOWN INS	13
2.8	INCERTEZAS DO SISTEMA	14
2.8.1	PROCESSOS GAUSSIANOS	15
2.8.2	PROPAGAÇÃO DAS INCERTEZAS DE ESTIMAÇÕES DOS ESTADOS DO IMU	15
3	DESENVOLVIMENTO E RESULTADOS OBTIDOS	17
3.1	INTRODUÇÃO	17
3.2	SIMULAÇÃO DE IMU	17
3.2.1	VALIDAÇÃO DO MODELO	18
3.2.2	SIMULAÇÕES PARA VALIDAÇÃO DA ESTIMAÇÃO DE POSIÇÃO	22
3.3	IMPLEMENTAÇÃO DO SISTEMA	26
3.3.1	CONSTRUÇÃO DO PROTÓTIPO DE IMU	26
3.4	DEFINIÇÃO DOS ESTADOS DO SISTEMA	28
3.5	IMPLEMENTAÇÃO DA PLACA-MÃE DO MICROCONTROLADOR	29
3.6	IMPLEMENTAÇÃO DA BIBLIOTECA PARA COMUNICAÇÃO COM O PC	31
3.6.1	DEFINIÇÃO DE REQUISIÇÕES FEITAS PELO PC	31
3.6.2	PROTOCOLO DE COMUNICAÇÃO	32
3.6.3	BIBLIOTECA	32
3.6.4	PROGRAMA PARA VISUALIZAÇÃO DA ESTIMAÇÃO	32
3.7	SENSORES	33
3.7.1	CARACTERIZAÇÃO	33
3.8	TEMPOS DE PROCESSAMENTO	38
3.9	TESTE DAS ESTIMAÇÕES	38
4	CONCLUSÕES	40

REFERÊNCIAS BIBLIOGRÁFICAS	41
ANEXOS	42
I DIAGRAMAS ESQUEMÁTICOS	43
II DESCRIÇÃO DO CONTEÚDO DO CD	52

LISTA DE FIGURAS

1.1	Esboço do sistema do scanner 3D com INS.....	1
1.2	Foto do protótipo de IMU construído em projeto anterior.....	2
1.3	Imagens sobrepostas obtidas para reconstrução de objeto 3D.....	3
1.4	Esboço do Sistema.	4
2.1	Representação do sistema de coordenadas fixas local.....	5
2.2	Representação do sistema de coordenadas do corpo.....	6
2.3	Transformação de coordenadas.....	6
2.4	Representação de transformação de translação.....	7
2.5	Ângulos de Euler pela convenção utilizada em Engenharia Aeroespacial.....	8
2.6	Design básico de um acelerômetro.....	12
2.7	Montagem dos sensores no IMU utilizado no <i>Strapdown</i> INS.....	13
2.8	Exemplos de funções de densidade de Distribuições Gaussianas.....	15
3.1	Fluxograma da Simulação de Estimação de Orientação.....	19
3.2	Gráficos de validação das propagações de incerteza.....	20
3.3	Gráficos de erros dos métodos de Integração Numérica.....	21
3.4	Gráficos com intervalos $\pm 3\sigma$ para análise de período de amostragem.....	23
3.5	Fluxograma da Simulação de Estimação de Posição.....	24
3.6	Gráficos da simulação de estimação da posição.....	25
3.7	Foto do protótipo de IMU construído.....	26
3.8	Rede de Petri ilustrando funcionamento do sistema.....	29
3.9	Árvore de alcançabilidade do sistema.....	29
3.10	Foto da placa-mãe construída.....	30
3.11	Formato das mensagens definidas para o sistema.....	32
3.12	Apresentação do programa de visualização dos estados estimados.....	33
3.13	Histogramas das conversões A/D com IMU parado.....	34
3.14	Experimento para determinação do Ganho de Escala dos Acelerômetros.....	35
3.15	Experimento para determinação do Ganho de Escala dos Girômetros.....	36
3.16	Estimações dos quaternions nas rotações de π radianos.....	37
3.17	Estimações obtidas com movimento teste.....	39
I.1	PCB da placa do segundo protótipo de IMU.....	44

LISTA DE TABELAS

3.1	Mensagens apresentadas no LCD	30
3.2	Possíveis requisições feitas ao sistema pelo PC.....	31
3.3	Valores de Ganho de Escala encontrados para os acelerômetros.....	35
3.4	Tempos de processamento das etapas realizadas pelo sistema.....	38

LISTA DE SIMBOLOS

Símbolos Latinos

W	Eixo de coordenadas global
B	Eixo de coordenadas do corpo
U	Matriz Identidade
A	Vetor com medições de aceleração
R	Matriz de transformação de Rotação
T	Vetor com informações de Translação

Símbolos Gregos

ω	Velocidade de rotação
δ	Erro de medição
ψ	Ângulo de azimute
θ	Ângulo de elevação
ϕ	Ângulo de rolagem
μ	Valor médio
σ	Desvio-padrão
ξ	Esperança
Ω	Vetor com medições de velocidade de rotação

Grupos Adimensionais

A/D	Conversão Analógico Digital
-------	-----------------------------

Subscritos

e	Erro
E	Identificador do eixo de coordenadas com $(0, 0, 0)$ no centro da terra
W	Identificador do eixo de coordenadas global
B	Identificador do eixo de coordenadas do corpo

Sobrescritos

\cdot	Variação temporal
$\bar{}$	Conjugado
$\check{}$	Medição
$\hat{}$	Estimação

Siglas

IMU	Unidade de Medição Inercial
INS	Sistema de Navegação Inercial

1 INTRODUÇÃO

Este capítulo apresenta a principal motivação do trabalho de graduação. Os objetivos são claramente apresentados, visando assim satisfazer um conjunto de características prescritas para este trabalho. Por fim, o manuscrito é apresentado.

Reconstruir significa tornar a construir algo, é exatamente isto que a reconstrução tridimensional (3D) faz. De porte de algum objeto é possível se reconstruir digitalmente as informações de formato e textura do mesmo, possibilitando utilizar esta reconstrução digital para diversos fins.

Na área de saúde utiliza-se a reconstrução 3D em órgãos e veias do corpo humano, como dito em [1]. Isto possibilita a análise e visualização de possíveis problemas ou mal-funcionamentos com extrema liberdade de referência de posição e orientação.

Além da saúde, outra área onde se utiliza com frequência a digitalização 3D é na engenharia. Um exemplo, pode ser no teste de estabilidade ou de resistência a passagem de fluídos de peças de aviões ou mesmo de carros através de simulações, sem que estes realmente necessitem entrar num túnel de vento para tal.

Já na arqueologia é interessante a utilização do processo na reconstrução de artefatos arqueológicos. Neste procedimento, os fragmentos encontrados podem ser digitalizados e trabalhados por computador para a reconstrução mais precisa do objeto antigamente existente, como apresentado em [2].

1.1 CONTEXTUALIZAÇÃO

O projeto do CNPQ, intitulado "Museu Virtual 3D" entre a UnB e a UFPR, propõe a reconstrução tridimensional utilizando-se um *scanner* 3D e uma central inercial. Esta é utilizada para rastreamento dos movimentos realizados pelo sistema, como mostrado na Fig. 1.1. Devido a uma pequena experiência anterior com sistemas de navegação inercial, ver [3], pôde-se desenvolver um protótipo para este projeto e obter alguns resultados experimentais para análise.

Como verificado, no protótipo construído anteriormente, não foi possível a validação dos modelos de estimação de estados devido a inúmeros problemas encontrados na construção do mesmo. O protótipo

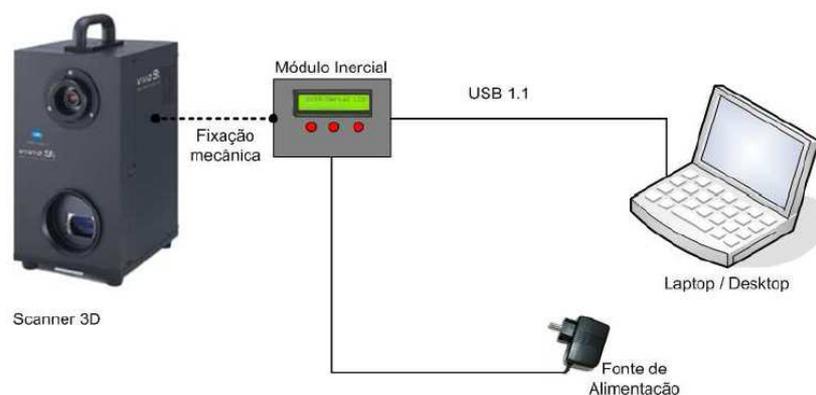


Figura 1.1: Esboço do sistema do scanner 3D com INS.

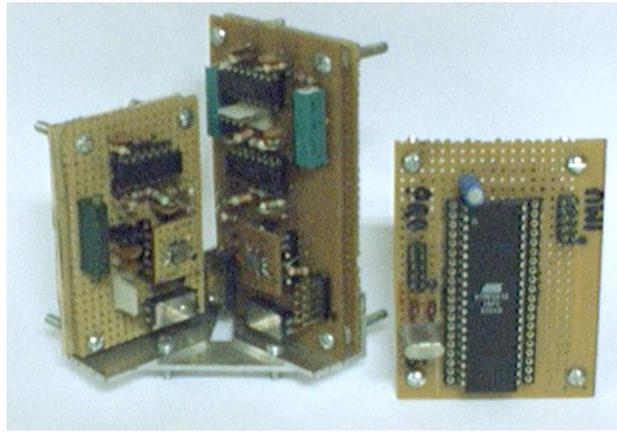


Figura 1.2: Foto do protótipo de IMU construído em projeto anterior

produzido possuía apenas os seis sensores dispostos ortogonalmente e enviava de forma analógica as informações à um microcontrolador. Este era responsável pela conversão dos sinais e envio destas conversões ao PC. A Fig. 1.2 mostra as placas implementadas, tanto a da unidade de medição inercial, ou IMU (do inglês *Inertial Measurement Unit*), que pode ser diferenciada por possuir duas placas fixadas ortogonalmente, quanto a do microcontrolador que executava as conversões A/D e enviava as mesmas ao PC. Diversos problemas encontravam-se nesta implementação, os principais foram:

- Comunicação com PC estava gerando grandes ruídos nas saídas analógicas dos sensores;
- Devido a má implementação dos filtros, com utilização de componentes sem precisão necessária, os sinais das medições chegavam aos conversores A/D muito alterados e não correspondiam fielmente ao que deveriam, acelerações e velocidades de rotação;
- As conexões entre as placas ortogonais e entre o IMU com o microcontrolador eram feitas por fios que estavam muito sujeitos a ruídos externos e como a conversão só era feita nas portas do microcontrolador os sinais chegavam com diversos ruídos indesejáveis.

Os problemas encontrados foram tratados, para que não se repetissem nesta nova implementação de sistema para navegação inercial. Algumas particularidades são diferentes nos dois protótipos, por exemplo, o fato de no primeiro não ser necessária a interação com o usuário e utilizar-se de fusão sensorial para melhorar a precisão, fato que não será tratado, à princípio, neste desenvolvimento.

1.2 DEFINIÇÃO DO PROBLEMA

Em reconstrução 3D, basicamente, são capturadas imagens do objeto, como as apresentadas na Fig. 1.3¹. Estas são adquiridas em diferentes posições e utilizando-se de algoritmos para a seleção do casamento dos pontos, e estratégias para a minimização dos erros, é possível a reconstrução do mesmo tridimensionalmente.

Para que os erros do casamento dos pontos seja reduzido faz-se necessário um bom conhecimento da posição e orientação do *scanner* 3D no momento da aquisição da imagem. Existem várias maneiras para o conhecimento destes parâmetros, uma delas pode ser através de alguma referência fixa, como nas mesas de reconstrução 3D. Onde estes parâmetros podem ser facilmente encontrados, já que existe uma referência e qualquer movimento executado pelo *scanner* pode ser relacionado a mesma.

¹Imagens adquiridas do material "Estimação Robusta Aplicada à Reconstrução 3D" do professor Geovanny Araújo Borges, ENE - UnB.

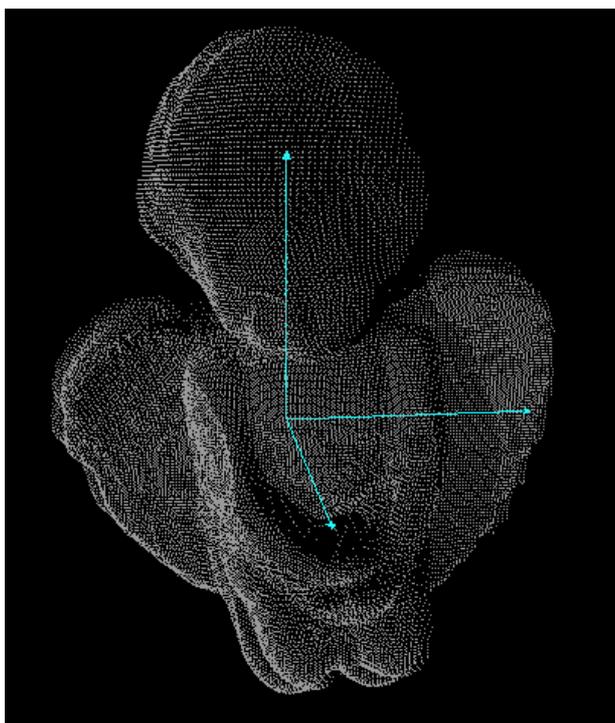


Figura 1.3: Imagens sobrepostas obtidas para reconstrução de objeto 3D

Porém, o fato do aparelho ficar fixo, preso, a uma mesa ou a alguma referência pode tornar-se num grande problema quando deseja-se digitalizar objetos no qual o acesso ou a movimentação do mesmo não é possível. Neste caso a implementação de um *scanner* 3D que possa ser manuseado facilmente e que não necessite de nenhuma fixação para determinação dos parâmetros de orientação e posição é de grande importância.

Mas como saber a orientação e posição de algum objeto sem se tê-lo preso a alguma referência? A resposta a esta questão pode ser através da utilização de informações inerciais do objeto. Informações como a aceleração e a força de *Coriolis* podem ser utilizadas para estimar estes parâmetros, sem que necessariamente o mesmo encontre-se preso.

1.3 OBJETIVOS DO PROJETO

O projeto propõe o desenvolvimento de uma central inercial, que utiliza uma IMU para a estimação dos parâmetros de posição e orientação do *scanner* 3D. Esse baseia-se na medição de acelerações e velocidades de rotação nos três eixos ortogonais. Estas informações podem ser integradas para estimar o estado atual do aparelho e propagar os erros de confiabilidade do sistema.

Algumas funcionalidades foram adicionadas para o bom funcionamento do sistema que são o acompanhamento através de um display LCD e a interação com o usuário. Esta se fará através de três botões, como mostrado na Fig. 1.4. Estes possibilitam a calibração dos sensores e o início e finalização da estimação do estado, que quando neste último não executa a propagação das incertezas, pois considera-se que o mesmo encontra-se estático.

Finalizando, a requisição das informações deve ser feita através da comunicação com o PC. Este deverá enviar mensagens de requisição para o sistema que deverá responder com os estados estimados juntamente com a informação de confiabilidade da estimação, covariância do estado. As requisições só serão respondidas quando a estimação dos parâmetros estiver parada.

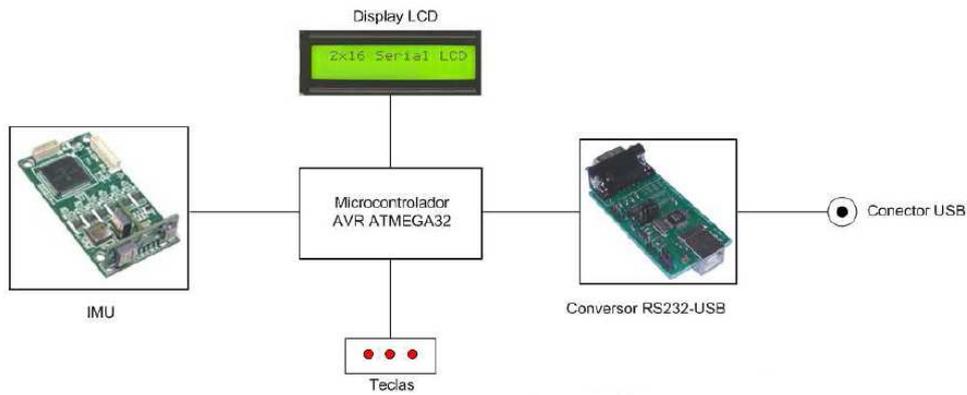


Figura 1.4: Esboço do Sistema.

1.4 APRESENTAÇÃO DO MANUSCRITO

No capítulo 2 são ministrados os principais assuntos relevantes ao tratamento do problema, como as transformações de coordenadas e as representações de orientação. Em seguida, o capítulo 3 descreve a metodologia empregada no desenvolvimento do projeto, como o planejamento e execução de simulações para a validação de métodos e modelos, a construção do protótipo da IMU e a integração dos sinais utilizando a estimação da atitude através de Quaternions. Os Resultados experimentais são discutidos também no capítulo 3 onde os gráficos encontrados pelas simulações e os testes feitos com o IMU são apresentados e analisados. Por fim, os projetos futuros e conclusões do projeto são mostradas no capítulo 4. Os anexos contém os esquemáticos e PCB das placas desenvolvidas.

2 FUNDAMENTOS TEÓRICOS

Neste capítulo serão abordados todos os assuntos importantes na implementação do IMU, como métodos de representações de orientação e processos estocásticos.

2.1 INTRODUÇÃO

Vários assuntos importantes para o tratamento do problema de estimação tiveram que ser estudados para que esta implementação fosse possível. As implementações de sistemas de navegação inercial, assim como a análise de comportamento de sistemas estocásticos e transformações de coordenadas foram muito relevantes para a resolução do problema e por isso estão em destaque na revisão bibliográfica.

Um outro assunto de extrema importância para uma boa estimação do estado do IMU foi a análise das diversas maneiras de se representar a orientação de um objeto, assim como os métodos de integração numérica utilizados. Estas maneiras estão diretamente conectadas aos problemas de propagação de erros e a acurácia do sistema.

2.2 SISTEMAS DE COORDENADAS

Os estados de navegação da central inercial são representados referenciados a algum sistema de coordenadas, o qual é definido de maneira diferente para cada implementação do sistema. Nesta implementação em questão foram utilizados dois sistemas de coordenadas, o sistema de coordenadas do corpo, ou **B** (do inglês *Body*), e o sistema de coordenadas fixas local, ou **W** (do inglês *World*).

2.2.1 Coordenadas fixas local

Para o tratamento do problema, é necessária somente a determinação da mudança de orientação e posição do *scanner* entre as imagens obtidas. Com isso, pode-se escolher um sistema de coordenadas fixo formado pelos eixos da primeira aquisição de imagem. Devido à maneira de calibração proposta para o sistema, apresentada em 3.7.1.1, a definição destes eixos de coordenadas deve-se, inicialmente, possuir o eixo Z_W na mesma direção e no sentido contrário da gravidade, como mostrado na Fig. 2.1, onde X_E , Y_E e Z_E definem um sistema de coordenadas ortogonal com ponto $(0, 0, 0)$ no centro da Terra.

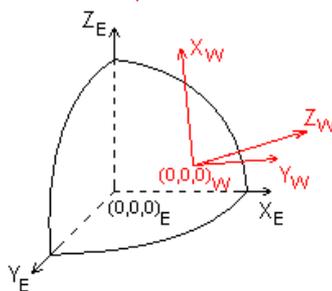


Figura 2.1: Representação do sistema de coordenadas fixas local

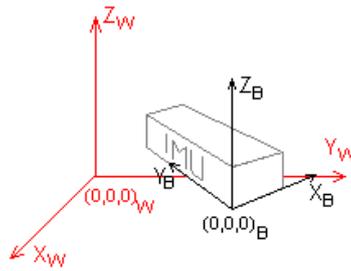


Figura 2.2: Representação do sistema de coordenadas do corpo

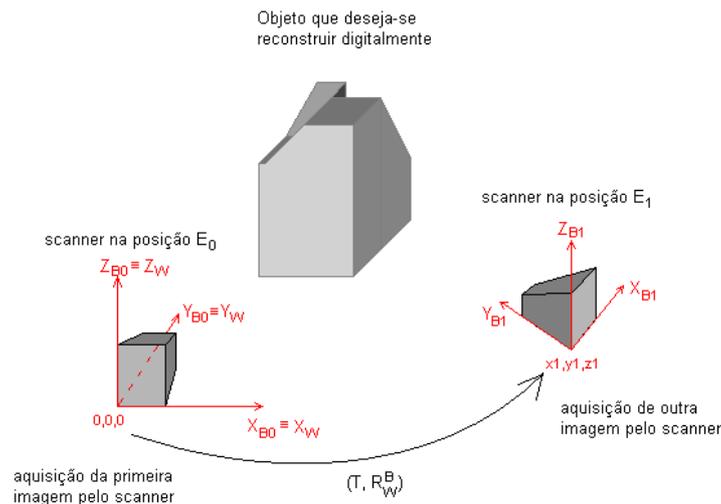


Figura 2.3: Transformação de coordenadas

2.2.2 Coordenadas do corpo

O sistema de coordenadas do corpo é fixo ao mesmo, isto quer dizer que qualquer movimento realizado no *scanner* será aplicado ao eixo de coordenadas do mesmo, valendo tanto para transformação de translação como para rotação. Este eixo de coordenadas está melhor representado na Fig. 2.2.

2.3 TRANSFORMAÇÃO DE COORDENADAS

As mudanças ocasionadas na posição e orientação do sistema entre as aquisições das imagens pode ser descrita por duas transformações. No caso, uma translação e uma rotação, as quais estão apresentadas na Fig. 2.3 por T e R_W^B . Estas transformações podem ser utilizadas para representar um ponto adquirido pelo sistema quando se encontrava no estado E_0 , apresentado na Fig. 2.3 no sistema de coordenadas B_0 ou W , para o sistema de coordenadas B_1 . Isto, possibilita que a reconstrução tridimensional do objeto seja possível com o casamento dos diversos pontos adquiridos.

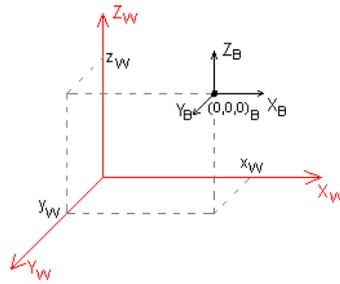


Figura 2.4: Representação de transformação de translação

2.4 REPRESENTAÇÃO DE TRANSLAÇÃO

A translação do sistema de coordenadas pode ser representada por três parâmetros. Isto deve-se pelo fato de ambos sistemas de coordenadas serem representações utilizando eixos ortogonais. Assim, pode-se apresentar a translação apenas com os valores da posição do ponto $(0, 0, 0)_B$ no sistema de coordenadas W , apresentado por $(x_W, y_W, z_W)_W$ na Fig. 2.4.

2.5 REPRESENTAÇÕES DE ORIENTAÇÃO

Retirando-se a translação ocasionada no *scanner*, pode-se representar a transformação de orientação de diversas formas. A mais popular destas é a que utiliza o método de Euler que provê uma descrição intuitiva da orientação, utilizando os ângulos de azimute ψ , elevação θ e rolagem ϕ . A seguir, têm-se outro método, que se tornou popular à pouco, conhecido por método de Quaternions, onde se utiliza quatro parâmetros para a representação da orientação, representando-se o eixo onde aplica-se a rotação em três dimensões e o ângulo da rotação. Um terceiro método seria utilizando a matriz de cossenos diretores, este método utiliza a determinação de uma matriz de transformação através dos ângulos de Euler ou Quaternions.

2.5.1 Representação de Orientação por Ângulos de Euler

A representação de orientação por ângulos de Euler é, simplificadamente, uma estratégia de utilizar uma sequência de três rotações para representar qualquer transformação de rotação possível. Fazendo assim com que os eixos, inicialmente iguais às dos eixos em W , fiquem coincidentes com o do corpo, sem translação, após as consecutivas rotações.

Várias maneiras de definir as três rotações são possíveis, e todas elas são equivalentes. Isto acontece porque por qualquer ordem de rotação que se utilize, contando que sejam em pelo menos dois eixos, a orientação final deve ser igual a orientação do objeto. Como mostrado em [4], várias convenções são utilizadas para a ordem das rotações, Euler adotou a convenção de utilizar as transformações em torno dos eixos do corpo na ordem Z_B , X_B e Z_B , verificando que após a primeira rotação o eixo X_B não coincide mais com o eixo X_W , e após a segunda, nem mesmo o eixo Z_B coincide mais com Z_W , isto, claro, não considerando rotações de 2π . Neste trabalho será adotada a mesma convenção utilizada em engenharia aeroespacial, na qual a ordem das rotações ocorre primeiro pelo eixo Z_B , depois em Y_B e por fim em X_B .

A primeira rotação, mostrada na Fig. 2.5(a), é aplicada através da matriz:

$$\mathbf{R}_Z(\phi) = \begin{bmatrix} \cos(\phi) & \text{sen}(\phi) & 0 \\ -\text{sen}(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

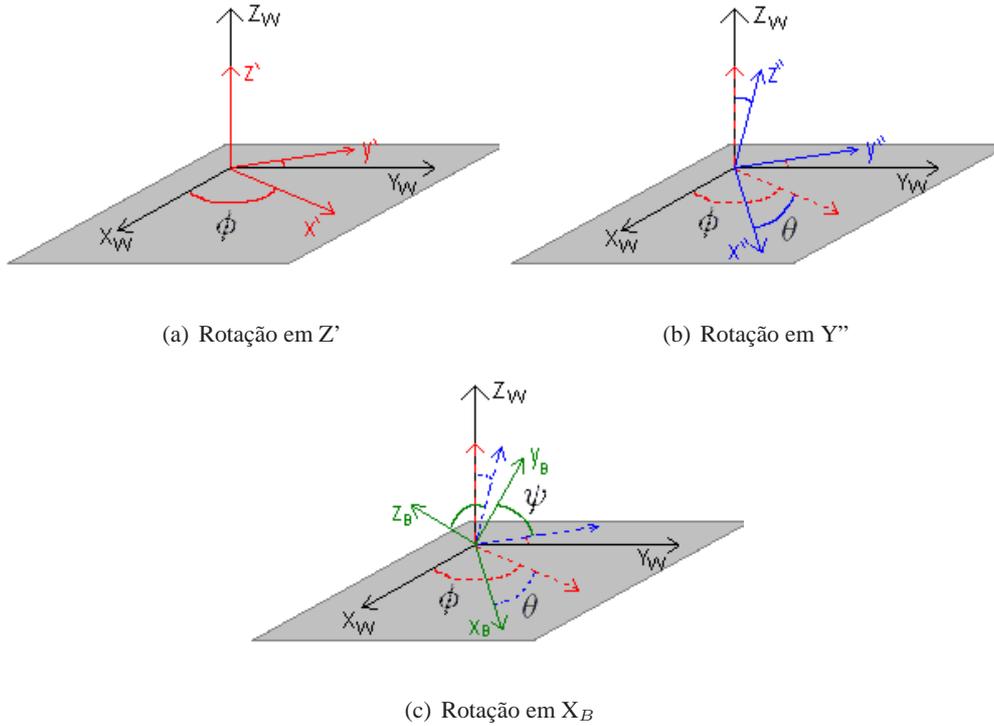


Figura 2.5: Ângulos de Euler pela convenção utilizada em Engenharia Aeroespacial

esta rotação é feita mantendo-se o eixo Z' , apresentado na Fig. 2.5(a), do corpo fixo e rotacionando o resto um ângulo ϕ . Em seguida, outra rotação de valor θ é aplicada, agora no eixo Y'' , apresentado na Fig. 2.5(b), do corpo, verificando que o eixo Y'' agora não coincide mais com o eixo Y_W fixo. Tal transformação é aplicada pela matriz:

$$\mathbf{R}_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.2)$$

após esta rotação, e por fim, aplica-se a rotação de valor ψ no eixo X_B do corpo com a matriz:

$$\mathbf{R}_X(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix} \quad (2.3)$$

a qual, finalmente, resulta na orientação desejada para o corpo, $[X_B \ Y_B \ Z_B]$. A ordem de aplicação destas rotações altera completamente o resultado final, isto significa que a comutatividade não existe para estas transformações. Percebe-se com isto que é de extrema importância a convenção da ordem em que as rotações são aplicadas sobre o corpo.

2.5.1.1 Matriz de Transformação de Cossenos Diretores utilizando Ângulos de Euler

Para trabalhar com variações na orientação de um objeto, frequentemente são utilizadas transformações das coordenadas do corpo para coordenadas fixas, uma maneira conveniente de transformação entre duas coordenadas é feita através da matriz de cossenos diretores (ou DCM do inglês *Direction Cosine Matrix*), como apresentado em [4].

Usando notação de matriz e os cossenos diretores $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ a transformação entre coordenadas pode ser

representada como:

$$\begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} = \mathbf{R}_B^W \cdot \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \cdot \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad (2.4)$$

Os cossenos diretores em termos dos ângulos de Euler, definidos como na Fig. 2.5, são definidos por:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cdot \cos(\psi) \\ \cos(\theta) \cdot \text{sen}(\psi) \\ -\text{sen}(\theta) \end{bmatrix} \quad (2.5)$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \text{sen}(\phi) \cdot \text{sen}(\theta) \cdot \cos(\psi) - \cos(\phi) \cdot \text{sen}(\psi) \\ \text{sen}(\phi) \cdot \text{sen}(\theta) \cdot \text{sen}(\psi) + \cos(\phi) \cdot \cos(\psi) \\ \text{sen}(\phi) \cdot \cos(\theta) \end{bmatrix} \quad (2.6)$$

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi) \cdot \text{sen}(\theta) \cdot \cos(\psi) + \text{sen}(\phi) \cdot \text{sen}(\psi) \\ \cos(\phi) \cdot \text{sen}(\theta) \cdot \text{sen}(\psi) - \text{sen}(\phi) \cdot \cos(\psi) \\ \cos(\phi) \cdot \cos(\theta) \end{bmatrix} \quad (2.7)$$

e são encontrados a partir das rotações na ordem utilizada na Fig. 2.5. Com a ordem correta da aplicação das rotações pode-se obter os vetores coluna através de

$$\mathbf{R}_B^W = \mathbf{R}_Z(\phi) \cdot \mathbf{R}_Y(\theta) \cdot \mathbf{R}_X(\psi) \quad (2.8)$$

Pelo fato dos cossenos diretores serem ortonormais a transformação inversa pode ser feita somente utilizando a transposta da matriz como:

$$(\mathbf{R}_B^W)^{-1} = \mathbf{R}_W^B = (\mathbf{R}_B^W)^T \quad (2.9)$$

esta transformação representa a transformação de coordenadas agora dos eixos globais para os eixos do corpo (\mathbf{R}_W^B).

2.5.2 Representação de Orientação por Quaternions

Segundo [5], *William Rowan Hamilton* propôs os quaternions quando pesquisava sobre álgebras que respeitavam o produto

$$|u \cdot v| = |u| \cdot |v| \quad (2.10)$$

com dimensões arbitrárias, em busca de uma generalização para os números complexos.

Os quaternions fazem parte dos números hipercomplexos, como apresentado em [6], e apesar de não apresentarem a propriedade comutativa, possuem a associativa.

2.5.2.1 Álgebra de Quaternions

Em analogia com os números complexos, os quaternions também podem ser escritos na maneira de soma de partes reais e imaginárias, como mostrado para os números complexos por:

$$\mathbf{Z} = z_1 \cdot \mathbf{1} + z_2 \cdot \mathbf{i} \quad (2.11)$$

e para os quaternions por:

$$\mathbf{Q} = q_1 \cdot \mathbf{1} + q_2 \cdot \mathbf{i} + q_3 \cdot \mathbf{j} + q_4 \cdot \mathbf{k} \quad (2.12)$$

sendo que a equação fundamental da álgebra dos quaternions se baseia em:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -\mathbf{1} \quad (2.13)$$

A representação do hipercomplexo \mathbf{Q} pode ser também expressada apenas pelos coeficientes como:

$$\mathbf{Q} = [q_1 \quad q_2 \quad q_3 \quad q_4] \quad (2.14)$$

ou pela seguinte matriz de complexos:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Z} & \mathbf{W} \\ -\overline{\mathbf{W}} & -\overline{\mathbf{Z}} \end{bmatrix} = \begin{bmatrix} q_1 \cdot \mathbf{1} + q_2 \cdot \mathbf{i} & q_3 \cdot \mathbf{1} + q_4 \cdot \mathbf{i} \\ -q_3 \cdot \mathbf{1} + q_4 \cdot \mathbf{i} & q_1 \cdot \mathbf{1} - q_2 \cdot \mathbf{i} \end{bmatrix} \quad (2.15)$$

Outra representação seria através de matrizes complexas 2×2 , apresentado por:

$$\mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.16)$$

$$\mathbf{I} = \begin{bmatrix} \mathbf{i} & 0 \\ 0 & -\mathbf{i} \end{bmatrix} \quad (2.17)$$

$$\mathbf{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2.18)$$

$$\mathbf{K} = \begin{bmatrix} 0 & \mathbf{i} \\ \mathbf{i} & 0 \end{bmatrix} \quad (2.19)$$

onde \mathbf{U} é a matriz identidade neste caso e não \mathbf{I} como normalmente é feito. Com as definições destas matrizes, pode-se chegar as identidades:

$$\mathbf{I}^2 = -\mathbf{U} \quad (2.20)$$

$$\mathbf{J}^2 = -\mathbf{U} \quad (2.21)$$

$$\mathbf{K}^2 = -\mathbf{U} \quad (2.22)$$

O conjugado de um quaternion é definido por:

$$\overline{\mathbf{Q}} = q_1 \cdot \mathbf{1} - q_2 \cdot \mathbf{i} - q_3 \cdot \mathbf{j} - q_4 \cdot \mathbf{k} \quad (2.23)$$

A soma de dois quaternions pode ser calculada como:

$$\mathbf{Q}_a + \mathbf{Q}_b = (q_{a1} + q_{b1}) \cdot \mathbf{1} + (q_{a2} + q_{b2}) \cdot \mathbf{i} + (q_{a3} + q_{b3}) \cdot \mathbf{j} + (q_{a4} + q_{b4}) \cdot \mathbf{k} \quad (2.24)$$

e o produto por:

$$\mathbf{Q}_a \otimes \mathbf{Q}_b = \begin{aligned} & (q_{a1} \cdot q_{b1} - q_{a2} \cdot q_{b2} - q_{a3} \cdot q_{b3} - q_{a4} \cdot q_{b4}) \cdot \mathbf{1} + \\ & (q_{a1} \cdot q_{b2} + q_{a2} \cdot q_{b1} + q_{a3} \cdot q_{b4} - q_{a4} \cdot q_{b3}) \cdot \mathbf{i} + \\ & (q_{a1} \cdot q_{b3} - q_{a2} \cdot q_{b4} + q_{a3} \cdot q_{b1} + q_{a4} \cdot q_{b2}) \cdot \mathbf{j} + \\ & (q_{a1} \cdot q_{b4} + q_{a2} \cdot q_{b3} - q_{a3} \cdot q_{b2} + q_{a4} \cdot q_{b1}) \cdot \mathbf{k} \end{aligned} \quad (2.25)$$

A norma de um quaternion pode ser calculada por:

$$n(\mathbf{Q}) = \sqrt{\mathbf{Q} \cdot \overline{\mathbf{Q}}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (2.26)$$

e possui a propriedade multiplicativa:

$$n(\mathbf{Q}_a \cdot \mathbf{Q}_b) = n(\mathbf{Q}_a) \cdot n(\mathbf{Q}_b) \quad (2.27)$$

Uma utilização interessante para o produto das normas é que este dá imediatamente a identidade de quatro quadrados de Euler, ver em [6], no qual muito se baseou a pesquisa de *Hamilton*, como pode ser visto em [5].

Quaternions também podem ser interpretados como um escalar e um vetor como:

$$\mathbf{Q} = q_1 \cdot \mathbf{1} + q_2 \cdot \mathbf{i} + q_3 \cdot \mathbf{j} + q_4 \cdot \mathbf{k} = (q_1, \mathbf{v}) \quad (2.28)$$

$$\mathbf{v} = [q_2 \quad q_3 \quad q_4] \quad (2.29)$$

com isto a multiplicação pode ser simplificada para:

$$\mathbf{Q}_a \otimes \mathbf{Q}_b = (q_{a1}, \mathbf{v}_a) \otimes (q_{b1}, \mathbf{v}_b) = (q_{a1} \cdot q_{b1} - \mathbf{v}_a \cdot \mathbf{v}_b, q_{a1} \cdot \mathbf{v}_b + q_{b1} \cdot \mathbf{v}_a + \mathbf{v}_a \times \mathbf{v}_b) \quad (2.30)$$

2.5.2.2 Rotações utilizando Quaternions

O método de especificar rotações e orientações de coordenadas computacionalmente utilizando quaternions foi introduzido por *Shoemake* em sua publicação de 1985, como dito em [7], a utilização desta representação provê grandes vantagens computacionais, pois utiliza funções lineares e o custo computacional para os cálculos é menor quando comparado aos ângulos de Euler.

A multiplicação de dois quaternions resulta na orientação representada por \mathbf{Q}_a , referenciando-se a Eq. 2.30, para a nova orientação acumulativa de \mathbf{Q}_a e \mathbf{Q}_b em termos de quaternions, como apresentado em [4]. Exemplificando, se \mathbf{Q}_b for um quaternion representando uma rotação baseada num intervalo, e \mathbf{Q}_a a acumulação destas rotações, então a multiplicação repetida destes quaternions resultará numa rotação suave de vários *frames*.

Assim, a rotação de um ângulo θ em torno de um vetor \mathbf{v} pode ser aplicada em um ponto \mathbf{p} através da multiplicação:

$$\mathbf{p}' = \mathbf{Q} \otimes (0, \mathbf{p}) \otimes \overline{\mathbf{Q}} \quad (2.31)$$

onde \mathbf{Q} é definido por:

$$\mathbf{Q} = (\cos(1/2 \cdot \theta), \mathbf{v} \cdot \sin(1/2 \cdot \theta)) \quad (2.32)$$

2.5.2.3 Matriz de Transformação de Cossenos Diretores utilizando Quaternions

Desenvolvendo-se a Eq. 2.31 pode-se chegar a DCM equivalente para a representação por quaternions, como:

$$\mathbf{R}_B^W(\mathbf{Q}) = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2 \cdot (q_2 \cdot q_3 + q_1 \cdot q_4) & 2 \cdot (q_2 \cdot q_4 - q_1 \cdot q_3) \\ 2 \cdot (q_2 \cdot q_3 - q_1 \cdot q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2 \cdot (q_3 \cdot q_4 + q_1 \cdot q_2) \\ 2 \cdot (q_2 \cdot q_4 + q_1 \cdot q_3) & 2 \cdot (q_3 \cdot q_4 - q_1 \cdot q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \quad (2.33)$$

esta implementação, apresentada em [8], não utiliza funções harmônicas como na implementação por ângulos de Euler e por este motivo reduz muito o tempo gasto com a geração da mesma.

Isto determina uma das grandes vantagens da utilização de quaternions sobre ângulos de Euler. Soamente quando o sistema necessita que os ângulos de Euler sejam informados esta vantagem deixa de existir, pois a transformação de quaternions para Euler custa muito computacionalmente, como visto em [4].

Na DCM apresentada na Eq. 2.33 o quaternion \mathbf{Q} deve ter norma unitária para que a transformação desejada seja aplicada. Apesar de ter-se utilizado um método diferente para determinação da DCM, a inversa mostrada na Eq. 2.9, ainda é válida pois os cossenos diretores não deixam de ser ortonormais.

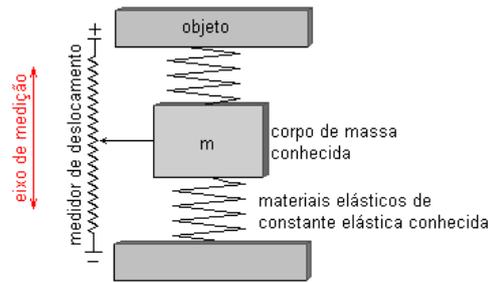


Figura 2.6: Design básico de um acelerômetro.

2.6 SENSORES INERCIAIS

Como apresentado em [9], o design de sensores inerciais é limitado somente pelas leis da física e a imaginação humana. Com isso, existem diversas implementações diferentes de giroscópios e acelerômetros, não necessariamente sendo todos utilizados para navegação inercial.

2.6.1 Girômetros

Os girômetros são utilizados em várias aplicações para medir o ângulo de rotação num veículo ou estrutura (*displacement gyroscopes*) ou, mais comumente, para medir a variação de rotação em torno de algum eixo definido (*rate gyroscopes*). A forma mais básica e original utiliza as propriedades inerciais de uma roda, ou rotor, girando em alta velocidade. Neste estado, o eixo de rotação continua apontando para a mesma direção independente de qualquer rotação que ocorra na base do sensor.

2.6.2 Acelerômetros

A grande vantagem de se utilizar medições de aceleração em relação a velocidade ou a posição está no fato desta não necessitar de referência externa e poder ser mensurada internamente. O princípio físico básico deste dispositivo baseia-se na famosa segunda lei de Newton que diz:

$$\mathbf{F} = m \cdot a \quad (2.34)$$

onde \mathbf{F} seria uma força aplicada ao corpo, m a massa e a a aceleração do mesmo. Com isto, pode-se, utilizando um pequeno corpo de massa conhecida, determinar qual o valor de sua aceleração. Para isto, prende-se o mesmo ao objeto através de materiais elásticos. Assim, pode-se determinar a força, e conseqüentemente a aceleração, conhecendo o deslocamento da massa em relação ao objeto. Este design do dispositivo pode ser melhor entendido com a Fig. 2.6.

2.7 SISTEMAS DE NAVEGAÇÃO INERCIAL

O desenvolvimento da navegação inercial possui um histórico relativamente curto, porém muito intenso, grande parte ocorreu durante a evolução da Guerra Fria, como explicitado por [9]. Desenvolveram-se, basicamente, duas implementações de Sistemas de Navegação Inercial, ou INS (do inglês *Inertial Navigation System*), mais conhecidas por *Gimbaled* e *Strapdown*.

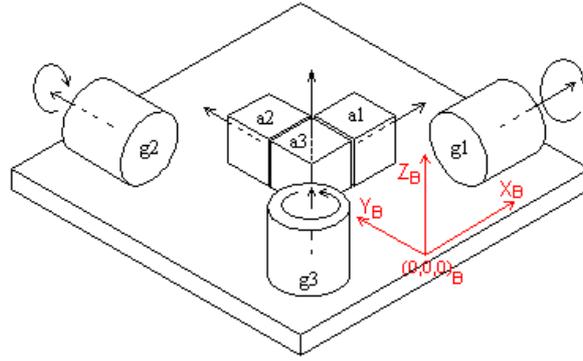


Figura 2.7: Montagem dos sensores no IMU utilizado no *Strapdown* INS.

2.7.1 Gimbaled INS

Esta tecnologia baseia-se em isolar a IMU das rotações da base em que está montada, para que assim as acelerações medidas sejam em atitudes conhecidas. Esta implementação foi muito utilizada nos primeiros INS's que surgiram pois possuíam baixa capacidade de processamento e portanto não poderiam realizar os cálculos necessários à tecnologia *Strapdown*.

2.7.2 Strapdown INS

A partir do desenvolvimento dos circuitos integrados, por volta de 1960 e 1970, foi possível a utilização de modelos onde o IMU estava fixado e rodava juntamente com a base em que estava montado. Na Fig. 2.7 é apresentada a configuração dos sensores de um IMU utilizado no INS *Strapdown*, onde a1, a2 e a3 são acelerômetros e g1, g2 e g3 girômetros.

Esta necessidade de melhoria do poder de processamento dos circuitos deve-se ao fato de se aplicar rotações matriciais. As quais precisam de grande capacidade de processamento para serem aplicadas rapidamente às medições obtidas pela IMU e, portanto, não necessitam mais de isolamento das rotações da base. Esta implementação, onde a IMU não se isola das rotações da base, é chamada *Strapdown*.

Atualmente é a configuração dos INS's mais utilizada, tanto devido pela sua facilidade de montagem física, quanto pelo baixo custo de fabricação se comparada à *Gimbaled*.

2.7.2.1 Relação dos Ângulos de Euler com as Medições dos Girômetros

Com as informações provenientes dos sensores posicionados em direções ortogonais, como na montagem *Strapdown*, é possível adquirir todas as medições necessárias para uma estimação da atitude do IMU. Isto pode ser obtido utilizando-se três girômetros, um posicionado ortogonalmente ao outro. Analisando o esquema apresentado na Fig. 2.7 pode-se visualizar melhor a idéia de posicionamento dos sensores. Assim, percebe-se que nesta configuração as medições são referentes à eixos fixos ao corpo (**B**) e não aos eixos globais (**W**).

Os ângulos de Euler ϕ , θ e ψ são atualizados com as informações dos girômetros a partir das seguintes equações:

$$\dot{\phi} = \omega_1 + \omega_2 \cdot \text{sen}(\phi) \cdot \text{tan}(\theta) + \omega_3 \cdot \text{cos}(\phi) \cdot \text{tan}(\theta) \quad (2.35)$$

$$\dot{\theta} = \omega_2 \cdot \text{cos}(\phi) - \omega_3 \cdot \text{sen}(\phi) \quad (2.36)$$

$$\dot{\psi} = \omega_2 \cdot \text{sen}(\phi) \cdot \text{sec}(\theta) + \omega_3 \cdot \text{cos}(\phi) \cdot \text{sec}(\theta) \quad (2.37)$$

que são conhecidas como equações *Gimbal*. Analisando estas equações, percebe-se que quando θ encontra-se perto da vertical, isto é, próximo de valores iguais a $\pm\pi/2$, ϕ e ψ tornam-se infinitos. Este problema é conhecido como *Gimbal Lock* e deve ser tratado pois leva a problemas numéricos.

2.7.2.2 Relação dos Quaternions com as Medições dos Girômetros

Como mostrado em [4] e [8], as medições provenientes dos girômetros podem ser relacionadas diretamente as variações dos quaternions, como:

$$\dot{q}_1 = -1/2 \cdot (q_2 \cdot \omega_1 + q_3 \cdot \omega_2 + q_4 \cdot \omega_3) \quad (2.38)$$

$$\dot{q}_2 = 1/2 \cdot (q_1 \cdot \omega_1 + q_3 \cdot \omega_3 - q_4 \cdot \omega_2) \quad (2.39)$$

$$\dot{q}_3 = 1/2 \cdot (q_1 \cdot \omega_2 + q_4 \cdot \omega_1 - q_2 \cdot \omega_3) \quad (2.40)$$

$$\dot{q}_4 = 1/2 \cdot (q_1 \cdot \omega_3 + q_2 \cdot \omega_2 - q_3 \cdot \omega_1) \quad (2.41)$$

Tendo conhecimento da variação dos quaternions em algum intervalo de tempo conhecido, a orientação pode ser estimada facilmente através de algum método de integração numérica. Com a atualização do quaternion, este pode deixar de ter norma unitária e devido a este fato um termo adicional é inserido na formulação das Eq.'s 2.38, 2.39, 2.40 e 2.41, resultando em:

$$\dot{q}_1 = -1/2 \cdot (q_2 \cdot \omega_1 + q_3 \cdot \omega_2 + q_4 \cdot \omega_3) + \lambda \cdot q_1 \quad (2.42)$$

$$\dot{q}_2 = 1/2 \cdot (q_1 \cdot \omega_1 + q_3 \cdot \omega_3 - q_4 \cdot \omega_2) + \lambda \cdot q_2 \quad (2.43)$$

$$\dot{q}_3 = 1/2 \cdot (q_1 \cdot \omega_2 + q_4 \cdot \omega_1 - q_2 \cdot \omega_3) + \lambda \cdot q_3 \quad (2.44)$$

$$\dot{q}_4 = 1/2 \cdot (q_1 \cdot \omega_3 + q_2 \cdot \omega_2 - q_3 \cdot \omega_1) + \lambda \cdot q_4 \quad (2.45)$$

com λ , como mostrado em [4], sendo:

$$\lambda = 1 - (q_1^2 + q_2^2 + q_3^2 + q_4^2) \quad (2.46)$$

Outras maneiras de se retirar este erro da integração são mostrados em [8], uma delas seria a normalização do quaternion a partir do momento que este passa de uma tolerância pré-estabelecida.

Percebe-se que pelas equações de atualização dos quaternions que o problema de *Gimbal Lock* não é encontrado nesta representação. Isto é outra grande vantagem quando comparado à parametrização de Euler.

2.8 INCERTEZAS DO SISTEMA

Como a explicação apresentada em [10], praticamente todos sistemas estão sujeitos a influências complicadas e não inteiramente conhecidas. Geralmente, quando se trabalha com vários graus de liberdade, é difícil se predizer com precisão arbitrária o estado do sistema em cada instante. Assim, o resultado de uma experiência, realizada praticamente com as mesmas condições de outra, retorna as medições de um dado observável diferente da primeira.

Apesar desta imprevisibilidade, percebeu-se que quando repetido várias vezes o mesmo experimento uma certa regularidade pode ser notada. Isto é utilizado para caracterizar os sistemas e estudar-los para obter-se uma melhor estimativa dos estados do mesmo.

Um processo estocástico (PE) se caracteriza por ser uma função aleatória do tempo, um exemplo no tempo discreto pode ser a sequência dos resultados de diversos lançamentos de um dado.

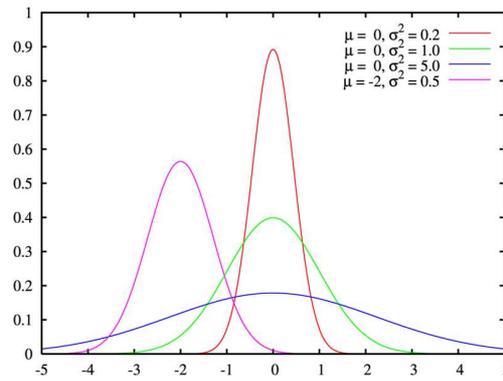


Figura 2.8: Exemplos de funções de densidade de Distribuições Gaussianas

2.8.1 Processos Gaussianos

Um PE é *Gaussiano*, ou normal, se a lei de probabilidade é normal. Isto significa, que se a regularidade do sistema, comentada acima quando executa-se várias vezes o experimento, se caracterizar como uma distribuição normal. Como visto em [11], as distribuições *Gaussianas* podem ser representadas por dois parâmetros, o valor médio (μ) e a variância (σ^2), definidas por:

$$\mu = \xi[x] \quad (2.47)$$

$$\sigma^2 = \xi[(x - \xi[x])^2] \quad (2.48)$$

onde $\xi[x]$ é a esperança de x , função que significa o valor esperado para a variável aleatória.

O valor esperado de x , ou $\xi[x]$, possui as seguintes propriedades:

$$\xi[x + y] = \xi[x] + \xi[y] \quad (2.49)$$

$$\xi[a \cdot x] = a \cdot \xi[x] \quad (2.50)$$

$$\xi[a + x] = a + \xi[x] \quad (2.51)$$

sendo x e y variáveis aleatórias e a sendo uma constante.

Nesta mesma distribuição, a função de densidade das diversas amostras colhidas com a execução de várias vezes do experimento, é apresentada como na Fig. 2.8¹. Como pode ser visto, esta distribuição possui uma propriedade interessante, que é o fato da função de densidade ser simétrica. Outra propriedade é de que qualquer processamento linear de um processo *Gaussiano*, resulta num processo *Gaussiano*, propriedade muito importante para a propagação das incertezas.

2.8.2 Propagação das Incertezas de Estimações dos Estados do IMU

Devido a singularidade encontrada na representação por ângulos de Euler, *Gimbal Lock*, e a maior demora no cálculo de funções harmônicas, foi escolhida a representação dos estados de orientação através de quaternions.

Com isto, como analisado em [12], pode-se analisar o erro de estimação por:

$$\mathbf{Q} = \hat{\mathbf{Q}} \otimes \mathbf{Q}_e \quad (2.52)$$

onde considera-se \mathbf{Q} como o estado real de orientação, $\hat{\mathbf{Q}}$ como o estado estimado e \mathbf{Q}_e como o erro de rotação. Com esta notação, \mathbf{Q}_e representa uma pequena rotação aplicada a $\hat{\mathbf{Q}}$ para que o mesmo seja igual a \mathbf{Q} .

¹Ilustração retirada do site http://en.wikipedia.org/wiki/Image:Normal_distribution_pdf.png

Assim, utilizando-se as Eq.'s 2.42, 2.43, 2.44 e 2.45, sem a compensação não-linear e pelo método de integração numérico de Euler², a atualização pode ser considerada linear e a propagação das incertezas serem facilmente calculadas. Para estes cálculos considera-se que as informações provenientes dos sensores são *Gaussianas* e sabendo-se que $\xi[x]$ possui as propriedades apresentadas nas Eq.'s 2.49, 2.50 e 2.51 e que σ^2 de uma constante é igual a *zero*, pode-se, baseado nas Eq.'s 2.42, 2.43, 2.44 e 2.45, encontrar \mathbf{Q}_e através de:

$$\mathbf{Q}_{e(n)} = \begin{bmatrix} q_{e1(n)} \\ q_{e2(n)} \\ q_{e3(n)} \\ q_{e4(n)} \end{bmatrix} = \begin{bmatrix} q_{e1(n-1)} - T^2/4 \cdot (\hat{q}_2^2 \cdot \sigma_{\omega_1}^2 + \hat{q}_3^2 \cdot \sigma_{\omega_2}^2 + \hat{q}_4^2 \cdot \sigma_{\omega_3}^2) \\ q_{e2(n-1)} + T^2/4 \cdot (\hat{q}_1^2 \cdot \sigma_{\omega_1}^2 + \hat{q}_3^2 \cdot \sigma_{\omega_3}^2 - \hat{q}_4^2 \cdot \sigma_{\omega_2}^2) \\ q_{e3(n-1)} + T^2/4 \cdot (\hat{q}_1^2 \cdot \sigma_{\omega_2}^2 + \hat{q}_4^2 \cdot \sigma_{\omega_1}^2 - \hat{q}_2^2 \cdot \sigma_{\omega_3}^2) \\ q_{e4(n-1)} + T^2/4 \cdot (\hat{q}_1^2 \cdot \sigma_{\omega_3}^2 + \hat{q}_2^2 \cdot \sigma_{\omega_2}^2 - \hat{q}_3^2 \cdot \sigma_{\omega_1}^2) \end{bmatrix} \quad (2.53)$$

onde $\sigma_{\omega_i}^2$ representa a variância do girômetro i e T o período de amostragem do sistema.

Este erro \mathbf{Q}_e é uma perturbação na DCM e pode ser expressado por:

$$\delta \mathbf{R}_B^W(\mathbf{Q}) \equiv \mathbf{R}_B^W(\mathbf{Q}_e) \quad (2.54)$$

$$\mathbf{R}_B^W = \delta \mathbf{R}_B^W \times \hat{\mathbf{R}}_B^W \quad (2.55)$$

²Método de integração numérica de primeira ordem.

3 DESENVOLVIMENTO E RESULTADOS OBTIDOS

Neste capítulo é apresentada a metodologia e os resultados obtidos em cada passo do trabalho.

3.1 INTRODUÇÃO

Como apresentado na seção 1.3, a central inercial deverá estimar a orientação e a posição do IMU, além de possuir interação com o usuário e comunicação com PC. Para validação da estimação do estado, foram feitas simulações com a propagação dos estados e das incertezas em movimentos pré-estabelecidos. Após a validação do modelo que será utilizado, começou-se a construção do protótipo de um IMU, utilizando três girômetros e três acelerômetros.

Por fim, estabelecida a construção do protótipo, modelou-se os possíveis estados do sistema e começou-se o desenvolvimento da placa-mãe para estimação dos parâmetros, interação com o usuário e comunicação com o PC. Com tudo funcionando satisfatoriamente foi implementada uma biblioteca no PC para aquisição de informações do sistema e um programa para demonstrar os movimentos executados com a central inercial.

3.2 SIMULAÇÃO DE IMU

No sistema desenvolvido, as medições dos sensores podem ser implementadas por velocidades angulares e acelerações com um valor aleatório associado. Essa modelagem dos sensores, de maneira simplista, fica:

$$\check{m} = m + \delta m \quad (3.1)$$

em que \check{m} é o valor medido, m o valor real e δm uma variável aleatória de valor médio nulo e que segue distribuição normal e variância de σ_m^2 :

$$\delta m \sim N(0, \sigma_m^2) \quad (3.2)$$

Vários outros parâmetros podem ser utilizados para a modelagem dos sensores, como os apresentados em [13], porém estes foram omitidos nesta primeira análise do problema. Assim, utilizando-se a modelagem proposta, pode-se definir $\check{\Omega}_B$ como:

$$\check{\Omega}_B = \begin{bmatrix} \check{\omega}_1 \\ \check{\omega}_2 \\ \check{\omega}_3 \end{bmatrix} \quad (3.3)$$

e sabendo-se as condições iniciais de orientação pode-se estimar os parâmetros da mesma por:

$$\dot{\mathbf{Q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = -\frac{1}{2} \cdot \begin{bmatrix} q_2 & q_3 & q_4 \\ -q_1 & q_4 & -q_3 \\ -q_4 & -q_1 & q_2 \\ q_3 & -q_2 & -q_1 \end{bmatrix} \times \check{\Omega}_B + \lambda \cdot \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (3.4)$$

retirado do desenvolvimento das Eq.'s 2.42, 2.43, 2.44 e 2.45, e apresentando λ pela Eq. 2.46.

Tendo-se estimado \mathbf{Q} pode-se utilizar a mesma modelagem para $\check{\mathbf{A}}_B$:

$$\check{\mathbf{A}}_B = \begin{bmatrix} \check{a}_1 \\ \check{a}_2 \\ \check{a}_3 \end{bmatrix} \quad (3.5)$$

e com isso, pode-se estimar a posição através da aplicação de $\mathbf{R}_B^W(\mathbf{Q})$ em $\check{\mathbf{A}}_B$. Sendo $\mathbf{R}_B^W(\mathbf{Q})$ calculado com \mathbf{Q} estimado no instante da aquisição das medições dos acelerômetros. Esta aplicação da transformação de coordenadas passa as medições para o eixo \mathbf{W} , assim a estimação da velocidade e posição do corpo pode ser feita por:

$$\begin{bmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{V}} \end{bmatrix} = \begin{bmatrix} \mathbf{V} \\ \mathbf{R}_B^W(\mathbf{Q}) \times \check{\mathbf{A}}_B - \mathbf{G} \end{bmatrix} \quad (3.6)$$

a qual também pode ser calculada por método de integração numérica, onde \mathbf{V} é o vetor de velocidades, \mathbf{X} o de posição e \mathbf{G} sendo um vetor definido por:

$$\mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (3.7)$$

com g igual ao valor da gravidade local.

3.2.1 Validação do modelo

Antes de começar o desenvolvimento em si do IMU, fez-se algumas simulações, utilizando-se a plataforma do MatLab, para validação dos modelos que serão utilizados na implementação. Estas simulações levaram em conta apenas os erros de incertezas das medições dos sensores, sem considerar qualquer problemas de desalinhamentos ou ruídos nas medições causados por problemas externos ao sistema.

3.2.1.1 Validação da estimação de orientação

Pelo fato de ter apresentado muitas desvantagens, a implementação por ângulos de Euler nem foi desenvolvida. A singularidade, *Gimbal Lock*, existente nesta representação, e as grandes não linearidades, com relação a propagação das incertezas, fez com que esta representação fosse descartada da implementação. Portanto, a única simulação feita foi a que utiliza quaternions.

Na simulação, a trajetória utilizada foi escolhida de maneira que a orientação do objeto, em quaternions, tenha como ser analisada analiticamente, assim como sua derivada. Isto torna-se necessário pelo fato de que deseja-se analisar os erros de estimação e os mesmos só podem ser analisados corretamente se nenhum método numérico de integração for utilizado na trajetória real.

Para validação da estimação de orientação, foi implementado em MatLab uma simulação onde variações conhecidas de orientação foram aplicadas num corpo e utilizando quaternions estimou-se o estado e a propagação das incertezas. Foram analisados se os erros de estimação encontravam-se dentro do intervalo de confiança de $\pm 3\sigma$. O qual foi encontrado fazendo-se a propagação das incertezas como apresentado na sessão 2.8.2.

A Fig. 3.1 mostra o fluxograma da simulação de estimação por quaternions, onde as medições dos girômetros foram criadas pela equação:

$$\check{\mathbf{\Omega}}_B = 2 \cdot \begin{bmatrix} q_1 \cdot \dot{q}_2 - q_2 \cdot \dot{q}_1 - q_4 \cdot \dot{q}_3 + q_3 \cdot \dot{q}_4 \\ q_1 \cdot \dot{q}_3 - q_3 \cdot \dot{q}_1 - q_2 \cdot \dot{q}_4 + q_4 \cdot \dot{q}_2 \\ q_1 \cdot \dot{q}_4 - q_4 \cdot \dot{q}_1 - q_3 \cdot \dot{q}_2 + q_2 \cdot \dot{q}_3 \end{bmatrix} + \begin{bmatrix} \delta\omega_1 \\ \delta\omega_2 \\ \delta\omega_3 \end{bmatrix} \quad (3.8)$$

desenvolvida pela equação apresentada em [8] com um valor aleatório adicionado.

A atualização dos quaternions pelas medições dos girômetros foi feita utilizando valores iniciais conhecidos e métodos de integração numérica, para resolução da Eq. 3.4 e a propagação das incertezas da Eq. 2.53, com os valores de σ_{w1}^2 , σ_{w2}^2 e σ_{w3}^2 igual as variâncias dos valores aleatórios inseridos nas rotações.

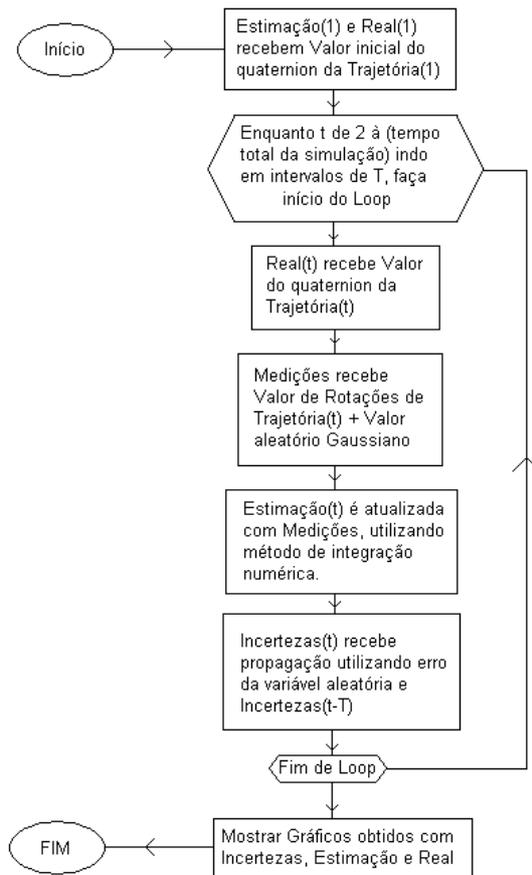
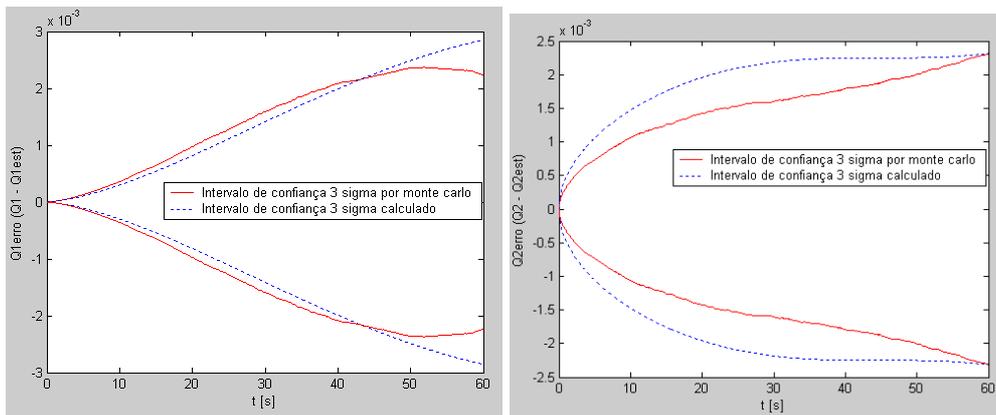
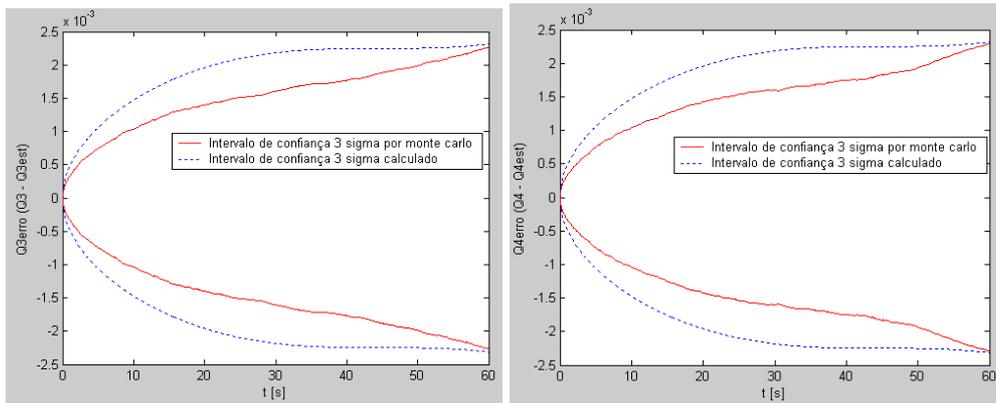


Figura 3.1: Fluxograma da Simulação de Estimãõo de Orientação



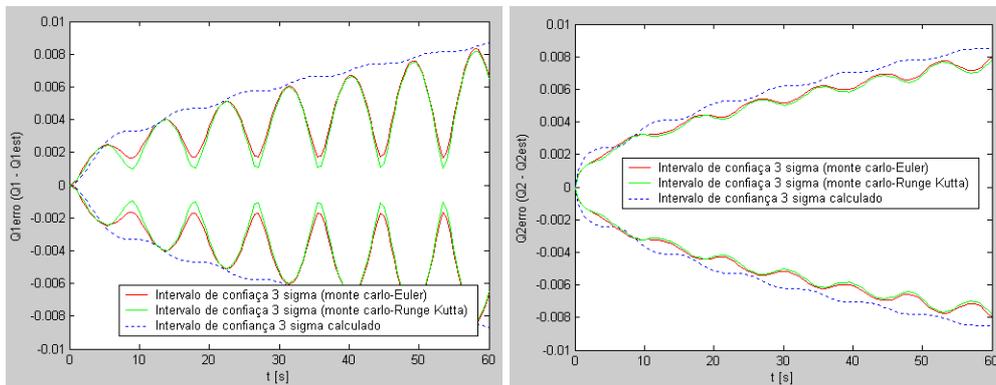
(a) Propagação das incertezas de q_1 .

(b) Propagação das incertezas de q_2 .



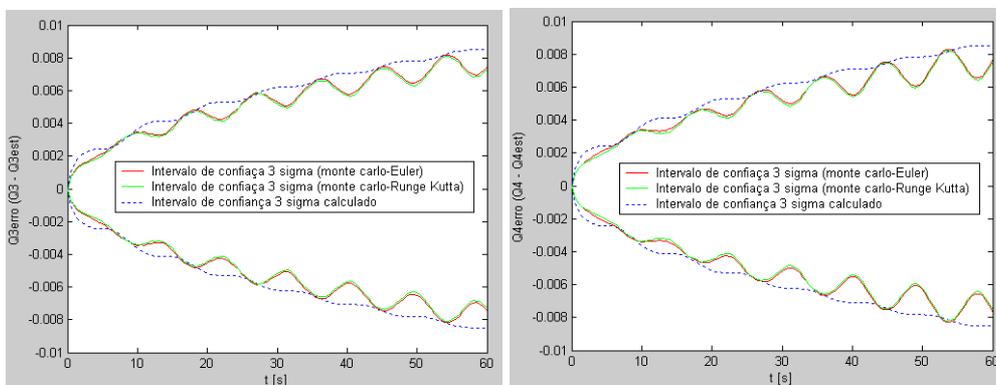
(c) Propagação das incertezas de q_3 .

(d) Propagação das incertezas de q_4 .



(e) Propagação das incertezas de q_1 com rotação rápida.

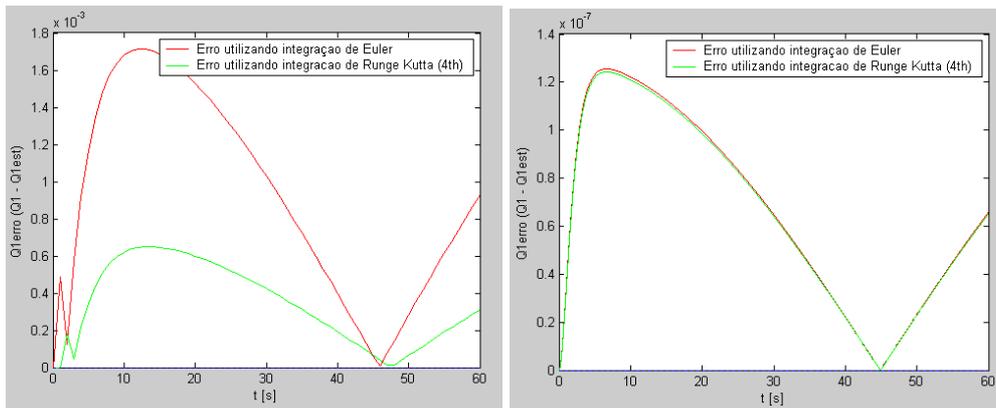
(f) Propagação das incertezas de q_2 com rotação rápida.



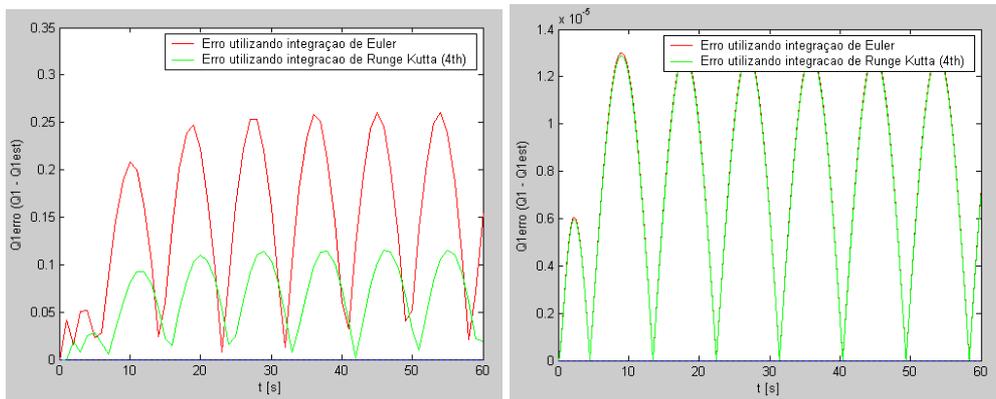
(g) Propagação das incertezas de q_3 com rotação rápida.

(h) Propagação das incertezas de q_4 com rotação rápida.

Figura 3.2: Gráficos de validação das propagações de incerteza



(a) Erros de q_1 na simulação de orientação, (b) Erros de q_1 na simulação de orientação, utilizando-se taxa de amostragem lenta e trajetória utilizando-se taxa de amostragem rápida e trajetória lenta.



(c) Erros de q_1 na simulação de orientação, (d) Erros de q_1 na simulação de orientação, utilizando-se taxa de amostragem lenta e trajetória utilizando-se taxa de amostragem rápida e trajetória rápida.

Figura 3.3: Gráficos de erros dos métodos de Integração Numérica

O modelo pôde ser validado pela análise dos gráficos, apresentado na Fig. 3.2, obtidos com os erros das estimções e o intervalo de $\pm 3\sigma$, obtido tanto pela Eq. 2.53, quanto por simulação de Monte Carlo com 1000 amostras.

O corpo simulado realizou rotações de frequências diferentes para os gráficos apresentados pelas Fig.'s 3.2(a), 3.2(b), 3.2(c) e 3.2(d) com os apresentados pelas Fig.'s 3.2(e), 3.2(f), 3.2(g) e 3.2(h). Isto foi feito para mostrar o efeito da influência da dinâmica da trajetória no erro da estimação. Nas Fig.'s 3.2(e), 3.2(f), 3.2(g) e 3.2(h), o objeto realizava uma rotação mais veloz e por isto o método de integração numérica de Runge-Kutta de 4^a ordem também foi utilizado para analisar o modelo.

3.2.1.2 Variação no Método de Integração Numérica

Dois métodos de integração numérica foram utilizados, o método de Euler, ou aproximação linear da integração, e o método de Runge-Kutta, apresentado em [14]. Este último, possui erro menor que o método linear, e isto garante uma melhor precisão na estimação do estado, o mesmo também garante maior convergência na integração. Porém, uma particularidade deve ser analisada na hora de se utilizar este método em microcontroladores, onde a velocidade de processamento não é abundante. Por ocupar mais tempo de processamento que o primeiro método, talvez o período de amostragem seja afetado por esta integração e isto faz com que um período maior de amostragem seja necessário para que o cálculo da integração numérica com Runge-Kutta seja possível.

Na simulação feita em PC este problema não é encontrado, pois não precisa-se de processamento em tempo real para que os resultados sejam coerentes. Com isto, realizou-se as diferentes integrações numéricas, para estimação dos parâmetros do quaternion na mesma trajetória, e obteve-se os erros mostrados na Fig. 3.3. Os quais foram encontrados comparando-se o resultado obtido pela propagação de estado através do método numérico com a trajetória real, analítica.

Analisando-se os gráficos, percebe-se que para a mesma trajetória os métodos de integração possuem erros parecidos quando a taxa de amostragem é suficientemente grande comparada à trajetória, mostrado pelas Fig.'s 3.3(b) e 3.3(d). Também pode se analisar pelos gráficos que os erros caem muito para ambos métodos quando a dinâmica do sistema é menor, ou seja, quando executadas trajetórias mais lentas. Isto deve-se ao fato de considerar-se que, pelo método de Euler, as derivadas são constantes entre as medições, fato que não é verdade para a trajetória utilizada. Por isto, quanto mais perto de constantes mais precisos serão os métodos numéricos.

3.2.1.3 Variação no Período de Amostragem

Com diferentes valores nas taxas de amostragem, verificou-se pelos gráficos, apresentados na Fig. 3.4, que a precisão da estimação aumenta com o aumento da taxa de amostragem. Mesmo com rotações mais velozes o erro caiu com o aumento da taxa de amostragem. Isto significa que o menor período possível para as amostragens deve ser escolhido. Alguns problemas devem ser levantados quanto a obtenção deste tempo, como será mostrado na subseção 3.2.2.1.

3.2.2 Simulações para Validação da Estimação de Posição

Após a validação do modelo que será utilizado para estimação de orientação do corpo, pode-se implementar por simulação a estimação da posição do corpo. Esta também se baseou em executar movimentos conhecidos num corpo e através da propagação do estado, considerando-se as medidas dos acelerômetros, verificar se os mesmos acompanham o esperado.

No caso da estimação de posição é importante ressaltar que tanto a propagação das incertezas de orientação quanto a das incertezas de aceleração afetam a estimação do estado. Portanto, se caso a propagação seja mal feita na orientação, esta simulação fica comprometida.

A Fig. 3.5 mostra o fluxograma da simulação de estimação do estado de posição de um corpo. Assim como na simulação de orientação, os sensores, agora de aceleração, foram implementados através da inserção de um valor aleatório gaussiano às acelerações executadas no corpo, sendo consideradas as acelerações em cada eixo das coordenadas do corpo.

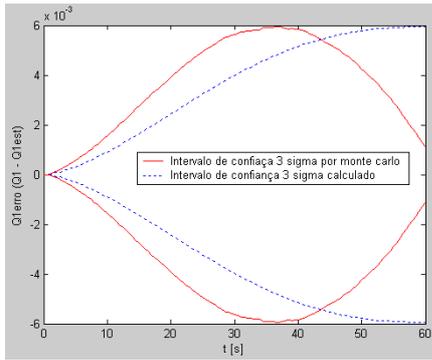
A propagação do estado de posição foi feito utilizando-se o processo de integração numérica de Euler, recebendo como entrada as medições dos acelerômetros dadas por:

$$\ddot{\mathbf{A}}_{\mathbf{B}} = \mathbf{R}_{\mathbf{W}}^{\mathbf{B}} \times \mathbf{A}_{\mathbf{W}} + \begin{bmatrix} \delta a_1 \\ \delta a_2 \\ \delta a_3 \end{bmatrix} \quad (3.9)$$

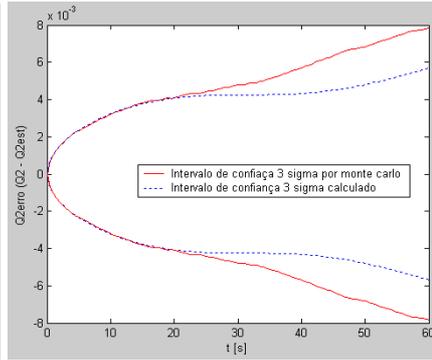
e utilizando a Eq. 3.6.

Analisando-se os gráficos obtidos, mostrados na Fig. 3.6, pode-se analisar que a medida que o erro dos quaternions aumenta, apresentado pelas Fig.'s 3.6(a), 3.6(b), 3.6(c) e 3.6(d), a estimação da posição, apresentadas pelas Fig.'s 3.6(e), 3.6(f) e 3.6(g), se compromete. Isto é facilmente compreendido, já que a estimação de posição depende da transformação de coordenadas feita por $\mathbf{R}_{\mathbf{B}}^{\mathbf{W}}(\mathbf{Q})$ e \mathbf{Q} quando estimado errado propaga através da DCM o erro para a estimação de posição.

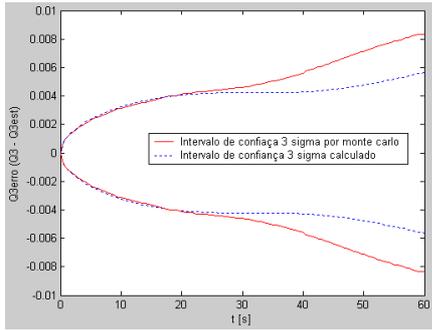
A propagação do intervalo de confiança $\pm 3\sigma$ para as estimativas de posição e velocidade não foram desenvolvidas ainda nesta implementação. Devido a isto, a análise do intervalo não foi feita para esta



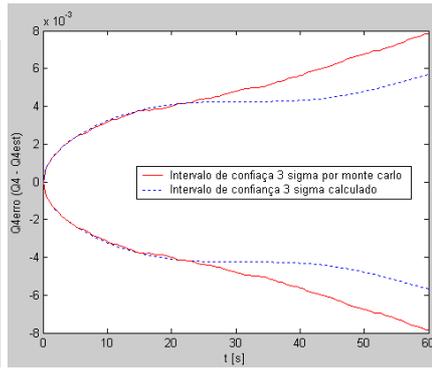
(a) Erro de q_1 utilizando trajetória mais lenta e taxa de amostragem menor.



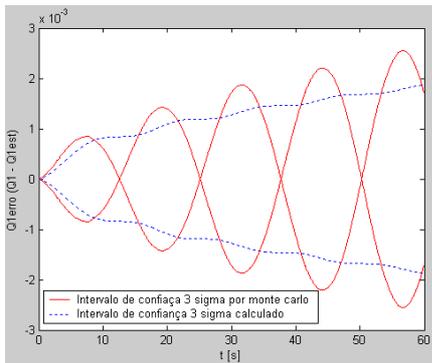
(b) Erro de q_2 utilizando trajetória mais lenta e taxa de amostragem menor.



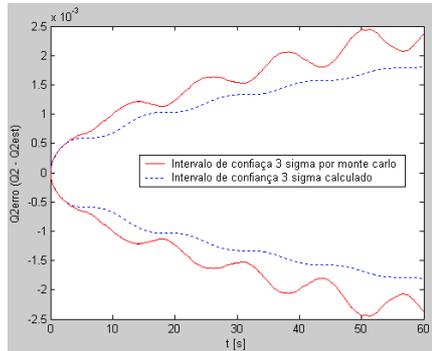
(c) Erro de q_3 utilizando trajetória mais lenta e taxa de amostragem menor.



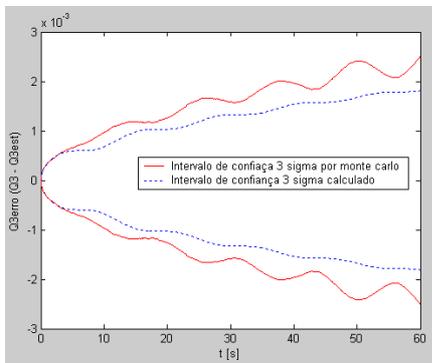
(d) Erro de q_4 utilizando trajetória mais lenta e taxa de amostragem menor.



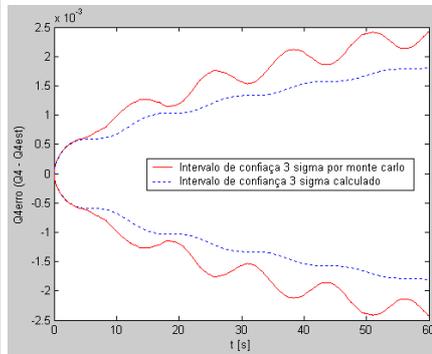
(e) Erro de q_1 utilizando trajetória mais rápida e taxa de amostragem maior.



(f) Erro de q_2 utilizando trajetória mais rápida e taxa de amostragem maior.



(g) Erro de q_3 utilizando trajetória mais rápida e taxa de amostragem maior.



(h) Erro de q_4 utilizando trajetória mais rápida e taxa de amostragem maior.

Figura 3.4: Gráficos com intervalos $\pm 3\sigma$ para análise de período de amostragem.

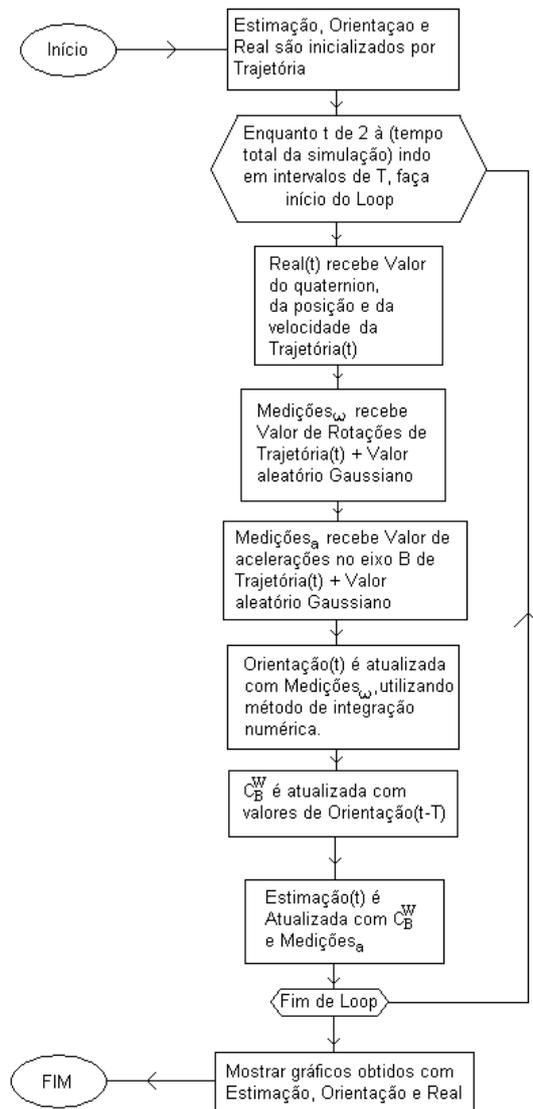
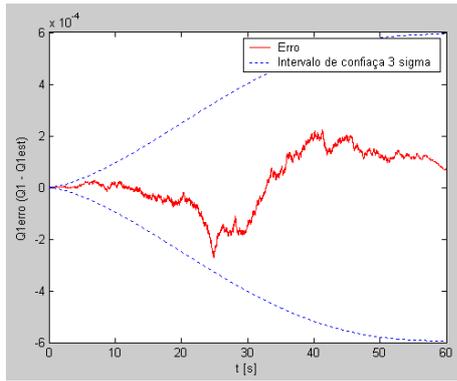
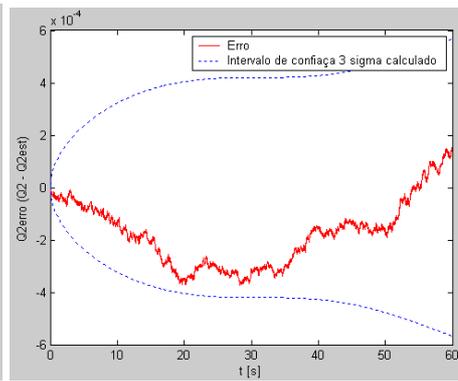


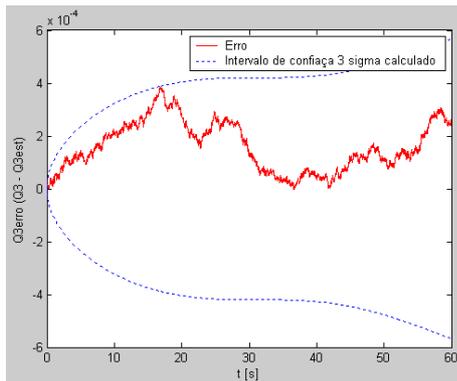
Figura 3.5: Fluxograma da Simulação de Estimação de Posição



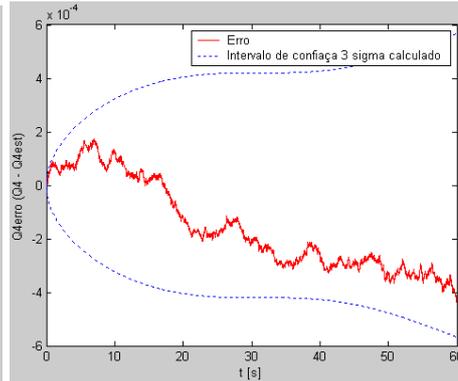
(a) Erro da estimação de q_1



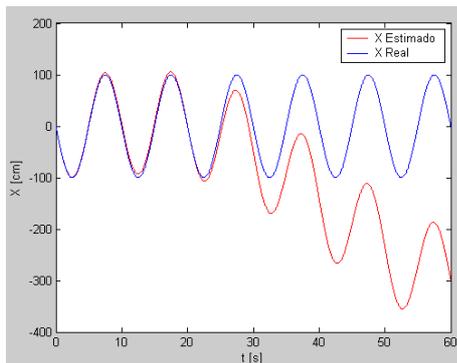
(b) Erro da estimação de q_2



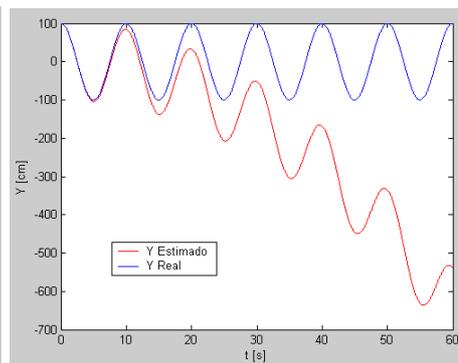
(c) Erro da estimação de q_3



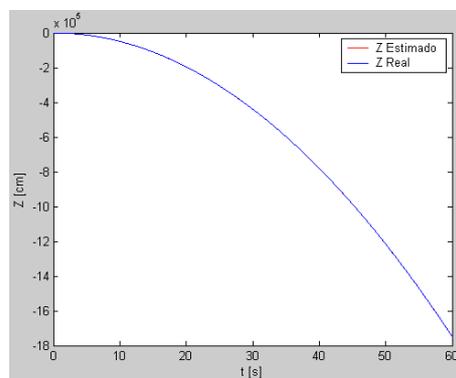
(d) Erro da estimação de q_4



(e) Valores encontrados para a estimação da posição no eixo X_W



(f) Valores encontrados para a estimação da posição no eixo Y_W



(g) Valores encontrados para a estimação da posição no eixo Z_W

Figura 3.6: Gráficos da simulação de estimação da posição.

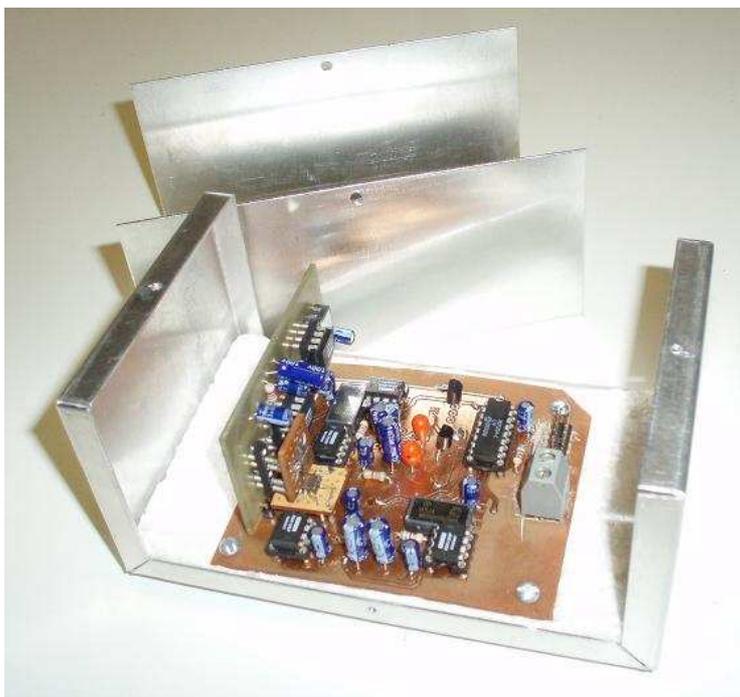


Figura 3.7: Foto do protótipo de IMU construído.

simulação.

3.2.2.1 Escolha do Método de Cálculo

Analisando os valores mais precisos das simulações feitas, escolheu-se a representação por quaternions, sendo estes, estimados através do método de integração numérica Runge-Kutta de quarta ordem¹. Este modelo escolhido, baseando-se nas simulações, deve ser validado na prática, já que a modelagem utilizada dos sensores não contempla toda a complexidade dos ruídos encontrados nos sinais reais.

Outro problema que pode ocorrer na implementação é devido ao tempo de cálculo do método de integração, que pode ser tão grande que o período de amostragem fica prejudicado. Talvez, utilizando-se um método mais rápido, como o de Euler², encontrem-se valores com maior precisão para a estimação.

3.3 IMPLEMENTAÇÃO DO SISTEMA

Tendo os modelos sido validados, começou-se a construção de um protótipo de IMU para uma central inercial *Strapdown*. A construção deste protótipo baseou-se na Fig. 2.7 onde os sensores são dispostos ortogonais uns aos outros e o eixo de medição de cada girômetro coincide com o de um dos acelerômetros.

3.3.1 Construção do Protótipo de IMU

Conhecendo-se algumas das dificuldades que seriam encontradas na construção de um protótipo para o sistema (IMU), apresentado na Fig. 3.7, foram tomados vários cuidados na disposição dos sensores, dos

¹método desenvolvido seguindo as orientações apresentadas na nota de aula da disciplina "Física dos Sistemas Dinâmicos", FEUP, site "<http://fisica.fe.up.pt/eic2107/acetatos/aula11.html>"

²Referindo-se a aproximação linear da integração e não aos ângulos de Euler

componentes eletrônicos e das trilhas. Na implementação, algumas considerações elétricas, dispostas por [15], foram utilizadas, são elas:

- Dispositivos digitais devem ser posicionados o mais perto possível do conector e da alimentação. Quanto maior a frequência de funcionamento, mais perto do conector deve estar o componente.
- Componentes totalmente analógicos devem ficar o mais distante possível dos componentes digitais, para assegurar que perturbações provenientes das transições de estado não fiquem acoplados nos sinais analógicos.
- No caso de conversores A/D utilizando tecnologia de registrador de aproximação sucessiva, ou SAR (do inglês *Successive Approximation Register*), um buffer digital deve ser utilizado para isolar o conversor da parte digital do circuito.
- Utilizar planos terra no circuito, dois planos serão necessários devido a dois sensores necessitarem ficar em outra placa (devido à ortogonalidade requerida).
- As trilhas de alimentação devem ser pelo menos três vezes maior que as trilhas normais.
- Todas trilhas, tanto analógicas quanto digitais, devem ser as mais curtas possíveis, para assim, minimizar a possibilidade de sinais estranhos se acoplarem aos sinais desejados.
- Terminais de entrada de componentes analógicos, devido a alta impedância, devem possuir trilhas afastadas das trilhas de mudança rápida (clocks, comunicações, etc.), devido a possibilidade de acoplamento, a criação de capacitor entre as trilhas e a injeção de correntes indesejadas.
- Capacitores By-Pass devem ser alocados em duas localidades:
 - Na alimentação (de 10 μF a 100 μF)
 - Um em cada componente ativo (tanto analógico quanto digital)
 - * Se faixa de trabalho menor que 1 Mhz, 1 μF
 - * Se faixa de trabalho maior que 1 Mhz, 0,1 μF

Além destas considerações, foram utilizados componentes de maior precisão para a filtragem do sinal e a alimentação dos sensores.

O ganho nos sinais dos sensores foi feito por um amplificador operacional de instrumentação, o *INA126*, que possui ganho padrão, com precisão neste valor. Este foi fácil de ser adquirido, porque é comumente utilizado em outros projetos do laboratório, o único problema deste componente é não ser *rail-to-rail* e com isso limitar a faixa útil de utilização dos sensores, ver anexos com esquemáticos dos circuitos implementados para este IMU.

Realizando-se movimentos suaves aos sensores, foi possível perceber-se que o ganho inserido ao sinal dos sensores pelo componente de instrumentação não ocasiona saturação das medições. Esta análise foi feita em todos os sensores utilizados, com isto, o componente pôde ser utilizado tanto para os acelerômetros, quanto para os girômetros. Uma referência de metade da alimentação foi inserida também pelo *INA126* às medições, a qual sendo gerada por um componente de precisão, o *LM336*, que gera uma referência de 2,5V com pequeno coeficiente de temperatura.

Pelo fato dos acelerômetros não possuírem sinal de referência para as medições e variar próximo a valores iguais a metade da alimentação, o sinal de 2,5V foi utilizado como referência para estes sensores, posteriormente, a calibração do sistema se encarrega de retirar o *Bias* dos sensores e isto excluía as diferenças entre a referência e a medição com o IMU parado.

Após o ganho e a inserção de referência, os sinais passam por um filtro passa-baixo passivo de primeira ordem que retira os ruídos indesejáveis. A frequência de corte definida ficou em 100 Hz, valor mais que suficiente para movimentos suaves do IMU.

Nesta implementação utilizou-se um conversor de 12-bits que se localiza na própria IMU, apenas sinais digitais trafegam em fios de comunicação, evitando assim o acoplamento de muitos ruídos externos indesejáveis às medições. O acréscimo de 2 bits nas conversões aumentou em muito a resolução das medições.

Com todas estas considerações, os problemas encontrados na primeira implementação foram praticamente resolvidos. Grande parte dos ruídos deixaram de acompanhar as medições e a validação prática dos modelos tornou-se possível.

3.4 DEFINIÇÃO DOS ESTADOS DO SISTEMA

Todos os processos de medição, calibração e evolução dos estados são feitos no microcontrolador, utilizando-se ponto flutuante e enviados ao PC quando requeridos. Isto facilita o atendimento da necessidade de se ter um sistema em tempo real, já que no microcontrolador isto é facilmente configurado.

Para o planejamento das funções, modelou-se os estados do sistema utilizando-se uma rede de Petri elementar e finita apresentada na Fig. 3.8. Esta rede foi definida abordando-se as requisições de interação com o usuário, comunicação com o PC e o fato de não se propagar as incertezas dos sensores quando o sistema encontrar-se parado, requisições estas melhor apresentadas na sessão 1.3. Para a construção da rede foram escolhidos as seguintes posições, transições e conjunto de arcos:

Posições:

- p_1 = iniciado
- p_2 = calibrando
- p_3 = calibrado
- p_4 = integrando
- p_5 = parado
- p_6 = respondendo

Transições:

- t_1 = botão A
- t_2 = botão B
- t_3 = botão C
- t_4 = fim da calibração
- t_5 = chegar requisição de estimação ou de incertezas
- t_6 = terminar de responder

Conjunto de Arcos:

$\{(p_1, t_1), (t_1, p_2), (p_2, t_4), (t_4, p_3), (t_4, p_5), (p_3, t_2), (t_2, p_4), (p_4, t_3), (t_3, p_5), (p_5, t_1), (p_5, t_2), (p_5, t_5), (t_5, p_6), (p_6, t_6), (t_6, p_5)\}$

Sendo a posição p_1 , simplesmente, a representação do sistema estar ligado e p_5 a marcação inicial, possui um *token* no início do funcionamento.

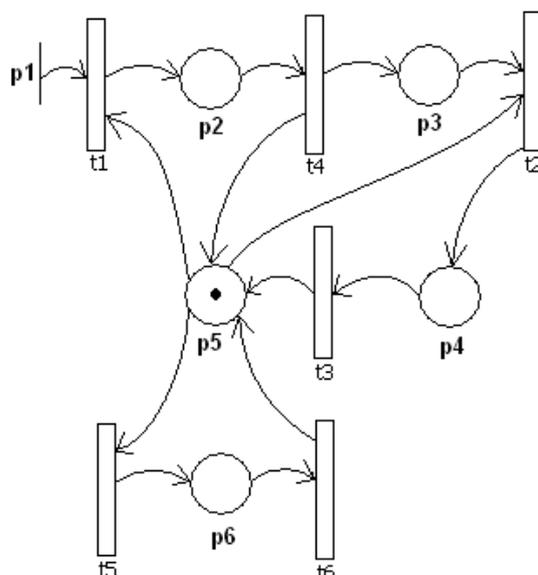


Figura 3.8: Rede de Petri ilustrando funcionamento do sistema

Analisando a rede criada pode-se criar a árvore de alcançabilidade apresentada na Fig. 3.9, com esta árvore pode-se visualizar todos os possíveis estados do sistema e percebe-se que este não encontrará nenhum estado indefinido por qualquer combinação de transições que ocorra, isto é o desejável para o bom funcionamento do mesmo. Também é possível obter através da análise da rede que o sistema é reversível, seguro e vivo, o que garante a passagem por todas posições e que em nenhum momento ocorrerá *Deadlock*.

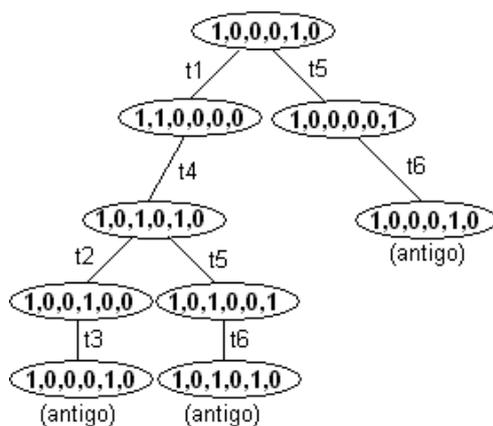


Figura 3.9: Árvore de alcançabilidade do sistema

3.5 IMPLEMENTAÇÃO DA PLACA-MÃE DO MICROCONTROLADOR

Para que as medições do IMU cheguem ao microcontrolador e sejam atendidas todas requisições, a placa-mãe, apresentada na Fig. 3.10, deve preencher os seguintes requisitos:

- O processamento das informações devem ser feitos num microcontrolador, devido a portabilidade, foi escolhido um da Atmel, da linha AVR ATmega32;

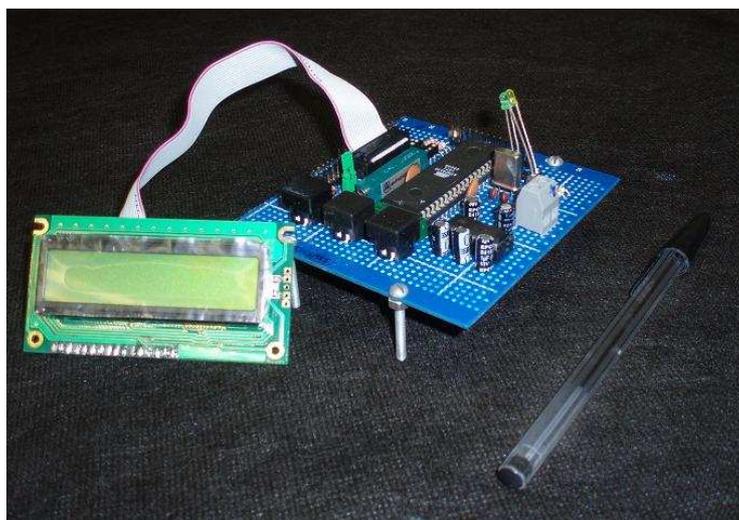


Figura 3.10: Foto da placa-mãe construída

- Deverá possibilitar a reprogramação do mesmo, o escolhido foi que será feita por programador BSD;
- Alguns botões para interação com o usuário, assim como um *BackLight LCD*;
- A comunicação com o PC deve ser feita através de porta USB 1.1 (baixa velocidade);
- LED's devem informar se o sistema está ligado e se ocorre comunicação com o PC;
- As informações do IMU devem chegar ao microcontrolador por comunicação SPI (do inglês *Serial Peripheral Interface*).

Tabela 3.1: Mensagens apresentadas no LCD

Mensagem	Interrupção	Significado
Ligado	Iniciar o sistema	Foi iniciado e ainda não está calibrado
Calibrando...	Apertar o primeiro botão	Indica que a calibração está em andamento
Calibrado!	Finalizar a calibração	Indica o fim da calibração
Integrando...	Apertar o segundo botão	Indica que a estimação do estado de orientação e posição está em andamento
Posição: X Y Z	Apertar o terceiro botão	Indica que a estimação está parada e mostra em centímetros a posição estimada até o momento da pausa

Para a comunicação USB foi utilizado um conversor Serial-USB, facilmente encontrado em lojas de acessórios para computador, com esta conversão foi possível comunicação em taxas acima do limite das portas seriais do PC, 115200 bps.

Foi definido a taxa de 333333 bps para a comunicação, pois esta taxa é configurável com precisão no *ATMega32* utilizando um cristal de 16 MHz e é aceitável pelo conversor *RS232* utilizado.

O botões são utilizados pelo usuário para informar ao sistema os momentos de começar a estimar o estado, parar a estimação e calibrar o sistema, sendo na calibração também alterado o estado para o estado inicial.

No *LCD* são apresentados os estados do sistema e os valores da estimação de posição quando no estado de estimação parado. As possíveis mensagens mostradas no *LCD* são apresentadas na Tab. 3.1. Com isto, a placa foi implementada atendendo a todos requisitos necessários, ver esquemático em anexo.

3.6 IMPLEMENTAÇÃO DA BIBLIOTECA PARA COMUNICAÇÃO COM O PC

Para que as estimações de estado tenham utilidade na reconstrução 3D, é necessário que estas informações sejam enviadas ao PC que fará o tratamento das informações provenientes do *scanner*. Estas informações são enviadas através de requisições feitas pelo PC ao sistema, o qual deverá responder da maneira esperada em um certo intervalo de tempo.

3.6.1 Definição de Requisições feitas pelo PC

Foram definidas possíveis requisições que poderão ser feitas pelo PC, estas definições foram criadas levando-se em consideração o melhor tratamento possível para o problema. As principais requisições seriam o envio das estimações de estado da central inercial, assim como também o envio da propagação das incertezas.

Pelo fato do *scanner* poder executar várias capturas e o computador não estar disponível em todos os locais, criou-se a possibilidade de se armazenar as estimações e as propagações de incertezas numa lista, e posteriormente serem enviadas ao PC quando requisitado.

Outra requisição seria a de informar qual o estado atual do sistema, informando, por exemplo, que este se encontra calibrando-se, ou integrando as medições, lembrando que as estimações de estado só podem ser enviadas quando o mesmo não encontra-se neste estado.

Na Tab. 3.2 são apresentadas todas possíveis requisições definidas para o sistema, requisições estas que serão feitas pela biblioteca e não utilizando a biblioteca. Também é apresentada a resposta da central para cada requisição.

Tabela 3.2: Possíveis requisições feitas ao sistema pelo PC

Requisição (ASCII enviado)	Resposta (tamanho da mensagem)
Estimações (E)	Valores de estimação do quaternion, da posição e da velocidade (40 bytes)
Propagação de Incertezas (P)	Incertezas da orientação propagadas (16 bytes)
Estado do Sistema (S)	Estado em que se encontra o sistema (1 byte)
Retirar Estimação da Fila (A)	Confirmação de item Retirado (1 byte)
Retirar Incerteza da Fila (R)	Confirmação de item Retirado (1 byte)

O *byte* de resposta da requisição de estado do sistema informa em cada *bit* o estado das posições da rede de Petri apresentada em 3.4. Sendo o primeiro *bit* para a p_1 (sempre 1), o segundo para p_2 , o terceiro para p_3 e assim consecutivamente.

Por fim, na resposta de confirmação de item retirado, quando na mesma retornar "S" significa que o item foi retirado da lista e quando "N" indica que lista não possui nenhum item.

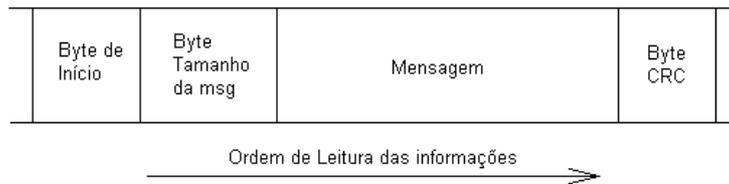


Figura 3.11: Formato das mensagens definidas para o sistema

3.6.2 Protocolo de comunicação

Definiu-se um protocolo de comunicação para que as mensagens erradas fossem descartadas e para que o sistema atendesse como esperado as requisições. Inicialmente, foram formatadas as mensagens como um byte informando o início da mensagem, um byte com o tamanho da mensagem, a mensagem em si e por fim um check byte para verificação da mensagem. Esta forma é apresentada na Fig. 3.11.

Para que não ocorra travamento, um tempo máximo de resposta foi definido para o sistema, após este tempo se não houver resposta uma outra requisição é feita para a central. Caso também a resposta que chegar do sistema não estiver correta, tanto pelo tamanho não coincidir ou pelo check byte estar incorreto, também será feita outra requisição para a central.

3.6.3 Biblioteca

Utilizando plataforma LINUX implementou-se uma biblioteca para a interface com a central inercial, esta criação baseou-se na implementação de funções para atender as todas requisições definidas na sub-sessão 3.6.1. Para o funcionamento da mesma faz-se necessário o módulo do kernel usb-serial, o qual, quando existente, é automaticamente iniciado assim que se conecta a central inercial ao PC.

Nesta foram implementadas funções para iniciar o sistema, finalizar-lo e fazer requerimentos de estimações ou incertezas. Estas funções podem ser chamadas apenas importando-se a biblioteca do sistema, a qual cria as estruturas "pEstimac" para a representação de estimação e "pErrQuat" para as incertezas propagadas dos quaternions.

É necessário chamar a função de inicialização do sistema antes de se fazer qualquer requerimento ao sistema, lembrando que somente no estado *parado* as estimações e incertezas são respondidas. A utilização das funções podem ser vistas no anexo, onde os valores retornados por cada é apresentado, como também o nome das chamadas.

3.6.4 Programa para Visualização da Estimação

Foi desenvolvido um programa, utilizando a biblioteca *OpenGL* e a biblioteca matricial *GMatrix*³, para uma visualização das estimações de orientação e posição do sistema. Estes valores são aplicados como rotação e translação de um corpo virtual, no momento paralelepípedo, e apresentados na interface gráfica do Linux.

Para que este programa funcione, é necessário que o módulo usb-serial esteja instalado no Linux, como para o funcionamento da biblioteca. Também é necessário um programa diferente carregado no microcontrolador, pois como apresentado na sessão 3.4 no estado *integrando* não é possível obter os valores de estimação e neste programa de visualização a requisição do estado é feita frequentemente.

Na implementação, foi necessária a criação de uma estrutura para o paralelepípedo, chamada de "pa-

³Criada por Geovanny Araújo Borges, Dep. de Engenharia Elétrica - UnB, Brasil.

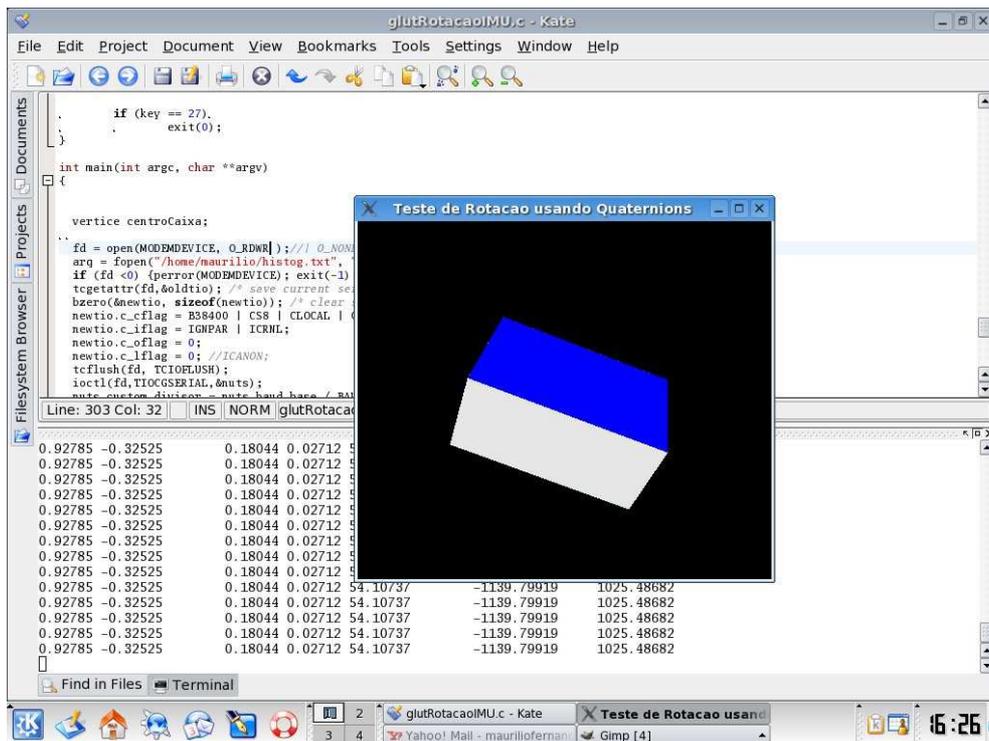


Figura 3.12: Apresentação do programa de visualização dos estados estimados.

ralelepi" e a implementação da DCM^T utilizando quaternions. A estrutura é formada por oito "vertice", estrutura com três parâmetros (representando X, Y e Z), e pelo centro de tipo "vertice" também. Para a visualização da orientação, o quaternion retornado pelo sistema é utilizado para construção da DCM e esta é aplicada sobre cada vértice de "paralelepípedo". Já para a posição é alterado o valor do centro do paralelepípedo e assim uma translação pode ser vista.

Como ocorre variação na posição do objeto, é necessário que se desloque o centro do mesmo até o ponto $(0, 0, 0)$ e nesta posição aplica-se a DCM^T , após isto, aplica-se a translação, para que assim a visualização dos estados enviados seja correta. A tela desenvolvida para o programa é apresentada na Fig. 3.12.

3.7 SENSORES

Para que a modelagem escolhida seja aplicável, é necessário que as medições apresentadas pelos sensores aproximem-se de distribuições *Gaussianas*. Para verificação deste fato, foram colhidas diversas amostras de conversões A/D dos sensores com o IMU parado e analisando os histogramas, apresentados na Fig. 3.13, pode-se considerar que as mesmas apresentam características semelhantes às da distribuição esperada.

3.7.1 Caracterização

Caracterizar estes sensores inerciais não é uma tarefa trivial, pois necessita de equipamentos que realizem movimentos com altíssima precisão e estes não são facilmente encontrados. Como apresentado em [16], a caracterização dos sensores do sistema pode ser feita uma vez e, utilizando-se as informações da gravidade e da rotação da Terra, determinar um valor para o estado inicial de orientação do sistema. Porém

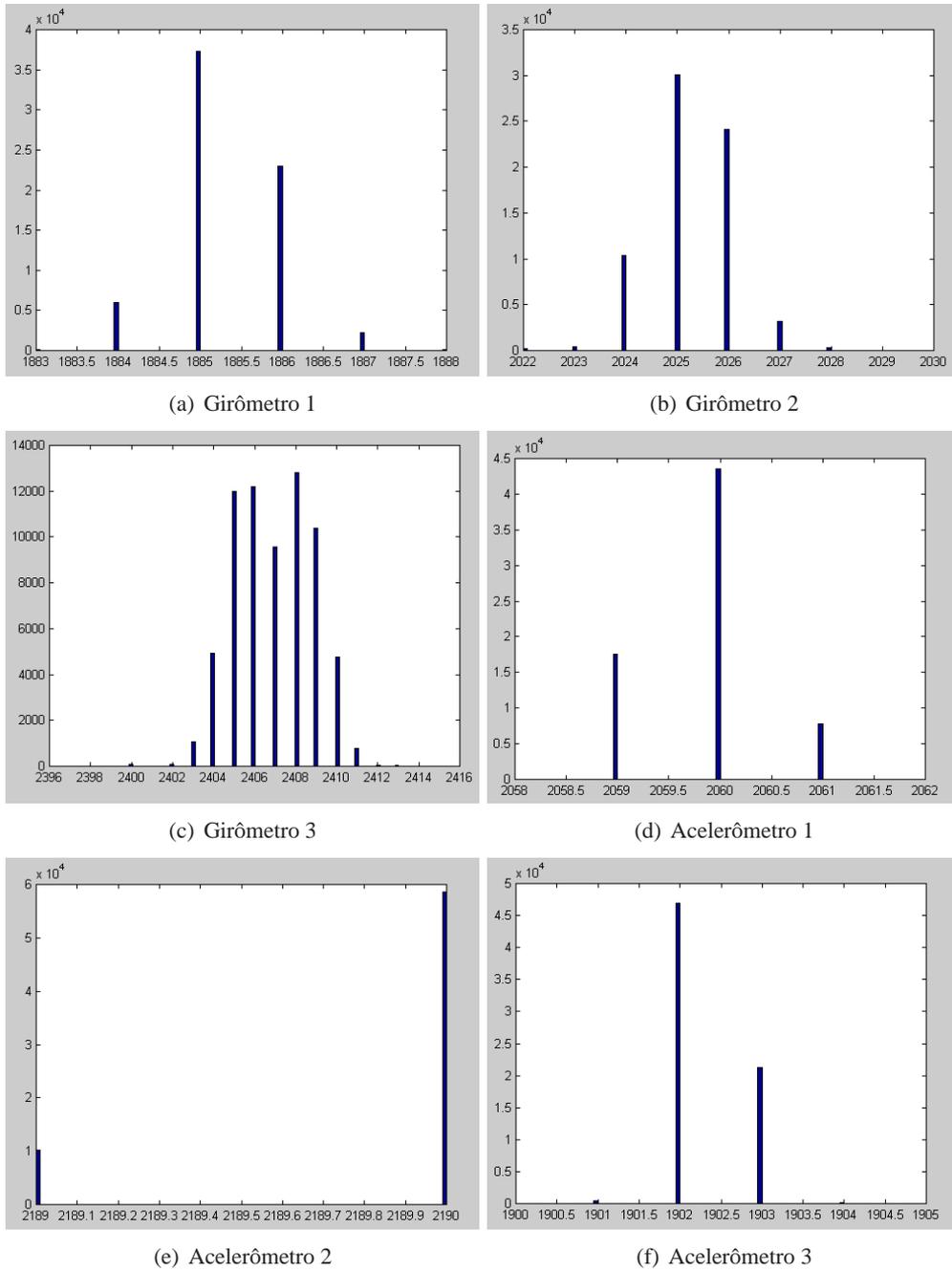


Figura 3.13: Histogramas das conversões A/D com IMU parado.

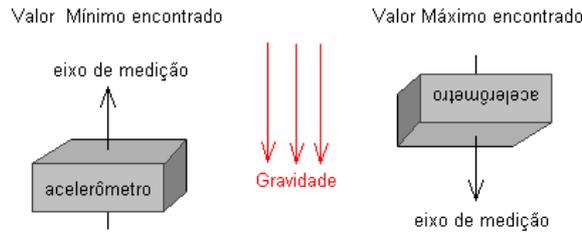


Figura 3.14: Experimento para determinação do Ganho de Escala dos Acelerômetros

os ruídos encontrados nos girômetros não permitem a determinação do sentido de rotação da Terra, o que indisponibiliza a utilização deste método para calibração.

Devido a isto, uma estimação fraca dos parâmetros de Ganho de Escala e *Bias* dos sensores pôde ser feita. Utilizando-se a gravidade, para os acelerômetros, e o programa desenvolvido para representação dos movimentos realizados, para os girômetros, encontrou-se valores de Ganho de Escala para ambos os sensores, já o offset será determinado pelo sistema.

3.7.1.1 Bias dos Sensores

Para determinação do *Bias* dos sensores utiliza-se o estado de *calibrando*, apresentado na seção 3.4, desenvolvido para o sistema. Neste, adquire-se duas mil amostras das conversões A/D dos sensores e com estes valores faz-se uma média, a qual é utilizada como offset constante dos sensores.

Para que a determinação deste parâmetro seja feita de maneira correta, é necessário que o IMU se encontre totalmente parado durante todo o processo, cerca de 6 segundos, e com o eixo Z_B na direção da gravidade. Esta posição de calibração deve ser a posição da aquisição da primeira imagem, para que assim o sistema defina os eixos $[X_W Y_W Z_W]$.

Por fim, no encerramento do processo de calibração, a mensagem "Calibrado!" aparece no *LCD* e o sistema vai para o estado *calibrado*, também apresentado na sessão 3.4.

3.7.1.2 Ganho de Escala dos Acelerômetros

Tabela 3.3: Valores de Ganho de Escala encontrados para os acelerômetros

Acelerômetro	Valor Máximo	Valor Mínimo	Ganho de Escala
a ₁	2288	1444	0,02303
a ₂	2420	1604	0,02382
a ₃	2410	1568	0,02309

Para o ganho de escala dos acelerômetros, realizou-se um experimento parecido com o apresentado em [16], onde se utiliza a aceleração da gravidade como referência para a conversão. Realizando movimentos lentos com os sensores, pode-se localizar um valor mínimo de medição e, rotacionando o sensor em π radianos, um valor máximo, como mostrado na Fig. 3.14. Com estes dois valores encontra-se o ganho de escala realizando uma regra de três simples:

$$\frac{V_{max} - V_{min}}{V_{medido}} = \frac{2 \cdot g}{ac} \quad (3.10)$$

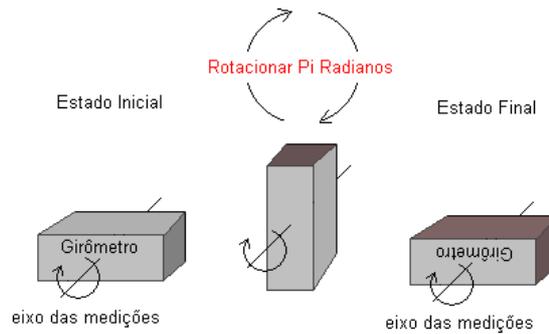


Figura 3.15: Experimento para determinação do Ganho de Escala dos Girômetros

onde g é o valor da gravidade local, ac o valor da aceleração que deseja-se encontrar, V_{max} o valor máximo encontrado no experimento e V_{min} o valor mínimo encontrado.

O *Bias* dos acelerômetros, encontrado pelo tratamento explicado na sessão 3.7.1.1, é retirado da amostra antes que o ganho de escala seja aplicado, assim, a medição pode ser representada por:

$$ac = (2 \cdot g / (V_{max} - V_{min})) \cdot (V_{medido} - Bias) \quad (3.11)$$

Por fim, utilizando-se um conversor A/D de 12 *Bits* obtém-se para cada um dos eixos do IMU os valores apresentados na tabela 3.3. Considerando a gravidade local como $9,72m/s^2$ calculou-se também o ganho de escala.

3.7.1.3 Ganho de Escala dos Girômetros

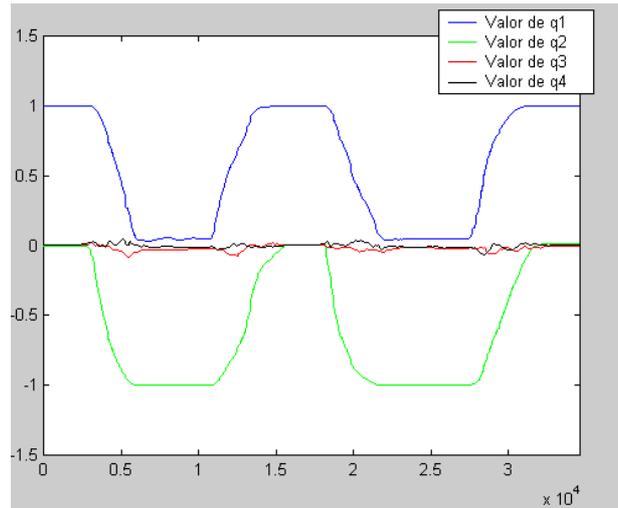
Com os girômetros a estimação destes parâmetros é mais complicada, não existe um valor de referência facilmente encontrado, como a gravidade no caso do acelerômetros. Com isto a estimação do Ganho de Escala é dificultada.

Assim, uma maneira encontrada para a obtenção deste valor foi através do programa criado para demonstrar os movimentos realizados com o IMU, este programa, como apresentado na sessão 3.6.4, ilustra a orientação atual do sistema na tela do PC, utilizando a biblioteca *OpenGL*. Com o sistema já implementado, e utilizando-se o método de integração de Runge-Kutta de quarta ordem é possível estimar um valor de ganho de escala para cada eixo por tentativa e erro.

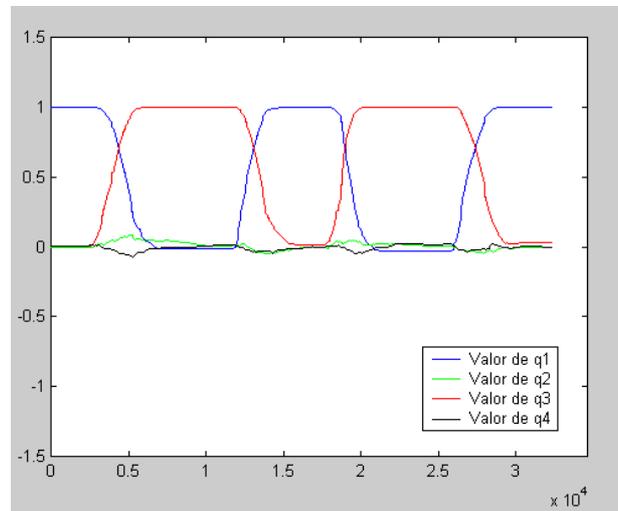
O procedimento é o seguinte:

- Analisando o datasheet do Girômetro estima-se um valor teórico inicial para o ganho de escala;
- Utilizando este valor no sistema, movimenta-se fazendo uma rotação de π radianos em algum dos eixos, como mostrado na Fig. 3.15;
- Analisando se o movimento apresentado no programa executou uma rotação maior ou menor que π radianos faz-se a correção do parâmetro;
- Repete-se este experimento utilizando o último valor escolhido até que se encontre um valor adequado ao movimento executado.

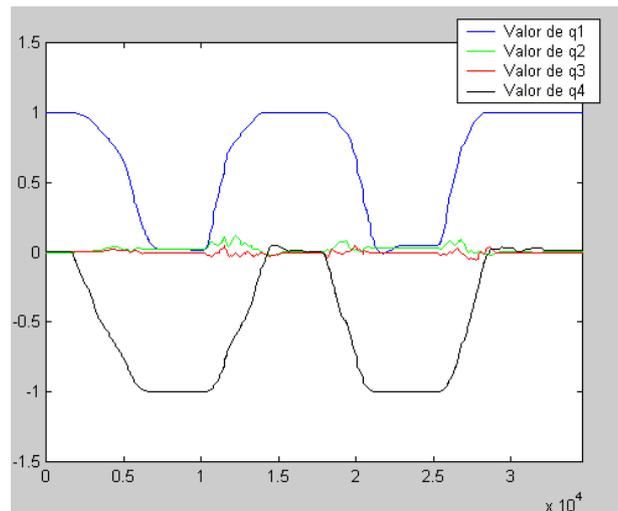
Após a execução deste procedimento, encontrou-se um valor de ganho de escala para o girômetros com que as rotações realizadas aproximassem de π radianos. Na Fig. 3.16 estão apresentados valores do quaternion estimado, encontrados com rotações iguais a executada na calibração.



(a) Rotações no eixo X do corpo.



(b) Rotações no eixo Y do corpo.



(c) Rotações no eixo Z do corpo.

Figura 3.16: Estimções dos quaternions nas rotações de π radianos.

Os movimentos realizados, manualmente, foram rotações iguais às utilizadas para determinação do ganho de escala. Analisando os valores que os quaternions devem assumir, q_1 variando entre 1 e 0 e os outros parâmetros podendo assumir valores de -1 , 1 e 0, dependendo se o movimento realizado foi de $+\pi$ radianos ou $-\pi$ radianos, pode-se analisar que o ganho de escala escolhido foi adequado.

3.8 TEMPOS DE PROCESSAMENTO

Como visto na sessão 3.2.1.3, o menor período de amostragem deve ser escolhido para que a estimação seja a mais precisa possível. Para encontrar-se este valor, determinou-se os tempos de processamento de cada etapa da resolução do problema. Assim, pôde-se encontrar um valor mínimo para o processamento total, dando assim, o melhor período de amostragem possível. Todos os processos estão sendo executados no microcontrolador, o que garante processamento em tempo real e utilizando ponto flutuante.

Com isso, os valores dos tempos foram encontrados utilizando o sistema já programado, alterando-se o estado de um *bit* não utilizado da porta *C* do microcontrolador à medida que o processo era iniciado ou finalizado. Com os tempos encontrados por osciloscópio, e listados na tabela 3.4, estimou-se um valor mínimo para o período de amostragem, o qual será utilizado para o período da interrupção que recolherá as medições inerciais.

Tabela 3.4: Tempos de processamento das etapas realizadas pelo sistema.

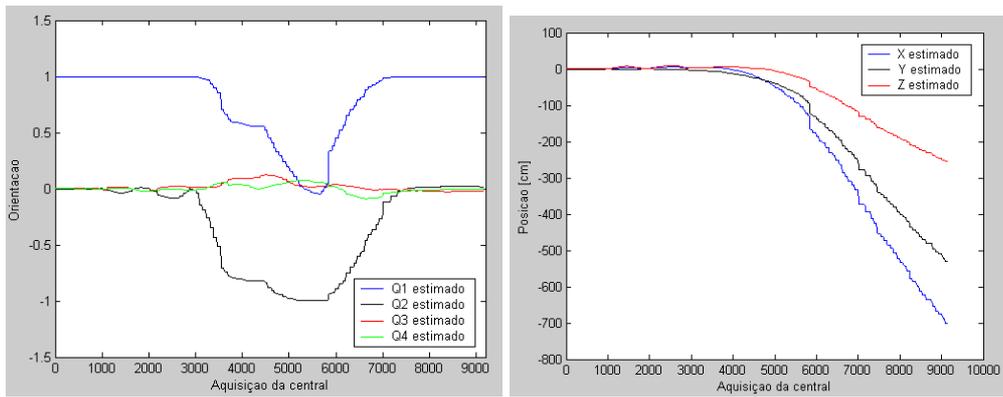
Processo	Tempo encontrado
Colher medições de todos os seis sensores	229 μs
Estimação de orientação por método de Euler	912 μs
Propagação das incertezas de orientação por método de Euler	310 μs
Estimação de orientação por método de Runge-Kutta de quarta ordem	2740 μs
Estimação da posição por método de Euler	484 μs
Envio de resposta com estimativas de posição e orientação	976 μs
Envio de resposta com estimativas de posição, velocidade e orientação	1394 μs
Envio de resposta com propagação de incertezas da orientação	580 μs
Construção da DCM por quaternions	804 μs
Tempo total de estimação utilizando método de Euler (posição e orientação)	2560 μs
Tempo total de estimação utilizando método de Runge-Kutta (orientação)	3730 μs

Com o tempo total de estimação, utilizando método de Euler para posição e orientação, pode-se escolher um período de cerca de 3 ms. Já com o tempo total encontrado pelo método de Runge-Kutta, aplicado somente na orientação e Euler para posição, pode-se utilizar um período de 5 ms. A diferença na taxa de amostragem pode justificar a utilização do método de Euler, porém a integração por Runge-Kutta garante maior convergência e por isso é mais confiável para movimentos aleatórios.

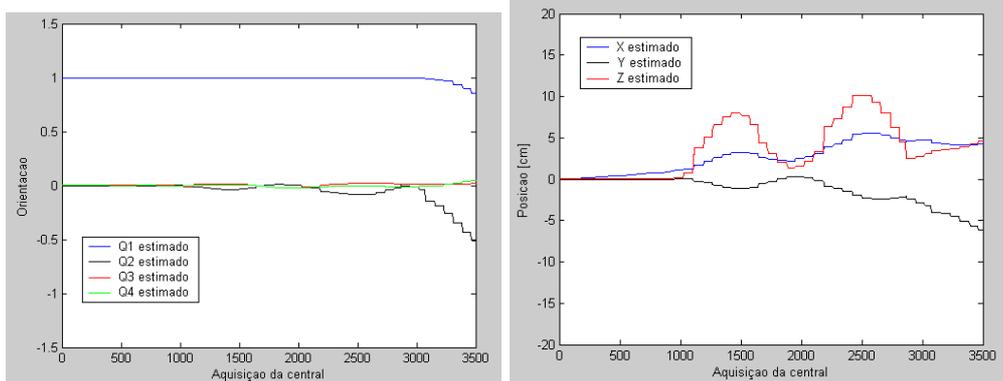
3.9 TESTE DAS ESTIMAÇÕES

Com parte do sistema definido, faltando apenas a propagação das incertezas de posição e velocidade, fez-se um teste de translação e rotação. Com os valores obtidos com o movimento teste traçou-se os gráficos apresentados na Fig. 3.17.

Neste teste, o movimento realizado manualmente foi uma oscilação, próxima de harmônica no eixo



(a) Estimacões da orientacão no movimento teste. (b) Estimacões da posicão no movimento teste.



(c) Estimacões da orientacão no começo do movimento teste. (d) Estimacões da posicão no começo do movimento teste.

Figura 3.17: Estimacões obtidas com movimento teste.

Z_B . Depois uma rotacão de π radianos em X_B e por fim outra rotacão de $-\pi$ radianos também no eixo X_B . Nas Fig.'s 3.17(a) e (b) não consegue-se perceber facilmente o movimento de translacão em Z_B devido a escala dos gráficos, porém este pode ser melhor visualizado nas Fig.'s 3.17(c) e (d) onde mostra-se as estimacões antes da execucao da rotacão em X_B , a qual, devido a calibracão implementada, propaga a aceleraçaõ da gravidade para os eixos X_W e Y_W .

4 CONCLUSÕES

Neste capítulo são apresentadas as conclusões mais pertinentes e propostas de trabalhos futuros.

No decorrer do desenvolvimento do projeto vários problemas encontrados na implementação do IMU anterior foram ultrapassados. Um destes foi a visualização de rotações do corpo, utilizando o IMU na prática e não somente em simulações. Estas, implementadas em MatLab, foram utilizadas para analisar alguns modelos e, assim, obter melhores modelagem e tratamento numérico para a atualização dos estados. Além disto, fracas calibrações em laboratório foram realizadas no sistema, para que assim, o mínimo de caracterização dos sensores fosse possível. Depois a construção do protótipo de IMU e de uma placa para interação com o usuário foram construídas e testadas. Por fim, uma biblioteca para atender as requisições do sistema foi implementada em Linux, além de um programa para visualização das estimações.

As simulações serviram para validar, de modo simplista, a propagação de incertezas utilizada no modelo dinâmico do sistema e apresentar as diferenças na propagação de erros com distintos métodos de integração numérica, utilizando-se diferentes dinâmicas na trajetória e diferentes taxas de amostragem. A simulação de estimação de posição foi implementada, porém sem a propagação das incertezas, assim como no sistema, onde a estimação da posição e de velocidade foram estimadas, mas também sem as incertezas das estimações.

Para o sistema, aplicou-se a teoria apresentada no decorrer do curso sobre Rede de Petri, com isto, um bom planejamento dos estados necessários ao funcionamento do sistema foi planejado. Com a facilidade de visualização da rede, previne-se estados indesejáveis para o mesmo e a não ocorrência de *Deadlocks*.

A biblioteca criada e o programa de visualização de movimentos, foram ambos implementados para serem utilizados em plataforma Linux e corresponderam de forma adequada aos objetivos. A utilização do módulo do *kernel* Linux, *usb-serial*, não mostrou grandes dificuldades para ser utilizada, assim a comunicação *usb* foi possível utilizando o protocolo *RS232*, facilmente implementado.

Por tratar-se de uma análise inicial sobre o problema de estimação de posições e orientações de um corpo utilizando sensores inerciais, vários trabalhos futuros são possíveis para melhoramento da implementação deste sistema. Um destes seria a utilização de sensores e circuitos com melhor precisão, para que assim possa se desenvolver e avaliar o algoritmo de calibração que utiliza as rotações da Terra e a gravidade para estimação de um estado inicial ao sistema.

Outra idéia seria utilizar magnetômetros para estimação de uma posição inicial e para futuras correções das estimações de orientação. Para as correções de incertezas, também é possível utilizar-se outros meios de localização, como o GPS.

Um tratamento que ficou pendente ao sistema foi o de propagação das incertezas da posição e velocidade, este tratamento deve ser executado futuramente para que o sistema responda a todos requisitos de seu correto funcionamento. Também poderia ser testado outro modelo para o cálculo das incertezas da orientação.

Pelo fato da limitação de velocidade do sistema, com um cristal de 16 Mhz, outro modelo de microcontrolador, ou processador, poderia ser utilizado para aquisição das informações do conversor A/D e de cálculo das estimações. Isto possibilitaria uma maior taxa de amostragem e um erro menor para o sistema, já que o que limitou esta taxa foram os processos realizados para estimação e não o conversor A/D.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MARCHANT, J. *3D reconstruction benefits IR lab*. October 2003. Diagnostic Imaging Online.
- [2] LEITAO, H. C. G. Digitalization and reconstruction of archaeological artifacts. *XIV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2001)*, p. 382, 2001.
- [3] MARQUES, M. F. *Sistema de localização 3D para veículos aéreos autônomos por meio de fusão de dados GPS e central inercial*. LCV - Ene, 2005.
- [4] COOKE M. J. ZYDA, D. R. P. R. B. M. J. M. Npsnet: Flight simulation dynamic modeling using quaternions. *In Presence*, v. 1, p. 404–420, 1994.
- [5] STILLWELL, J. The story of the 120 cell. *Notices of the AMS*, v. 48, p. 17–25, 2001.
- [6] WEISSTEIN, E. W. *Quaternion*. March 2004. From MathWorld - A Wolfram Web Resource. <http://mathworld.wolfram.com/Quaternion.html>.
- [7] HART J. K. FRANCIS, L. H. K. J. C. Visualizing quaternion rotation. *ACM Transactions on Graphics*, v. 13, p. 256–276, 1994.
- [8] FANG, B. G. Z. A. C. *Digital Simulation of Rotational Kinematics*. Goddard Space Flight Center Greenbelt, Md., 1969.
- [9] GREWAL, L. R. W. M. S. *Global Positioning Systems, Inertial Navigation, and Integration*. [S.l.]: John Wiley & Sons, Inc., 2001.
- [10] ANTENEODO, C. *Processos Estocásticos*. Julho 2004. V Escola do CBPF.
- [11] JAZWINSKI, A. H. *Stochastic Processes and Filtering Theory*. [S.l.]: Academic Press, 1970.
- [12] GEBRE-EGZIABHER G. H. ELKAIM, J. D. P. D.; PARKINSON, B. W. A gyro-free quaternion-based attitude determination system suitable for implementation using low cost sensors. *Proceedings of IEEE 2000 Position Location and Navigation Symposium*, v. 1, p. 185–192, 2000.
- [13] IV J. H. WALL, D. M. B. W. S. F. Characterization of various imu error sources and the effect on navigation performance. In: *2005 ION GNSS Conference*. [S.l.: s.n.], 2005.
- [14] PRESS S. A. TEUKOLSKY, W. T. V. W. H.; FLANNERY, B. P. *Numerical Recipes in C, The Art of Scientific Computing - Second Edition*. [S.l.]: Cambridge University Press, 1988.
- [15] BAKER, B. C. *Layout Tips for 12-Bit AD Converter Application*. [S.l.], 1980. Application Note 688.
- [16] ROGERS, R. M. *Applied Mathematics in Integrated Navigation Systems, Second Edition*. [S.l.]: AIAA, American Institute of Aeronautics and Astronautics, Inc., 2003.

I. DIAGRAMAS ESQUEMÁTICOS

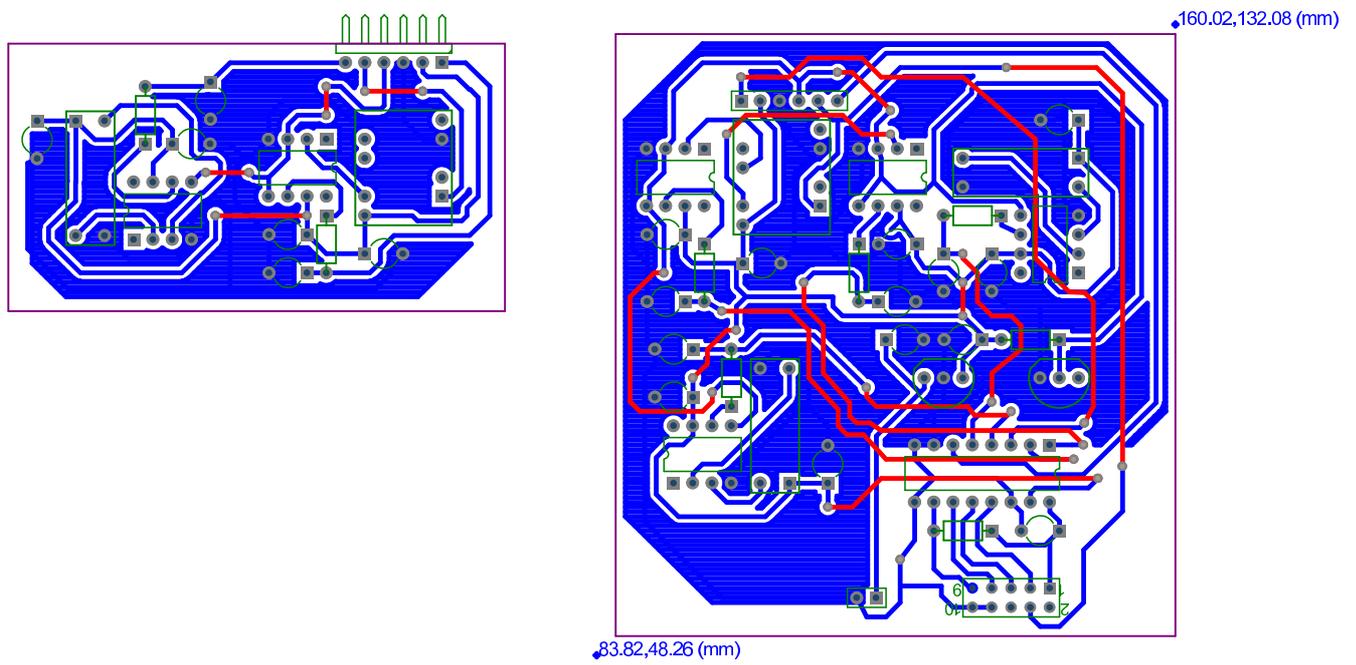
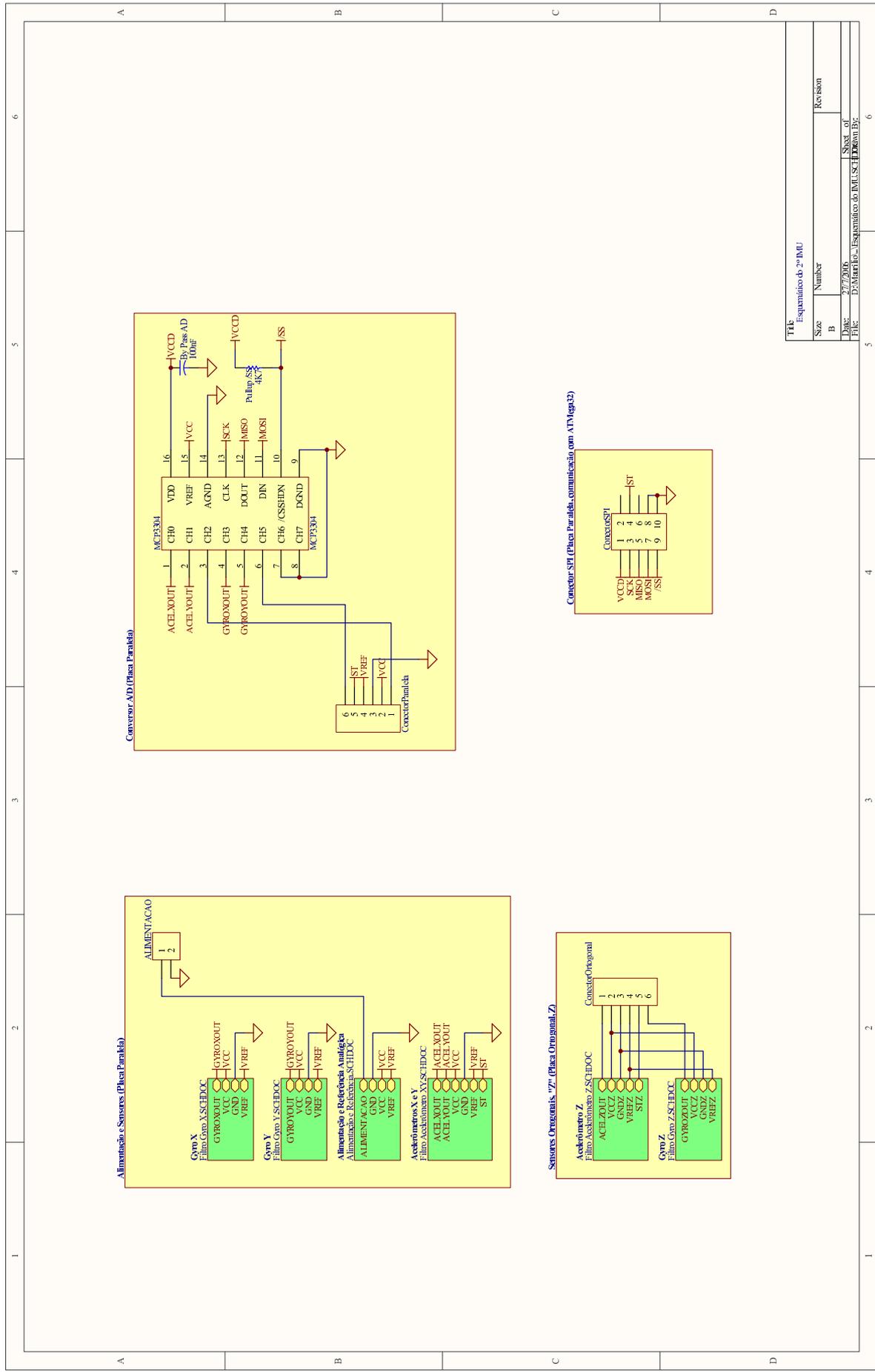
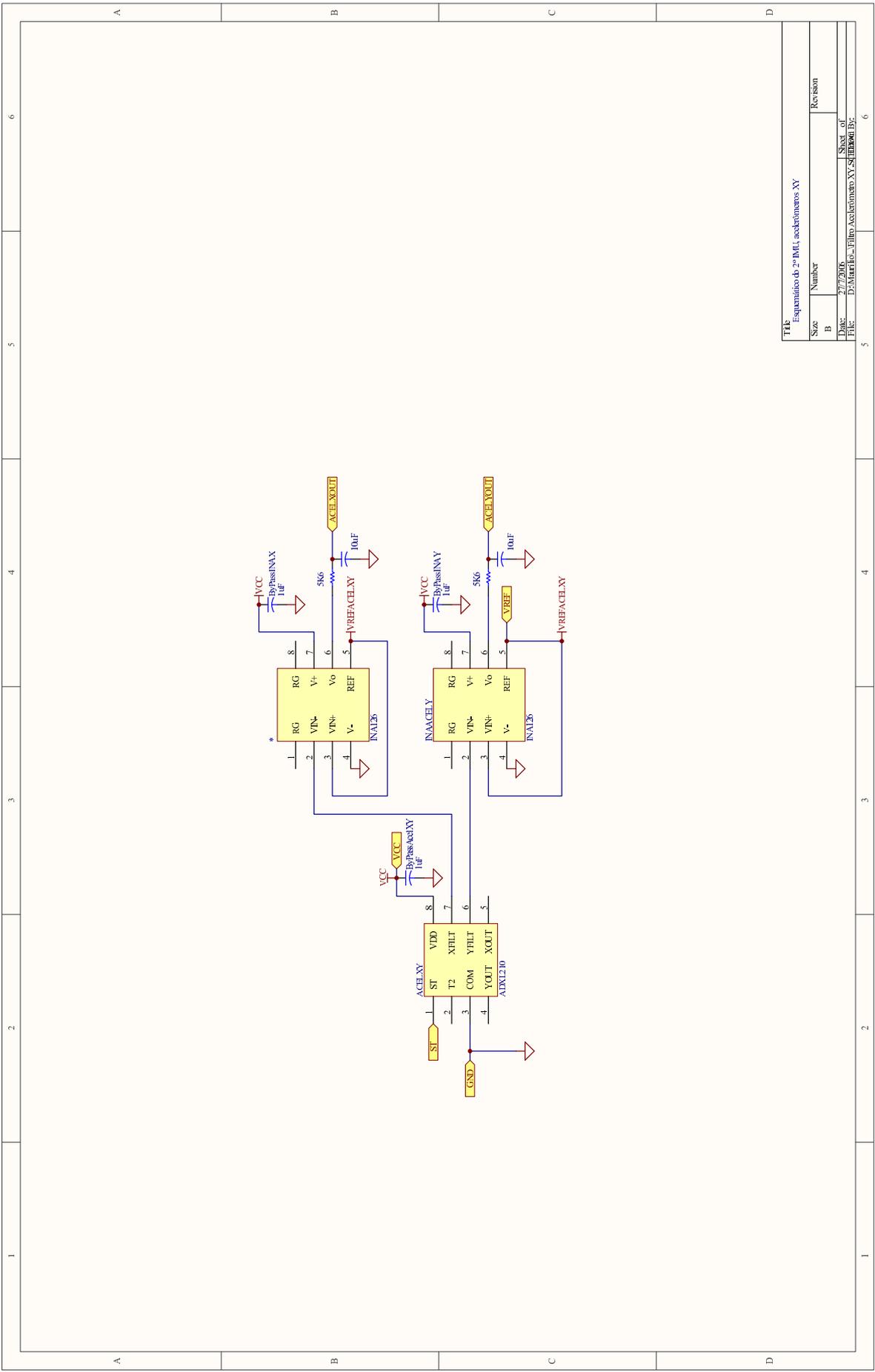
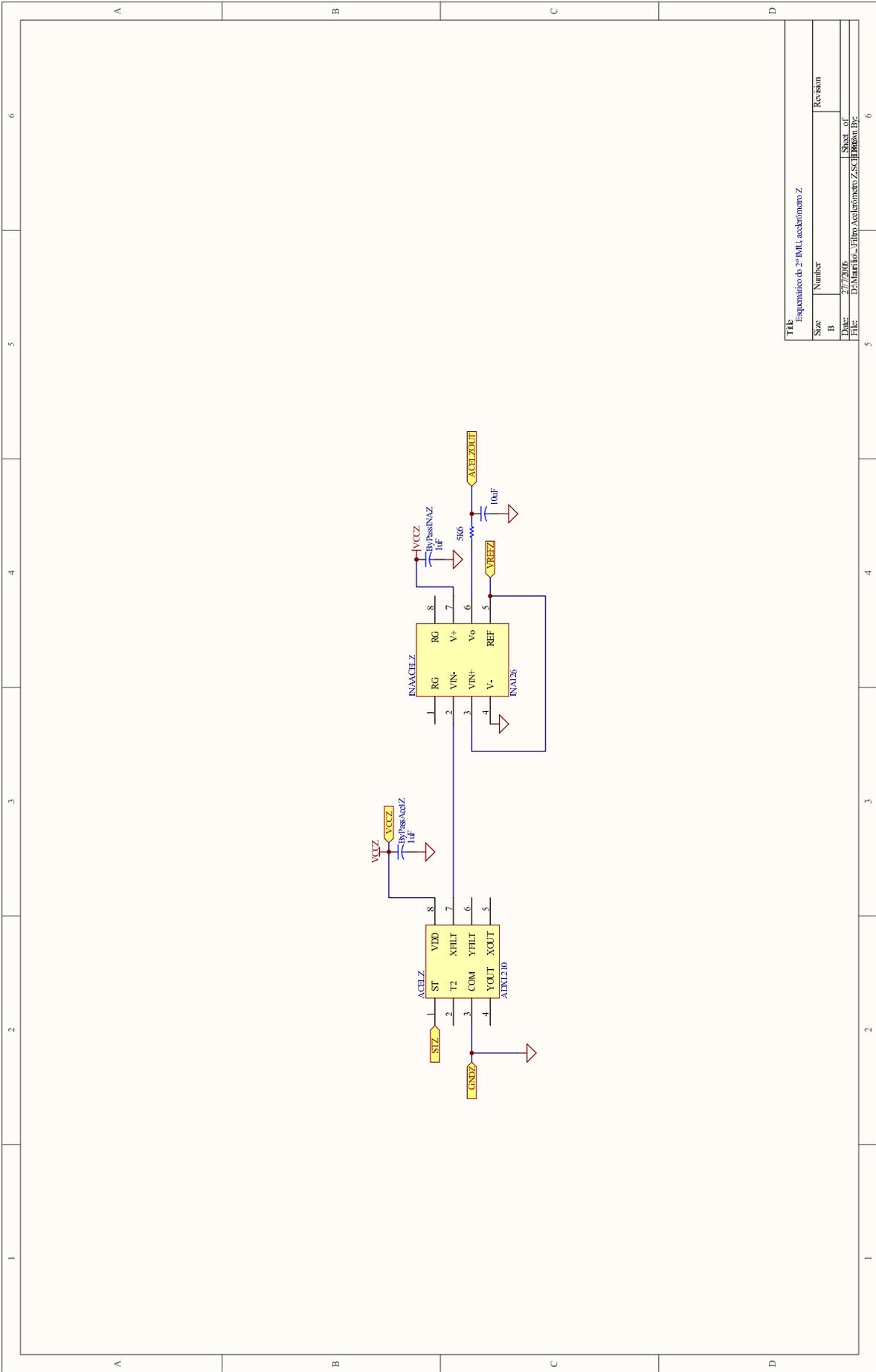


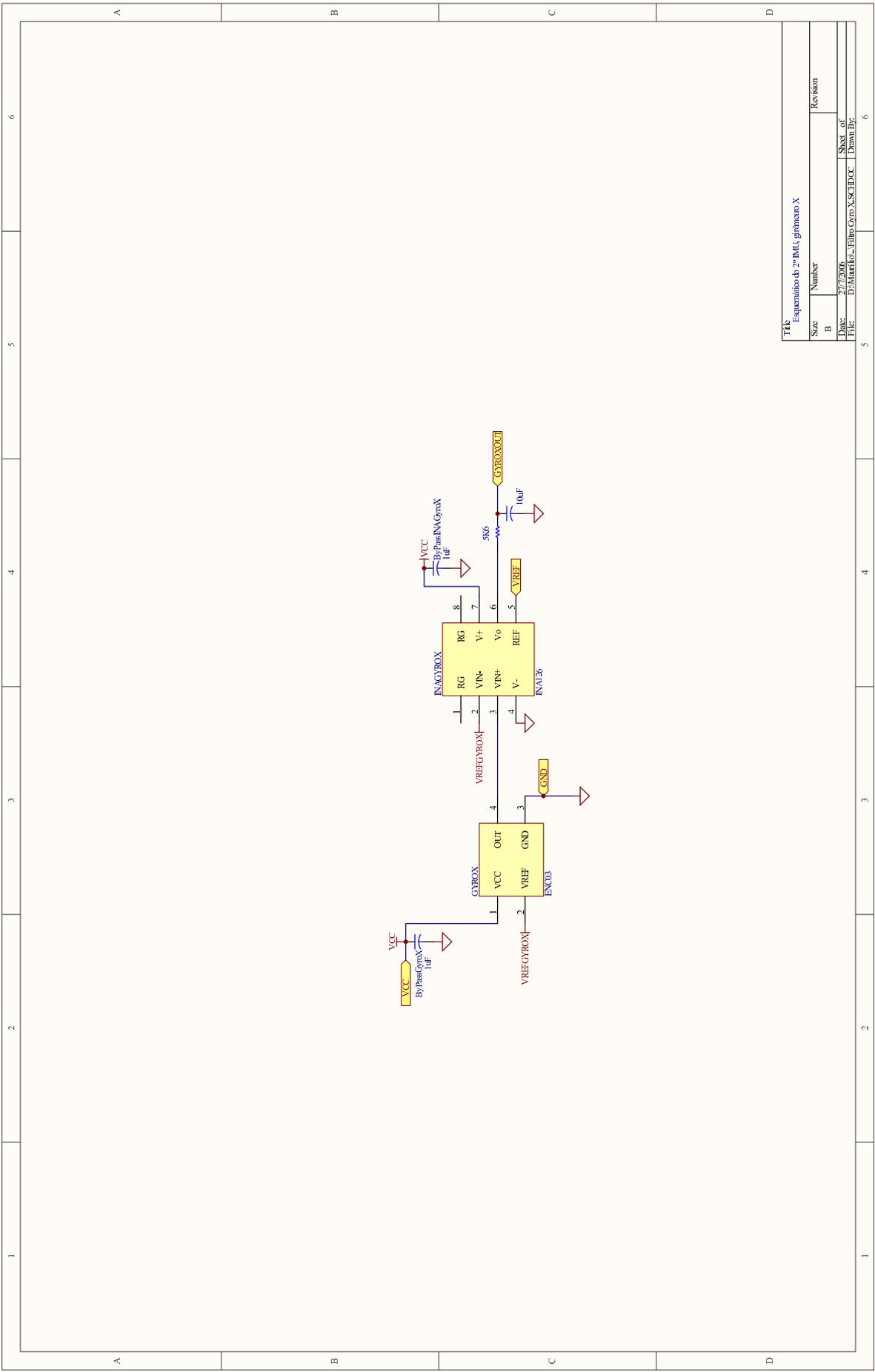
Figura I.1: PCB da placa do segundo protótipo de IMU





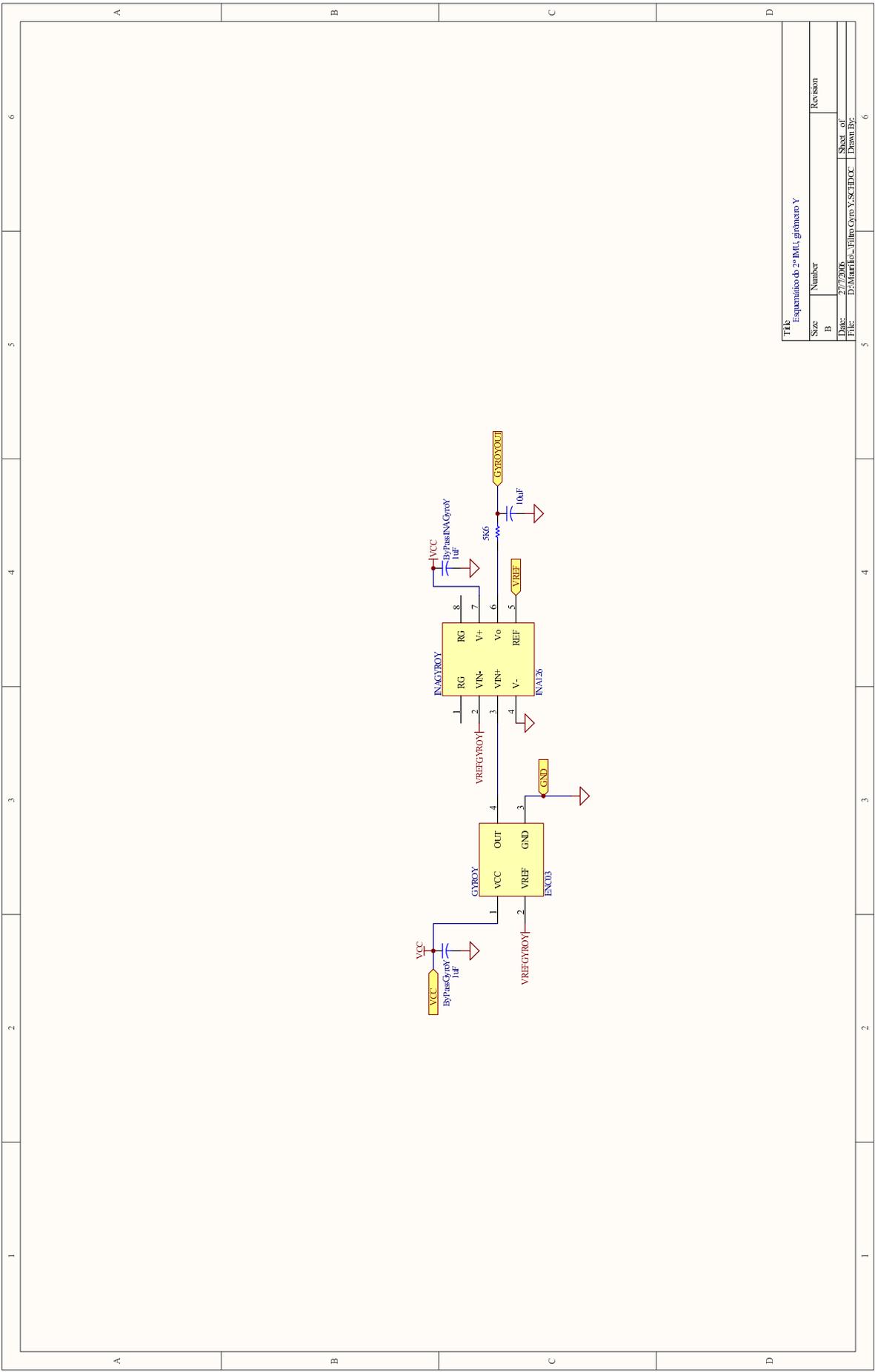
TIT: Esquema de 2º MLI, acelerómetro XY	
Size: B	Number: Revision
Date: 27/7/2016	Sheet of: 6
File: D:\Materiales\Elmto Acústico\2º MLI\2º MLI.dwg	



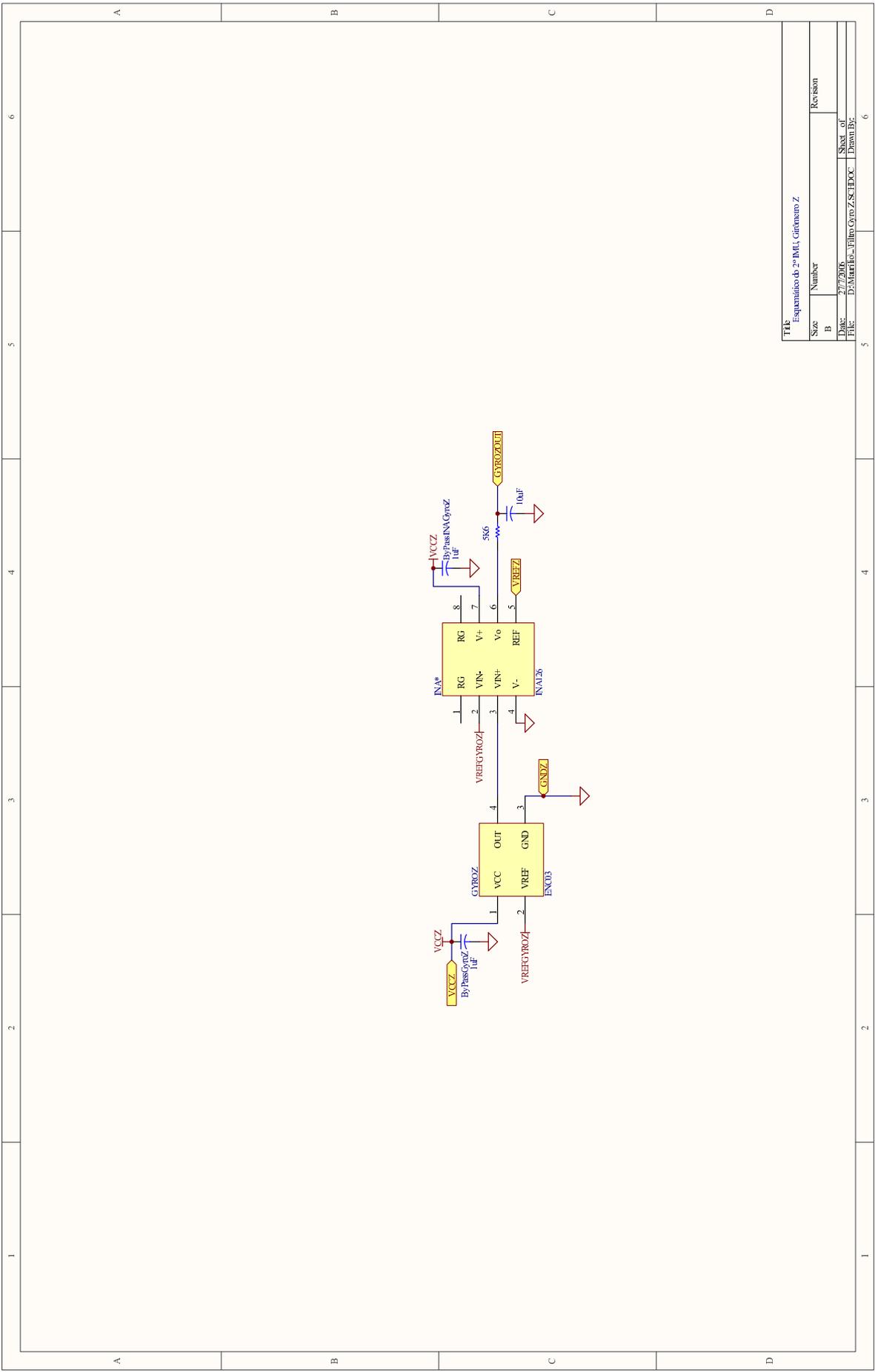


TITLE: Esquema do 2º MCU, giroscópio X

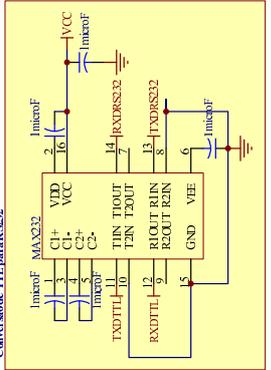
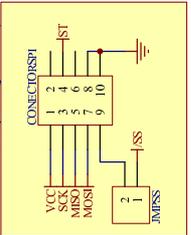
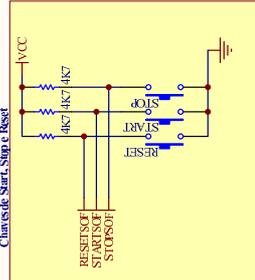
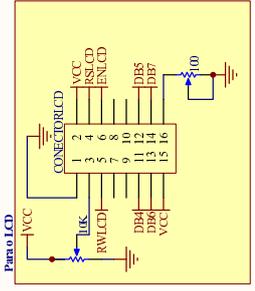
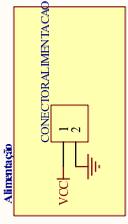
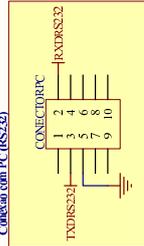
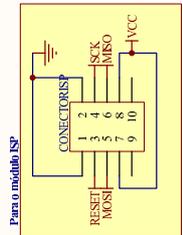
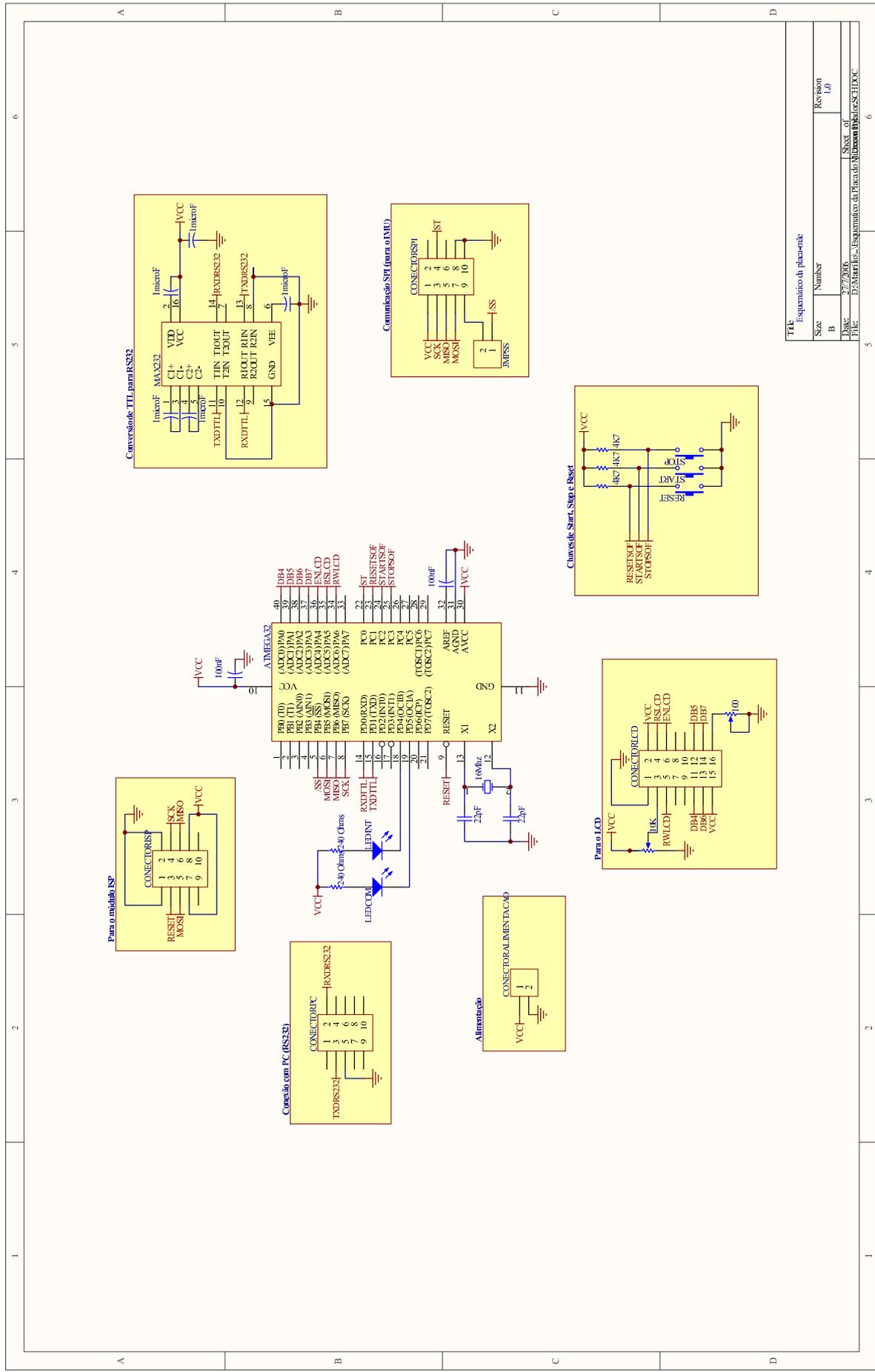
Size	Number	Revision
B	27/7/2016	
Drawn	D:\Marta\..._Filipe_Gyro_X_SCHDOC	Sheet of
File:		Drawn By:



TTL			
Esquema do 2º MCU, giroscópio Y			
Size:	Number	Revision	
B	27/7/2016	Sheet of	
File:	D:\Materiais\Elmex\Giro Y\SCHEM.DOC	Drawn by:	



TTL			
Esquema de 2º MLU Girómetro Z			
Size:	Number	Revision	
B	27/2016	Sheet of	
Date:	27/2016	Drawn by:	
File:	D:\Materiales\Elm\Gyro Z\CHOC	Sheet of	6



TTL - Esquematizado de placa emble

Size	Number	Revision
B		1.0
Date:	27/7/2006	Sheet of
File:	D:\Materiais... Esquematico da Placa do Módulo de Comunicação RS232.TXD.C	

II. DESCRIÇÃO DO CONTEÚDO DO CD

O CD possui:

- Documentações feitas;
- Esquemáticos de circuitos;
- Simulações implementadas;
- Códigos-fonte para os programas implementados no microcontrolador e no PC;
- Vídeos de experimentos executados.