

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM ENGENHARIA DE DEFESA

CAP JOSÉ ADALBERTO FRANÇA JUNIOR

SIMULAÇÃO E IMPLEMENTAÇÃO EM TEMPO REAL DE SISTEMAS
DE NAVEGAÇÃO INERCIAL INTEGRADOS INS/GPS

Rio de Janeiro
2009

INSTITUTO MILITAR DE ENGENHARIA

CAP JOSÉ ADALBERTO FRANÇA JUNIOR

**SIMULAÇÃO E IMPLEMENTAÇÃO EM TEMPO REAL DE
SISTEMAS DE NAVEGAÇÃO INERCIAL INTEGRADOS
INS/GPS**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia de Defesa do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia de Defesa.

Orientador: Prof. Jorge Audrin Morgado, Dr.-Ing
Co-orientador: Prof. Marcos Ferreira Duarte Pinto,
MSc

Rio de Janeiro
2009

c2009

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

F814s França Junior, José Adalberto
 Simulação e Implementação em Tempo Real de Sistemas de Navegação Inercial Integrados INS/GPS /
 José Adalberto França Junior. - Rio de Janeiro: Instituto Militar de Engenharia, 2009.
 112 p.: il., graf., tab.

Dissertação (mestrado) - Instituto Militar de Engenharia - Rio de Janeiro, 2009.

1. Sistemas de Navegação Inercial. 2. Integração (INS/GPS). 3. Sensores Iniciais. I. Título. II. Instituto Militar de Engenharia.

CDD 623.89

INSTITUTO MILITAR DE ENGENHARIA

CAP JOSÉ ADALBERTO FRANÇA JUNIOR

**SIMULAÇÃO E IMPLEMENTAÇÃO EM TEMPO REAL DE SISTEMAS
DE NAVEGAÇÃO INERCIAL INTEGRADOS INS/GPS**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia de Defesa do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia de Defesa.

Orientador: Prof. Jorge Audrin Morgado, Dr.-Ing

Co-orientador: Prof. Marcos Ferreira Duarte Pinto, MSc

Aprovada em 30 de Janeiro de 2009 pela seguinte Banca Examinadora:

Prof. Jorge Audrin Morgado, Dr.-Ing do IME - Presidente

Prof. Marcos Ferreira Duarte Pinto, MSc do IPqM

Prof. Pedro Cunha Campos Roquette, D.C. do IPqM

Prof. Max Suell Dutra, Dr.-Ing da COPPE/UFRJ

Prof. Alessandro Jacoud Peixoto, D.Sc. da CEFET/RJ

Rio de Janeiro
2009

Eu dedico este trabalho ao meu filho Micael e à minha esposa Elaine que tanto me apoiaram nesta jornada.

AGRADECIMENTOS

Durante a realização deste trabalho de dissertação em Pós-graduação em Engenharia de Defesa contei com a ajuda e a colaboração de diversas pessoas. Por isso, quero explicitar meus agradecimentos:

Ao meu orientador Cap Audrin pelo modo seguro, compreensivo e amigo com que orientou este trabalho.

Ao meu co-orientador Prof. Marcos Pinto que me incentivou e me apoiou nos momentos mais difíceis.

Ao Prof. Renato Durão e ao Prof. Pedro Roquette que sempre me ajudaram em todas as dúvidas sobre Engenharia Inercial e programação.

Ao Ten Cel Pellanda por ter me colocado no caminho certo no início de minha dissertação.

Ao Ten Cel Ades pela dedicação e ensinamento que transmitiu no decorrer do curso.

A todos os amigos que encontrei e os amigos que fiz durante os anos do curso de pós-graduação no IME e no IPqM, pelas inúmeras discussões e troca de idéias.

Aos meus pais, por serem os grandes responsáveis por minha formação humana e intelectual.

Em especial à minha esposa que eu admiro muito e que me ajudou bastante neste trabalho e ao meu filho Micael de onde tirei minhas inspirações.

E finalmente a Deus por ter me dado clareza e discernimento sobre como enfrentar os diversos obstáculos que tive em minha jornada.

SUMÁRIO

| | |
|--|-----------|
| LISTA DE ILUSTRAÇÕES | 8 |
| LISTA DE TABELAS | 12 |
| 1 INTRODUÇÃO | 17 |
| 1.1 Apresentação do Problema | 17 |
| 1.2 Aplicações da Navegação Inercial | 18 |
| 1.3 Objetivos do Trabalho Proposto | 19 |
| 1.4 Estudo Bibliográfico | 20 |
| 1.5 Convenções Adotadas | 20 |
| 2 SISTEMAS DE NAVEGAÇÃO INERCIAL | 22 |
| 2.1 Navegação Inercial | 22 |
| 2.2 Acelerômetros e Giroscópios | 24 |
| 2.3 Parâmetros de Navegação | 25 |
| 2.4 Sistemas de Coordenadas | 27 |
| 2.5 Equações de Navegação | 28 |
| 2.5.1 Transformação de Coordenadas | 29 |
| 2.6 Taxa de Rotação Angular da Terra (<i>Earth Rate</i>) | 30 |
| 2.7 Taxa de Transporte (<i>Transport Rate</i>) | 31 |
| 2.8 Equação da dinâmica da posição | 33 |
| 2.9 Equação da Dinâmica da Atitude | 33 |
| 2.10 Equação da Dinâmica da Velocidade | 34 |
| 2.11 Equações da Navegação Inercial | 35 |
| 2.12 Discretização das equações de Navegação | 36 |
| 3 INTEGRAÇÃO INS/GPS COM O USO DO FILTRO DE KALMAN | 38 |
| 3.1 Integração INS/GPS | 38 |
| 3.2 Desenvolvimento das equações da dinâmica dos erros e o Filtro de Kalman .. | 39 |
| 3.2.1 Dinâmica dos erros em Posição | 40 |
| 3.2.2 Dinâmica dos erros em Velocidade | 40 |
| 3.2.3 Dinâmica dos erros em Atitude | 41 |

| | | |
|----------|---|------------|
| 3.2.4 | Equacionamento do Filtro de Kalman | 42 |
| 4 | ESTIMAÇÃO DE ERROS DOS SENSORES E ALINHAMENTO | 46 |
| 4.1 | Tipos de erros dos sensores inerciais | 46 |
| 4.2 | Calibração | 47 |
| 4.3 | Alinhamento | 47 |
| 4.3.1 | Alinhamento grosso | 48 |
| 4.3.2 | Alinhamento grosso com sensores tipo MEMS | 51 |
| 5 | IMPLEMENTAÇÃO DO ALGORITMO | 53 |
| 5.1 | Modelo implementado em Simulink | 53 |
| 5.2 | Implementação em tempo real com o <i>xPC Target</i> | 56 |
| 5.3 | Gerador de trajetória - O simulador <i>FlightGear</i> | 59 |
| 6 | RESULTADOS PRÁTICOS | 73 |
| 6.1 | Simulador de Movimentos | 73 |
| 6.2 | Trajetória com a kombi | 83 |
| 7 | CONCLUSÕES E PROPOSTAS PARA FUTUROS TRABALHOS | 109 |
| 7.1 | Conclusões | 109 |
| 7.2 | Propostas para Trabalhos Futuros | 109 |
| 7.2.1 | Gerador de Trajetórias | 109 |
| 7.2.2 | Implementação em tempo real com diferentes RTOS e avaliação de robustez | 110 |
| 7.2.3 | Refinamento do algoritmo INS/GPS quanto à falta de GPS e tratamento de ruídos | 110 |
| 8 | REFERÊNCIAS BIBLIOGRÁFICAS | 111 |

LISTA DE ILUSTRAÇÕES

| | | |
|----------|--|----|
| FIG.2.1 | Sistema de Navegação Inercial <i>Strapdown</i> | 23 |
| FIG.2.2 | Sistema de Navegação Inercial <i>Gimbaled</i> | 23 |
| FIG.2.3 | IMU tipo MEMS da <i>Spark Fun Electronics</i> SEN-00842 | 25 |
| FIG.2.4 | Parâmetros de Atitude | 26 |
| FIG.2.5 | Sistema de Coordenadas | 27 |
| FIG.2.6 | Movimento norte-sul no plano meridional | 31 |
| FIG.2.7 | Movimento leste-oeste no plano equatorial | 32 |
| FIG.3.1 | Implementação do navegador integrado na configuração de realimentação | 39 |
| FIG.5.1 | Algoritmo de navegação inercial integrado com o GPS | 53 |
| FIG.5.2 | Algoritmo de navegação inercial puro - INS | 54 |
| FIG.5.3 | Algoritmo do Filtro de <i>Kalman</i> | 55 |
| FIG.5.4 | Algoritmo de navegação inercial integrado com GPS implementado para a geração de código no <i>xPC Target</i> | 57 |
| FIG.5.5 | Algoritmo da comunicação serial do <i>xPC Target</i> com a IMU400 | 57 |
| FIG.5.6 | Algoritmo da comunicação serial do <i>xPC Target</i> com o GPS 18-5Hz <i>Garmin</i> | 58 |
| FIG.5.7 | Modelo de demonstração do Simulink "asbdhc2.mdl" | 59 |
| FIG.5.8 | Modelo "asbdhc2.mdl" modificado para operar como Gerador de trajetória | 60 |
| FIG.5.9 | Simulação de vôo com o <i>FlightGear</i> a partir dos parâmetros calculados pelo INS/GPS | 61 |
| FIG.5.10 | Análise do ângulo de <i>roll</i> com o INS puro | 62 |
| FIG.5.11 | Análise do ângulo de <i>pitch</i> com o INS puro | 63 |
| FIG.5.12 | Análise do ângulo de <i>head</i> com o INS puro | 63 |
| FIG.5.13 | Análise da latitude com o INS puro | 64 |
| FIG.5.14 | Análise da longitude com o INS puro | 64 |
| FIG.5.15 | Análise da altitude com o INS puro | 65 |
| FIG.5.16 | Análise da velocidade norte V_n com o INS puro | 65 |
| FIG.5.17 | Análise da velocidade leste V_e com o INS puro | 66 |
| FIG.5.18 | Análise da velocidade vertical V_d com o INS puro | 66 |

| | | |
|----------|--|----|
| FIG.5.19 | Análise do ângulo de <i>roll</i> com o INS/GPS | 67 |
| FIG.5.20 | Análise do ângulo de <i>pitch</i> com o INS/GPS | 67 |
| FIG.5.21 | Análise do ângulo de <i>head</i> com o INS/GPS | 68 |
| FIG.5.22 | Análise da latitude com o INS/GPS | 68 |
| FIG.5.23 | Análise da longitude com o INS/GPS | 69 |
| FIG.5.24 | Análise da altitude com o INS/GPS | 69 |
| FIG.5.25 | Análise da velocidade norte V_n com o INS/GPS | 70 |
| FIG.5.26 | Análise da velocidade leste V_e com o INS/GPS | 70 |
| FIG.5.27 | Análise da velocidade vertical V_d com o INS/GPS | 71 |
| FIG.5.28 | Trajetória do vôo feito no <i>FlightGear</i> plotada no <i>Google Earth</i> a partir dos dados calculados pelo INS/GPS | 72 |
| | | |
| FIG.6.1 | Simulador de movimentos de 2 eixos Contraves | 73 |
| FIG.6.2 | PC104 utilizado com o <i>xPC Target</i> | 74 |
| FIG.6.3 | Análise do ângulo de <i>roll</i> gerado pelo simulador de movimentos | 76 |
| FIG.6.4 | Análise do ângulo de <i>pitch</i> gerado pelo simulador de movimentos | 76 |
| FIG.6.5 | Análise do ângulo de <i>head</i> gerado pelo simulador de movimentos | 77 |
| FIG.6.6 | Análise do erro em latitude gerado pelo simulador de movimentos | 77 |
| FIG.6.7 | Análise do erro em longitude gerado pelo simulador de movimentos | 78 |
| FIG.6.8 | Análise da altitude gerada pelo simulador de movimentos | 78 |
| FIG.6.9 | Análise da velocidade norte V_n gerada pelo simulador de movimentos | 79 |
| FIG.6.10 | Análise da velocidade leste V_e gerada pelo simulador de movimentos | 79 |
| FIG.6.11 | Análise da velocidade vertical V_d gerada pelo simulador de movimentos | 80 |
| FIG.6.12 | Análise da velocidade angular ω_x lida da IMU | 80 |
| FIG.6.13 | Análise da velocidade angular ω_y lida da IMU | 81 |
| FIG.6.14 | Análise da velocidade angular ω_z lida da IMU | 81 |
| FIG.6.15 | Análise da aceleração linear A_x lida da IMU | 82 |
| FIG.6.16 | Análise da aceleração linear A_y lida da IMU | 82 |
| FIG.6.17 | Análise da aceleração linear A_z lida da IMU | 83 |
| FIG.6.18 | kombi usada no experimento | 84 |
| FIG.6.19 | Sistema de Navegação Inercial sobre a kombi | 84 |
| FIG.6.20 | Trajetória percorrida pela kombi | 87 |

| | | |
|----------|--|-----|
| FIG.6.21 | Análise da velocidade angular ω_x lida da IMU (kombi) | 88 |
| FIG.6.22 | Análise da velocidade angular ω_y lida da IMU (kombi) | 88 |
| FIG.6.23 | Análise da velocidade angular ω_z lida da IMU (kombi) | 89 |
| FIG.6.24 | Análise da aceleração linear A_x lida da IMU (kombi) | 89 |
| FIG.6.25 | Análise da aceleração linear A_y lida da IMU (kombi) | 90 |
| FIG.6.26 | Análise da aceleração linear A_z lida da IMU (kombi) | 90 |
| FIG.6.27 | Análise da latitude lida pelo GPS 18-5Hz <i>Garmin</i> | 91 |
| FIG.6.28 | Análise da longitude lida pelo GPS 18-5Hz <i>Garmin</i> | 91 |
| FIG.6.29 | Análise da altitude lida pelo GPS 18-5Hz <i>Garmin</i> | 92 |
| FIG.6.30 | Análise da velocidade norte (V_n) lida pelo GPS 18-5Hz <i>Garmin</i> | 92 |
| FIG.6.31 | Análise da velocidade leste (V_e) lida pelo GPS 18-5Hz <i>Garmin</i> | 93 |
| FIG.6.32 | Análise da velocidade vertical (V_d) lida pelo GPS 18-5Hz <i>Garmin</i> | 93 |
| FIG.6.33 | Análise do <i>head</i> lido pelo GPS 18-5Hz <i>Garmin</i> | 94 |
| FIG.6.34 | Análise do "Track Status" do GPS 18-5Hz <i>Garmin</i> | 95 |
| FIG.6.35 | Análise do número de satélites rastreados do GPS 18-5Hz <i>Garmin</i> | 95 |
| FIG.6.36 | <i>Roll</i> calculado pelo algoritmo INS/GPS | 96 |
| FIG.6.37 | <i>Pitch</i> calculado pelo algoritmo INS/GPS | 97 |
| FIG.6.38 | Comparação entre o <i>head</i> calculado pelo algoritmo e o <i>head</i> do GPS <i>Garmin</i> | 97 |
| FIG.6.39 | Comparação entre a latitude calculada pelo algoritmo e a latitude do GPS <i>Garmin</i> | 98 |
| FIG.6.40 | Comparação entre a longitude calculada pelo algoritmo e a longitude do GPS <i>Garmin</i> | 98 |
| FIG.6.41 | Comparação entre a altitude calculada pelo algoritmo e a altitude do GPS <i>Garmin</i> | 99 |
| FIG.6.42 | Comparação entre o V_n calculado pelo algoritmo e o V_n do GPS <i>Garmin</i> | 99 |
| FIG.6.43 | Comparação entre o V_e calculado pelo algoritmo e o V_e do GPS <i>Garmin</i> | 100 |
| FIG.6.44 | Comparação entre o V_d calculado pelo algoritmo e o V_d do GPS <i>Garmin</i> | 100 |
| FIG.6.45 | Comparação das latitudes calculadas pelo algoritmo com os valores dos marcos | 101 |

| | |
|--|-----|
| FIG.6.46 Comparação das longitudes calculadas pelo algoritmo com os valores dos marcos | 102 |
| FIG.6.47 Comparação das altitudes calculadas pelo algoritmo com os valores dos marcos | 102 |
| FIG.6.48 Comparação da latitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos | 103 |
| FIG.6.49 Comparação entre o erro da latitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos | 104 |
| FIG.6.50 Comparação da longitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos | 104 |
| FIG.6.51 Comparação entre o erro da longitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos | 105 |
| FIG.6.52 Comparação da altitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos | 105 |
| FIG.6.53 Comparação entre o erro da altitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos | 106 |
| FIG.6.54 dados da trajetória percorrida pelo kombi - GPS <i>Garmin</i> e Marcos | 107 |
| FIG.6.55 dados da trajetória percorrida pelo kombi - INS/GPS e Marcos | 107 |
| FIG.6.56 dados da trajetória percorrida pelo kombi - GPS <i>Garmin</i> , INS/GPS e Marcos | 108 |

LISTA DE TABELAS

| | | |
|---------|---|----|
| TAB.2.1 | Comparativo de Plataformas | 24 |
| TAB.3.1 | Características isoladas dos sistemas INS e GPS | 38 |
| TAB.5.1 | Níveis de ruído adicionados aos sensores para a simulação | 62 |
| TAB.6.1 | Especificações da IMU400CD-200 da CrossBow | 74 |
| TAB.6.2 | Especificações do GPS 18-5Hz Garmin | 85 |
| TAB.6.3 | Especificações do NAV420 da CrossBow | 86 |

LISTA DE ABREVIATURAS E SÍMBOLOS

ABREVIATURAS

| | |
|------|--|
| INS | - <i>Inertial Navigation System</i> |
| GPS | - <i>Global Positioning Systems</i> |
| IMU | - de Medição Inercial (<i>Inertial Measurement Unit</i>) |
| RTW | - <i>Real Time Workshop</i> |
| VANT | - Veículo Aéreo Não Tripulado |
| VSA | - Veículo Submarino Autônomo |
| VTA | - Veículo Terrestre Autônomo |
| RTOS | - <i>Real Time Operational System</i> |
| MEMS | - <i>Micro Electro Mechanical Systems</i> |
| PIG | - <i>Pipeline Inspection Gauge</i> |
| MTC | - Matriz de Transformação de Coordenadas |
| DSP | - <i>Digital Signal Processor</i> |

SÍMBOLOS

| | |
|----------------------|--|
| φ | - ângulo de latitude |
| λ | - ângulo de longitude |
| h | - altitude ao nível do mar |
| V_n | - velocidade norte (North) no referencial navegacional |
| V_e | - velocidade leste (East) no referencial navegacional |
| V_d | - velocidade vertical (Down) no referencial navegacional |
| ϕ | - ângulo de rolagem <i>roll</i> |
| θ | - ângulo de arfagem <i>pitch</i> |
| ψ | - ângulo de guinagem <i>head</i> |
| C_b^n | - matriz de transformação de coordenadas do referencial b para o referencial n |
| $\underline{\omega}$ | - vetor velocidade angular |
| Ω | - matriz anti-simétrica do vetor $\underline{\omega}$ |
| $\underline{\gamma}$ | - vetor gravidade normal |
| \underline{f} | - força específica ou aceleração não gravitacional |
| ϵ | - erros de desalinhamento em atitude |

RESUMO

O objetivo deste trabalho é a implementação em tempo real de um algoritmo de navegação inercial integrado com o GPS (INS/GPS) e o desenvolvimento de um ambiente de simulação para prototipação rápida deste mesmo algoritmo. Também foi usado para a validação e testes do algoritmo de navegação inercial o simulador de vôo *FlightGear* integrado ao MATLAB. Com o *FlightGear* tem-se um excelente gerador de trajetórias para se analisar o comportamento do algoritmo de navegação inercial em diferentes dinâmicas de veículos.

A plataforma escolhida para a simulação e implementação em tempo real do algoritmo de navegação foi o MATLAB/Simulink com os *toolboxes Real Time Workshop* (RTW) e *xPC Target*, onde o RTW gera o código em C a partir do modelo desenvolvido em Simulink para a plataforma de tempo real *xPC Target*. A escolha desta ferramenta, deveu-se ao fato da simplicidade de se construir modelos no Simulink e na possibilidade de geração de código C com diretrizes de tempo real a partir deste modelo.

Como estudo de caso, foram realizados dois experimentos reais: um teste em um simulador de movimentos de 2 eixos da Contraves e outro teste em um aparato montado sobre uma viatura tipo kombi que percorreu um trajeto pré-definido. Os resultados foram gravados e analisados. O equipamento usado nestes experimentos foi um computador PC104 que processou em tempo real os dados de uma Unidade de Medidas Iniciais (IMU) do tipo "MEMS" e de um GPS que fornece posição, velocidade e *head*.

ABSTRACT

The goal of this work is the implementation in real time of a inertial navigation system algorithm integrated with GPS (INS/GPS) and the development of a simulation enviroment for rapid prototyping. For tests and validate of this algorithm was used the flight simulator FlightGear integrated with Matlab. With Flightgear, one has an exelent trajectory generator to analyse the behaviour of the inertial navigation algorithm in differents vehicles dynamics.

The plataform choosed for the simulation and the real time implementation of the algorithm was MATLAB/Simulink with Real Time Workshop (RTW) and XPC Target toolboxes, where RTW generates the C code from the INS/GPS Simulink model for the xPC Target plataform. The choice of this tool, was due to the fact that is more simple to build Simulink models and one has the possibility of generate C code with real time assertives from this high level model.

Two real experiments were made: a test in a Contraves movement simulator of 2 axes and another test in a vehicle with the inertial navigation system mounted on. this vehicle travelled a pre-set path, and its results were analysed and recorded. The equipment used in this experiments were a PC104 computer wich processed in real time the data from a MEMS Inertial Measure Unit (IMU) and a GPS wich reads position, velocity and heading.

1 INTRODUÇÃO

1.1 APRESENTAÇÃO DO PROBLEMA

Navegação é definida como a determinação da posição, velocidade e orientação angular (atitude) de um veículo em relação a um sistema de coordenadas. Um INS (Inertial Navigation System) ou Sistema de Navegação Inercial é um sistema que usa um computador e sensores de movimento para calcular continuamente a posição, a orientação e a velocidade de um veículo sem a necessidade de referências externas como o GPS (*Global Positioning System*), por exemplo, tornando-o auto suficiente. Esta vantagem torna o INS único e bastante atraente para aplicações de navegação.

Sistemas de navegação inercial têm uma enorme gama de aplicações, e podem ser usados em diferentes tipos de veículos, como: aeronaves, mísseis, submarinos, torpedos, navios, plataformas flutuantes, carros, etc.

Como todo processo, a navegação inercial tem vantagens e desvantagens. Imprecisão na trajetória como resultado de erros acumulados com o tempo, é sua principal desvantagem. Erros estes que são consequência, principalmente, da propagação dos erros de medição dos sensores e dos erros de integração numérica.

Esta desvantagem pode ser atenuada se usarmos o INS integrado com outros sensores não inerciais, como o GPS, o baroaltímetro, o magnetômetro, etc. Se optarmos pelo GPS para integrar o INS, por exemplo, temos um Sistema de Navegação Inercial Integrado com o GPS (INS/GPS). Sabendo que o GPS também tem suas componentes de erro nas informações fornecidas, o Filtro de Kalman é normalmente usado para fazer esta fusão e assim fornecer com a maior exatidão possível os parâmetros desejados.

O Filtro de Kalman determina, segundo um critério de otimização, os dados de navegação a partir das medidas dos sensores inerciais, de um modelo determinístico que descreve a dinâmica dos erros de um INS, e de um modelo que descreve o comportamento estocástico dos erros das medidas dos sensores inerciais. Portanto, a qualidade das estimativas obtidas pelo Filtro de Kalman depende da exatidão das medidas fornecidas pela IMU (Unidade de Medidas Iniciais) e do desempenho dos modelos adotados. O desempenho do INS portanto, está diretamente relacionado à qualidade dos sensores inerciais: acelerômetros e giroscópios.

Em se tratando de simulação e implementação em hardware, os avanços nesta área

são significativos, principalmente no que se refere à linguagens de alto nível. Um exemplo é a ferramenta *Real Time Workshop* do software Matlab da Mathworks. Até então, o desenvolvimento de aplicações em tempo real exigia um bom conhecimento de C/C++, programação em tempo real e as vezes um profundo conhecimento de métodos numéricos. Um interessante modo de se contornar isso é oferecido pelo Matlab/Simulink/*Real Time Workshop*, ou simplesmente RTW.

O RTW (*Real Time Workshop*) é um gerador automático de código C com diretiwas de tempo real para o Simulink. Com o Simulink é possível criar, simular e analisar sistemas dinâmicos complexos com simples conexões entre blocos funcionais, sendo a maioria já disponíveis em várias bibliotecas pré-configuradas dentro de uma interface gráfica amigável. A principal vantagem de seu uso reside na notação matemática familiar para expressar o problema bem como na visualização e análise de qualquer sinal do modelo. Ao se usar esta ferramenta, o usuário pode se concentrar estritamente nos problemas de modelagem e controle ao invés de gastar um tempo considerável em problemas de programação. É o RTW, então, que controla a tradução do modelo feito em Simulink para uma série de funções em C que também é compilado e montado para então se obter um programa.

1.2 APLICAÇÕES DA NAVEGAÇÃO INERCIAL

O desenvolvimento da navegação inercial iniciou-se por volta de 1900 na Alemanha e nos Estados Unidos visando aplicações militares. Com o domínio das técnicas, sua utilização foi expandida para o meio civil.

Das inúmeras aplicações dos sistemas iniciais podem-se destacar:

- Navegação, guiagem e controle de navios, submarinos e aeronaves;
- Guiagem de torpedos e mísseis de curto, médio e longo alcance;
- Geodésia e cartografia;
- Robótica, VANT (Veículo Aéreo não Tripulado), VSA (Veículo Submarino Autônomo) e VTA (Veículo Terrestre Autônomo);
- Estabilização de plataformas (de tiro, de petróleo, de equipamentos de filmagens);
- Missões espaciais (lançamento de satélites);
- Estabilização de satélites;

- Acompanhamento de trajetória de perfuração petrolífera;
- Guiagem, navegação e controle de veículos quando empregados em regiões inóspitas ou em situações táticas;
- Sistema de geo-referenciamento de dutos e PIG (*Pipeline Inspection Gauge*);
- Veículos terrestres autônomos;
- Indústria de entretenimentos;
- Mapeamento Topográfico;
- Detecção de falhas estruturais por meio de análise de vibrações.

1.3 OBJETIVOS DO TRABALHO PROPOSTO

O objetivo deste trabalho é a simulação e implementação em tempo real de um algoritmo de navegação inercial integrado com o GPS.

Para isso, optou-se por usar o algoritmo discretizado por ser mais adequado para ser implementado em hardware e também para a programação em tempo real. A grande vantagem de se utilizar algoritmos discretos é a possibilidade de se usar um tempo de amostragem não tão pequeno, possibilitando o uso de processadores modestos e mais baratos e ainda sim obter resultados bastante próximos da realidade, embora nos dias de hoje a velocidade de processamento não seja um problema tão vivenciado quanto nas décadas de 60 a 90.

A plataforma escolhida para a simulação e implementação em tempo real do algoritmo de navegação foi o MATLAB/Simulink com os *toolboxs Real Time Workshop* e *xPC Target*. O *Real Time Workshop* gera e otimiza códigos portáveis e customizáveis ANSI C ou C++ de modelos do Simulink para criar implementações específicas de modelos que operam em tempo real. O código gerado pode ser executado em PCs, microcontroladores e DSPs e ainda em vários RTOS (Sistema Operacional de Tempo Real) diferentes.

O *xPC Target* é um ambiente para a prototipagem, teste e desenvolvimento de sistemas que funcionam em tempo real para computadores padrões. Ele utiliza um computador *Target*, separado do computador *Host*, para rodar aplicações em tempo real. Um computador target pode ser praticamente qualquer computador com um Intel 386 ou 486 ou AMD K5 ou K6/Athlon, pode ser um computador de mesa ou um computador industrial como um PC104, o qual será usado neste trabalho. A partir de uma simulação criada

no Simulink, pode-se gerar um código executável a partir do RTW, tendo como alvo o próprio *xPC Target*. O código executável e o *xPC Target real-time Kernel* são então gravados do computador *Host* para o computador *Target*. Em seguida, pode-se executar a aplicação em tempo real, com a possibilidade de variação "online" dos parâmetros do modelo.

1.4 ESTUDO BIBLIOGRÁFICO

Os algoritmos de navegação inercial *strapdown* encontram-se em (SAVAGE, 1998a), (SAVAGE, 1998b), (BEKIR, 2007) e (TITTERTON & WESTON, 1997). Recentemente novas abordagens para acomodar modernas tecnologias computacionais foram feitas por (LITMANOVICH et al., 2000).

As equações de observação do Filtro de Kalman para a integração entre INS e GPS encontram-se em (SHIN, 2001), (BEKIR, 2007) e (TITTERTON & WESTON, 1997), onde é descrito o modelo em espaço de estados dos erros de posição entre as medidas do INS e as medidas do GPS.

O uso do Simulink e dos toolboxes Real Time Workshop e xPC Target do MATLAB podem ser encontrados em (DABNEY & HARMAN, 2003), (MATHWORKS-SIMULINK, 1999), (MATHWORKS-RTW, 2007) e (MATHWORKS-XPC, 2008).

1.5 CONVENÇÕES ADOTADAS

As notações utilizadas nessa dissertação seguirão as convenções listadas abaixo:

- a) Matrizes são representadas por letras maiúsculas;
- b) Vetores são representados por letras minúsculas e sublinhado;
- c) Os referenciais que estão envolvidos em transformações de coordenadas são representadas em sobreescrito e subscrito. Por exemplo, C_b^n é a matriz de transformação de coordenadas do referencial do corpo para o referencial de navegação. Para vetores representando movimentos angulares, o subscrito representa a referência e o eixo alvo enquanto o sobreescrito representa o referencial o qual o vetor é projetado. Por exemplo, $\underline{\omega}_{ib}^n$ representa o vetor velocidade angular do referencial do corpo em relação ao referencial inercial projetado no referencial de navegação.
- d) Os símbolos utilizados pelos operadores são definidos como:

- \cdot - derivada temporal
- -1
- T - matriz inversa
- T - matriz transposta
- \times - produto vetorial
- \cdot - produto escalar
- $^{\wedge}$ - quantidade estimada
- δ - quantidade de erro entre o valor estimado e o valor real

2 SISTEMAS DE NAVEGAÇÃO INERCIAL

Neste capítulo serão apresentados os conceitos fundamentais ligados ao processo de navegação inercial além da descrição das equações de navegação do INS (Sistema de Navegação Inercial). Serão abordados os conceitos relativos aos referenciais de navegação, aos sensores inerciais, os princípios de navegação inercial, os algoritmos de navegação e suas equações. Mais informações a respeito podem ser encontradas nas referências bibliográficas já citadas e nos trabalhos de dissertação de mestrado de (DE SOUZA PINTO, 2007) e (ROCHA, 2006).

2.1 NAVEGAÇÃO INERCIAL

Navegação inercial é o processo pelo qual se adquirem informações sobre a posição, velocidade e atitude de um veículo com relação a um dado referencial, utilizando informações fornecidas por sensores inerciais tais como acelerômetros e giroscópios. Os sistemas de navegação inercial se baseiam nas leis da mecânica de *Newton*. Medindo-se as acelerações e velocidades angulares de um corpo, torna-se possível calcular as mudanças de velocidade, posição e atitude através de sucessivas integrações numéricas.

Um Sistema de Navegação Inercial é formado por 2 unidades básicas:

- Unidade de Medidas Inerciais (IMU): onde são montados os sensores inerciais, sendo três acelerômetros, que fornecem as medidas das componentes da aceleração linear, e três giroscópios, sensores que fornecem as componentes da velocidade angular;
- Unidade de Processamento: onde as medidas dos sensores inerciais são processadas. As velocidades lineares e as posições são obtidas a partir da integração dos sinais dos acelerômetros, enquanto que a integração dos sinais dos giroscópios fornece a atitude do veículo.

Os eixos de sensibilidade dos instrumentos de um sistema de navegação inercial podem ter sua orientação estabilizada no espaço por meio de uma plataforma servo-controlada (plataforma *Gimballed*) ou podem ser fixados à estrutura do veículo que o transporta (plataforma *Strapdown*). Esta última exige um esforço computacional relativamente elevado para a determinação de sua orientação no espaço em tempo real, quando comparado com os sistemas *gimballed* (BRITING, 1971).

A FIG.2.1 mostra um INS do tipo *Strapdown* e a FIG.2.2 um INS do tipo *Gimballed*, enquanto a tabela 3.1 compara esses dois tipos de plataformas.

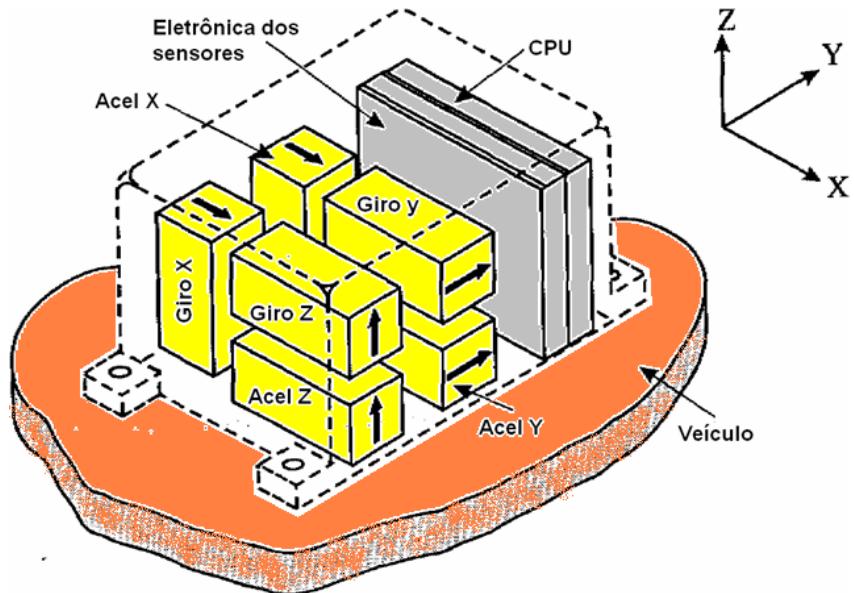


FIG.2.1: Sistema de Navegação Inercial *Strapdown*

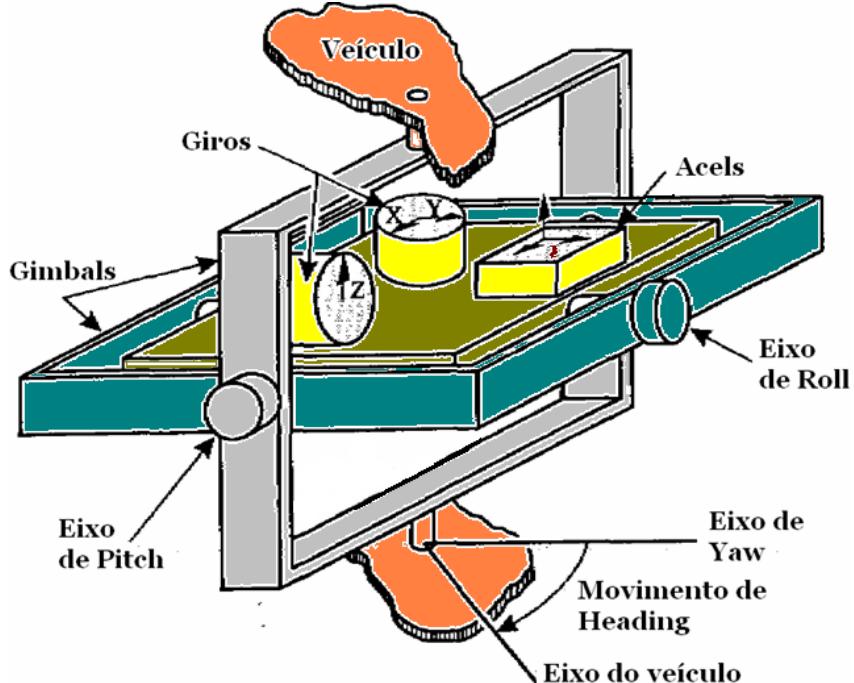


FIG.2.2: Sistema de Navegação Inercial *Gimbaled*

TAB.2.1: Comparativo de Plataformas

| | Plataforma <i>gimbal</i> | Plataforma <i>strapdown</i> |
|--|--------------------------------------|------------------------------------|
| Referência de altitude | Mecânica | Computacional |
| Medida dos acelerômetros | Referencial de Navegação | Referencial do corpo |
| Propagação de erros devido à incertezas nas condições iniciais | Os erros em altitude não se propagam | Os erros em altitude se propagam |
| Dimensões físicas | Comparativamente maior | Reduzido |
| Custo | Comparativamente maior | Reduzido |
| Consumo de energia | Comparativamente maior | Reduzido |
| Calibração a bordo | Possível | Não é possível |

2.2 ACELERÔMETROS E GIROSCÓPIOS

Os acelerômetros são os sensores inerciais que fornecem a medida da força específica que atua no veículo, que é a resultante das ações da aceleração inercial e da aceleração da gravidade. Portanto, a partir da medida da força específica e do modelo do campo gravitacional da Terra determina-se a aceleração linear, informação que é integrada para determinação da velocidade e posição do veículo.

Os giroscópios fornecem a medida da velocidade angular. Este dado é utilizado para a determinação da altitude do veículo. A tecnologia de giroscópios possui três gerações. A primeira geração, denominada giroscópios mecânicos, tem como princípio de funcionamento as propriedades de um disco girante. Apresentam maior exatidão, maior volume e maior custo que as demais classes, sendo utilizada principalmente em aplicações navais. A segunda geração, a dos giroscópios ópticos, tem seu funcionamento baseado nas propriedades da luz. Apresentam menor custo, volume e exatidão que os giros mecânicos, sendo principalmente utilizados em aplicações aeronáuticas. A terceira geração é formada pelos sensores baseados na tecnologia MEMS (*Micro Electro Mechanical Systems*). Tais sensores apresentam os menores custos e volume, e a pior qualidade de medidas dentre todas as tecnologias. A tecnologia MEMS para medida da rotação consiste em placas cerâmicas vibrantes que utilizam a força de *Coriolis* para medir a taxa independente da aceleração. Os acelerômetros são componentes de silício mecânicos miniaturizados que usam a capacidade diferencial para medir aceleração.

A FIG.2.3 mostra uma IMU tipo MEMS com acelerômetros e giroscópios embutidos.

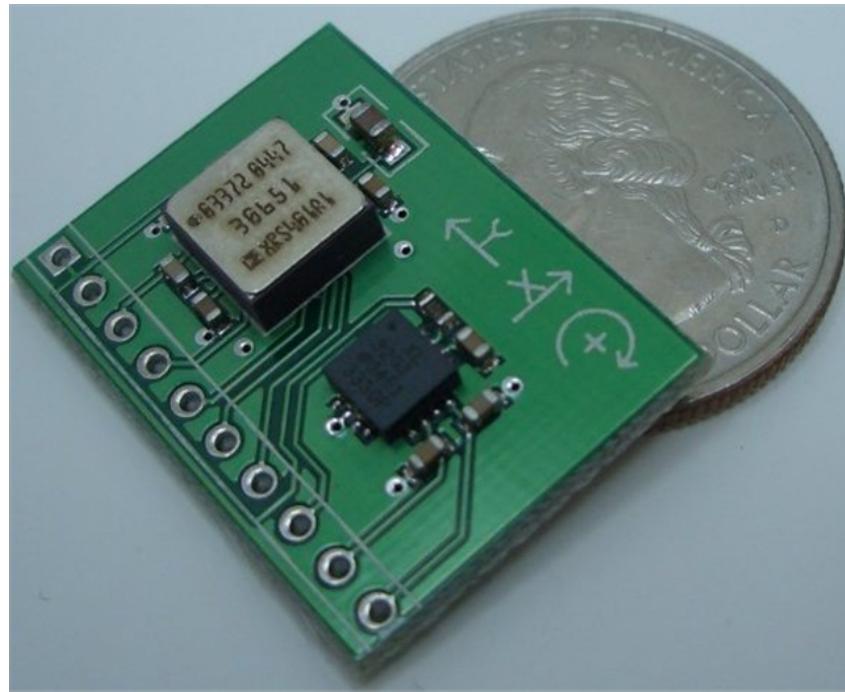


FIG.2.3: IMU tipo MEMS da *Spark Fun Electronics* SEN-00842

2.3 PARÂMETROS DE NAVEGAÇÃO

Os parâmetros de navegação são constituídos pelos parâmetros de **posição**, que definem a posição do corpo, parâmetros de **velocidade**, que especificam a variação da posição do corpo com o tempo, e parâmetros de **atitude**, que especificam a orientação angular do corpo.

A posição de um veículo (corpo) em relação à Terra é descrita através de três parâmetros (FIG.2.5): latitude (φ), longitude (λ) e altitude (H). A altitude é definida como a distância vertical acima da Terra, ao longo da perpendicular à superfície da Terra. A longitude é definida como a distância angular, medida no plano equatorial, entre o plano que contém o eixo polar e o meridiano correspondente à posição do veículo e o que contém o eixo polar e o meridiano que passa por Greenwich, na Inglaterra. Para definir latitude, tem-se que primeiramente definir a posição de referência na superfície como a interseção com a superfície da Terra da perpendicular à superfície da Terra baixada a partir da posição real do veículo. Latitude é o ângulo medido em relação ao plano equatorial a partir da posição de referência na superfície.

Os parâmetros de velocidade (FIG.2.5) são em geral expressos em termos de componentes verticais e horizontais do movimento translacional do veículo em relação à Terra. A componente vertical da velocidade (V_d) é definida como a variação temporal da alti-

tude. A componente horizontal é a projeção, no plano tangente à superfície da Terra na posição de referência do corpo, do vetor velocidade do corpo em relação à Terra. Esta componente, por sua vez, é expressa em duas componentes, nos sentidos norte (V_n) e leste (V_e).

Os parâmetros de atitude que se encontram na FIG.2.4 são:

- Ângulo de rolagem (ϕ - *roll*): é o ângulo medido em torno do eixo longitudinal do veículo.
- Ângulo de arfagem (θ - *pitch*): é definido como o ângulo, medido no plano vertical (plano que contém o eixo longitudinal do corpo e que é perpendicular ao plano tangente à superfície da Terra na posição de referência do corpo), entre o eixo longitudinal do corpo e um plano que, contendo o Centro de Massa (C.M.) do corpo, é paralelo ao plano tangente à superfície da Terra na posição de referência do corpo.
- Ângulo de guinagem (ψ - *yaw*): é o ângulo, medido num plano paralelo ao plano horizontal (plano tangente à superfície da Terra na posição de referência do corpo), contendo o C.M. do corpo, entre uma direção de referência e o eixo longitudinal do corpo. Se a direção de referência é o Norte Geográfico, o ângulo de guinagem recebe a denominação de ângulo de rumo (*head*). A denominação *head* será a usada durante todo o trabalho.

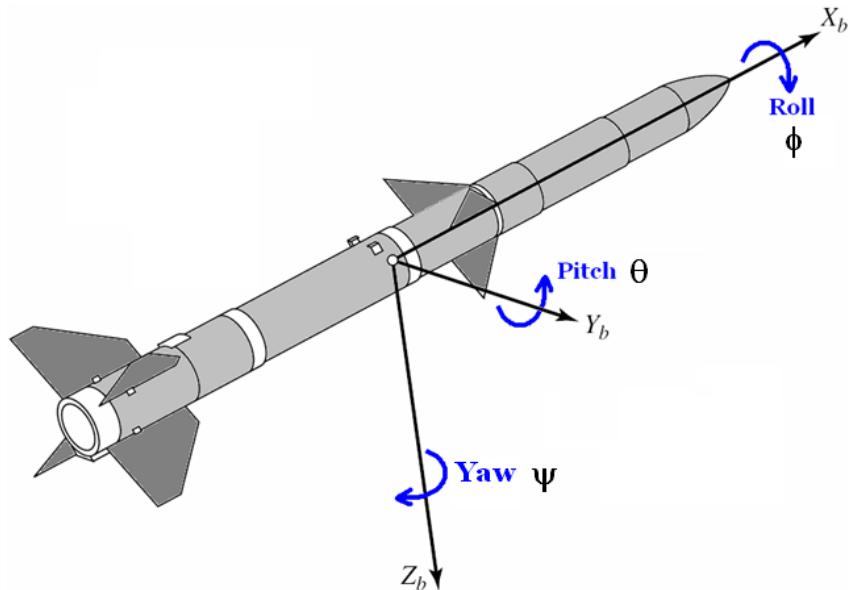


FIG.2.4: Parâmetros de Atitude

2.4 SISTEMAS DE COORDENADAS

No desenvolvimento dos algoritmos de navegação é essencial que sejam bem estabelecidos os sistemas de coordenadas adotados. Eventualmente, equações que são desenvolvidas em um referencial são transportadas para outro. Os principais sistemas de coordenadas são apresentados a seguir e mostrados na FIG.2.5.

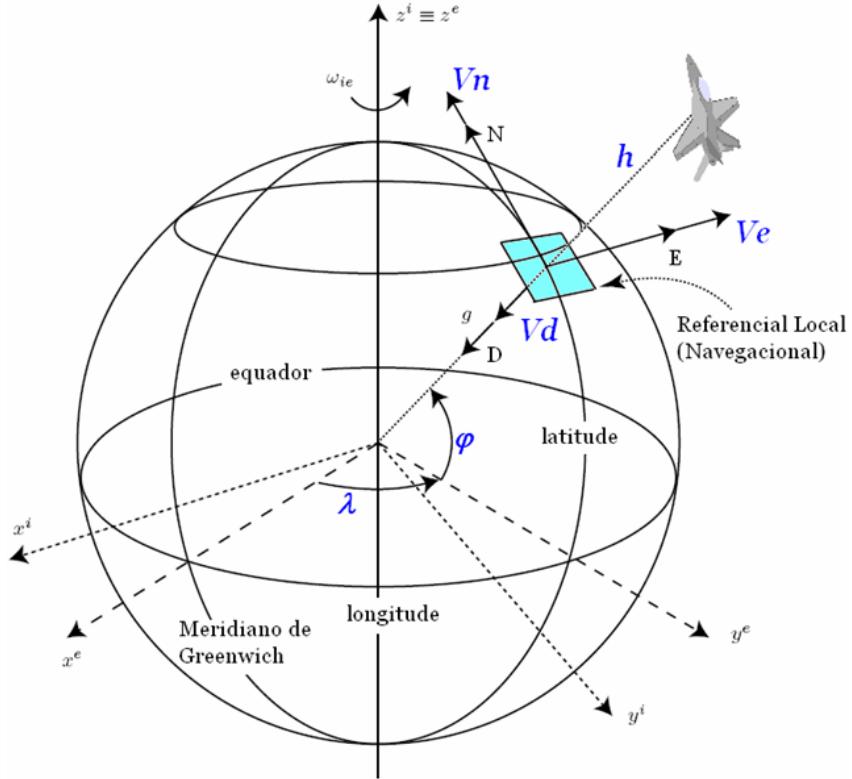


FIG.2.5: Sistema de Coordenadas

Referencial Inercial (I):

- Sistema de coordenadas fixo nas estrelas;
- Origem: centro da Terra;
- Eixo Z_i coincidente com o eixo de rotação da Terra;
- Eixos Y_i e X_i completam o triedro ortogonal.

Referencial Terra / Earth (E):

- O triedro definido pelos eixos X_e , Y_e e Z_e gira junto com a Terra.
- Origem: centro da Terra;

- Eixo Xe: intersecção entre a linha do equador e o meridiano de Greenwich;
- Eixo Ze: direção e sentido do eixo de rotação da Terra;
- Eixo Ye: completa o triedro pela regra da mão direita.

Referencial do Veículo / Corpo / Body (B):

- Neste caso o triedro é fixo e se movimenta junto com o veículo (figura 2.4).
- Origem: centro de gravidade do veículo;
- Eixo Xb (*roll*): sentido longitudinal do veículo;
- Eixo Yb (*pitch*): sentido transversal do veículo;
- Eixo Zb (*head*): completa o triedro pela regra da mão direita, apontando para baixo.

Referencial Navegacional (N) / Geográfico / Local (L):

- É o referencial normalmente adotado para a saída de resultados, pois seus eixos fornecem valores de latitude e longitude de forma direta.
- Origem: centro de gravidade do veículo;
- Eixo Xn (N): tangente ao elipsóide, apontando para o pólo norte geográfico;
- Eixo Yn (E): tangente ao elipsóide, completando o triedro pela regra da mão direita, apontando para a direção leste.
- Eixo Zn (D): vertical para baixo, perpendicular à superfície do elipsóide de referência;

2.5 EQUAÇÕES DE NAVEGAÇÃO

Será desenvolvido agora as equações que descrevem a evolução temporal dos parâmetros de navegação em função dos dados recebidos pelos sensores iniciais (acelerômetros e giroscópios).

2.5.1 TRANSFORMAÇÃO DE COORDENADAS

MTC (Matriz de Transformação de Coordenadas) são aquelas que transformam um dado vetor representado em um dado referencial "a" no mesmo vetor representado pelo referencial "b" como mostra a equação abaixo:

$$V^b = C_a^b \cdot V^a \quad (2.1)$$

Onde:

- C_a^b é a MTC que transforma do referencial "a" para o "b",
- V^b é o vetor representado pelo referencial "b" e,
- V^a é o vetor representado pelo referencial "a".

Uma rotação de um referencial qualquer no espaço pode ser vista como uma sequência de três rotações: Uma rotação de um ângulo ϕ em torno do eixo X, outra de um ângulo θ em torno do eixo Y e outra de um ângulo ψ em torno do eixo Z. Suas respectivas matrizes de rotação podem ser escritas como segue:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (2.2)$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.3)$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Então, define-se a MTC do referencial "n" (navegacional) para o referencial "b" (corpo) sendo:

$$C_n^b = R_x(\phi) \cdot R_y(\theta) \cdot R_z(\psi) \quad (2.5)$$

onde ϕ , θ e ψ são as três componentes dos ângulos cardânicos *roll*, *pitch* e *yaw* respectivamente. Sendo C_n^b uma matriz ortonormal, temos que:

$$C_b^n = (C_n^b)^T \quad (2.6)$$

Combinando 2.2 a 2.6 temos:

$$C_b^n = \left[\begin{array}{c|c|c} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\theta s\phi c\psi \\ \hline c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ \hline -s\theta & s\phi c\theta & c\phi c\theta \end{array} \right] \quad (2.7)$$

Como C_b^n é ortonormal, suas linhas formam vetores ortogonais. Podemos então estabelecer relações entre elas e achar facilmente os ângulos cardânicos como se segue:

$$\phi = \tan^{-1} \left(\frac{c_{32}}{c_{33}} \right) \quad (2.8)$$

$$\theta = -\tan^{-1} \left(\frac{c_{31}}{\sqrt{1 - c_{31}^2}} \right) \quad (2.9)$$

$$\psi = \tan^{-1} \left(\frac{c_{21}}{c_{11}} \right) \quad (2.10)$$

onde, c_{ij} representa o elemento (i,j) da MTC.

2.6 TAXA DE ROTAÇÃO ANGULAR DA TERRA (*EARTH RATE*)

A Terra gira em torno de seu eixo polar (eixo Z) com um taxa de $\omega_e = 7.29211585e - 5 rad/s$. Esta rotação representa a taxa de rotação angular da Terra relativo ao referencial inercial (I) e projetada no referencial da Terra (E). É representada por:

$$\underline{\omega}_{ie}^e = \begin{pmatrix} 0 \\ 0 \\ \omega_e \end{pmatrix} \quad (2.11)$$

É fácil verificar na FIG.2.6 que esta mesma taxa de rotação angular pode ser representada no referencial navegacional (N), nos dando:

$$\underline{\omega}_{ie}^n = \begin{pmatrix} \omega_e \cos \varphi \\ 0 \\ -\omega_e \sin \varphi \end{pmatrix} \quad (2.12)$$

Se, no entanto estivermos no hemisfério sul (referencial navegacional (N) abaixo da linha do equador) temos:

$$\underline{\omega}_{ie}^n = \begin{pmatrix} \omega_e \cos \varphi \\ 0 \\ \omega_e \sin \varphi \end{pmatrix} \quad (2.13)$$

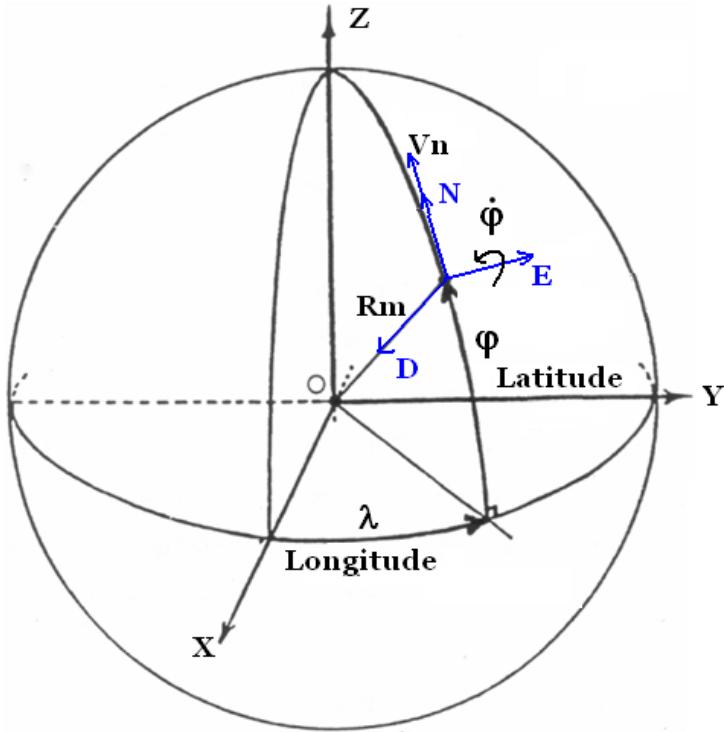


FIG.2.6: Movimento norte-sul no plano meridional

2.7 TAXA DE TRANSPORTE (*TRANSPORT RATE*)

Para se determinar a posição de um veículo na superfície da Terra, é mais fácil e conveniente usar coordenadas angulares (latitude e longitude) do que coordenadas cartesianas (distâncias X, Y e Z). Entretanto, para usarmos tais coordenadas, devemos primeiro converter as velocidades lineares do veículo em velocidades angulares. Considere um veículo que se move na superfície da Terra elipsoidal (altura zero) e cujas componentes das velocidades nas direções Norte (N) e Leste (E) são v_n e v_e respectivamente. Neste caso, observando as figuras FIG.2.6 e FIG.2.7 temos que as velocidades angulares $\dot{\varphi}$ e $\dot{\lambda}$ são:

$$\dot{\varphi} = \frac{v_n}{R_m} \quad (2.14)$$

$$\dot{\lambda} = \frac{v_e}{R_p \cos \varphi} \quad (2.15)$$

onde:

- R_m é o raio de curvatura meridional da Terra (BEKIR, 2007)
- R_p é o raio de curvatura principal do plano perpendicular ao plano meridional (BEKIR, 2007)
- $\dot{\varphi}$ é a velocidade angular sobre o eixo Leste (E) devido à v_n
- $\dot{\lambda}$ é a velocidade angular sobre o eixo polar devido à v_e

A uma altura h , acima da Terra elipsoidal, as equações 2.14, 2.15 devem ser atualizadas para:

$$\dot{\varphi} = \frac{v_n}{R_m + h} \quad (2.16)$$

$$\dot{\lambda} = \frac{v_e}{(R_p + h)\cos\varphi} \quad (2.17)$$

Para expressar estas velocidades angulares no referencial navegacional (N), $\dot{\lambda}$ deve ser projetada nos eixos Norte (N) e vertical (D) (FIG.2.7).

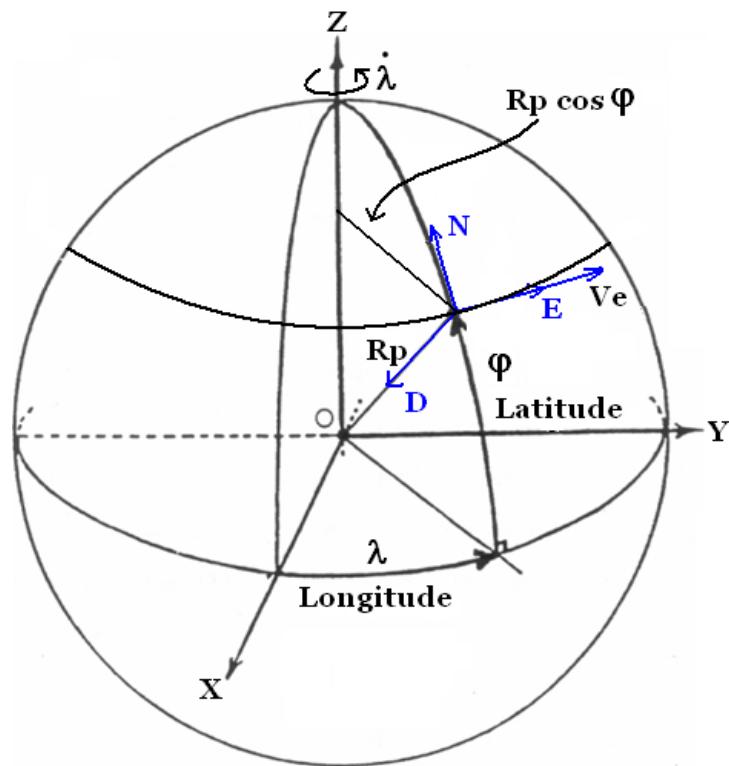


FIG.2.7: Movimento leste-oeste no plano equatorial

O vetor velocidade angular do referencial navegacional (N) em relação ao referencial da Terra (E) projetado no navegacional (N) é então:

$$\underline{\omega}_{en}^n = \begin{pmatrix} \dot{\lambda} \cos \varphi \\ -\dot{\varphi} \\ -\dot{\lambda} \sin \varphi \end{pmatrix} \quad (2.18)$$

substituindo 2.16 e 2.17 em 2.18, temos:

$$\underline{\omega}_{en}^n = \begin{pmatrix} \frac{v_e}{R_p+h} \\ -\frac{v_n}{R_m+h} \\ -\frac{v_e \tan \varphi}{R_p+h} \end{pmatrix} \quad (2.19)$$

2.8 EQUAÇÃO DA DINÂMICA DA POSIÇÃO

A posição do veículo no referencial de navegação (N) é expressa em coordenadas esféricas por

$$\underline{r}^n = \begin{pmatrix} \varphi \\ \lambda \\ h \end{pmatrix} \quad (2.20)$$

e sua velocidade no referencial de navegação (N) é definida por

$$\underline{v}^n = \begin{pmatrix} v_n \\ v_E \\ v_D \end{pmatrix} = \begin{pmatrix} R_m + h & 0 & 0 \\ 0 & (R_p + h) \cos \varphi & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} \quad (2.21)$$

Conseqüentemente, a derivada temporal da posição pode ser obtida por

$$\dot{\underline{r}}^n = \begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} \frac{1}{R_m+h} & 0 & 0 \\ 0 & \frac{1}{(R_p+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} v_n \\ v_E \\ v_D \end{pmatrix} \quad (2.22)$$

2.9 EQUAÇÃO DA DINÂMICA DA ATITUDE

A dinâmica da atitude são definidas por (SCHWARZ & WEI, 2000)

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b = C_b^n (\Omega_{ib}^b - \Omega_{in}^b) \quad (2.23)$$

ou em (ROGERS, 2000)

$$\dot{C}_b^n = -\Omega_{bn}^n C_b^n \quad (2.24)$$

onde Ω representa a matriz na forma anti-simétrica (*skew symmetric*) do vetor $\underline{\omega}$. Então, Ω_{ib}^b é a forma anti-simétrica de $\underline{\omega}_{ib}^b$ que é a saída dos giroscópios *Strapdown*. Ω_{bn}^n é obtido por:

$$\underline{\omega}_{bn}^n = \underline{\omega}_{in}^n - C_b^n \underline{\omega}_{ib}^b \quad (2.25)$$

sendo $\underline{\omega}_{in}^n$ obtido por:

$$\underline{\omega}_{in}^n = \underline{\omega}_{ie}^n + \underline{\omega}_{en}^n \quad (2.26)$$

com $\underline{\omega}_{ie}^n$ e $\underline{\omega}_{en}^n$ definidas nas equações 2.13 e 2.19 respectivamente

2.10 EQUAÇÃO DA DINÂMICA DA VELOCIDADE

Para se obter as equações de dinâmica das velocidades, parte-se de:

$$\underline{v}^n = C_e^n \underline{r}^e \quad (2.27)$$

Substituindo $\underline{r}^e = C_i^e(\underline{r}^i - \Omega_{ie}^i \underline{r}^i)$ na equação acima chega-se a:

$$\underline{v}^n = C_i^n(\underline{r}^i - \Omega_{ie}^i \underline{r}^i) \quad (2.28)$$

Conseqüentemente, a dinâmica da velocidade pode ser obtida como se segue:

$$\begin{aligned} \dot{\underline{v}}_n &= C_i^n \Omega_{ni}^i (\underline{r}^i - \Omega_{ie}^i \underline{r}^i) + C_i^n (\underline{r}^i - \Omega_{ie}^i \underline{r}^i) \\ &= C_i^n \ddot{\underline{r}}^i + C_i^n (\Omega_{ni}^i - \Omega_{ie}^i) \dot{\underline{r}}^i - C_i^n \Omega_{ni}^i \Omega_{ie}^i \underline{r}^i \end{aligned} \quad (2.29)$$

onde Ω_{ie}^i representa a taxa de rotação da Terra e é considerada constante. Substituindo $\dot{\underline{r}}^i = C_e^i(\dot{\underline{r}}^e + \Omega_{ie}^e \underline{r}^e)$ e $I = C_n^i C_i^n$ na equação anterior obtém-se:

$$\begin{aligned} \dot{\underline{v}}_n &= C_i^n \ddot{\underline{r}}^i + C_i^n (\Omega_{ni}^i - \Omega_{ie}^i) C_n^i C_i^n C_e^i (\dot{\underline{r}}^e + \Omega_{ie}^e \underline{r}^e) - C_i^n \Omega_{ni}^i \Omega_{ie}^i \underline{r}^i = \\ &= C_i^n \ddot{\underline{r}}^i + (\Omega_{ni}^n - \Omega_{ie}^n) C_e^n \dot{\underline{r}}^e + (\Omega_{ni}^n - \Omega_{ie}^n) C_e^n \Omega_{ie}^e \underline{r}^e - C_i^n \Omega_{ni}^i \Omega_{ie}^i \underline{r}^i = \\ &= C_i^n \ddot{\underline{r}}^i + (\Omega_{ni}^n - \Omega_{ie}^n) C_e^n \dot{\underline{r}}^e + \Omega_{ni}^n C_e^n \Omega_{ie}^e \underline{r}^e - \Omega_{ie}^n C_e^n \Omega_{ie}^e \underline{r}^e - C_i^n \Omega_{ni}^i \Omega_{ie}^i \underline{r}^i \end{aligned} \quad (2.30)$$

Usando a equação:

$$\Omega_{ni}^n C_e^n \Omega_{ie}^e \underline{r}^e = C_i^n C_n^i \Omega_{ni}^n C_i^n C_e^n \Omega_{ie}^e C_e^i C_e^n \underline{r}^e = C_i^n \Omega_{ni}^i \Omega_{ie}^i \underline{r}^i$$

e a Equação 2.27, a equação da dinâmica de velocidade pode ser reduzida a:

$$\begin{aligned}\dot{\underline{v}}^n &= C_i^n \dot{\underline{r}}^i + (\Omega_{ni}^n - \Omega_{ie}^n) \underline{v}^n - \Omega_{ie}^n C_e^n \Omega_{ie}^e \underline{r}^e \\ &= C_i^n \dot{\underline{r}}^i - (2\Omega_{ie}^n + \Omega_{en}^n) \underline{v}^n - C_e^n \Omega_{ie}^e \Omega_{ie}^e \underline{r}^e\end{aligned}\quad (2.31)$$

Substituindo $\ddot{\underline{r}}^i = \underline{f}^i + \underline{\bar{g}}^i$ na equação anterior, chega-se a:

$$\begin{aligned}\dot{\underline{v}}^n &= C_i^n (\underline{f}^i + \underline{\bar{g}}^i) - (2\Omega_{ie}^n + \Omega_{en}^n) \underline{v}^n - C_e^n \Omega_{ie}^e \Omega_{ie}^e \underline{r}^e = \\ &= C_i^n \underline{f}^i - (2\Omega_{ie}^n + \Omega_{en}^n) \underline{v}^n + C_e^n (\underline{\bar{g}}^e - \Omega_{ie}^e \Omega_{ie}^e \underline{r}^e) = \\ &= C_b^n \underline{f}^b - (2\Omega_{ie}^n + \Omega_{en}^n) \underline{v}^n + \underline{g}^n\end{aligned}\quad (2.32)$$

onde \underline{f} é o vetor força específica definido como a diferença entre a aceleração verdadeira e a aceleração devido a gravidade, $\underline{\bar{g}}$ é o vetor aceleração gravitacional e \underline{g} é o vetor gravidade aparente (TITTERTON & WESTON, 1997).

A equação anterior pode ser escrita como:

$$\dot{\underline{v}}^n = C_b^n \underline{f}^b - (2\underline{\omega}_{ie}^n + \underline{\omega}_{en}^n) \times \underline{v}^n + \underline{g}^n \quad (2.33)$$

Deduções similares podem ser encontradas em (BEKIR, 2007).

É importante considerar o significado físico dos diversos termos da equação 2.33. Pode-se observar que a taxa de variação da velocidade é composta dos seguintes termos:

- Da força específica atuando no veículo, como medida da tríade de acelerômetros montado na mesma (\underline{f}^b);
- Do efeito de *Coriolis* (a correção para a aceleração causada pela velocidade do veículo sobre a superfície de rotação da Terra - w_{ie}^n);
- Da aceleração centrípeta do veículo resultando no seu movimento no sistema de referência Terra (w_{en}^n);
- Da compensação da força gravitacional aparente atuando no veículo. Esta inclui a força gravitacional causada pela atração entre massas (\underline{g}^n) e a aceleração centrípeta do veículo resultante da rotação da Terra.

2.11 EQUAÇÕES DA NAVEGAÇÃO INERCIAL

Em síntese, utilizando as Equações 2.22, 2.24 e 2.33, as equações de navegação inercial podem ser escritas como:

$$\begin{pmatrix} \dot{\underline{r}}^n \\ \dot{\underline{v}}^n \\ \dot{\underline{C}}_b^n \end{pmatrix} = \begin{pmatrix} D^{-1}\underline{v}^n \\ C_b^n \underline{f}^b - (2\underline{\omega}_{ie}^n + \underline{\omega}_{en}^n) \times \underline{v}^n + \underline{g}^n \\ -\Omega_{bn}^n C_b^n \end{pmatrix} \quad (2.34)$$

onde

$$D^{-1} = \begin{pmatrix} \frac{1}{R_m+h} & 0 & 0 \\ 0 & (\frac{1}{R_p+h} \cos \varphi) & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

2.12 DISCRETIZAÇÃO DAS EQUAÇÕES DE NAVEGAÇÃO

Para aproximar o sistema de equações diferenciais 2.34 contínuas no tempo em um sistema de equações diferença discretas, podemos usar o método Trapezoidal, por ter uma boa aproximação nas integrações para o caso da navegação inercial.

O método trapezoidal é baseado na seguinte aproximação (FRANKLIN et al., 1998):

Considere a equação diferencial contínua:

$$\dot{u}(t) = e(t) \quad (2.35)$$

Então, uma aproximação para esta equação seria:

$$u_k = u_{k-1} + \frac{T}{2}(e_k + e_{k-1}) \quad (2.36)$$

onde:

$T = t_k - t_{k-1}$ é o tempo de amostragem em segundos;

$t_k = kT$ para um tempo de amostragem constante;

k é um inteiro;

u_k e u_{k-1} são os valores de $u(t)$ em t_k e t_{k-1} respectivamente e

e_k e e_{k-1} são os valores de $e(t)$ em t_k e t_{k-1} respectivamente.

Portanto, colocando as equações 2.34 na forma da equação 2.35, temos:

$$\begin{pmatrix} \dot{\underline{r}}^n \\ \dot{\underline{v}}^n \\ \dot{\underline{C}}_b^n \end{pmatrix} = \begin{pmatrix} e^r(t) \\ e^v(t) \\ e^{Cbn}(t) \end{pmatrix} \quad (2.37)$$

onde:

$$e^r(t) = D^{-1}\underline{v}^n;$$

$$e^v(t) = C_b^n \underline{f}^b - (2\underline{\omega}_{ie}^n + \underline{\omega}_{en}^n) \times \underline{v}^n + \underline{g}^n \text{ e}$$

$$e^{Cbn}(t) = -\Omega_{bn}^n C_b^n$$

logo, a forma discreta de 2.34 é:

$$\begin{pmatrix} \underline{r}_k^n \\ \underline{v}_k^n \\ C_{b,k}^n \end{pmatrix} = \begin{pmatrix} \underline{r}_{k-1}^n + \frac{T}{2}(e_k^r + e_{k-1}^r) \\ \underline{v}_{k-1}^n + \frac{T}{2}(e_k^v + e_{k-1}^v) \\ C_{b,k-1}^n + \frac{T}{2}(e_k^{Cbn} + e_{k-1}^{Cbn}) \end{pmatrix} \quad (2.38)$$

3 INTEGRAÇÃO INS/GPS COM O USO DO FILTRO DE KALMAN

Neste capítulo será apresentado o método de integração entre INS e GPS como solucionador das principais desvantagens em se utilizar sozinho o INS ou o GPS. Serão desenvolvidas também as equações dinâmicas dos erros do Sistema de Navegação Inercial baseado na análise das pequenas perturbações. Estas equações serão então implementadas no modelo do Filtro de *Kalman* que fará a fusão INS/GPS compensando assim os erros isolados do INS puro e do GPS.

3.1 INTEGRAÇÃO INS/GPS

O navegador integrado INS/GPS apresenta um desempenho superior quando comparado com os sistemas GPS e INS operando isoladamente (INS puro). Isto se deve ao fato dos sistemas apresentarem características complementares. Na tabela 3.1 são apresentadas as principais características dos sistemas GPS e INS (TITTERTON & WESTON, 1997).

TAB.3.1: Características isoladas dos sistemas INS e GPS

| | Vantagens | Desvantagens |
|------------|---|--|
| INS | - Informação de posição, velocidade e altitude - Alta taxa de transmissão - Autonomia - Imune à interferências | - Erro crescente com o tempo - Custo elevado dos sensores de alta exatidão |
| GPS | - Erro limitado - Exatidão dos dados de posição e velocidade | - Baixa taxa de transmissão - Sujeito à interferências externas - Ausência de informação de altitude |

A principal desvantagem do INS que faz com que ele necessite de algum auxílio externo é o erro acumulativo que as sucessivas integrações das equações de navegação fornece. Tendo em vista que sensores, por melhores que sejam, nunca estão desprovidos de erros em suas leituras, um Sistema de Navegação Inercial operando sozinho irá fatalmente fornecer informações errôneas com o passar do tempo. O tempo em que um INS sem um auxílio externo consegue fornecer informações confiáveis é diretamente proporcional à qualidade de seus sensores inercias.

Num sistema integrado INS/GPS, os dados fornecidos pelo GPS são utilizados como atualizações do INS, mantendo os erros deste sistema limitados. A integração destes

sistemas é normalmente implementada através do Filtro de *Kalman* (FK). Baseado em modelos que descrevem a propagação dos erros, o FK obtém estimativas dos erros em posição, velocidade e atitude. A figura 3.1 mostra o diagrama da integração INS/GPS utilizada:

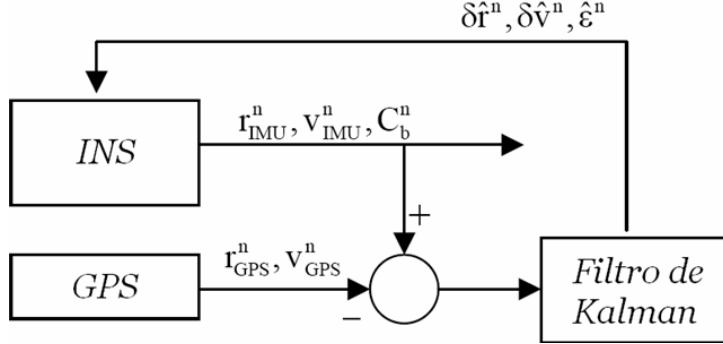


FIG.3.1: Implementação do navegador integrado na configuração de realimentação

3.2 DESENVOLVIMENTO DAS EQUAÇÕES DA DINÂMICA DOS ERROS E O FILTRO DE KALMAN

Para se chegar a um modelo em espaço de estados do Filtro de *Kalman*, será utilizada a análise dos erros, ou seja, o vetor de estados estimado pelo filtro será um vetor de erros estimados. Com esta modelagem, temos a vantagem de trabalhar com equações lineares, o que não aconteceria se quiséssemos modelar o Filtro de *Kalman* estimando os próprios parâmetros de navegação, pois as equações da dinâmica destes parâmetros (equação 2.34) são não lineares.

A análise da dinâmica dos erros utiliza o método das perturbações (TITTERTON & WESTON, 1997). Neste método, a posição, a velocidade e a atitude (MTC) estimadas pelo INS são descritas pelas equações:

$$\hat{r}^n = \underline{r}^n + \delta \underline{r}^n \quad (3.1)$$

$$\hat{v}^n = \underline{v}^n + \delta \underline{v}^n \quad (3.2)$$

$$\hat{C}_b^n = (I - E^n) C_b^n \quad (3.3)$$

$$\underline{\gamma}^n = \underline{g}^n + \delta \underline{g}^n \quad (3.4)$$

Onde,

\hat{r}^n , \hat{v}^n e \hat{C}_b^n são os valores calculados pelo INS puro (valores com erro);

r^n , v^n e C_b^n são os valores definidos no capítulo 2 (valores ideais);

δr^n e δv^n são os erros (perturbações) em posição e velocidade respectivamente;

γ é o vetor gravidade normal e E^n é a matriz anti-simétrica dos erros em atitude

$$\underline{\epsilon}^n = [\epsilon_N \ \epsilon_E \ \epsilon_D]^T.$$

$$E^n = (\epsilon^n \times) = \begin{pmatrix} 0 & -\epsilon_D & \epsilon_E \\ \epsilon_D & 0 & -\epsilon_N \\ -\epsilon_E & \epsilon_N & 0 \end{pmatrix} \quad (3.5)$$

O desenvolvimento da equação 3.3 encontra-se em (BRITING, 1971).

3.2.1 DINÂMICA DOS ERROS EM POSIÇÃO

As equações linearizadas dos erros em posição podem ser obtidas perturbando-se a equação 2.22. Como estas equações são funções da posição e da velocidade, as equações dos erros são obtidas usando-se as derivadas parciais:

$$\delta \dot{r}^n = F_{rr} \delta r^n + F_{rv} \delta v^n \quad (3.6)$$

onde

$$F_{rr} = \begin{pmatrix} \frac{\partial \dot{\varphi}}{\partial \varphi} & \frac{\partial \dot{\varphi}}{\partial \lambda} & \frac{\partial \dot{\varphi}}{\partial h} \\ \frac{\partial \dot{\lambda}}{\partial \varphi} & \frac{\partial \dot{\lambda}}{\partial \lambda} & \frac{\partial \dot{\lambda}}{\partial h} \\ \frac{\partial \dot{h}}{\partial \varphi} & \frac{\partial \dot{h}}{\partial \lambda} & \frac{\partial \dot{h}}{\partial h} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{-v_N}{(R_m+h)^2} \\ \frac{v_E \operatorname{sen} \varphi}{(R_p+h) \cos^2 \varphi} & 0 & \frac{-v_E}{(R_p+h)^2 \cos \varphi} \\ 0 & 0 & 0 \end{pmatrix}; \quad (3.7)$$

$$F_{rv} = \begin{pmatrix} \frac{\partial \dot{\varphi}}{\partial v_N} & \frac{\partial \dot{\varphi}}{\partial v_E} & \frac{\partial \dot{\varphi}}{\partial v_D} \\ \frac{\partial \dot{\lambda}}{\partial v_N} & \frac{\partial \dot{\lambda}}{\partial v_E} & \frac{\partial \dot{\lambda}}{\partial v_D} \\ \frac{\partial \dot{h}}{\partial v_N} & \frac{\partial \dot{h}}{\partial v_E} & \frac{\partial \dot{h}}{\partial v_D} \end{pmatrix} = \begin{pmatrix} \frac{1}{R_m+h} & 0 & 0 \\ 0 & \frac{1}{(R_p+h) \cos \varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (3.8)$$

3.2.2 DINÂMICA DOS ERROS EM VELOCIDADE

Com relação à equação 2.33, podemos modificá-la para uma versão cujos termos são os calculados pelo INS:

$$\hat{v}^n = \hat{C}_b^n \hat{f}^b - (2\hat{\omega}_{ie}^n + \hat{\omega}_{en}^n) \times \hat{v}^n + \gamma^n \quad (3.9)$$

Procedendo de forma análoga aos erros em posição podemos chegar à seguinte equação do erro em velocidade (SHIN, 2001):

$$\delta \dot{\underline{v}}^n = F_{vr} \delta \underline{r}^n + F_{vv} \delta \underline{v}^n + (\underline{f}^n \times) \underline{\epsilon}^n + C_b^n \delta \underline{f}^b \quad (3.10)$$

onde:

$\delta \underline{f}^b$ é o vetor de erros na medida da aceleração;

$$(\underline{f}^n \times) = \begin{pmatrix} 0 & -a_d & a_e \\ a_d & 0 & -a_n \\ -a_e & a_n & 0 \end{pmatrix}; \quad (3.11)$$

$$F_{vr} = \left(\begin{array}{c|c} -2v_E \omega_e \cos \varphi - \frac{v_E^2}{(R_p+h)\cos^2 \varphi} & 0 \\ \hline 2\omega_e(v_N \cos \varphi - v_D \sin \varphi) + \frac{v_E v_N}{(R_p+h)\cos^2 \varphi} & 0 \\ \hline 2v_E \omega_e \sin \varphi & 0 \end{array} \middle| \begin{array}{c} \frac{-v_N v_D}{(R_m+h)^2} + \frac{v_E^2 \tan \varphi}{(R_p+h)^2} \\ \hline \frac{-v_E v_D}{(R_p+h)^2} - \frac{v_N v_E \tan \varphi}{(R_p+h)^2} \\ \hline \frac{v_E^2}{(R_p+h)^2} + \frac{v_N^2}{(R_m+h)^2} - \frac{2\gamma}{(R+h)} \end{array} \right); \quad (3.12)$$

$$F_{vv} = \left(\begin{array}{c|c|c} \frac{v_D}{R_m+h} & -2\omega_e \sin \varphi - 2\frac{v_E \tan \varphi}{R_p+h} & \frac{v_N}{R_m+h} \\ \hline 2\omega_e \sin \varphi + \frac{v_E \tan \varphi}{R_p+h} & \frac{v_D + v_N \tan \varphi}{R_p+h} & 2\omega_e \cos \varphi + \frac{v_E}{R_p+h} \\ \hline -2\frac{v_N}{R_m+h} & -2\omega_e \cos \varphi - 2\frac{v_E}{R_p+h} & 0 \end{array} \right); \quad (3.13)$$

com

$$\gamma = \gamma_0 \left(\frac{R}{R+h} \right)^2 \quad (3.14)$$

e

$$R = \sqrt{R_p R_m} \quad (3.15)$$

3.2.3 DINÂMICA DOS ERROS EM ATITUDE

De forma semelhante, a partir da aplicação das perturbações na equação 2.24 é obtida a equação de propagação de erro em altitude:

$$\dot{\underline{\epsilon}}^n = F_{er} \delta \underline{r}^n + F_{ev} \delta \underline{v}^n - (\underline{\omega}_{in}^n \times) \underline{\epsilon}^n - C_b^n \delta \underline{\omega}_{ib}^b \quad (3.16)$$

onde:

$\delta \underline{\omega}_{ib}^b$ é o vetor de erros na medida da velocidade angular;

$$F_{er} = \begin{pmatrix} -\omega_e \sin \varphi & 0 & \frac{-v_E}{(R_p+h)^2} \\ 0 & 0 & \frac{v_N}{(R_m+h)^2} \\ -\omega_e \cos \varphi - \frac{v_E}{(R_p+h)\cos^2 \varphi} & 0 & \frac{v_E \tan \varphi}{(R_p+h)^2} \end{pmatrix}; \quad (3.17)$$

$$F_{ev} = \begin{pmatrix} 0 & \frac{1}{R_p+h} & 0 \\ \frac{-1}{R_m+h} & 0 & 0 \\ 0 & \frac{-\tan\varphi}{R_p+h} & 0 \end{pmatrix} \quad (3.18)$$

3.2.4 EQUACIONAMENTO DO FILTRO DE KALMAN

Combinando as equações 3.6, 3.9 e 3.16, podemos construir o seguinte sistema de equações contínuas:

$$\dot{\underline{x}} = F\underline{x} + G\underline{u}, \quad (3.19)$$

onde $\underline{x} = [\delta\underline{r}^n \ \delta\underline{v}^n \ \underline{\epsilon}^n]^T$ é o vetor de estados estimados, $\underline{u} = [\delta\underline{f}^b \ \delta\underline{\omega}_{ib}^b]^T$ é o vetor de ruídos,

$$F = \begin{pmatrix} F_{rr} & F_{rv} & 0 \\ F_{vr} & F_{vv} & (\underline{f}^n \times) \\ F_{er} & F_{ev} & -(\underline{\omega}_{in}^n \times) \end{pmatrix} \quad (3.20)$$

e

$$G = \begin{pmatrix} 0 & 0 \\ C_b^n & 0 \\ 0 & -C_b^n \end{pmatrix} \quad (3.21)$$

As componentes de \underline{u} são modeladas como ruído branco cuja matriz de covariância é dada por:

$$Q = \begin{pmatrix} \sigma_{ax}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{ay}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{az}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{wx}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{wy}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{wz}^2 \end{pmatrix} \quad (3.22)$$

onde σ_a^2 e σ_w^2 são as variâncias dos ruídos dos acelerômetros e dos giroscópios respectivamente.

A versão discreta de 3.19 é dada por:

$$\underline{x}_{k+1} = \Phi_k \underline{x}_k + G_k \underline{u}_k, \quad (3.23)$$

O método analítico para encontrar a matriz de transição de estados Φ é dado por:

$$\Phi_k = \mathcal{L}^{-1}[(sI - F)^{-1}] \quad (3.24)$$

onde \mathcal{L} representa a transformada de *Laplace* inversa e s é o parâmetro da transformada de *Laplace*. No entanto, como na implementação do INS usamos um tempo de amostragem Δt bem pequeno, podemos utilizar a seguinte aproximação:

$$\Phi_k = e^{(F\Delta t)} \approx I + F\Delta t \quad (3.25)$$

Adotando esta mesma consideração sobre o tempo de amostragem, podemos também achar uma aproximação para Q_k (SHIN, 2001):

$$Q_k = E[\underline{w}_k \underline{w}_k^T] \approx \Phi_k G Q G^T \Phi^T \Delta t \quad (3.26)$$

De posse do modelo estocástico do processo (equação 3.19), precisamos agora da equação de observação para completarmos a modelagem do Filtro de *Kalman*:

$$\underline{z}_k = H_k \underline{x}_k + \underline{\epsilon}_k \quad (3.27)$$

Onde o vetor de observações \underline{z}_k é definido como a diferença entre os parâmetros de posição e velocidade determinados pelos sistemas INS e GPS. Considerando um GPS que fornece dados de posição (latitude, longitude e altitude) e velocidades (v_N , v_E e v_D) temos:

$$\underline{z}_k = \begin{pmatrix} \varphi_{INS} - \varphi_{GPS} \\ \lambda_{INS} - \lambda_{GPS} \\ h_{INS} - h_{GPS} \\ v_{NINS} - v_{NGPS} \\ v_{EINS} - v_{EGPS} \\ v_{DINS} - v_{DGPS} \end{pmatrix} \quad (3.28)$$

e, por conseguinte:

$$H_k = \left(\begin{array}{c|c|c} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ \hline 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \end{array} \right) \quad (3.29)$$

O vetor $\underline{\epsilon}_k$ na equação 3.27 representa os erros das medidas do GPS. Adotando-se o ruído branco como modelo para as suas componentes temos:

$$R_k = E[\underline{e}_k \underline{e}_k^T] = \begin{pmatrix} \sigma_\varphi^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\lambda^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_h^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_N}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_E}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v_D}^2 \end{pmatrix} \quad (3.30)$$

Sendo σ_φ^2 , σ_λ^2 , σ_h^2 , $\sigma_{v_N}^2$, $\sigma_{v_E}^2$ e $\sigma_{v_D}^2$ as variâncias dos dados de latitude, longitude, altitude, velocidade norte, velocidade leste e velocidade vertical respectivamente.

A implementação do Filtro de *Kalman* pode ser dividida em dois estágios, atualização e predição. Na fase de atualização é calculado primeiramente o ganho de *Kalman* K_k , para logo após se atualizar o estado x_k e a matriz de covariância dos erros P_k baseados nas estimativas iniciais \underline{x}_k^- e P_k^- .

ATUALIZAÇÃO:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (3.31)$$

$$\hat{\underline{x}}_k = \underline{x}_k^- + K_k (\underline{z}_k - H_k \underline{x}_k^-) \quad (3.32)$$

$$P_k = (I - K_k H_k) P_k^- \quad (3.33)$$

Na fase de predição os estados e a matriz de covariância dos erros são estimados uma amostra de tempo a frente.

PREDIÇÃO:

$$\hat{\underline{x}}_{k+1} = \Phi_k \hat{\underline{x}}_k \quad (3.34)$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \quad (3.35)$$

De posse do vetor de estados estimados $\underline{x}_k = [\delta\underline{r}^n \ \delta\underline{v}^n \ \underline{\epsilon}^n]^T$, devemos então usá-lo para compensar os parâmetros de navegação no INS. Das equações 3.1 e 3.2 temos que :

$$\underline{r}^n = \hat{\underline{r}}^n - \delta\underline{r}^n \quad (3.36)$$

$$\underline{v}^n = \hat{\underline{v}}^n - \delta \underline{v}^n \quad (3.37)$$

Da equação 3.3 e sabendo que para erros em atitude muito pequenos a seguinte aproximação é válida:

$$(I - E^n)^{-1} = (I + E^n) \quad (3.38)$$

temos que:

$$C_b^n = (I + E^n) \hat{C}_b^n \quad (3.39)$$

sendo \underline{r}^n , \underline{v}^n e C_b^n os dados de navegação compensados (corrigidos).

4 ESTIMAÇÃO DE ERROS DOS SENSORES E ALINHAMENTO

Neste capítulo serão descritos os diversos tipos de erros dos sensores iniciais. Também serão mostradas as técnicas de alinhamento de IMUs tanto para sensores de alta exatidão quanto para sensores de baixo custo do tipo "MEMS". Para tanto será apresentado o algorítimo de inicialização e de alinhamento estático.

4.1 TIPOS DE ERROS DOS SENSORES INERCIAIS

Os erros contidos em acelerômetros e giroscópios de um Sistema de Navegação Inercial *Strapdown* são constituídos de duas partes: uma determinística e uma outra estocástica (randômica). A parte determinística pode ser descrita pelo *bias_{offset}*, pelo *fator de escala* e pelos *desalinhamentos*. A parte estocástica é representada pelo ruído e pelo *bias_{drift}*, onde este pode ser subdividido em mais duas partes: *bias_{repetibilidade}* e *bias_{instabilidade}*.

Pode-se observar portanto, que o bias é a principal componente de erro de um sensor inercial. O bias é a componente do sinal de saída que não está relacionada com a entrada à qual o sensor está submetido. Como equacionamento, podemos representar o bias da seguinte forma:

$$bias = bias_{offset} + bias_{repetibilidade} + bias_{instabilidade} \quad (4.1)$$

onde:

bias_{offset} é uma componente determinística e não variante com o tempo;

bias_{repetibilidade} é caracterizado como um valor constante no tempo, que é alterado sempre que o sensor é colocado em operação, ou seja, sempre que ele é ligado;

bias_{instabilidade} é descrito como uma variação aleatória do bias em baixa freqüência, sendo normalmente modelado como um processo *Markov* de primeira ordem.

O *fator de escala* é definido como a taxa de variação da saída em relação à entrada, é um parâmetro determinístico devendo ser estimado e compensado.

O *ruído* é um sinal de alta freqüência que interfere na medida do sensor, normalmente é eliminado por processos convencionais de filtragem.

Vimos no Capítulo 2 que os sensores iniciais são dispostos ortogonalmente entre si. No entanto, esta configuração ortogonal nunca é alcançada fisicamente, havendo sempre

um desalinhamento entre os sensores. Isso faz com que a medida de um dos sensores seja afetada pela entrada nos outros. Também é um parâmetro determinístico devendo ser estimado e compensado.

4.2 CALIBRAÇÃO

Chamaremos de calibração a estimação dos parâmetros determinísticos dos sensores. No entanto, uma melhor definição de calibração está no processo de comparação das saídas de instrumentos de medidas com valores de referência conhecidos e na determinação de coeficientes que forçam esta saída a coincidir com estes valores de referência. Para se determinar com exatidão todos os parâmetros de calibração, são necessários equipamentos especiais, como simuladores de movimento de alta precisão. Também é necessário conhecer a gravidade local e a posição (latitude, longitude e altitude) no ambiente em que se encontra o simulador de movimentos.

Existem normas que descrevem os procedimentos de estimação dos parâmetros de calibração de uma IMU (IEEE, 1977), (IEEE, 1998) e (IEEE, 2005). Em geral, é obtido um número de medidas dos sensores iniciais igual ao número de parâmetros do modelo, formando um sistema de equações, cuja solução fornece as estimativas dos parâmetros do modelo. O resultado da calibração será a determinação do $bias_{offset}$ de uma determinada IMU. Esta componente do $bias$ será usada no algoritmo de navegação para corrigir os sinais dos sensores

4.3 ALINHAMENTO

Para o algoritmo de navegação inercial fornecer os parâmetros de navegação, ele precisa das condições iniciais destes mesmos parâmetros. As condições iniciais dos parâmetros de posição (latitude, longitude e altitude) e de velocidade (v_N , v_E e v_D) podem ser fornecidas pelo GPS quer o veículo esteja em movimento ou parado. Já os parâmetros de atitude (*roll*, *pitch* e *head*) devem ser estimados pelo algoritmo de inicialização, que utiliza os próprios sinais dos sensores e as referências da gravidade local e da velocidade angular da terra.

O processo de alinhamento então está no algoritmo de inicialização que é responsável por fornecer os parâmetros iniciais de atitude. Para isso, o algoritmo deve primeiramente determinar o $bias_{repetibilidade}$ dos sensores, que somado ao $bias_{offset}$, deve corrigir os sinais destes. O procedimento para se determinar o $bias_{repetibilidade}$ pode variar de acordo com a

tecnologia empregada na IMU. Devido ao tamanho reduzido e ao processo de fabricação, os sensores MEMS apresentam uma grande sensibilidade a variações ambientais, causando um alto nível de instabilidade, uma baixa repetibilidade do bias, e um alto nível de ruído no sinal de saída. Portanto para a estimativa de *bias* de uma IMU MEMS, técnicas mais elaboradas de estimativa destes parâmetros são necessárias.

Existem basicamente dois processos de alinhamento, o alinhamento grosso e o alinhamento fino. Para IMUs cujos bias e ruídos têm valores menores que a taxa de rotação da terra, como IMUs de alta precisão ($bias \leq 0.01\text{graus}/h$) o alinhamento grosso seguido do alinhamento fino (ROCHA, 2006) pode ser aplicado para se estimar a atitude inicial. No alinhamento grosso a determinação do $bias_{repetibilidade}$ é feita adquirindo-se as médias dos sinais lidos durante 2 a 3 minutos com a IMU em estado estacionário. No alinhamento fino o $bias_{repetibilidade}$ já é obtido de forma mais precisa com o uso do Filtro de *Kalman*.

Para o caso da inicialização estática, o alinhamento de uma IMU está na determinação da MTC C_b^n que é calculada em dois passos: nivelamento e guinagem. O nivelamento é a obtenção dos ângulos de atitude *roll* e *pitch* usando as saídas dos acelerômetros e portanto usando o vetor gravidade local como referência. A guinagem consiste na obtenção do ângulo *head* usando os sinais dos giroscópios e portanto usando como referência a velocidade angular da terra.

Para IMUs de baixo custo (MEMS) que não leem velocidade angular da terra, já que o erro de seus giroscópios são muitos superiores a taxa de rotação do planeta ($bias \geq 0.1\text{graus}/s$), a informação de *head* só é medida por instrumentos externos como o magnetômetro e seu alinhamento portanto requer técnicas diferentes das utilizadas em IMUs de alto custo.

4.3.1 ALINHAMENTO GROSSEIRO

O alinhamento aqui proposto considera que o veículo encontra-se em repouso. Com esta consideração temos que a aceleração linear e a velocidade angular no referencial do corpo (sinais dos acelerômetros e giroscópios sentidos pela IMU) são dadas por:

$$\underline{f}^b = \underline{g}^b = C_n^b \underline{g}^n \quad (4.2)$$

$$\underline{\omega}_{ib}^b = \underline{\omega}_{ie}^b = C_n^b \underline{\omega}_{ie}^n \quad (4.3)$$

onde:

\underline{g}^n é o vetor aceleração da gravidade no referencial navegacional e é dado por:

$$\underline{g}^n = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.4)$$

$\underline{\omega}_{ie}^n$ é a velocidade angular da terra no referencial navegacional e dada por:

$$\underline{\omega}_{ie}^n = \begin{pmatrix} \omega_e \cos \varphi \\ 0 \\ -\omega_e \sin \varphi \end{pmatrix} \quad (4.5)$$

com:

g sendo a componente vertical da aceleração local da gravidade da terra;

ω_e a taxa de rotação da terra (0.72921151e-4 rad/s) e

φ a latitude.

C_n^b é a MTC do referencial local para o referencial do corpo:

$$C_n^b = (C_b^n)^{-1} = (C_b^n)^T \quad (4.6)$$

com C_b^n dada por:

$$C_b^n = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} = \left[\begin{array}{c|cc|c} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\theta s\phi c\psi \\ \hline c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{array} \right] \quad (4.7)$$

Admitindo que existam erros nas leituras dos sensores, os sinais fornecidos nas equações 4.2 e 4.3 são na realidade estimativas das grandezas reais. Assim:

$$\hat{\underline{g}}^b = C_n^b \underline{g}^n \quad (4.8)$$

$$\hat{\underline{\omega}}_{ie}^b = C_n^b \underline{\omega}_{ie}^n \quad (4.9)$$

ou:

$$\begin{pmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{pmatrix} = \begin{pmatrix} c_{11} & c_{21} & c_{31} \\ c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.10)$$

$$\begin{pmatrix} \hat{\omega}_x \\ \hat{\omega}_y \\ \hat{\omega}_z \end{pmatrix} = \begin{pmatrix} c11 & c21 & c31 \\ c12 & c22 & c32 \\ c13 & c23 & c33 \end{pmatrix} \begin{pmatrix} \omega_e \cos \varphi \\ 0 \\ -\omega_e \sin \varphi \end{pmatrix} \quad (4.11)$$

sendo \hat{a}_x , \hat{a}_y e \hat{a}_z as componentes das acelerações lidas pelos acelerômetros X, Y e Z e $\hat{\omega}_x$, $\hat{\omega}_y$ e $\hat{\omega}_z$ as componentes das velocidades angulares lidas pelos giroscópios X, Y e Z.

De 4.10 temos que:

$$\hat{a}_x = c31(g) \implies \hat{c}31 = \frac{\hat{a}_x}{g} \quad (4.12)$$

$$\hat{a}_y = c32(g) \implies \hat{c}32 = \frac{\hat{a}_y}{g} \quad (4.13)$$

$$\hat{a}_z = c33(g) \implies \hat{c}33 = \frac{\hat{a}_z}{g} \quad (4.14)$$

e de 4.11 temos:

$$\hat{\omega}_x = c11\omega_e \cos \varphi - c31\omega_e \sin \varphi \implies \hat{c}11 = \frac{\hat{\omega}_x}{\omega_e \cos \varphi} + \frac{\hat{a}_x}{g} \tan \varphi \quad (4.15)$$

$$\hat{\omega}_y = c12\omega_e \cos \varphi - c32\omega_e \sin \varphi \implies \hat{c}12 = \frac{\hat{\omega}_y}{\omega_e \cos \varphi} + \frac{\hat{a}_y}{g} \tan \varphi \quad (4.16)$$

$$\hat{\omega}_z = c13\omega_e \cos \varphi - c33\omega_e \sin \varphi \implies \hat{c}13 = \frac{\hat{\omega}_z}{\omega_e \cos \varphi} + \frac{\hat{a}_z}{g} \tan \varphi \quad (4.17)$$

Como C_b^n é uma matriz ortonormal, suas linhas formam vetores ortogonais, ou seja:

$$\hat{c}2 = \hat{c}3 \times \hat{c}1 \quad (4.18)$$

logo:

$$\begin{pmatrix} c21 \\ c22 \\ c23 \end{pmatrix} = \begin{pmatrix} c31 \\ c32 \\ c33 \end{pmatrix} \times \begin{pmatrix} c11 \\ c12 \\ c13 \end{pmatrix} = \begin{pmatrix} 0 & -c33 & c32 \\ c33 & 0 & -c31 \\ -c32 & c31 & 0 \end{pmatrix} \begin{pmatrix} c11 \\ c12 \\ c13 \end{pmatrix} \quad (4.19)$$

então,

$$\hat{c}21 = -\hat{c}12\hat{c}33 + \hat{c}13\hat{c}32 \quad (4.20)$$

$$\hat{c}22 = \hat{c}11\hat{c}33 - \hat{c}13\hat{c}31 \quad (4.21)$$

$$\hat{c}23 = -\hat{c}11\hat{c}32 + \hat{c}12\hat{c}31 \quad (4.22)$$

De posse de todas as estimativas dos elementos de C_b^n , substituímos os valores encontrados na equação 4.7, então:

$$c32 = \sin\phi\cos\theta, \quad c33 = \cos\phi\cos\theta \implies \quad (4.23)$$

$$\implies \frac{c32}{c33} = \tan\phi \implies \hat{\phi} = \tan^{-1}\left(\frac{\hat{c}32}{\hat{c}33}\right) \quad (4.24)$$

$$c21 = \cos\theta\sin\psi, \quad c11 = \cos\theta\cos\psi \implies \quad (4.25)$$

$$\implies \frac{c21}{c11} = \tan\psi \implies \hat{\psi} = \tan^{-1}\left(\frac{\hat{c}21}{\hat{c}11}\right) \quad (4.26)$$

$$c31 = -\sin\theta, \quad c11 = \cos\theta\cos\psi \implies \quad (4.27)$$

$$\implies \frac{c31}{c11} = -\frac{\tan\theta}{\cos\psi} \implies \hat{\theta} = \tan^{-1}\left(\frac{-\hat{c}31\cos\hat{\psi}}{\hat{c}11}\right) \quad (4.28)$$

4.3.2 ALINHAMENTO GROSSEIRO COM SENSORES TIPO MEMS

O procedimento acima mostra que a obtenção inicial dos parâmetros de atitude depende dos sinais $\hat{\omega}_x$, $\hat{\omega}_y$ e $\hat{\omega}_z$ lidos pelo giroscópio com a IMU parada, ou seja, estes sinais são as componentes da velocidade angular da terra no referencial do corpo. Como explicado anteriormente, se esta IMU for do tipo MEMS, as informações do giroscópio, no alinhamento estático, não serão confiáveis e os parâmetros de atitude iniciais não serão estimados corretamente.

Portanto, para IMUs MEMS, a inicialização da atitude (*roll* e *pitch*) é feita apenas com os sinais dos acelerômetros que utilizam a gravidade como referência. A inicialização do *head* deve ser feita com uma referência externa como o magnetômetro, ou com um GPS que forneça a informação de *head*.

Novamente considerando o alinhamento estático, os sinais medidos pelos acelerômetros \underline{a}^b são dados pela equação 4.10, logo:

$$\underline{\underline{a}}^b = \begin{pmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{pmatrix} = g \begin{pmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{pmatrix} \quad (4.29)$$

Resolvendo para os ângulos de atitude:

$$\begin{pmatrix} \sin\phi \\ \cos\phi \end{pmatrix} = \frac{1}{\sqrt{\hat{a}_y^2 + \hat{a}_z^2}} \begin{pmatrix} \hat{a}_y \\ \hat{a}_z \end{pmatrix} \quad (4.30)$$

$$\begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix} = \frac{1}{g} \begin{pmatrix} -\hat{a}_x \\ \hat{a}_y \sin\phi + \hat{a}_z \cos\phi \end{pmatrix} \quad (4.31)$$

portanto,

$$\hat{\phi} = \tan^{-1}\left(\frac{\hat{a}_y}{\hat{a}_z}\right) \quad (4.32)$$

$$\hat{\theta} = \tan^{-1}\left(\frac{-\hat{a}_x}{\sqrt{\hat{a}_y^2 + \hat{a}_z^2}}\right) \quad (4.33)$$

5 IMPLEMENTAÇÃO DO ALGORITMO

Neste capítulo será mostrado a implementação do algoritmo em MATLAB/Simulink juntamente com as ferramentas de geração de código de tempo real, *Real Time Workshop* e *xPC Target*. Será mostrado também um gerador de trajetória constituído de um simulador de vôo gráfico, o *FlightGear*, integrado a um *joystick* para validar e testar o algoritmo de navegação.

5.1 MODELO IMPLEMENTADO EM SIMULINK

O Simulink é um programa utilizado para modelagem, simulação e análise de sistemas dinâmicos (MATHWORKS-SIMULINK, 1999). O programa se aplica a sistemas lineares e não lineares, contínuos e/ou discretos no tempo. Utiliza uma interface gráfica com o usuário para construção dos modelos a partir de diagramas em blocos, através de operações de clique-e-arraste do mouse. Com esta interface pode-se criar modelos da mesma forma que se cria diagramas de blocos (DABNEY & HARMAN, 2003). A versão utilizada do MATLAB/Simulink para a implementação do algoritmo foi a R2008a.

A implementação em Simulink do algoritmo de navegação inercial integrado com o GPS pode ser vista na figura 5.1:

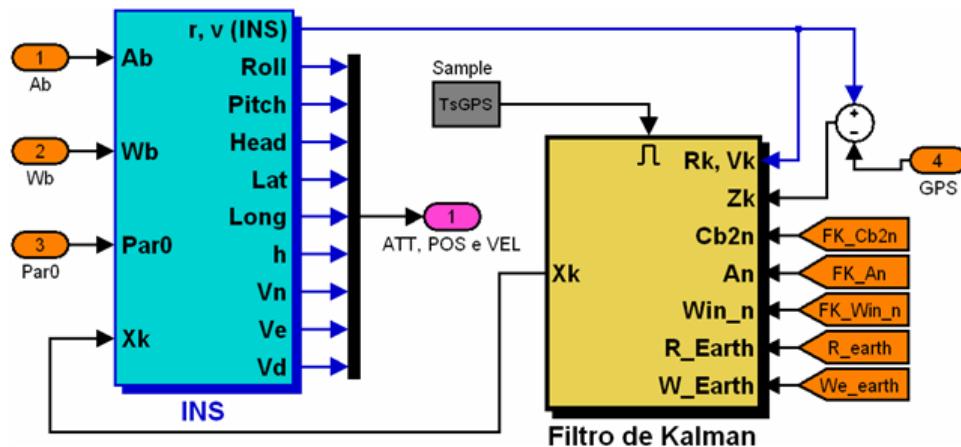


FIG.5.1: Algoritmo de navegação inercial integrado com o GPS

O algoritmo foi implementado em tempo discreto. Para isso a opção "solver options" da aba "*Simulation → Configuration parameters*" do Simulink está em: **Type: Fixed**

step e *Solver: discrete (no continuous state)*). Estas opções também são necessárias para o uso do *Real Time Workshop*.

Pode ser observado na figura 5.1 que a entrada do algoritmo são os sinais de acelerômetros (Ab), sinais de giroscópios (Wb), sinais de GPS e os parâmetros iniciais do veículo ($Par0$). As saídas são os parâmetros de navegação (*roll*, *pitch*, *head*, Latitude, Longitude, Altitude, Vn , Ve e Vd). Pode-se notar também dois blocos principais, O bloco de navegação inercial puro (INS) e o bloco do Filtro de *Kalman*. Os dois blocos estão integrados exatamente como mostrado na figura 3.1 do capítulo 3.

O bloco "INS" da figura 5.1 é detalhado na figura 5.2. Os blocos 1, 2, 3 e 4 estão dentro de um "loop" principal que é executado a cada tempo de amostragem do algoritmo.

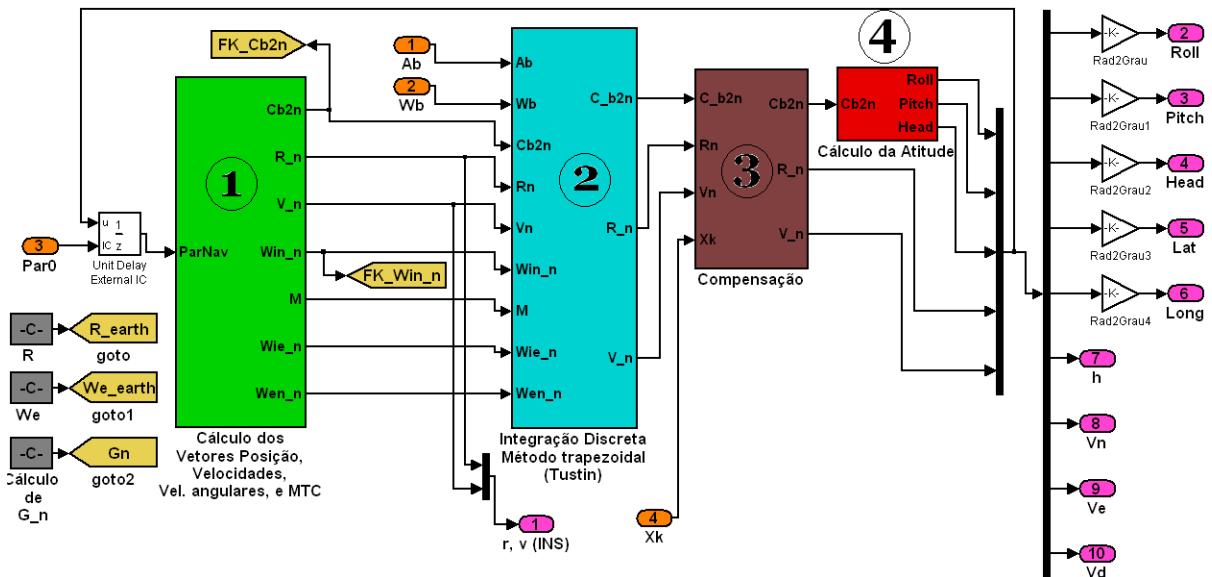


FIG.5.2: Algoritmo de navegação inercial puro - INS

Nos blocos 1 e 2 da figura 5.2 está implementado o sistema de equações 2.38. Estes blocos calculam, a cada tempo de amostragem, os parâmetros de navegação que são atualizados na saída do algoritmo e retornam ao início do "loop".

O bloco 3 faz a compensação (correção) dos parâmetros de navegação calculados através do vetor de erros X_k estimado pelo Filtro de *Kalman*. Neste bloco estão implementadas as equações 3.36, 3.37 e 3.39. Este bloco só é executado a cada atualização do GPS, como o GPS geralmente tem uma taxa de atualização menor que a da IMU, a compensação só ocorre em intervalos de tempo superiores ao tempo de amostragem do sistema. Portanto, quanto melhor a atualização do GPS (ou quanto menor seu tempo de amostragem), mais rapidamente os erros vão ser compensados e melhor será a exatidão no cálculo dos parâmetros de navegação.

Para IMUs de alta precisão, a velocidade de atualização do GPS não chega a ser tão importante, já que os erros destes sensores são tão pequenos que o algoritmo de navegação puro consegue manter um alto nível de exatidão dos parâmetros calculados por longos períodos de tempo sem o sinal de GPS. Para IMUs de baixo custo tipo MEMS, no entanto, a dependência do GPS é bem maior, e poucos segundos sem este sinal já é o suficiente para corromper as informações de navegação.

O bloco 4 possui duas funções: calcular os ângulos de Euler a partir da Matriz de Transformação de Coordenadas (MTC) e ortonormalizar a mesma. Sendo este algoritmo um processo numérico, uma matriz ortonormal vai perdendo esta sua característica a medida que vai sendo manipulada. Para resolver isto pode-se apenas desmontá-la e remontá-la a cada passo, ou seja, extrai-se os ângulos de Euler da MTC no final de sua operação (final do "loop") e no início do "loop" monta-se a MTC novamente, já que uma MTC criada a partir de quaisquer três ângulos sempre será ortonormal.

A figura 5.3 mostra o bloco "Filtro de Kalman" da figura 5.1 mais detalhado.

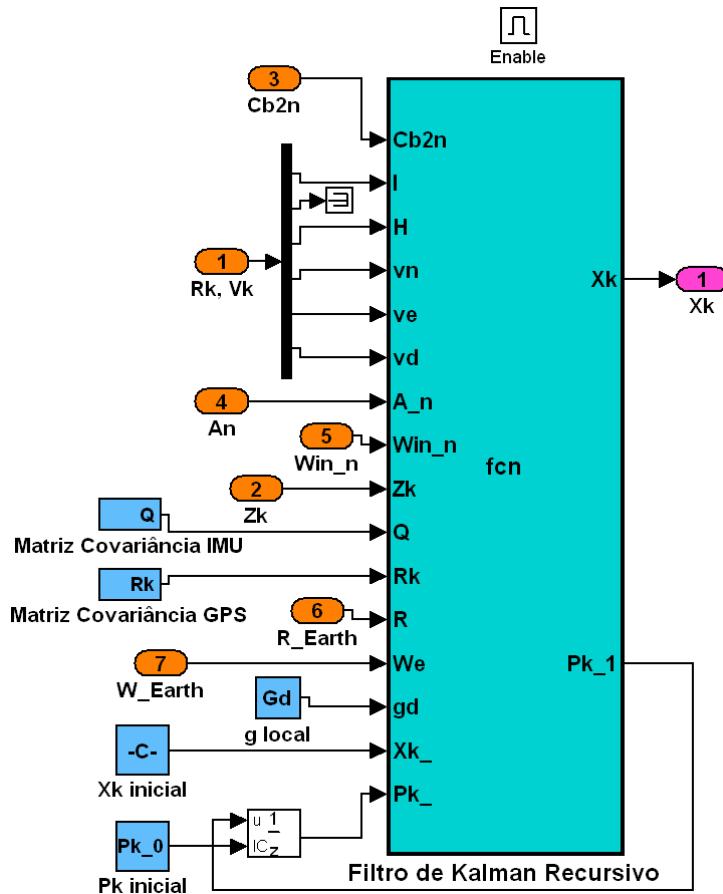


FIG.5.3: Algoritmo do Filtro de *Kalman*

Está implementado neste bloco o modelo de estimativa dos erros dos parâmetros de

navegação já apresentado no capítulo 3, mais especificamente, as equações 3.23, 3.27 e 3.31 a 3.35. Como o Filtro de *Kalman* faz a estimativa dos erros baseado nas leituras do GPS e da saída do INS puro, ele só é executado no maior tempo de amostragem do sistema, ou seja, no tempo de amostragem do GPS. A saída deste bloco é o vetor de erros X_k usado para compensar os parâmetros de navegação do INS (bloco 3 da figura 5.2).

5.2 IMPLEMENTAÇÃO EM TEMPO REAL COM O *XPC TARGET*

Atualmente, além da simulação, o Simulink está sendo bastante utilizado na implementação em hardware de modelos com o propósito de serem executados em tempo real. A ferramenta ("toolbox") usada para isto é o *Real Time Workshop* também da *Mathworks* e integrada ao Simulink. O *Real Time Workshop* gera e otimiza códigos portáveis e customizáveis ANSI C ou C++ de modelos do Simulink para criar implementações específicas de modelos que operam em tempo real. O código gerado pode rodar em PCs, microcontroladores e DSPs e ainda rodar em qualquer sistema operacional de tempo real (RTOS).

O *xPC Target* (MATHWORKS-XPC, 2008) é um ambiente para a prototipagem, teste e desenvolvimento de sistemas que funcionam em tempo real para computadores padrões. Ele utiliza um computador "*Target*" (alvo), separado do computador "*Host*" (principal), para rodar aplicações em tempo real. Um computador "*Target*" pode ser praticamente qualquer computador com um *Intel 386* ou *486* ou *AMD K5* ou *K6/Athlon*, pode ser um computador de mesa ou um computador industrial como um *PC104* o qual foi usado neste trabalho. A partir de um modelo criado no Simulink, pode-se gerar um código executável a partir da "toolbox" *Real Time Workshop* (MATHWORKS-RTW, 2007), tendo como alvo o próprio *xPC Target*. O código executável e o *xPC Target real-time Kernel* é então gravado do computador "*Host*" para o "*Target*". Em seguida, pode-se executar a aplicação em tempo real, com a possibilidade de variação "*online*" dos parâmetros do modelo.

Ao se criar um modelo no Simulink em que a finalidade é a geração de código C para a implementação em tempo real, deve-se ter o cuidado com a escolha dos blocos a serem inseridos no modelo, já que nem todos os blocos do Simulink geram código. Muitos problemas encontrados durante a criação de um modelo complexo no Simulink podem ser sanados com a utilização dos "*S-function Blocks*". "*S-function Blocks*" que geram código em C são dois: "*C-file S-Function Block*" e "*Embedded MATLAB function*". Este último chega a ser mais interessante, pois pode ser escrito na própria linguagem do MATLAB,

sendo no entanto um pouco limitado nos comandos que podem ser utilizados.

Se o alvo escolhido para a geração de código for o *xPC Target*, os blocos de comunicação externa do modelo devem ser do próprio *xPC Target*. A figura 5.4 mostra o modelo em Simulink que foi utilizado para uso no *xPC Target*.

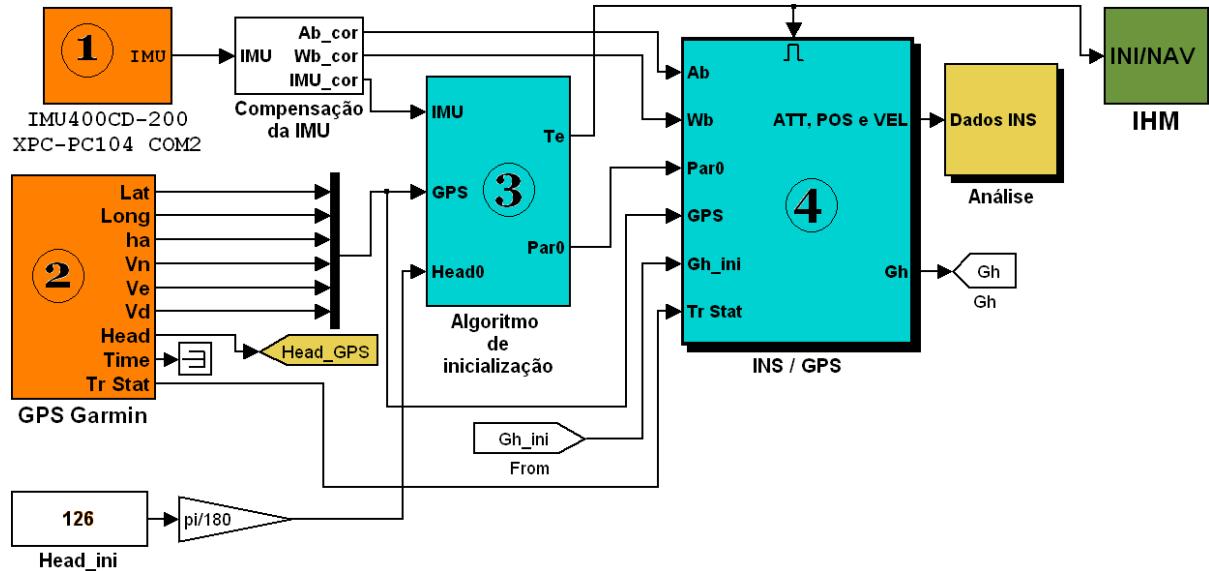


FIG.5.4: Algoritmo de navegação inercial integrado com GPS implementado para a geração de código no *xPC Target*

Detalhando os blocos 1 e 2 da figura 5.4, temos as figuras 5.5 e 5.6 respectivamente. Os blocos denominados "Baseboard Serial" são os blocos de comunicação serial do *xPC Target*:

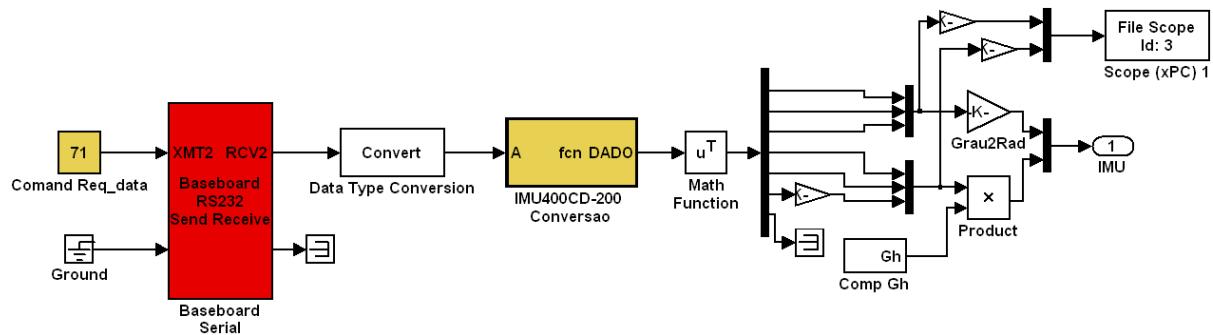


FIG.5.5: Algoritmo da comunicação serial do *xPC Target* com a IMU400

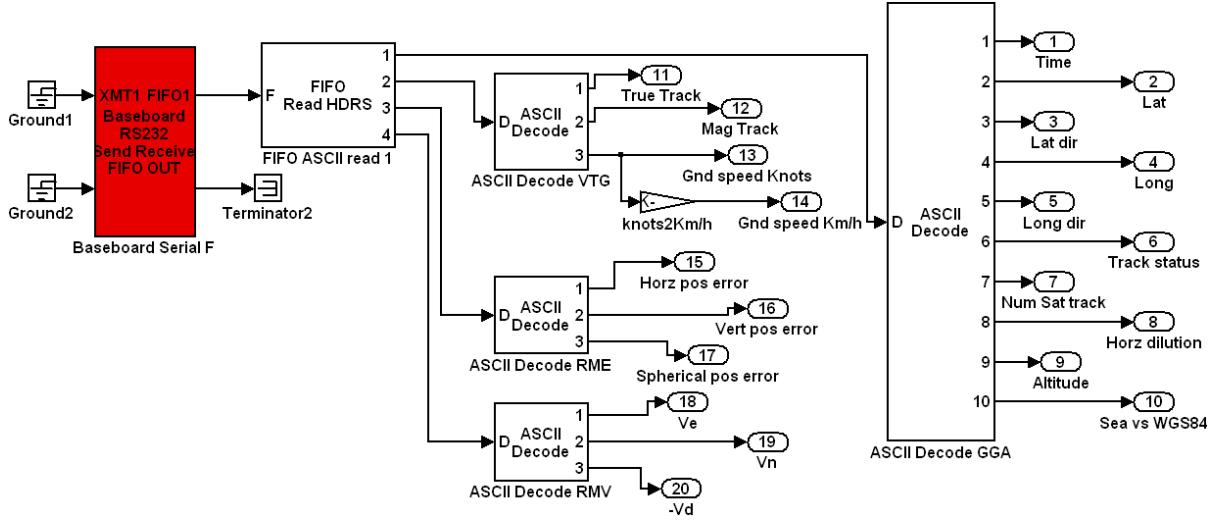


FIG.5.6: Algoritmo da comunicação serial do *xPC Target* com o GPS 18-5Hz *Garmin*

Uma vez que o modelo esteja implementado no Simulink deve-se configurar o *xPC Target Explorer* com o comando "*xpcexplr*". O *xPC Target Explorer* é uma interface gráfica para configurar o PC alvo e alguns parâmetros de execução do código. O principal parâmetro a se configurar é o modo de geração do código C a partir do modelo em Simulink, onde pode-se escolher os modos "*BootFloppy*", "*DOSLoader*" e "*StandAlone*". Com os dois primeiros modos é necessário que na geração do código o computador "*Host*" (computador onde está instalado o Matlab) esteja conectado via rede "*Ethernet*" ou conexão serial ao computador "*Target*" (computador que irá executar o código em tempo real), pois quando o código é gerado, também é carregado para o PC alvo. Também é necessário que o computador target possua um leitor de *diskete*, pois o *kernel* do *xPC Target* é carregado no PC alvo através deste disco. O terceiro modo ("*StandAlone*"), o qual foi o utilizado nos experimentos práticos deste trabalho, é o modo mais prático, pois não necessita de conexão entre o PC "*Target*" e o PC "*Host*" e nem precisa de leitor de *diskete*. A geração de código neste modo gera de forma independente o *kernel* do *xPC Target* e o código do modelo executável na forma de dois arquivos que podem ser carregados para o PC alvo através de um "*Flashdisk*" ou do próprio HD (*Hard Disk*) do computador alvo. Este modo, no entanto, não é habilitado no toolbox do xPC Target por padrão. Para se usar este modo deve-se adquirir a toolbox *xPC Target Embedded Option*.

Estando configurado o *xPC Target Explorer* no modo "*StandAlone*" pode-se então gerar o código. O código é gerado através da interface de configuração do *Real Time Workshop*. Esta interface é aberta no próprio modelo do Simulink em "*Simulation* → *Configuration Parameters* → *Real Time Workshop*". Nesta interface escolhe-se o Sis-

tema Operacional de Tempo Real que no caso é o *xPC Target* e depois gera-se o código clicando no botão "*Build*".

5.3 GERADOR DE TRAJETÓRIA - O SIMULADOR *FLIGHTGEAR*

A fim de validar e analisar os resultados do algoritmo de navegação inercial integrado com o GPS, faz-se necessário um gerador de trajetórias. Este gerador de trajetórias deve, a partir de uma entrada de sinais de acelerômetros e giroscópios, gerar parâmetros de navegação que se sabe estarem corretos. Então, a partir destas mesmas entradas, use-se o algoritmo INS/GPS para gerar os parâmetros de navegação e compará-los com os parâmetros do gerador.

Uma ferramenta que se adequa perfeitamente aos requisitos deste gerador é um simulador gráfico de vôo onde se tem acesso à modelagem dinâmica da aeronave a ser pilotada virtualmente com um *joystick*. O MATLAB possui tal ferramenta integrada, o *FlightGear*, que é um simulador de vôo multi plataforma, de desenvolvimento cooperativo e de código aberto sobre a licença GNU (*General Public License*). A utilização do *FlightGear* no MATLAB se dá através do Simulink com a *Toolbox Aerospace Blockset* (MATHWORKS-AEROSPACE, 2008).

A figura 5.7 é um exemplo do Simulink que utiliza o *FlightGear*, o modelo é intitulado "*Fly the DeHavilland Beaver*" (asbdhc2.mdl). A figura 5.8 é uma versão modificada e adaptada deste exemplo para operar com o algoritmo de navegação inercial.

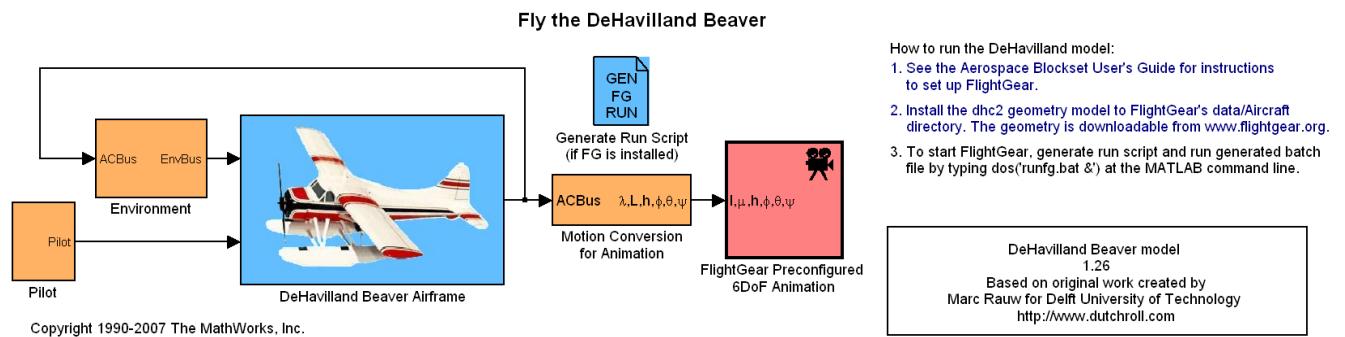


FIG.5.7: Modelo de demonstração do Simulink "asbdhc2.mdl"

Fly the DeHavilland Beaver com Rotação da Terra

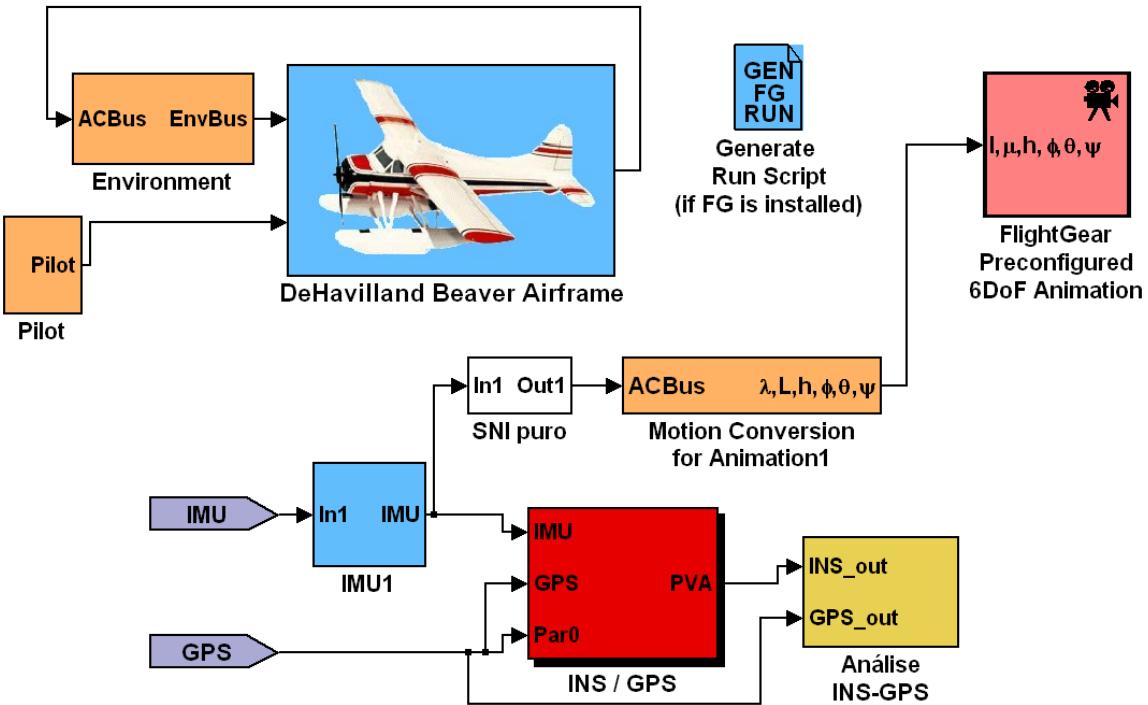


FIG.5.8: Modelo "asbdhc2.mdl" modificado para operar como Gerador de trajetória

O modelo "Fly the DeHavilland Beaver" funciona da seguinte forma: A entrada do modelo são os sinais do joystick (*roll*, *pitch*, *head* e Acelerador) inseridos no bloco "Pilot". Estes sinais são processados pela modelagem da aeronave (bloco "DeHavilland Beaver Airframe") que calcula continuamente os parâmetros de navegação. Estes parâmetros servem de entrada para o simulador *FlightGear*, onde os dados de Atitude (*roll*, *pitch* e *head*) e Posição (Latitude, Longitude e Altitude) são usados na simulação gráfica do vôo (figura 5.9). Na modelagem estão incluídos além da aerodinâmica do avião, os efeitos do vento e da gravidade.

A idealização de se usar este modelo como gerador de trajetória veio da possibilidade de se extrair da modelagem do avião (bloco "DeHavilland Beaver Airframe") os sinais dos sensores inerciais, ou seja, os sinais de acelerômetros e giroscópios. De posse destes sinais, utilizamo-os então como entrada do algoritmo de navegação inercial, e a partir dos parâmetros de navegação calculados por este, usamos o *FlightGear* para gerarmos a trajetória que quisermos com o *joystick*.



FIG.5.9: Simulação de vôo com o *FlightGear* a partir dos parâmetros calculados pelo INS/GPS

É importante observar que os sinais de acelerômetros e giroscópios extraídos da modelagem do avião, são sinais ideais e não representam a realidade das leituras de sensores inerciais. Para nos aproximarmos desta realidade, adicionamos ruído Gaussiano aos 3 sinais de acelerações lineares e aos 3 sinais de velocidades angulares. Estes ruídos inseridos têm a mesma magnitude dos ruídos encontrados na IMU400, que é um sensor inercial tipo MEMS fabricado pela *Crossbow* (IMU400-DATASHEET).

De posse da IMU simulada, falta agora o GPS simulado. Para se adquirir o GPS simulado, usa-se o algoritmo de navegação puro (sem o Filtro de *Kalman*) para calcular os parâmetros de navegação a partir dos sinais de sensores ideais, ou seja, sem o ruído adicionado. Os parâmetros calculados então estão livres de ruídos, e serão considerados como parâmetros verdadeiros, sendo usados como sinais de GPS (posição e velocidade). A estes parâmetros também são adicionados o ruído característico de um GPS comercial, que no caso foi o receptor GPS *Garmim 18-5Hz* (GARMIN INTERNATIONAL, 2005).

A tabela 5.1 mostra os níveis de ruídos adicionados aos sinais dos sensores:

TAB.5.1: Níveis de ruído adicionados aos sensores para a simulação

| Modelo do sensor utilizado | Sinal do sensor | Ruídos Gaussianos adicionados |
|----------------------------|------------------|--|
| IMU400 | a_x | Desvio Padrão: $5e - 4 \text{ g}$; Média: 0 |
| | a_y | Desvio Padrão: $5e - 4 \text{ g}$; Média: 0 |
| | a_z | Desvio Padrão: $5e - 4 \text{ g}$; Média: 0 |
| | w_x | Desvio Padrão: 0.3 graus/s ; Média: 0 |
| | w_y | Desvio Padrão: 0.3 graus/s ; Média: 0 |
| | w_z | Desvio Padrão: 0.3 graus/s ; Média: 0 |
| Garmin GPS18-5Hz | <i>Latitude</i> | Desvio Padrão: $1e - 4 \text{ graus}$; Média: 0 |
| | <i>Longitude</i> | Desvio Padrão: $1e - 4 \text{ graus}$; Média: 0 |
| | <i>Altitude</i> | Desvio Padrão: $10m$; Média: 0 |
| | V_n | Desvio Padrão: $1e - 2 \text{ m/s}$; Média: 0 |
| | V_e | Desvio Padrão: $1e - 2 \text{ m/s}$; Média: 0 |
| | V_d | Desvio Padrão: $1e - 2 \text{ m/s}$; Média: 0 |

As taxas de amostragem utilizadas para os sensores na simulação foram de 100Hz (0.01s) para a IMU e 1Hz (1s) para o GPS.

Os gráficos a seguir foram obtidos de um vôo feito com o modelo da figura 5.8. As figuras 5.10 - 5.18 mostram os parâmetros de navegação calculados pelo INS Puro, ou seja, sem a atualização do GPS com a correção do Filtro de Kalman, juntamente com os parâmetros de navegação verdadeiros (sinal de GPS), e logo após o erro entre estes dois sinais:

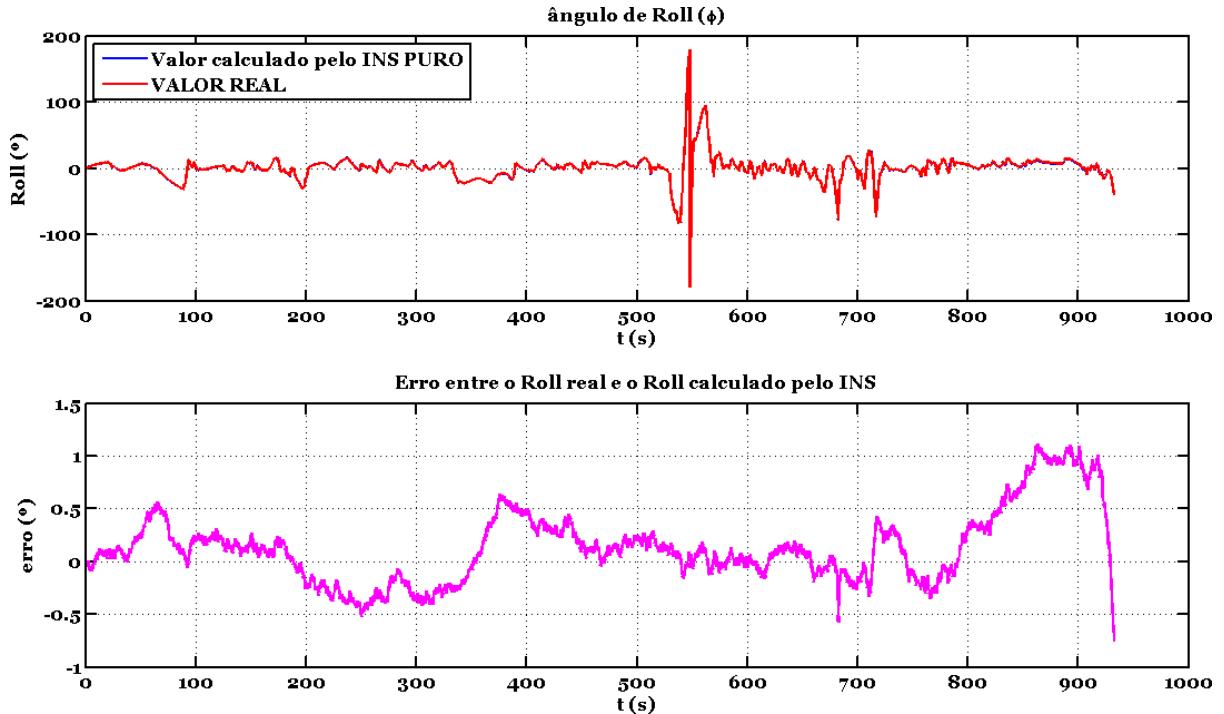


FIG.5.10: Análise do ângulo de *roll* com o INS puro

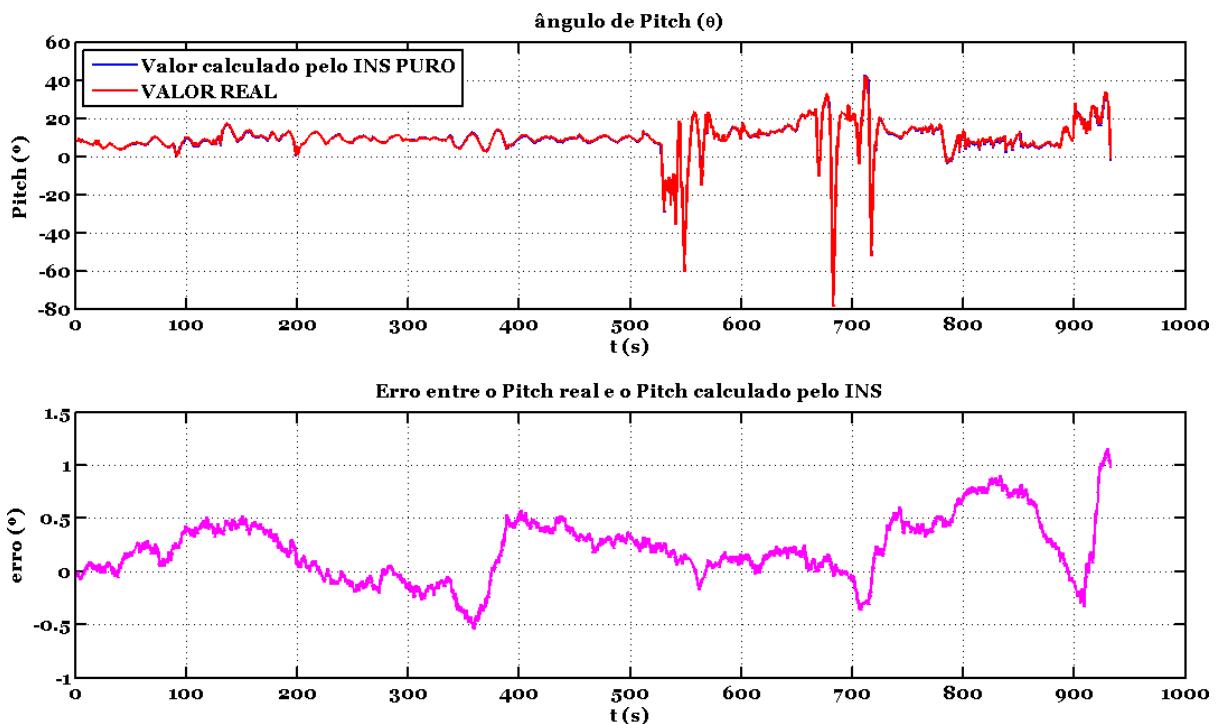


FIG.5.11: Análise do ângulo de *pitch* com o INS puro

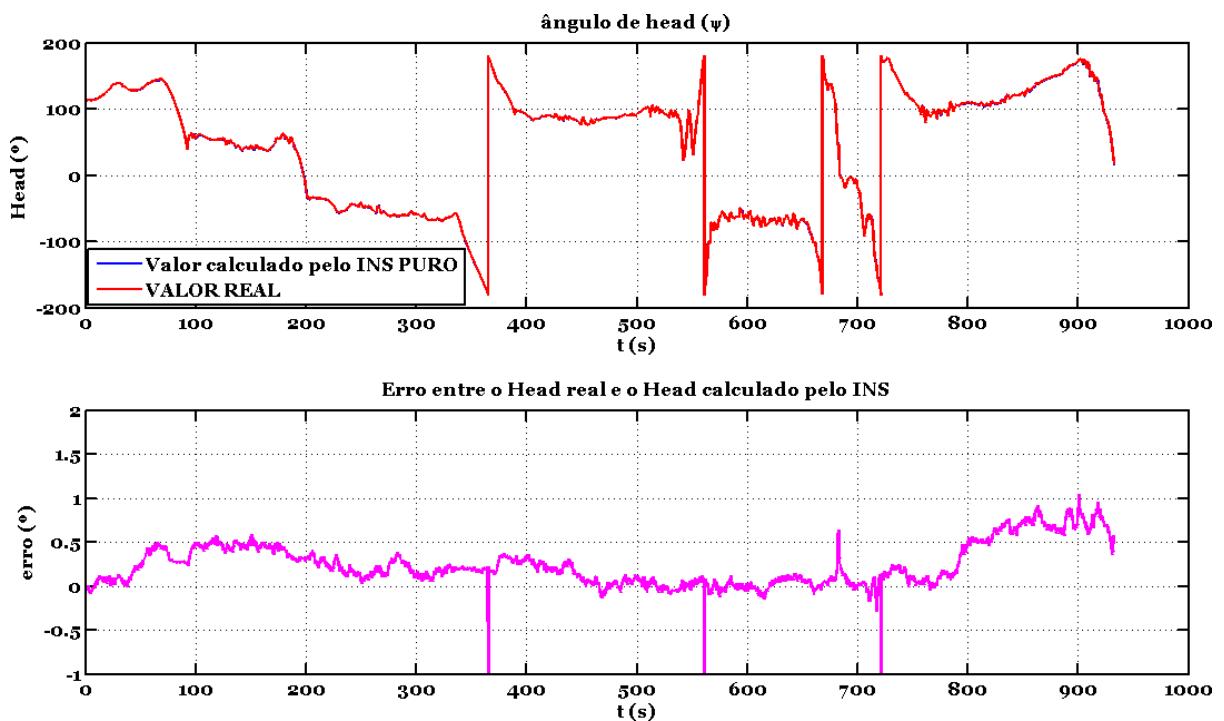


FIG.5.12: Análise do ângulo de *head* com o INS puro

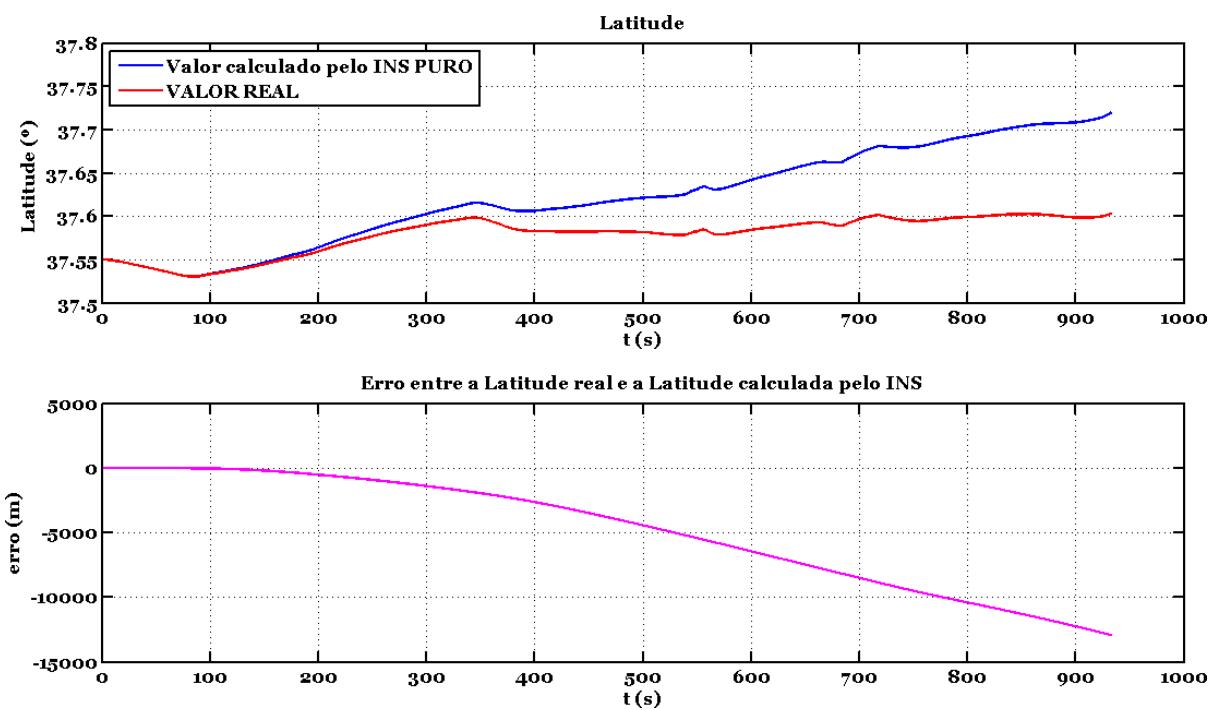


FIG.5.13: Análise da latitude com o INS puro

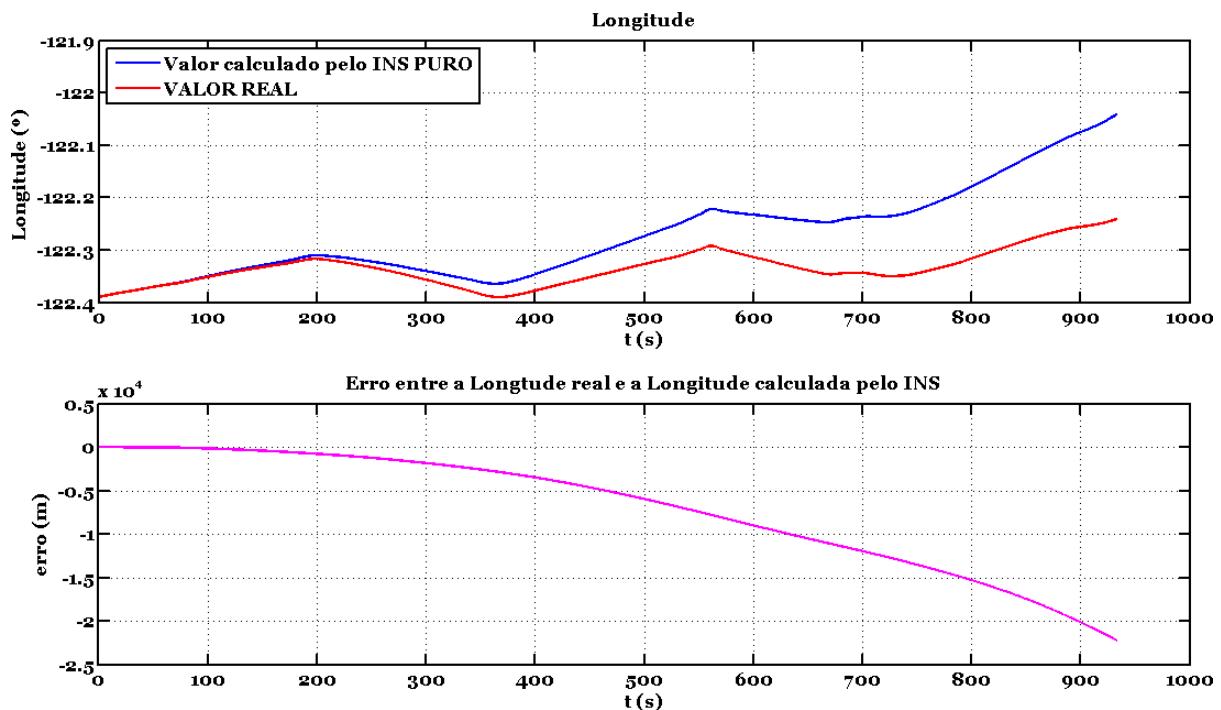


FIG.5.14: Análise da longitude com o INS puro

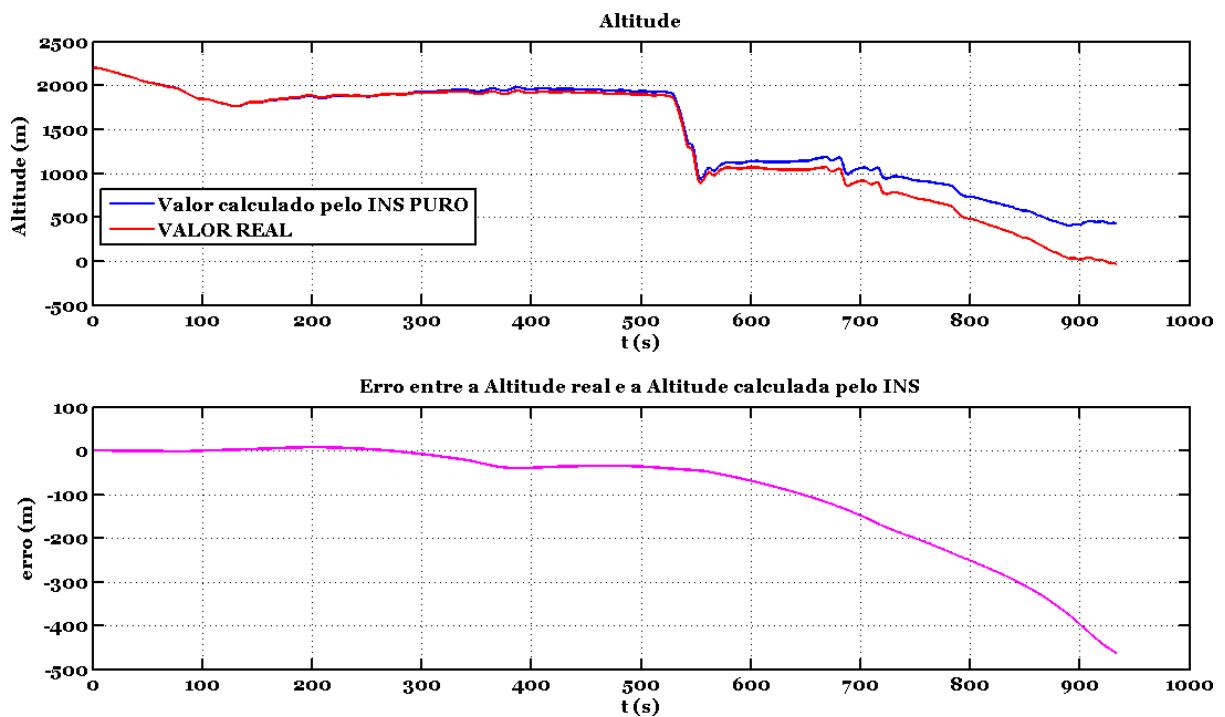


FIG.5.15: Análise da altitude com o INS puro

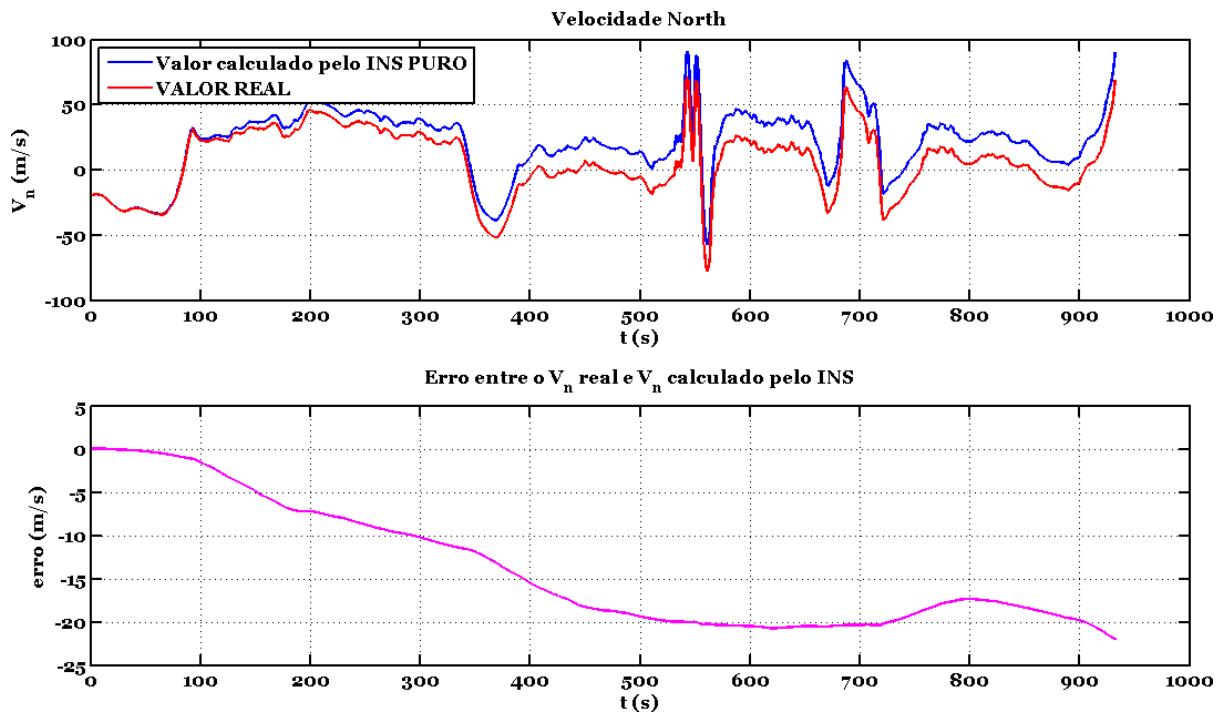


FIG.5.16: Análise da velocidade norte V_n com o INS puro

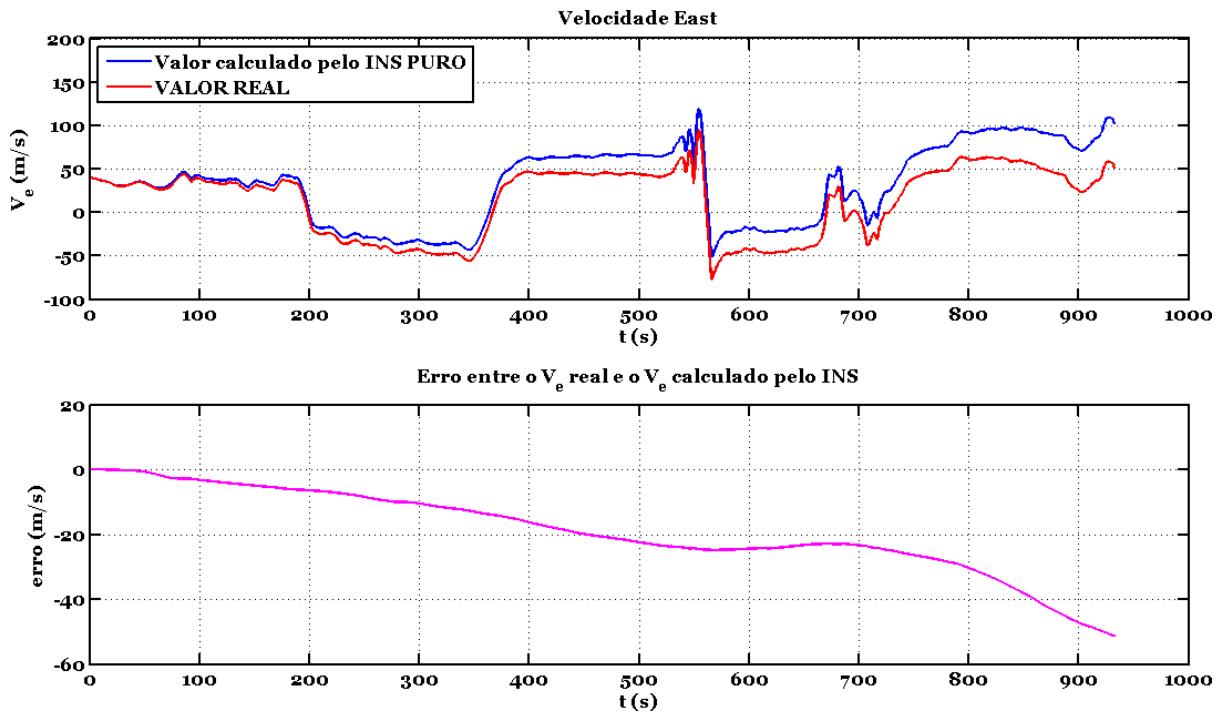


FIG.5.17: Análise da velocidade leste V_e com o INS puro

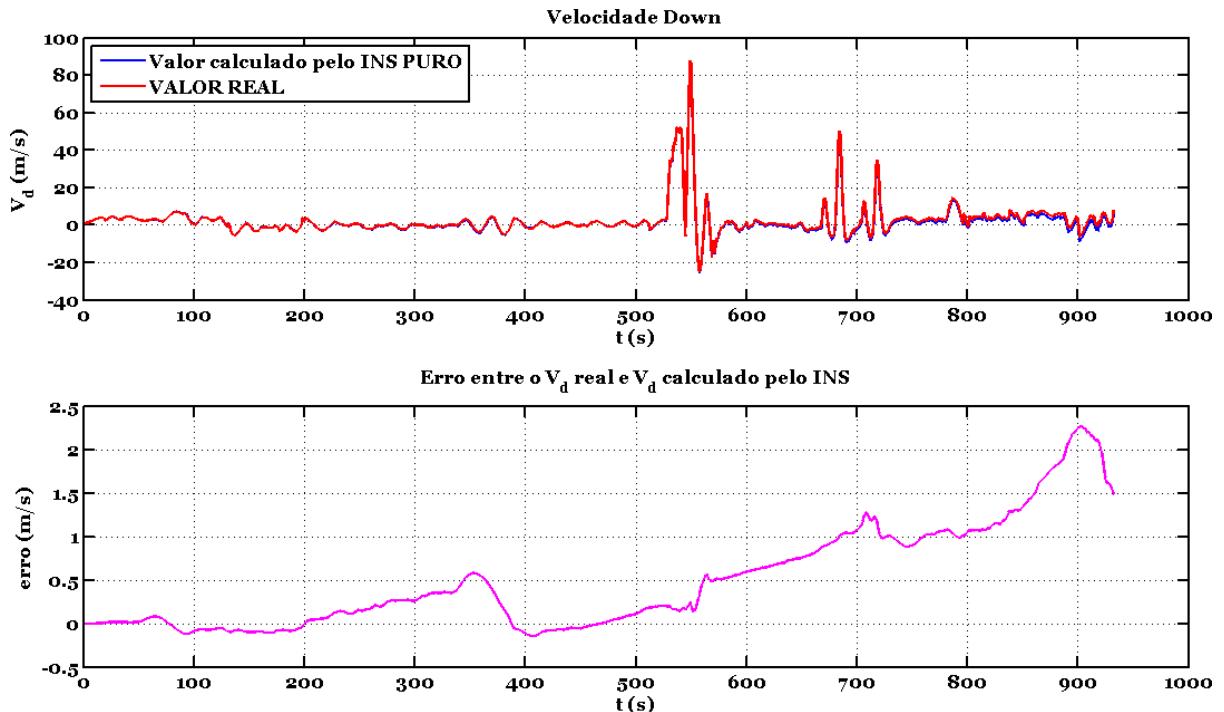


FIG.5.18: Análise da velocidade vertical V_d com o INS puro

As figuras 5.19 - 5.27 mostram os parâmetros de navegação calculados pelo INS integrado com o GPS, juntamente com os parâmetros de navegação verdadeiros (sinal de GPS), e logo após o erro entre estes dois sinais:

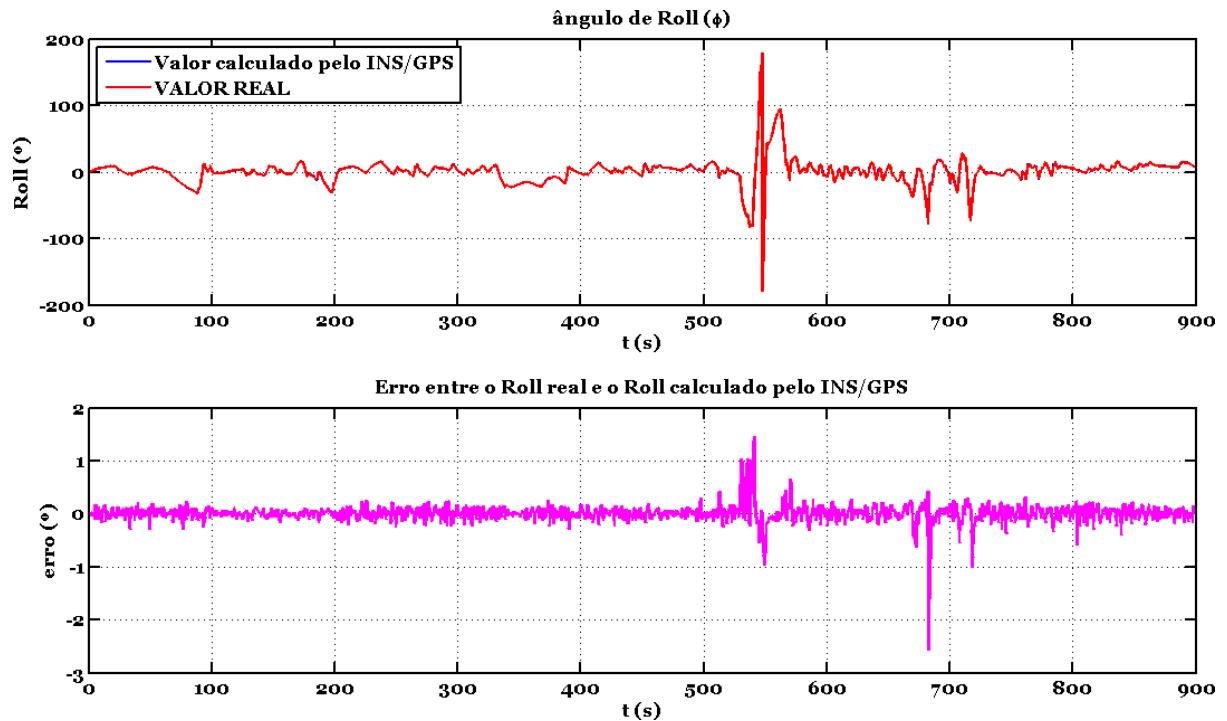


FIG.5.19: Análise do ângulo de *roll* com o INS/GPS

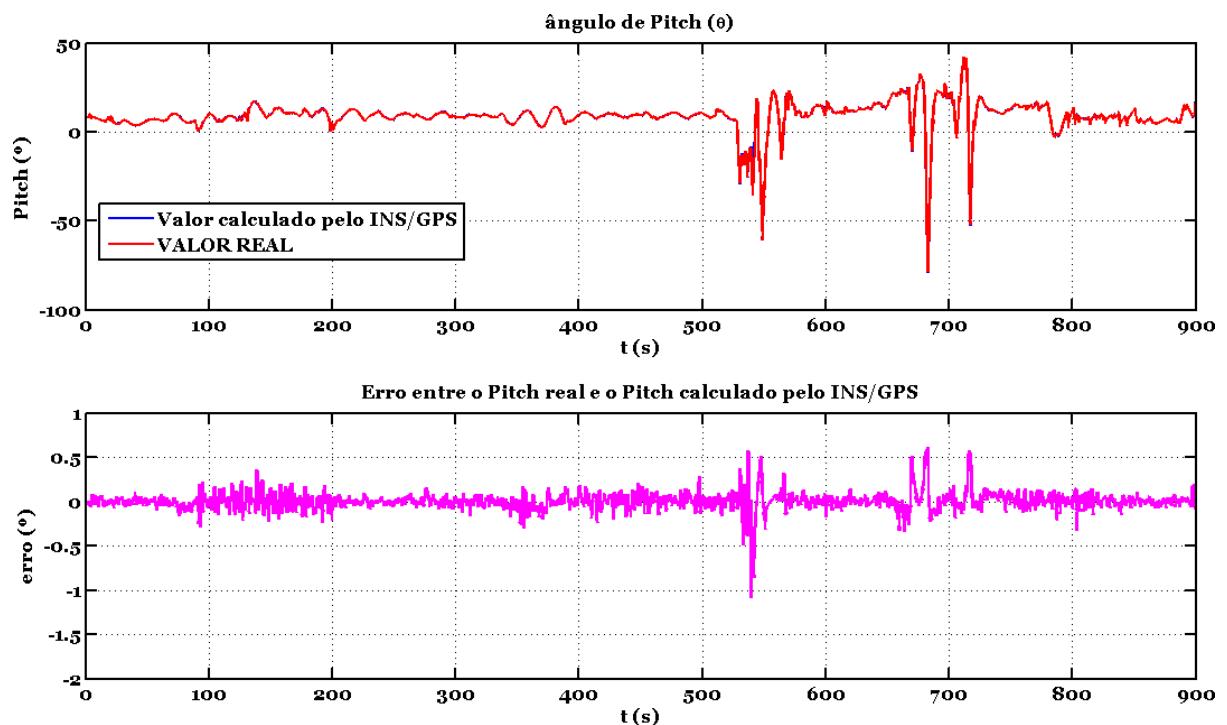


FIG.5.20: Análise do ângulo de *pitch* com o INS/GPS

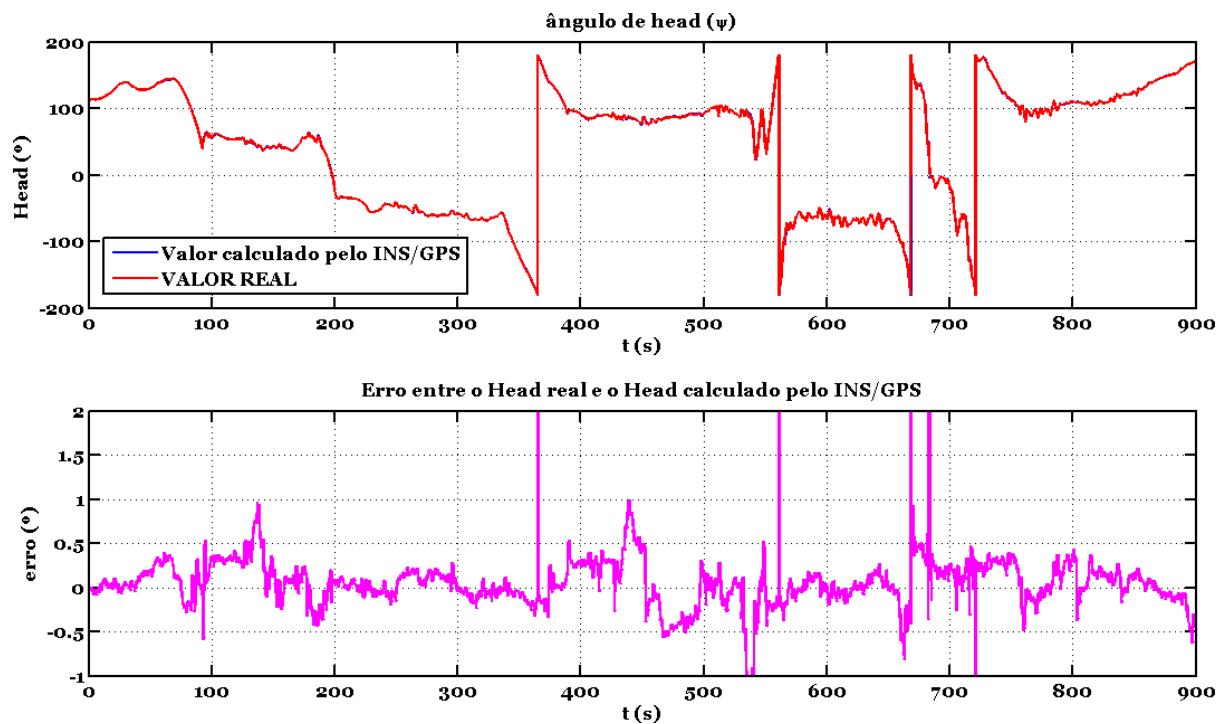


FIG.5.21: Análise do ângulo de *head* com o INS/GPS

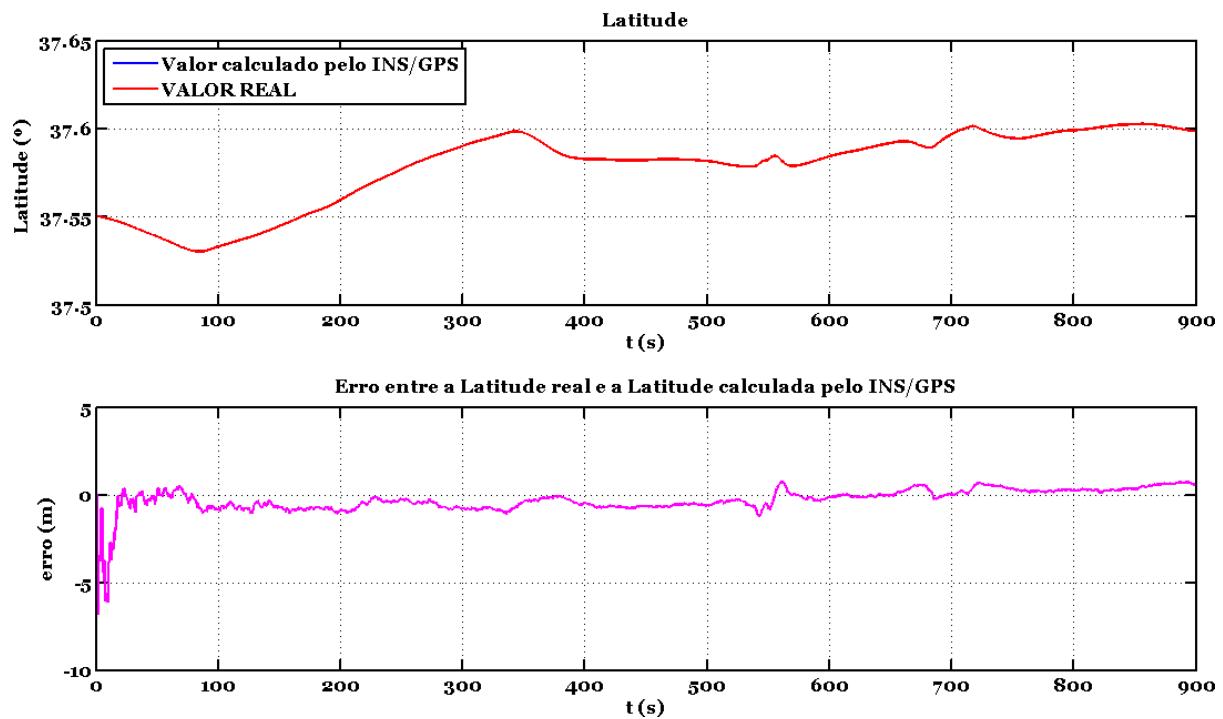


FIG.5.22: Análise da latitude com o INS/GPS

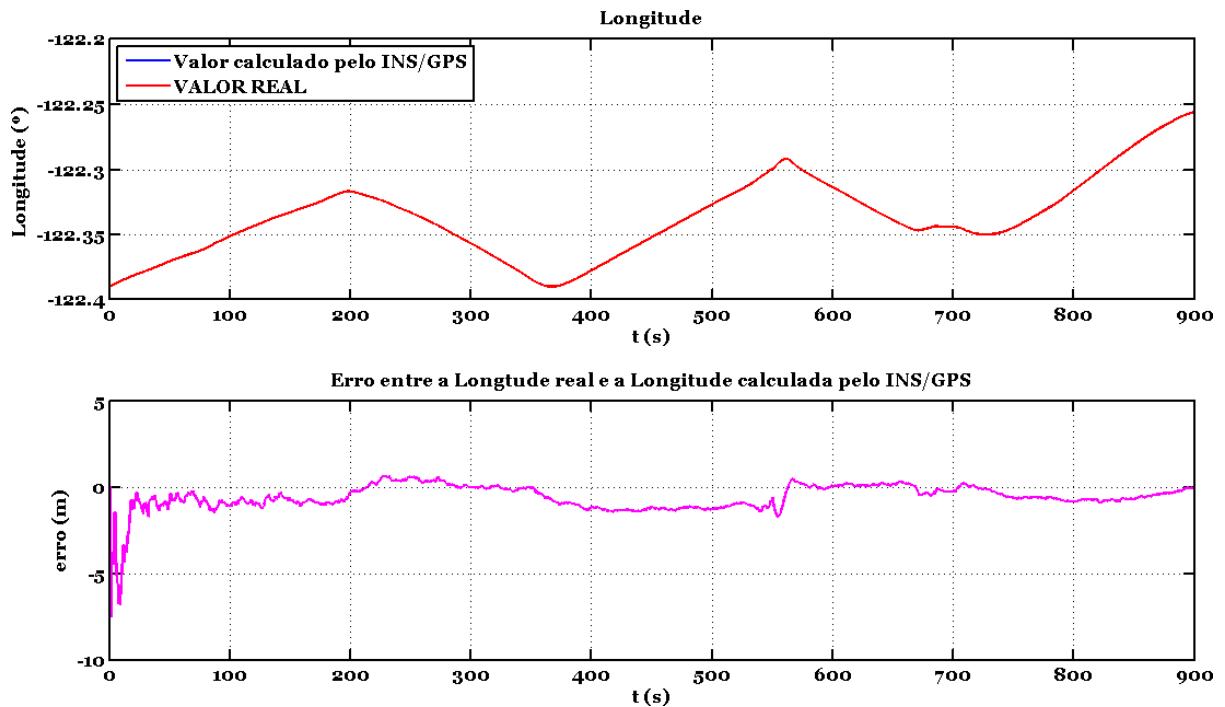


FIG.5.23: Análise da longitude com o INS/GPS

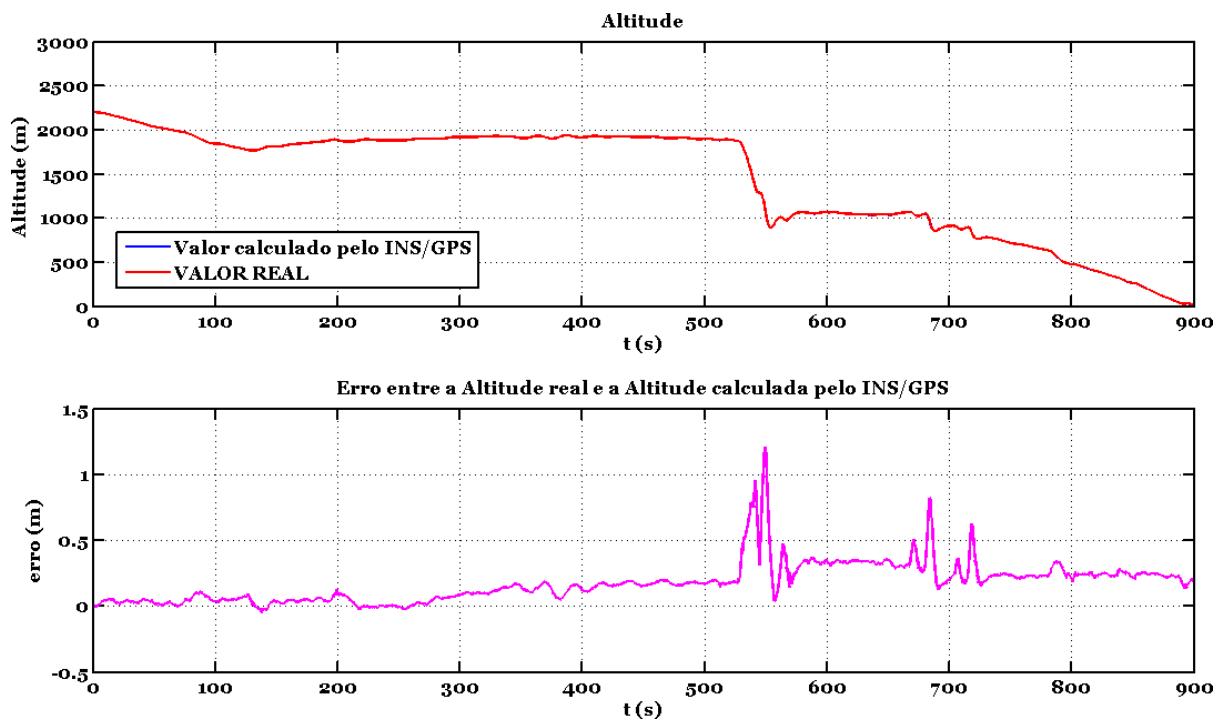


FIG.5.24: Análise da altitude com o INS/GPS

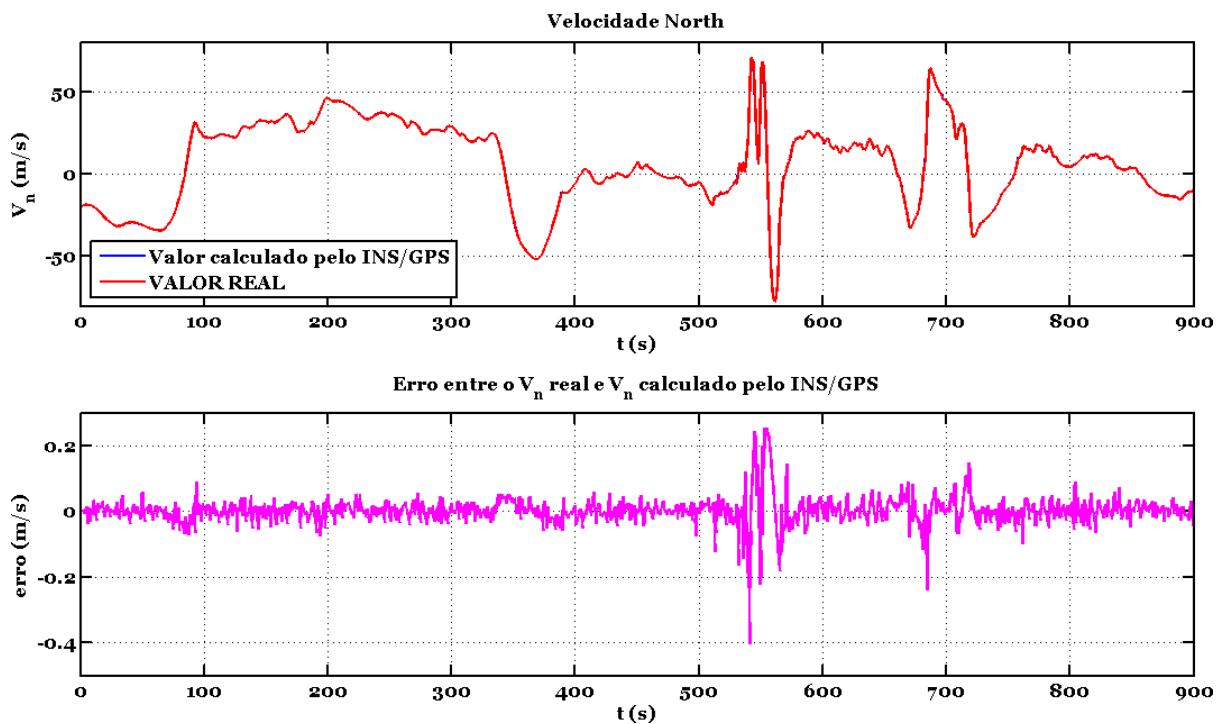


FIG.5.25: Análise da velocidade norte V_n com o INS/GPS

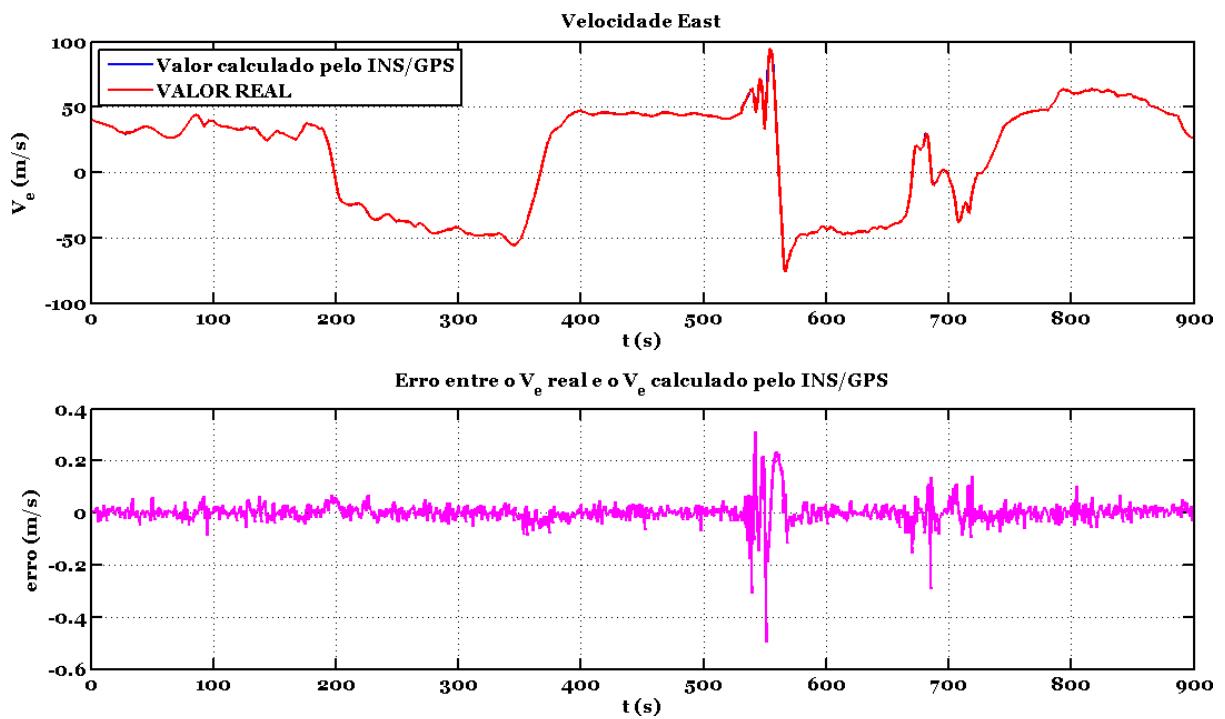


FIG.5.26: Análise da velocidade leste V_e com o INS/GPS

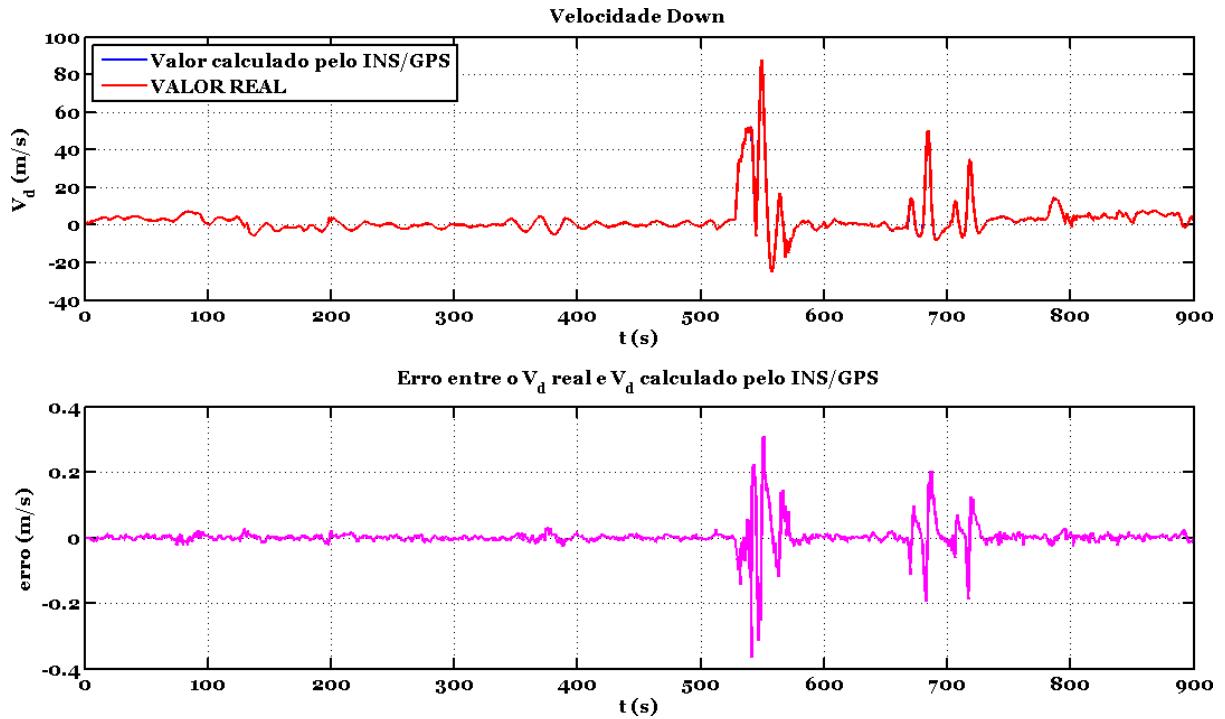


FIG.5.27: Análise da velocidade vertical V_d com o INS/GPS

A análise destes gráficos nos permite concluir que para este nível de ruído dos sensores, característico dos MEMS, a navegação inercial sozinha, isto é, sem o auxílio do GPS, não retorna os dados com a exatidão necessária por tempo suficiente, como é mostrado nos gráficos das figuras 5.10 a 5.18. Com o auxílio do GPS vemos que este erro se mantém praticamente constante e próximo de zero ao longo de todo o percurso, como é observado nas figuras 5.19 a 5.27.

A trajetória descrita pela aeronave pode ser melhor visualizada se plotarmos o par Latitude e Longitude numa ferramenta como o *Google Earth*. O Matlab possui o comando "*kmlwrite*", onde de posse dos vetores latitude e longitude, constrói-se um arquivo compatível com o *Google Earth*, estes dados do arquivo são então lidos pelo *Google Earth* e plotado através do comando "*winopen*". A figura 5.28 mostra estes dados em vermelho calculados a partir do INS/GPS exatamente na região onde foi feito o vôo no *FlightGear*:

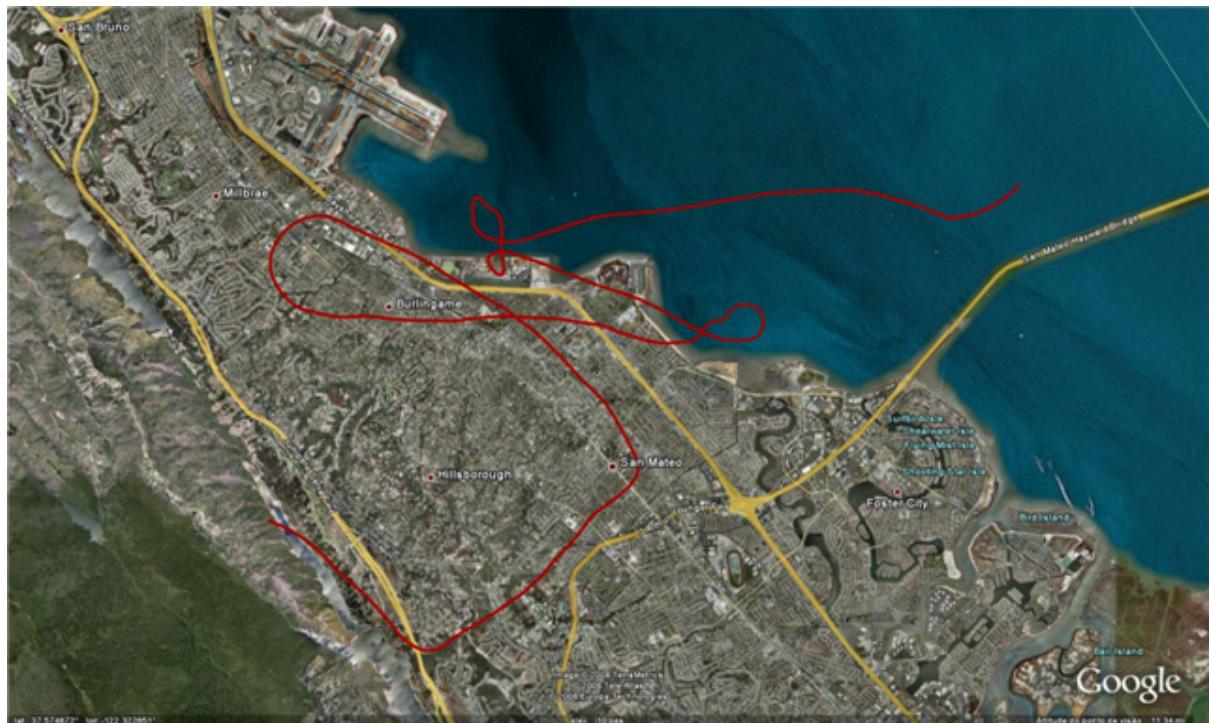


FIG.5.28: Trajetória do vôo feito no *FlightGear* plotada no *Google Earth* a partir dos dados calculados pelo INS/GPS

6 RESULTADOS PRÁTICOS

A seguir serão mostrados os experimentos práticos realizados com sensores reais e o algoritmo de navegação implementado em tempo real num PC104. Primeiramente será mostrado o experimento realizado em um simulador de movimentos de 2 eixos da Contraves, e em seguida os resultados de navegação de uma trajetória feita por uma viatura tipo kombi.

6.1 SIMULADOR DE MOVIMENTOS

O simulador de movimentos utilizado para testes foi uma mesa da Contraves situada no Laboratório de Sistemas Inerciais (LabSIn) do Instituto de Pesquisas da Marinha (figura 6.1).

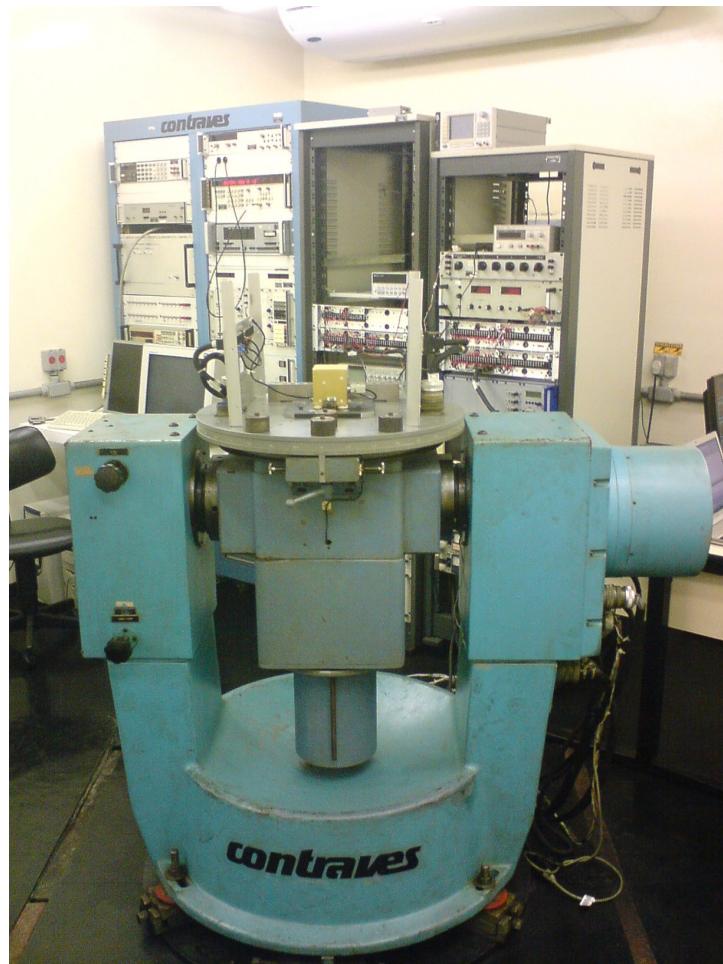


FIG.6.1: Simulador de movimentos de 2 eixos Contraves

Sobre o simulador de movimentos pode-se observar a IMU utilizada, uma IMU400CD-100 da *CrossBow* do tipo MEMS. Este modelo proporciona interfaces de saídas analógicas (12-bit DAC) e digital (RS-232). A aquisição dos dados foi feita através da porta de comunicação serial RS-232, com um protocolo proprietário da *Crossbow*, onde a temperatura já está compensada (calibração de fábrica, armazenada em EPROM interna). Sua taxa máxima configurável de amostragem é de 133Hz, embora neste trabalho foi utilizada uma taxa de amostragem de 20Hz. O tempo necessário na inicialização para se obter dados válidos é menor que 1 segundo, embora se recomende um "warm-up" mínimo de 30s. A Tabela 6.1 fornece as principais especificações da IMU (IMU400-DATASHEET).

TAB.6.1: Especificações da IMU400CD-200 da CrossBow

| Características do Giro | Valor | Características do Acelerômetro | Valor |
|--|-------------|---------------------------------|------------|
| Range Roll, Pitch, Head ($^{\circ}/sec$) | ± 200 | Range X, Y, Z (g) | ± 4 |
| Bias Roll, Pitch, Head ($^{\circ}/sec$) | $< \pm 1.0$ | Bias X, Y, Z (mg) | $< \pm 12$ |
| Scale Factor Accuracy (%) | < 1 | Factor Accuracy (%) | < 1 |
| Non-Linearity (% FS) | < 0.3 | Non-Linearity (% FS) | < 1 |
| Resolution ($^{\circ}/sec$) | < 0.05 | Resolution (mg) | < 0.6 |
| Bandwidth (Hz) | > 25 | Bandwidth (Hz) | > 75 |
| Random Walk ($^{\circ}/hr^{1/2}$) | < 4.5 | Random Walk (m/s/hr $^{1/2}$) | < 1 |

A figura 6.2 mostra o PC104 utilizado para processar o algoritmo de navegação Inercial com o *xPC Target*. A configuração deste PC é um Pentium II 400MHz com 256MB de memória RAM e 40GB de HD.

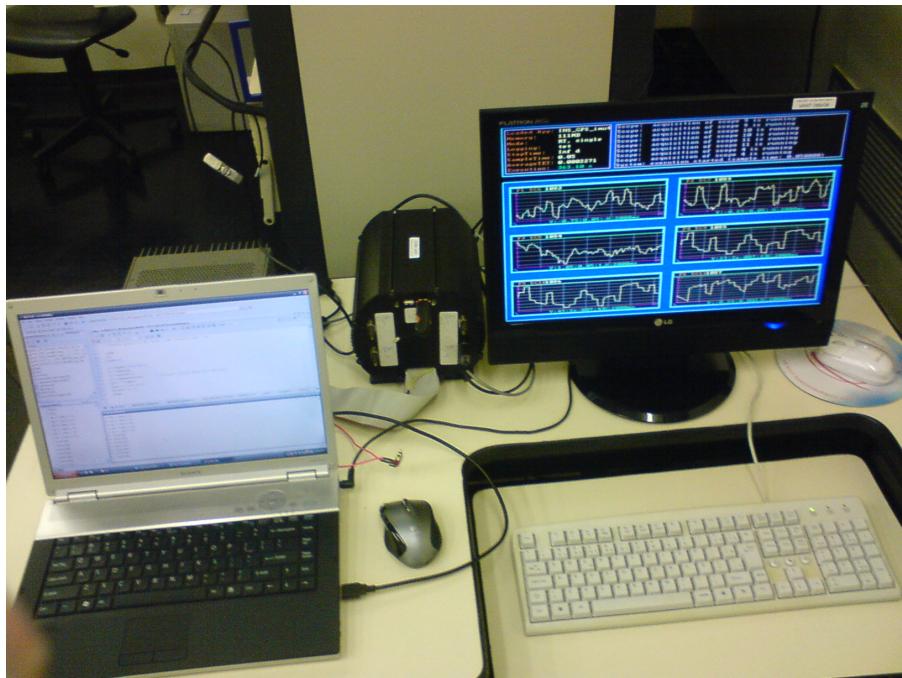


FIG.6.2: PC104 utilizado com o *xPC Target*

Foi realizado o seguinte experimento com a mesa: Os dois eixos de liberdade da mesa foram escolhidos como os eixos de rotação em *roll* e *head*. O eixo de *pitch* portanto é simulado sem movimento. Não foi utilizado GPS real, por isso, foi simulado no algoritmo um GPS com a mesma magnitude de erro dado pela tabela 3.1 e com os seguintes valores fixos para a latitude, longitude, altitude, V_n , V_e e V_d respectivamente: $[23\ 43\ 0\ 0\ 0\ 0]^T$. A taxa de amostragem utilizada para a leitura do GPS simulado foi de 5Hz (0.2 segundos) que é a máxima taxa de amostragem permitida para o GPS 18-5Hz *Garmin*. Também não foi utilizado magnetômetro, portanto, o Head inicial foi inicializado manualmente no algoritmo em 90° assim como na própria mesa.

Inicialmente, fixamos a mesa em um valor de -15° para o *roll*, e 90° para o *head*. Como a mesa não possui grau de liberdade em *pitch*, este valor mantém-se fixo em 0° . Em seguida executamos o algoritmo INS/GPS no PC104 com o *xPC Target*.

Quando o algoritmo é iniciado, a primeira fase do processamento é o algoritmo de inicialização. Mais especificamente, foi implementado apenas o alinhamento grosso descrito na seção 4.3.2 com uma leitura de 1200 amostras da IMU. Como a taxa de amostragem lida da IMU foi de 20Hz (0.05 segundos) o tempo de alinhamento da IMU é de 60 segundos. Após este tempo é verificada a resposta do algoritmo. Confirmando os -15° para o Roll, e 90° para o Head, iniciamos o seguinte movimento da mesa: com o eixo de Head sempre fixo, posicionamos sucessivamente, após certos intervalos de tempo, o ângulo de Roll em 0° , -30° , -60° e -90° . Os parâmetros de navegação calculados pelo algoritmo podem então ser observados nas figuras 6.3 à 6.11, enquanto que as figuras 6.12 a 6.17 mostram os valores lidos diretamente da IMU:

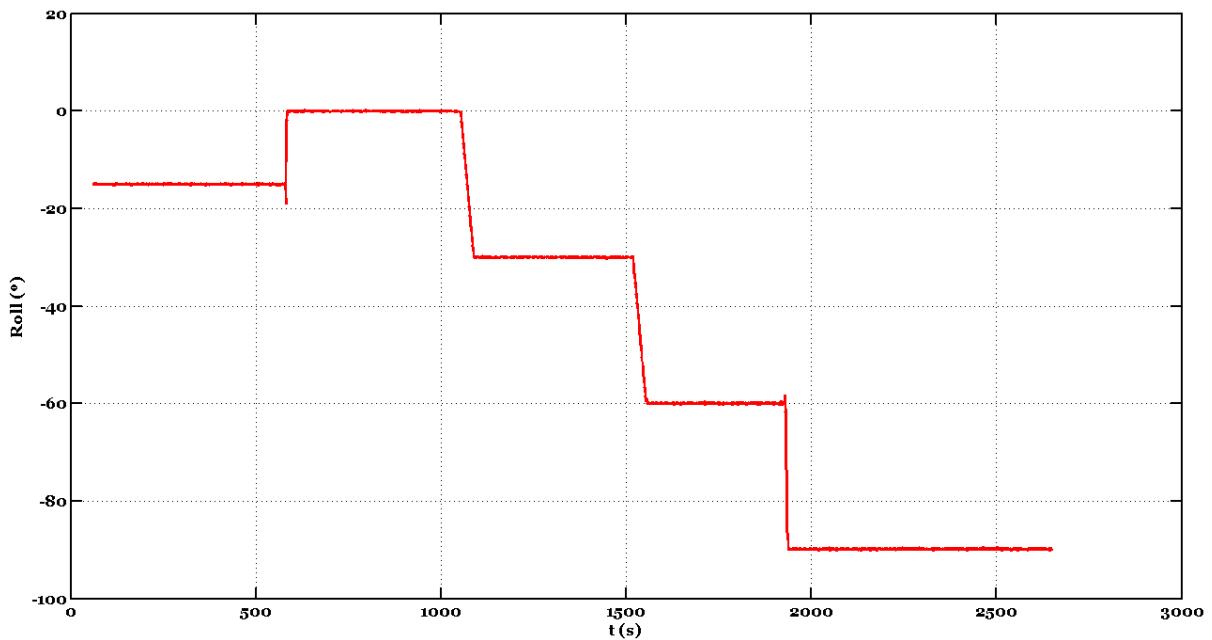


FIG.6.3: Análise do ângulo de *roll* gerado pelo simulador de movimentos

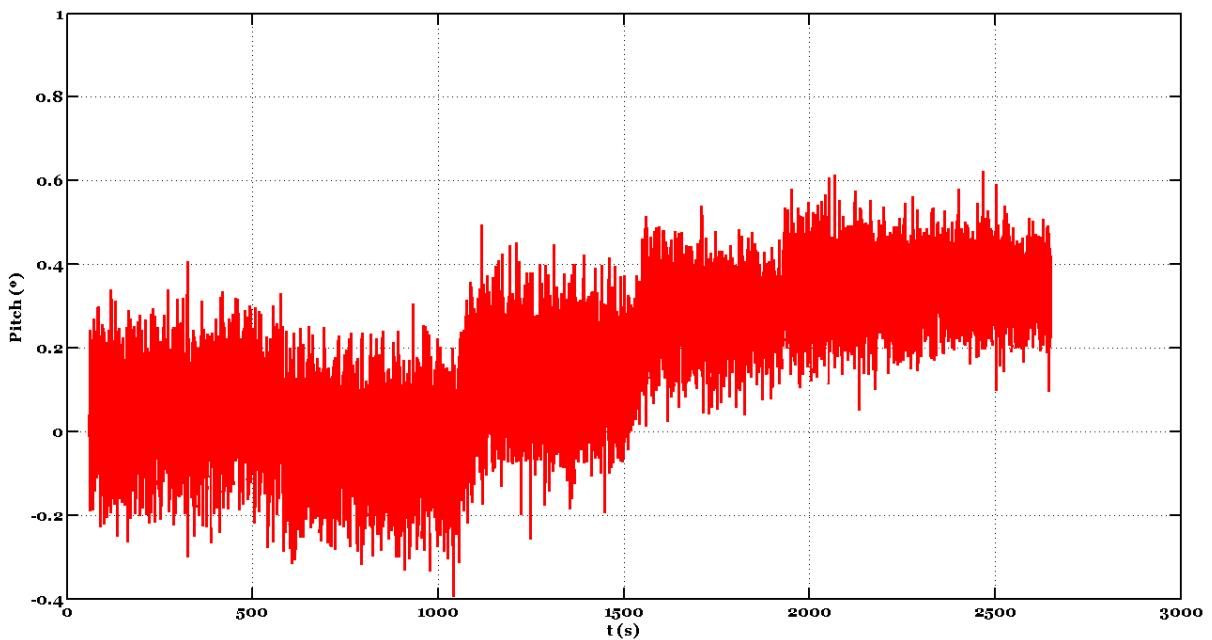


FIG.6.4: Análise do ângulo de *pitch* gerado pelo simulador de movimentos

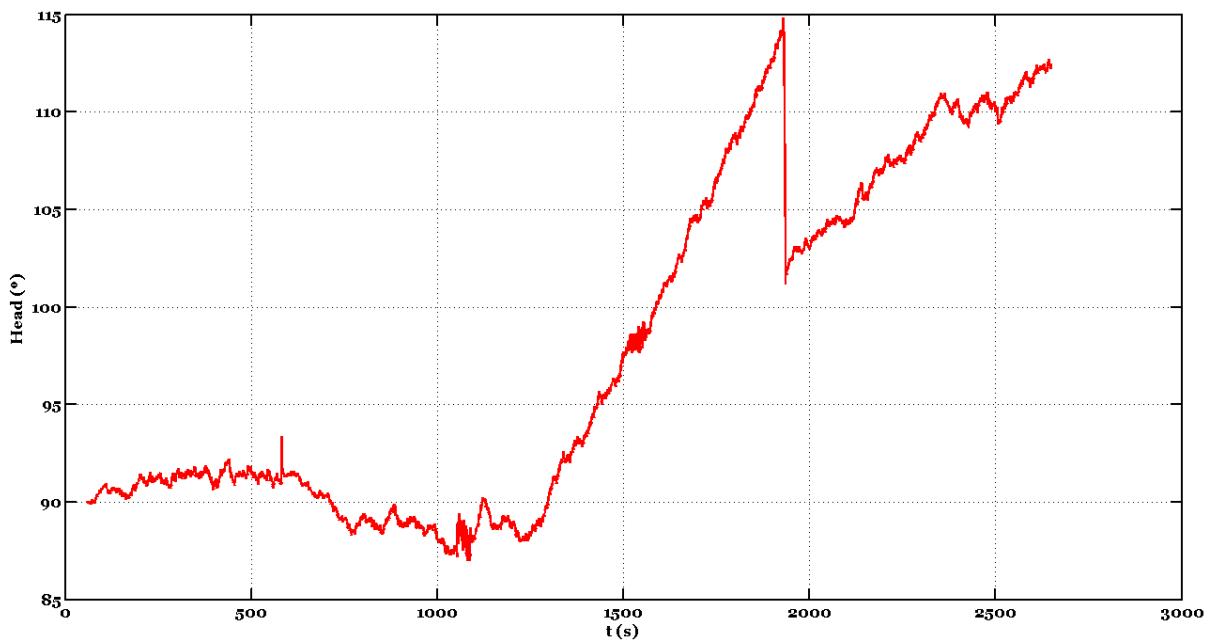


FIG.6.5: Análise do ângulo de *head* gerado pelo simulador de movimentos

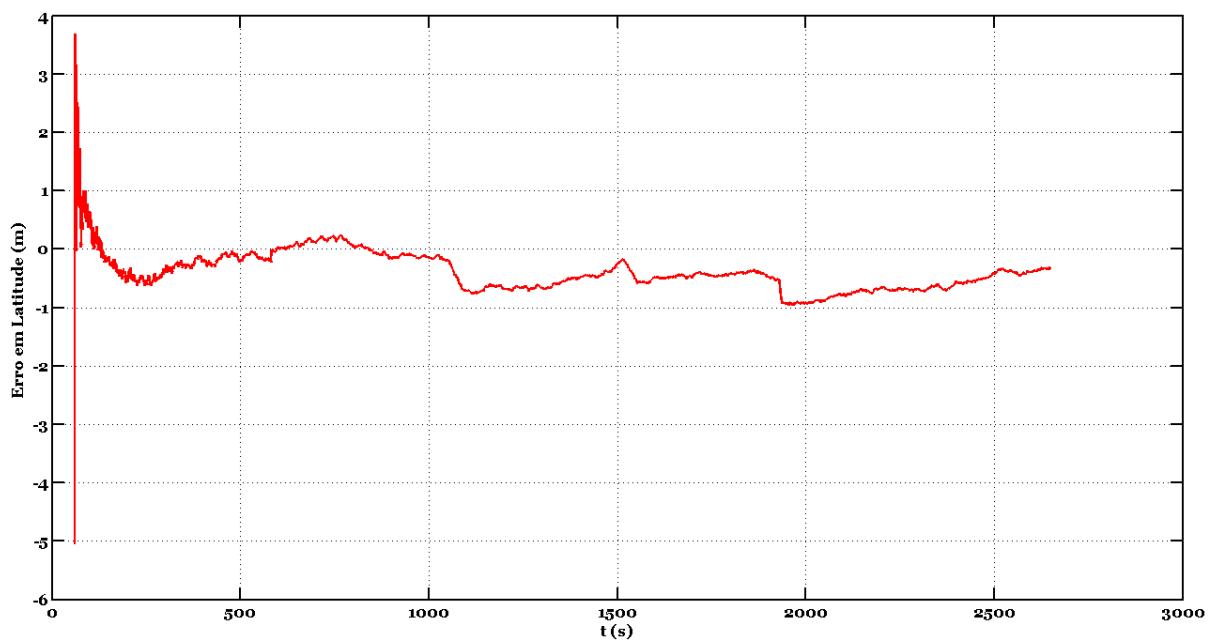


FIG.6.6: Análise do erro em latitude gerado pelo simulador de movimentos

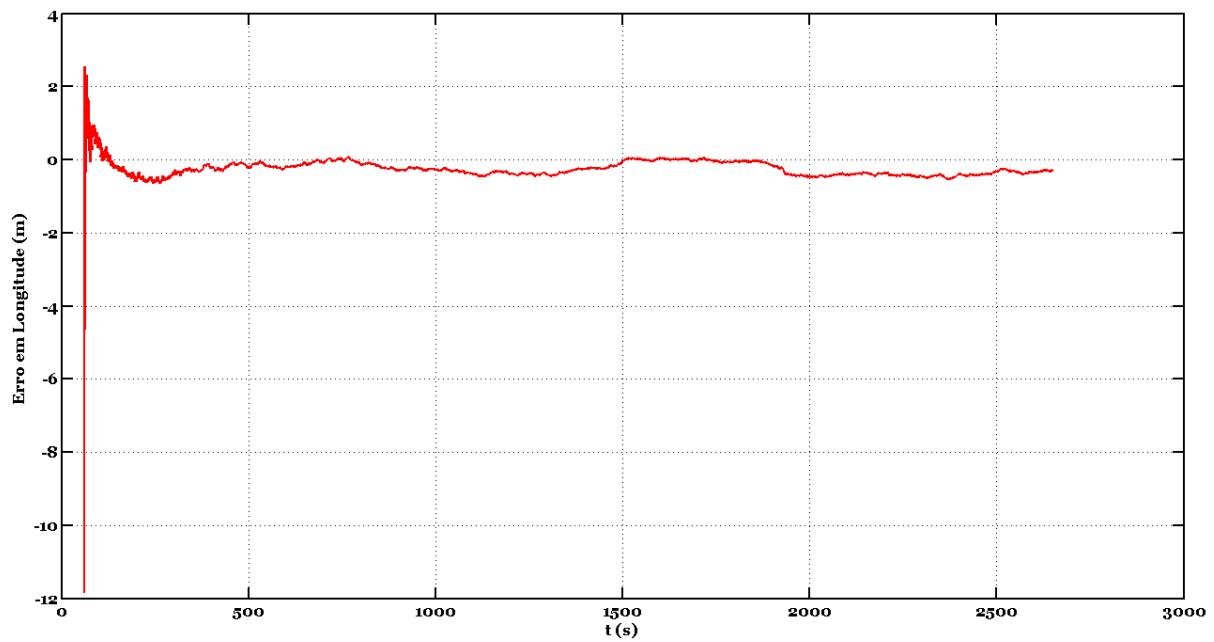


FIG.6.7: Análise do erro em longitude gerado pelo simulador de movimentos

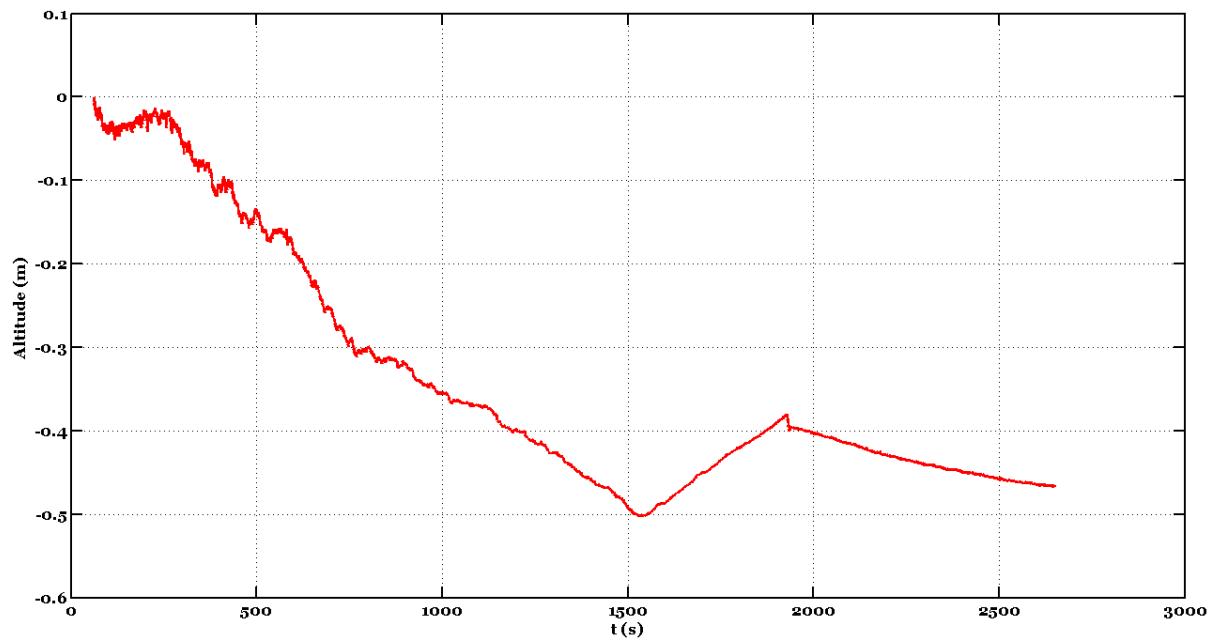


FIG.6.8: Análise da altitude gerada pelo simulador de movimentos

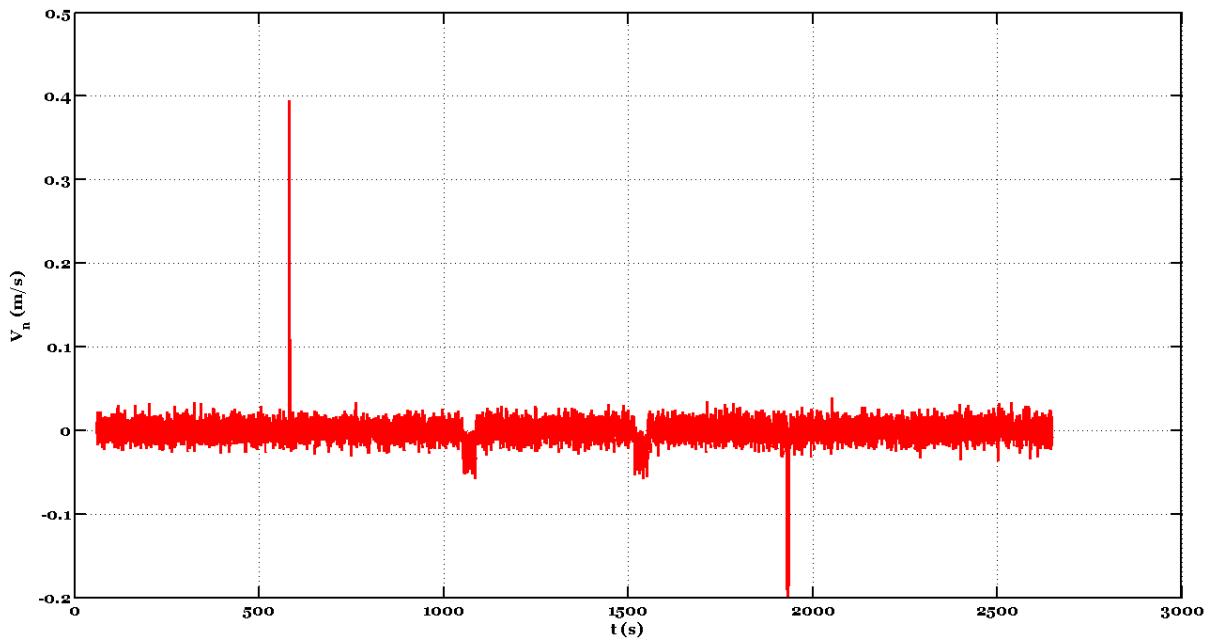


FIG.6.9: Análise da velocidade norte V_n gerada pelo simulador de movimentos

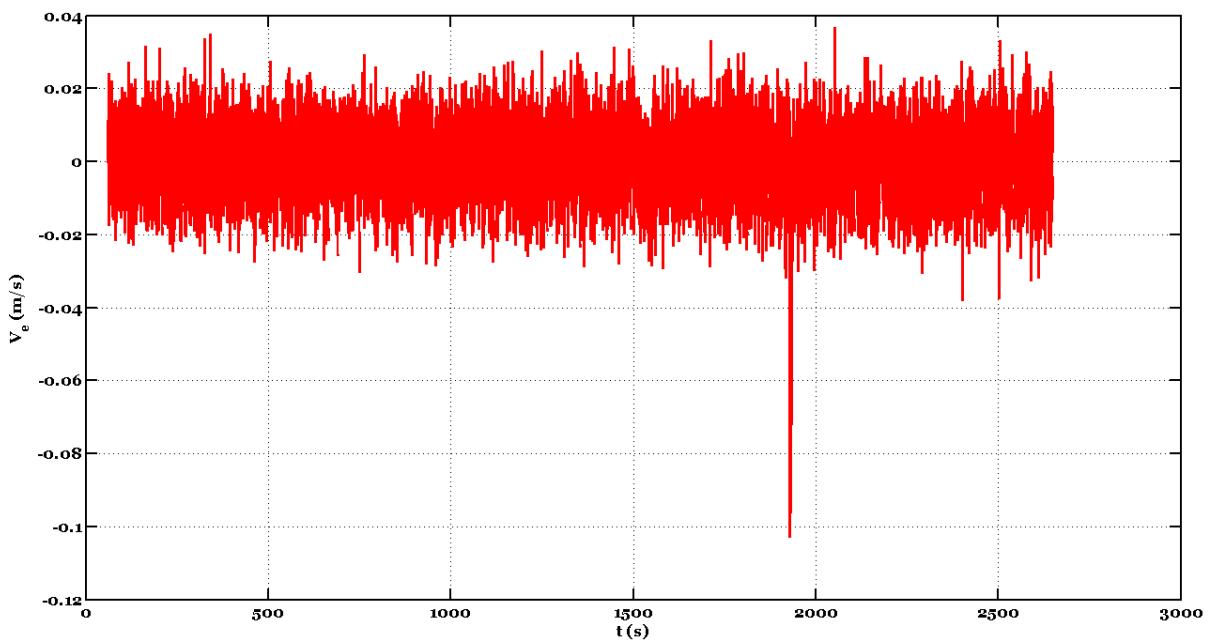


FIG.6.10: Análise da velocidade leste V_e gerada pelo simulador de movimentos

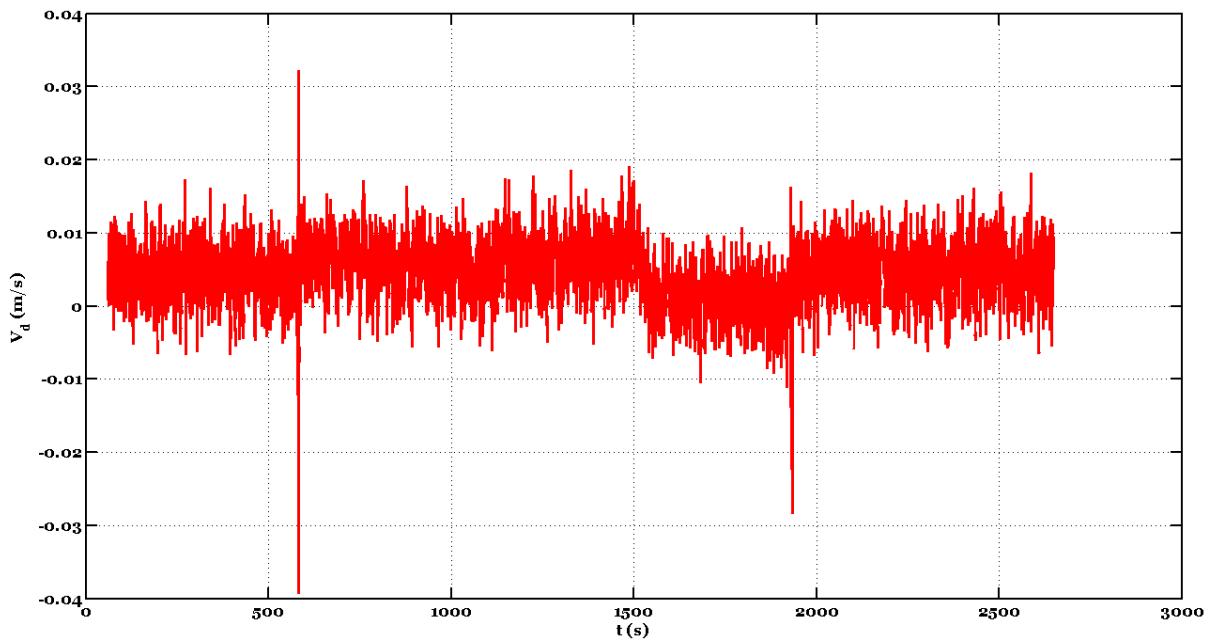


FIG.6.11: Análise da velocidade vertical V_d gerada pelo simulador de movimentos

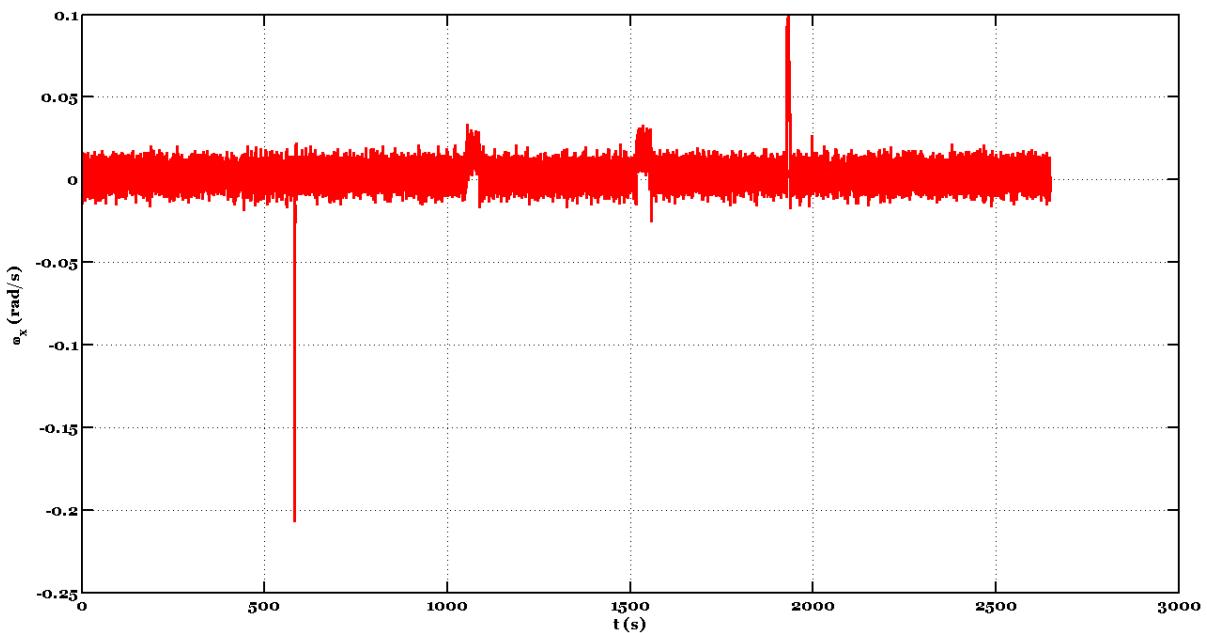


FIG.6.12: Análise da velocidade angular ω_x lida da IMU

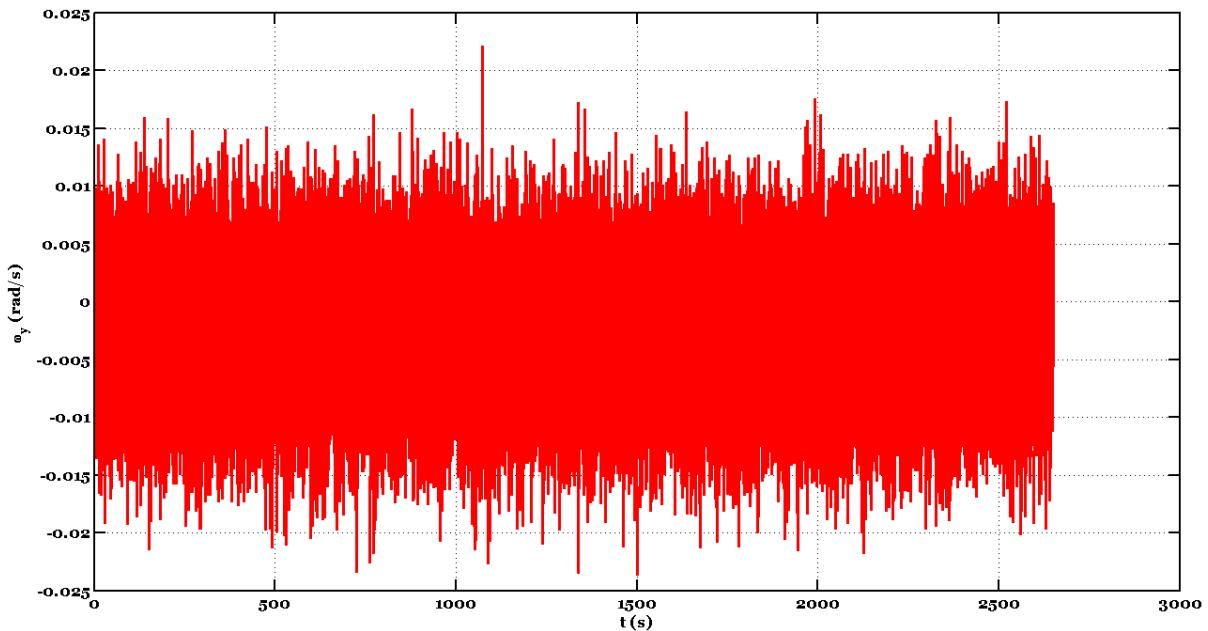


FIG.6.13: Análise da velocidade angular ω_y lida da IMU

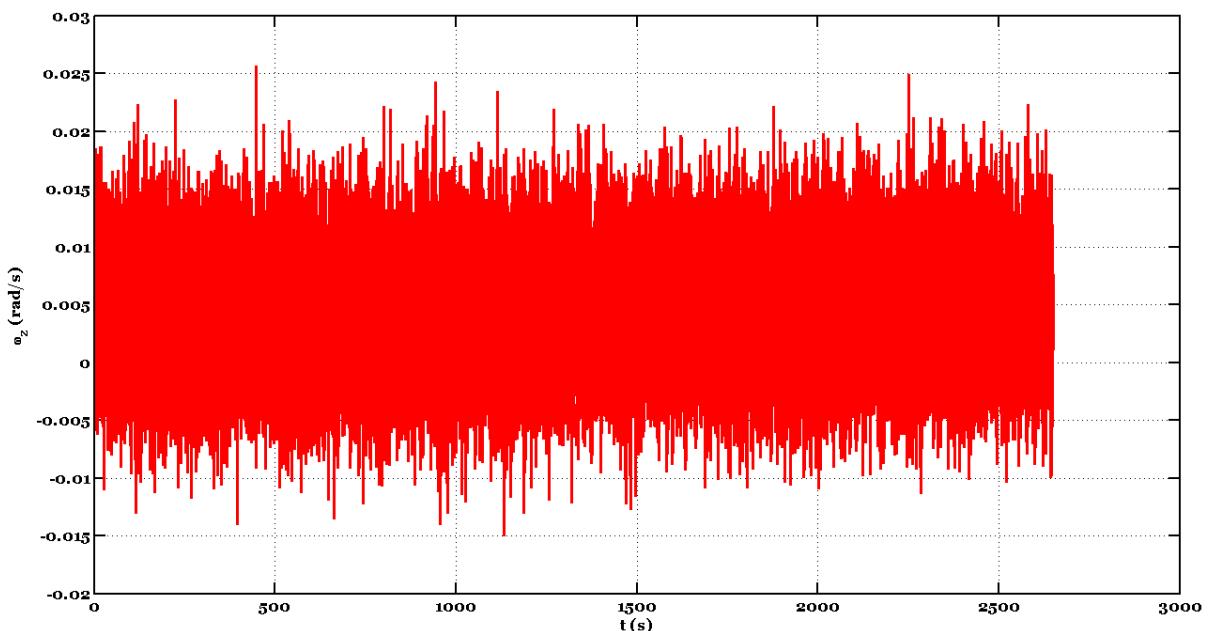


FIG.6.14: Análise da velocidade angular ω_z lida da IMU

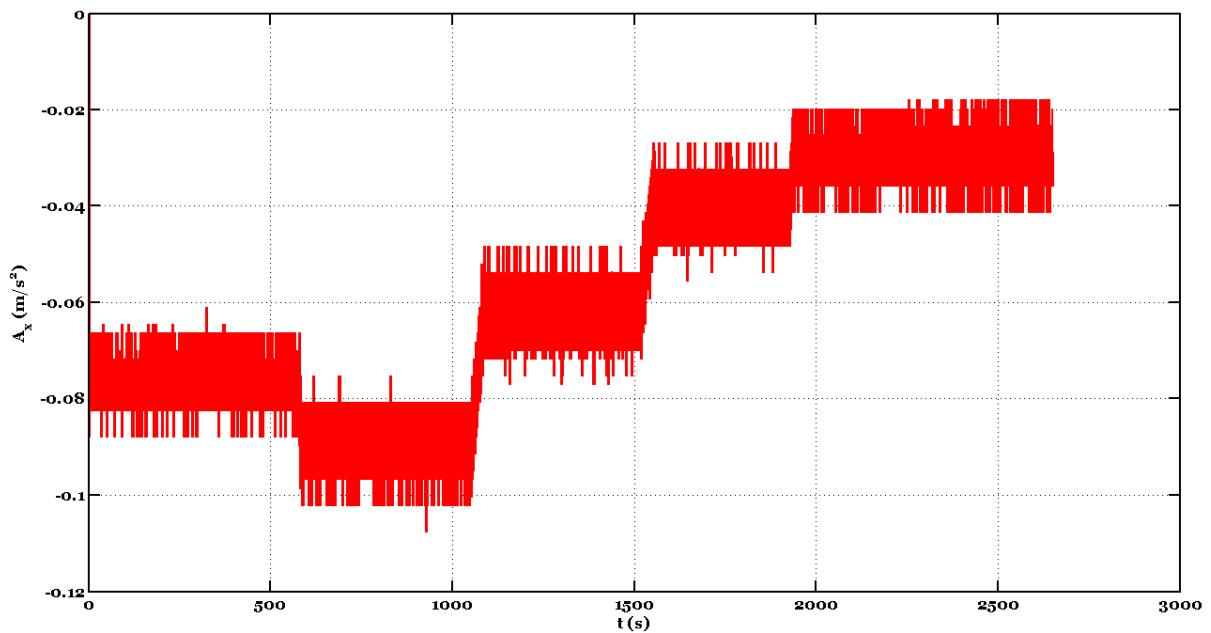


FIG.6.15: Análise da aceleração linear A_x lida da IMU

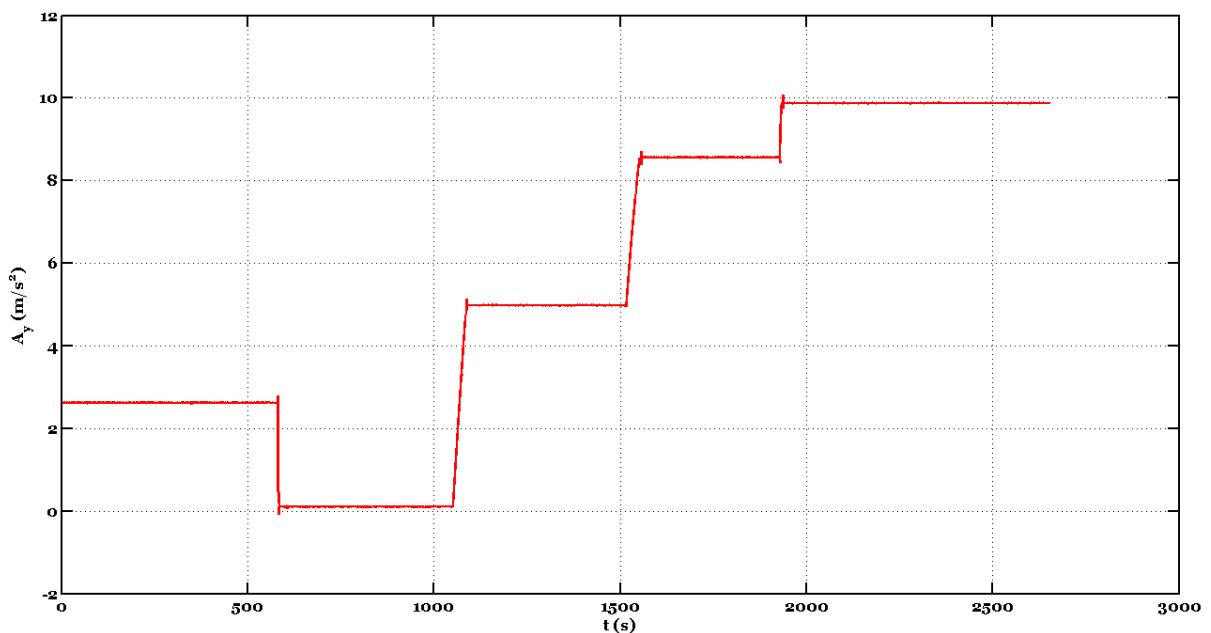


FIG.6.16: Análise da aceleração linear A_y lida da IMU

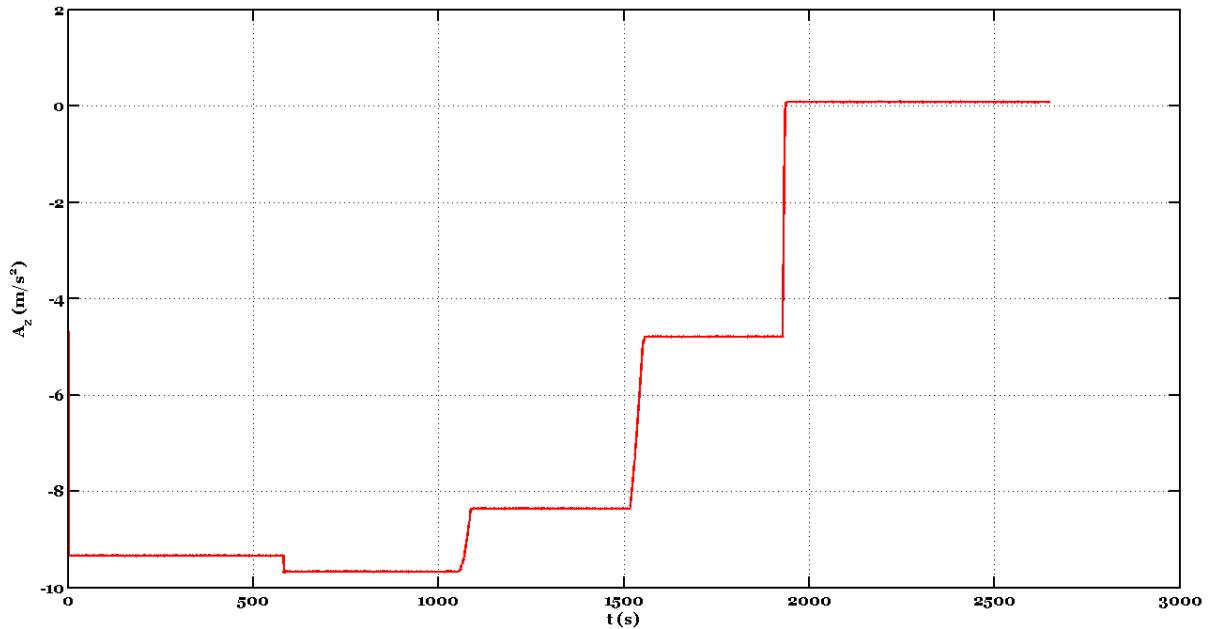


FIG.6.17: Análise da aceleração linear A_z lida da IMU

Pode-se notar pela figura 6.3 que o parâmetro de *roll* calculado pelo algoritmo segue o movimento previsto. O parâmetro de *pitch* no entanto, como pode ser observado na figura 6.4, diverge do valor esperado que deveria ser nulo durante todo o experimento, isto se deve ao desalinhamento entre os eixos da IMU com os eixos de rotação do simulador de movimentos. E ainda, como não foi utilizado nenhuma compensação (calibração) desta IMU específica, ocorrem os acoplamentos observados nas figuras 6.4 e 6.16.

Outro parâmetro que diverge bastante do esperado é o ângulo de *head* (figura 6.5) que deveria permanecer em 90° durante o experimento. No entanto, o parâmetro *head* é uma problemática inerente aos sensores MEMS. Enquanto o valor da gravidade serve como referência para a estimativa dos erros em *roll* e *pitch*, a referência para a estimativa do erro em *head* vem da velocidade angular da terra, como sensores MEMS não lêem este dado, o parâmetro de *head* fica comprometido. É por este motivo que sistemas inerciais que utilizam sensores MEMS devem usar também sensores externos como o magnetômetro. Um outro aspecto que dificulta a estimativa do *head* é a baixa dinâmica deste parâmetro, quanto menor a dinâmica do sistema no ângulo de *head* pior será a estimativa deste.

6.2 TRAJETÓRIA COM A KOMBI

Nesta Seção será mostrado os resultados alcançados com o Sistema de Navegação Inercial montado sobre uma kombi, como pode ser visto na figura 6.18 e mais detalhadamente na figura 6.19.



FIG.6.18: kombi usada no experimento

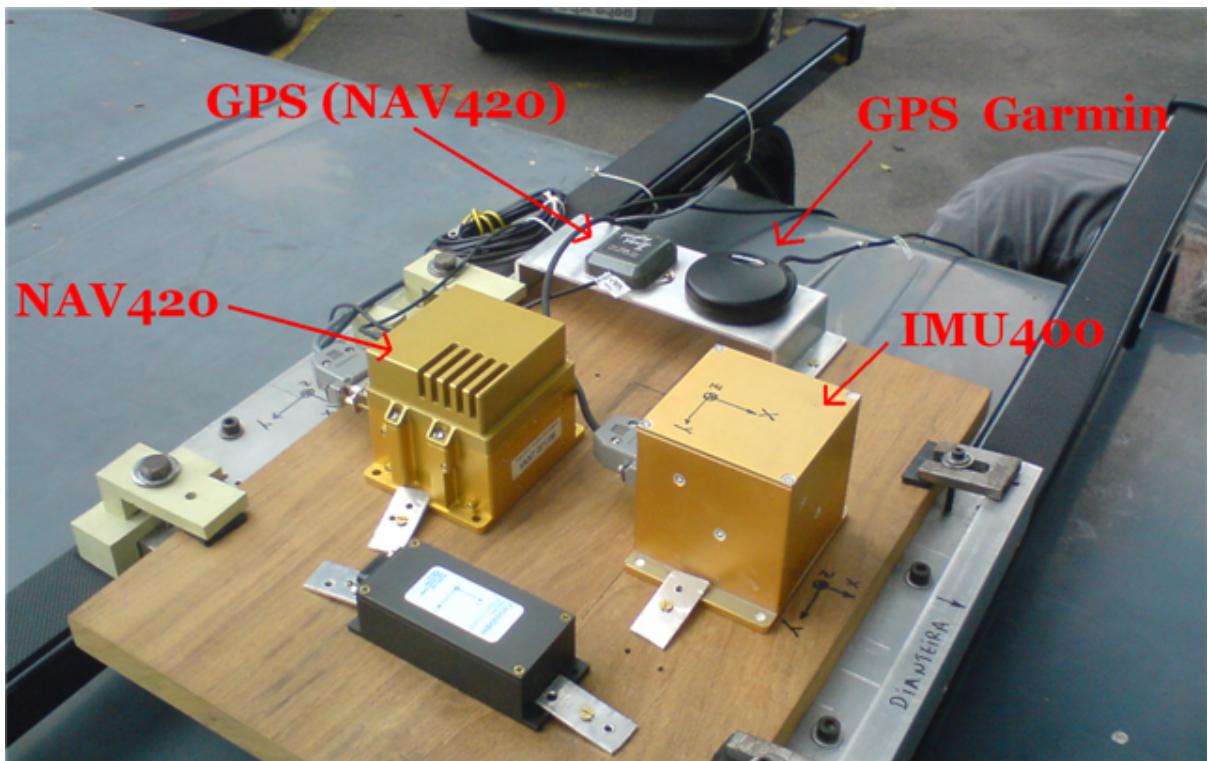


FIG.6.19: Sistema de Navegação Inercial sobre a kombi

O sensor inercial utilizado foi a mesma IMU400CD-200 usada no simulador de movimento.

mentos (tabela 6.1). No entanto, neste experimento, o sensor já foi previamente calibrado com as técnicas descritas na seção 4.2, e sua compensação foi incorporada no algoritmo de navegação. O GPS utilizado foi o GPS 18-5Hz da *Garmin* de interface RS-232 que dá informações de posição (latitude, longitude e altitude), velocidade (V_n , V_e e V_d) e *head*. Suas principais especificações podem ser vistas na tabela 6.2 (GARMIN INTERNATIONAL, 2005). Também pode ser visto na figura 6.19 o NAV420 da *CrossBow* juntamente com seu GPS (NAV420-DATASHEET). o NAV420 é um Sistema de Navegação Inercial com GPS e magnetômetro integrado baseado em sensores MEMS, proprietário da *CrossBow* e de código fechado, e ele será usado para compararmos os parâmetros de navegação calculados por este com os calculados pelo nosso. As principais especificações do NAV420 podem ser vistas na tabela 6.3

TAB.6.2: Especificações do GPS 18-5Hz Garmin

| Características do GPS | Valor |
|--------------------------|---|
| Receiver | WAAS-enabled; 12 parallel channel GPS receiver continuously tracks and uses up to 12 satellites to compute and update your position |
| GPS position accuracy | < 15 meters, 95% typical |
| GPS velocity accuracy | 0.1 knot RMS steady state |
| Dynamics | 999 knots, 6 g's |
| Measurement Pulse Output | 5 Hz pulse, programmable width, 1 microsecond accuracy |
| Operating temperature | 30°C to 80°C |

TAB.6.3: Especificações do NAV420 da CrossBow

| Características do NAV420 | Valor |
|---------------------------------|------------------------------------|
| Performance | |
| Update Rate (Hz) | 2-100 (Programmable) |
| Position/Velocity | |
| Position Accuracy (m) | 3 (Internal GPS, not augmented) |
| X,Y Velocity Accuracy (m/s rms) | < 0.4 (GPS available) |
| Z Velocity Accuracy (m/s rms) | < 0.5 (GPS available) |
| Attitude | |
| Range: Roll, Pitch (°) | ± 180, ± 90 |
| Accuracy (° rms) | < 0.75 (GPS available) |
| Accuracy (° rms) | < 2.5 (GPS unavailable) |
| Heading | |
| Range (°) | ± 180 |
| Accuracy (° rms) | < 3.0 |
| Angular Rate | |
| Range: Roll, Pitch, Yaw (°/sec) | ± 200 |
| Bias: Roll, Pitch, Yaw (°/sec) | < ± 0.1 (Kalman filter stabilized) |
| Bias: Roll, Pitch, Yaw (°/sec) | < ± 0.75 (Kalman filter off) |
| Acceleration | |
| Input Range: X/Y/Z (g) | ± 4 |
| Bias: X/Y/Z (mg) | < ± 15 |

Neste experimento também não foi utilizado o magnetômetro para o cálculo mais preciso do *head*, para isso, foi utilizado o *head* dado pelo GPS, ou seja, ao invés de compensar este parâmetro com a estimativa do Filtro de *Kalman*, compensa-se diretamente com o sinal de *head* do GPS. O *head* inicial foi inserido manualmente no algoritmo, já que a informação de head do GPS só é disponibilizada quando o veículo está em movimento, e na inicialização (alinhamento) o veículo deve estar parado. O PC104 foi embarcado dentro da viatura, assim como um laptop conectado à uma de suas portas seriais que serviu como IHM (Interface Homem Máquina) do processo.

A trajetória percorrida pela kombi pode ser vista na figura 6.20. Para podermos comparar os parâmetros de navegação calculados pelo algoritmo com resultados concretos, adicionamos marcos durante o trajeto sinalizados por cones onde já eram conhecidas as posições de latitude, longitude e altitude. Estes dados foram levantados pela equipe do IpQM com um DGPS cuja precisão possui erro de centímetros. Com as informações de latitude e longitude dos cones, estes marcos foram plotados no *Google Earth* através do Matlab com os comandos "*kmlwrite*" e "*winopen*" e são os representados por um losango na figura 6.20.

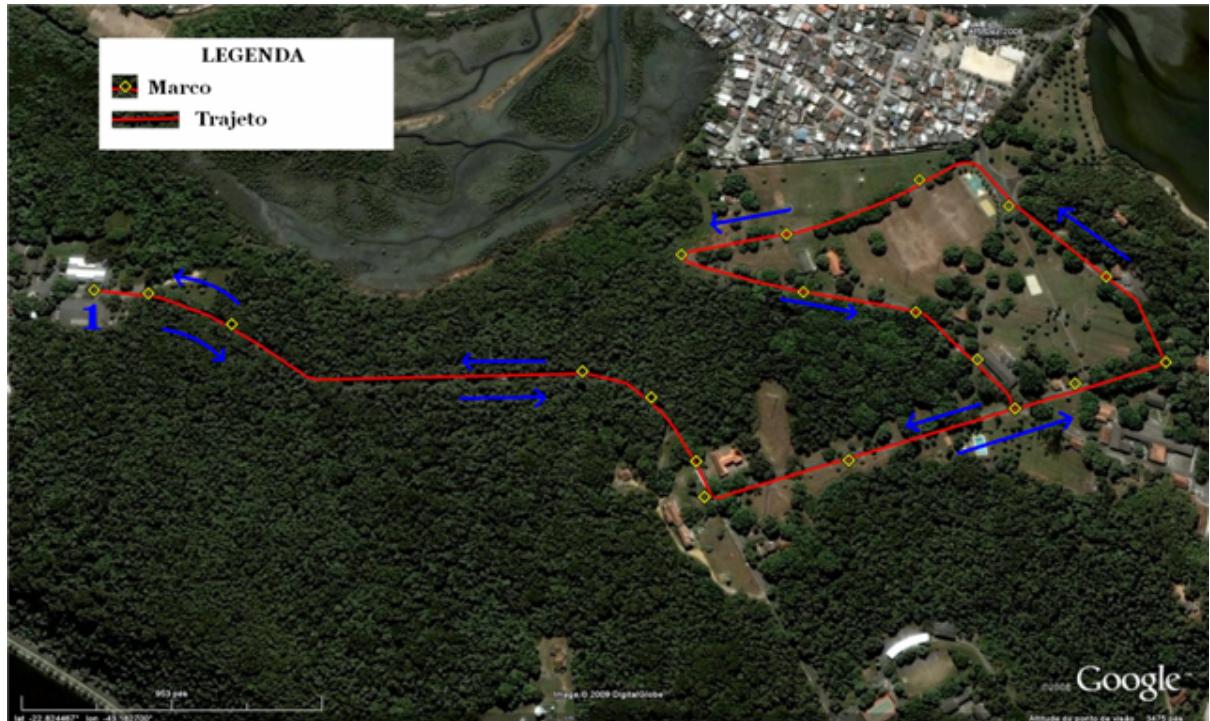


FIG.6.20: Trajetória percorrida pela kombi

A kombi fez o seguinte trajeto: posicionou-se no marco 1 (losango 1 da figura 6.20). Com o veículo desligado, ligou-se os sensores (IMU400, NAV420 e GPS *Garmin*) e logo após ligou-se o PC104, o qual iniciou o algoritmo. Quando o algoritmo passou do alinhamento para a navegação, o que pode ser verificado através da IHM, ligou-se a kombi e a navegação começou fazendo a trajetória descrita pelas setas da figura 6.20. Ao passar pelos marcos anotava-se o tempo exato de passagem por estes. Os tempos associados aos marcos são necessários para se comparar com a posição calculada pelo algoritmo associada aos mesmos tempos anotados.

Durante todo o trajeto foram gravados os parâmetros de navegação calculados pelo algoritmo e os calculados pelo NAV420. Também foram gravados os dados da IMU400 e do GPS *Garmin* para depois serem analisados e pós-processados.

As figuras 6.21 a 6.26 mostram os dados gravados da IMU400 durante a trajetória. Vale ser ressaltado para a análise destes gráficos que o alinhamento durou 170 segundos e o instante em que se ligou o carro foi em aproximadamente 260 segundos.

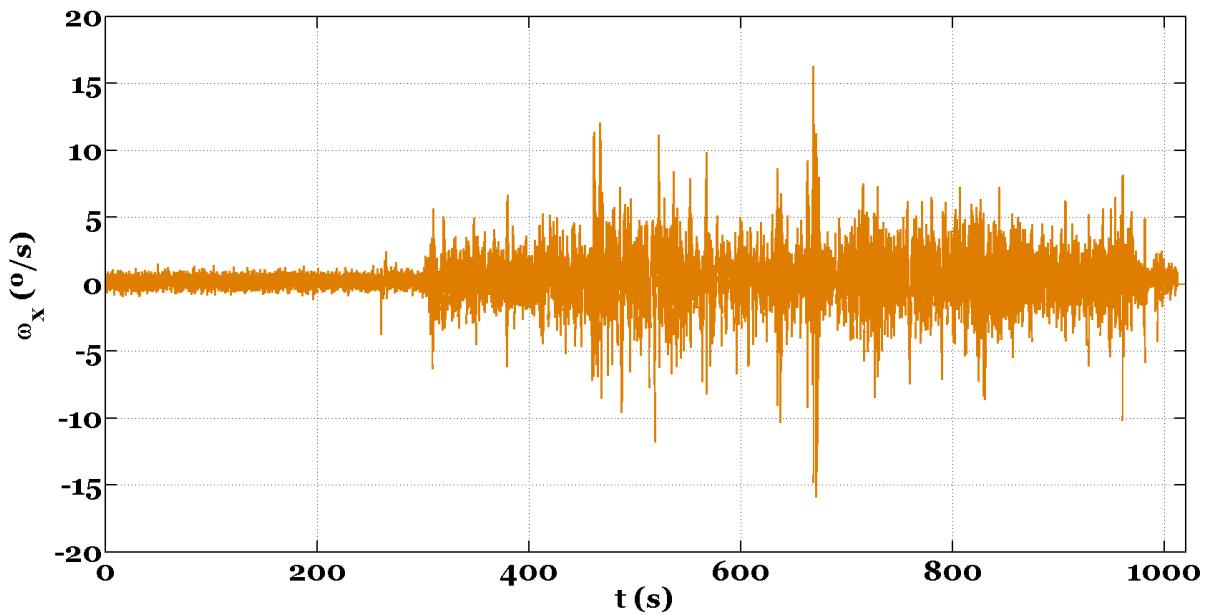


FIG.6.21: Análise da velocidade angular ω_x lida da IMU (kombi)

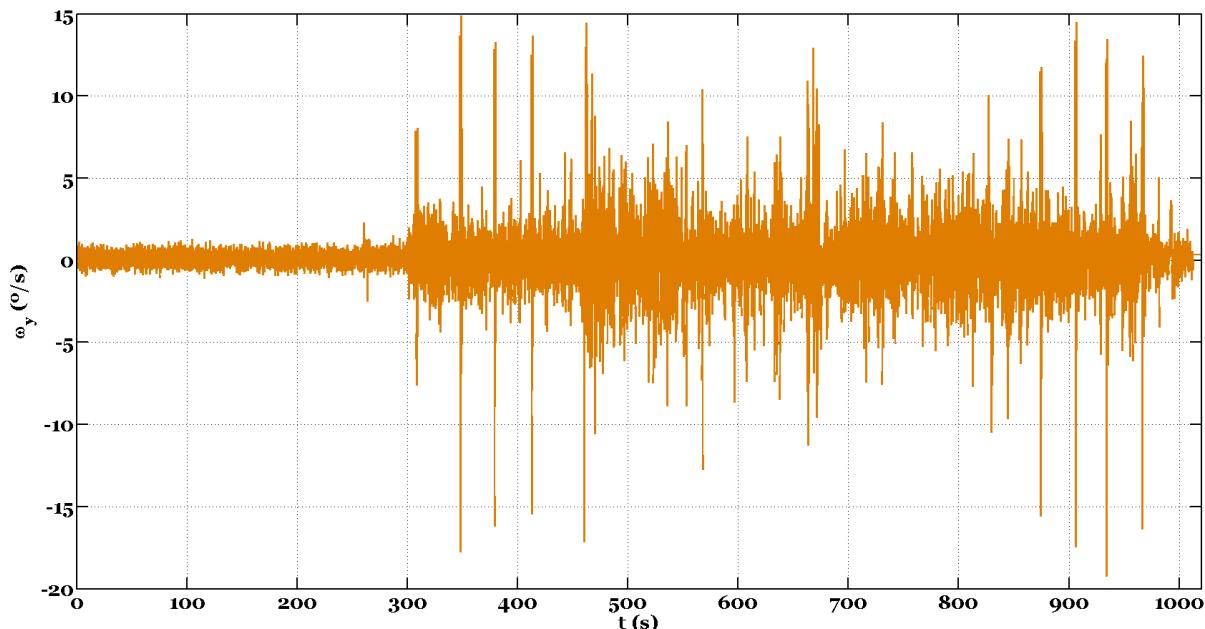


FIG.6.22: Análise da velocidade angular ω_y lida da IMU (kombi)

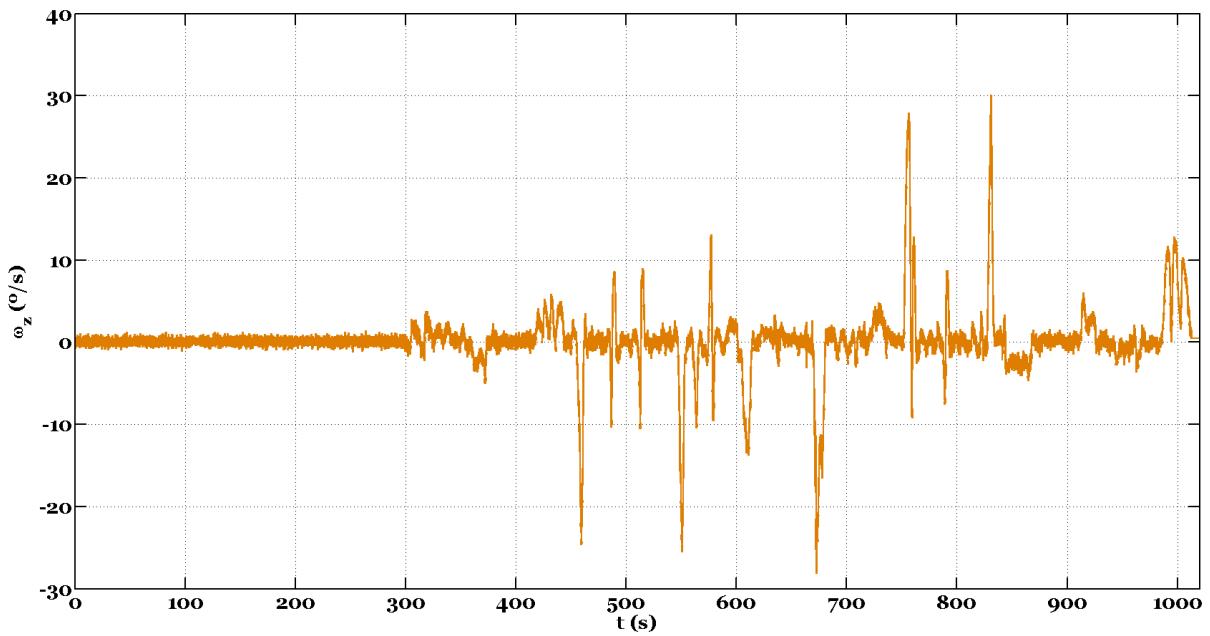


FIG.6.23: Análise da velocidade angular ω_z lida da IMU (kombi)

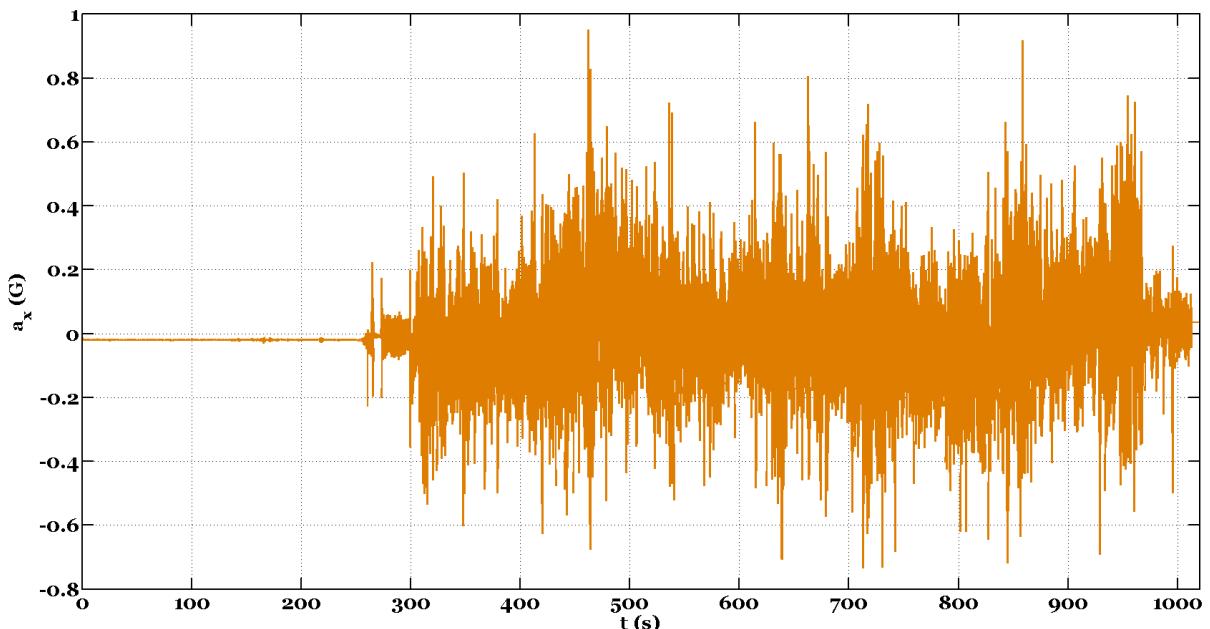


FIG.6.24: Análise da aceleração linear A_x lida da IMU (kombi)

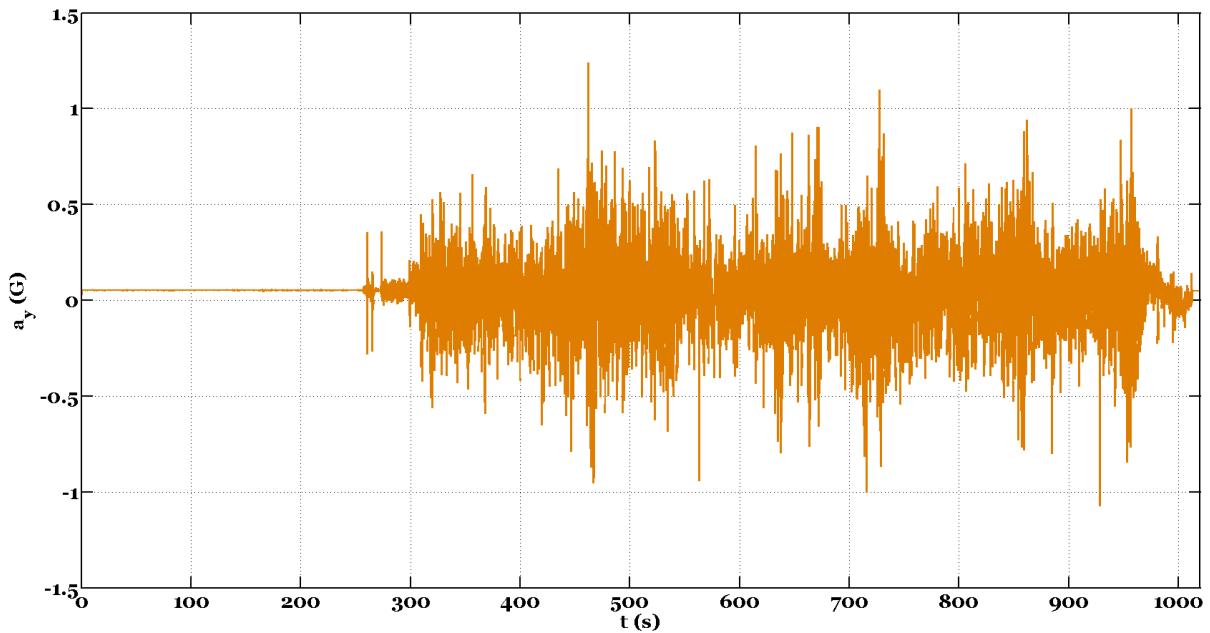


FIG.6.25: Análise da aceleração linear A_y lida da IMU (kombi)

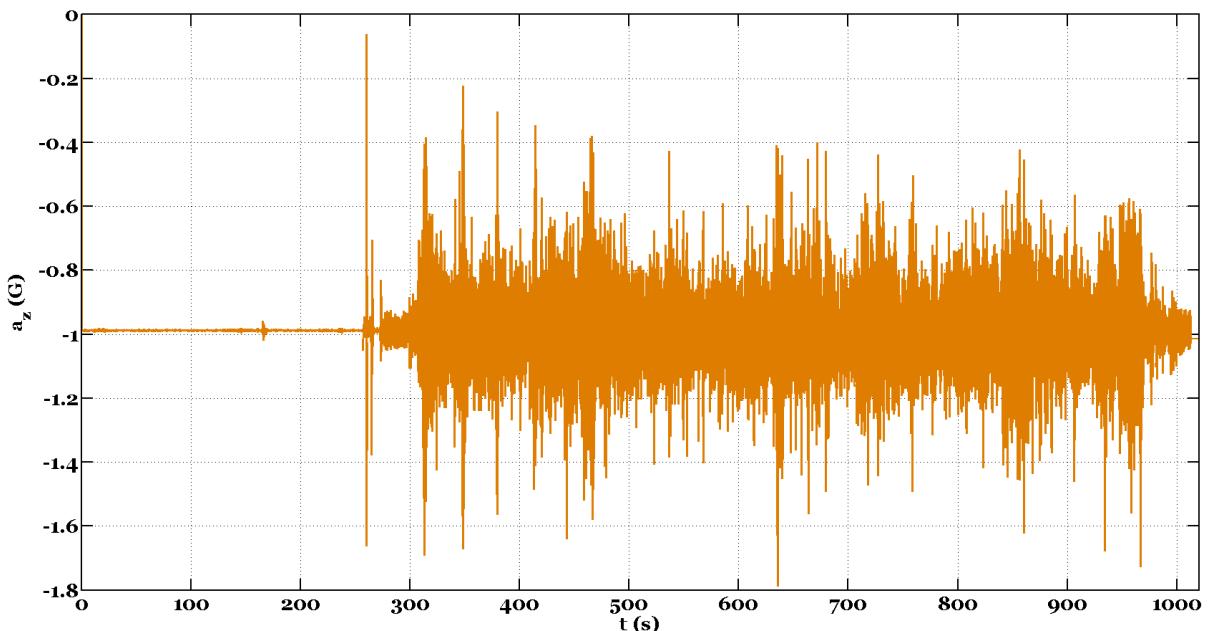


FIG.6.26: Análise da aceleração linear A_z lida da IMU (kombi)

A seguir, são mostradas nas figuras 6.27 a 6.33 os dados gravados do GPS:

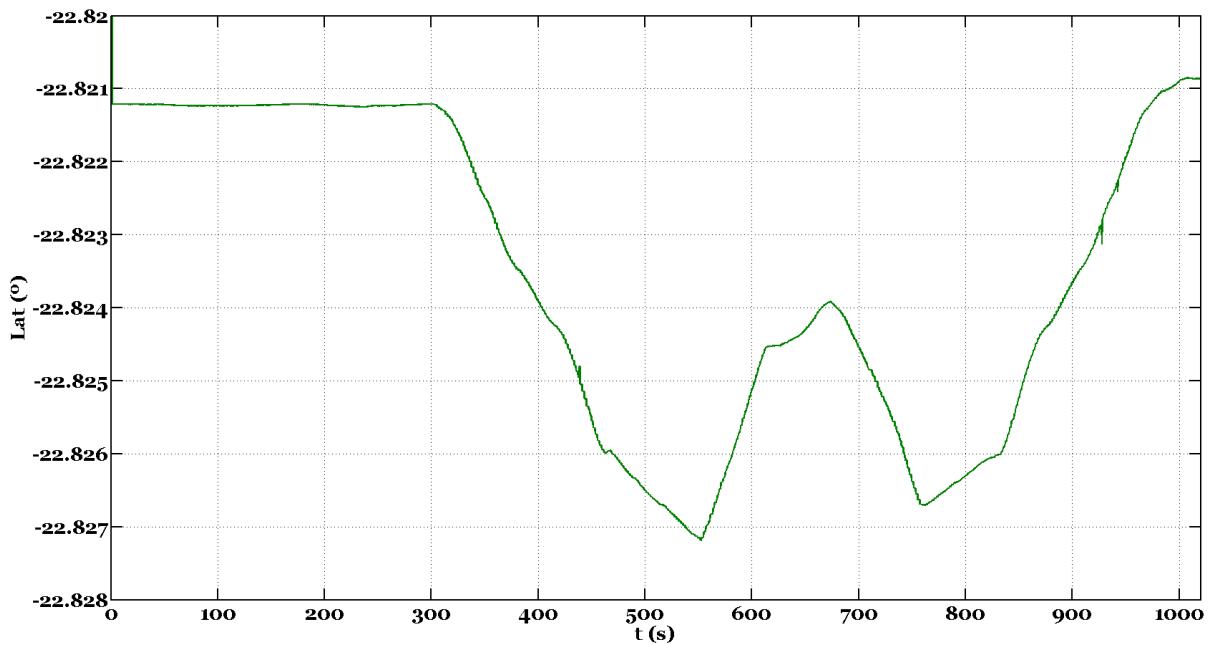


FIG.6.27: Análise da latitude lida pelo GPS 18-5Hz *Garmin*

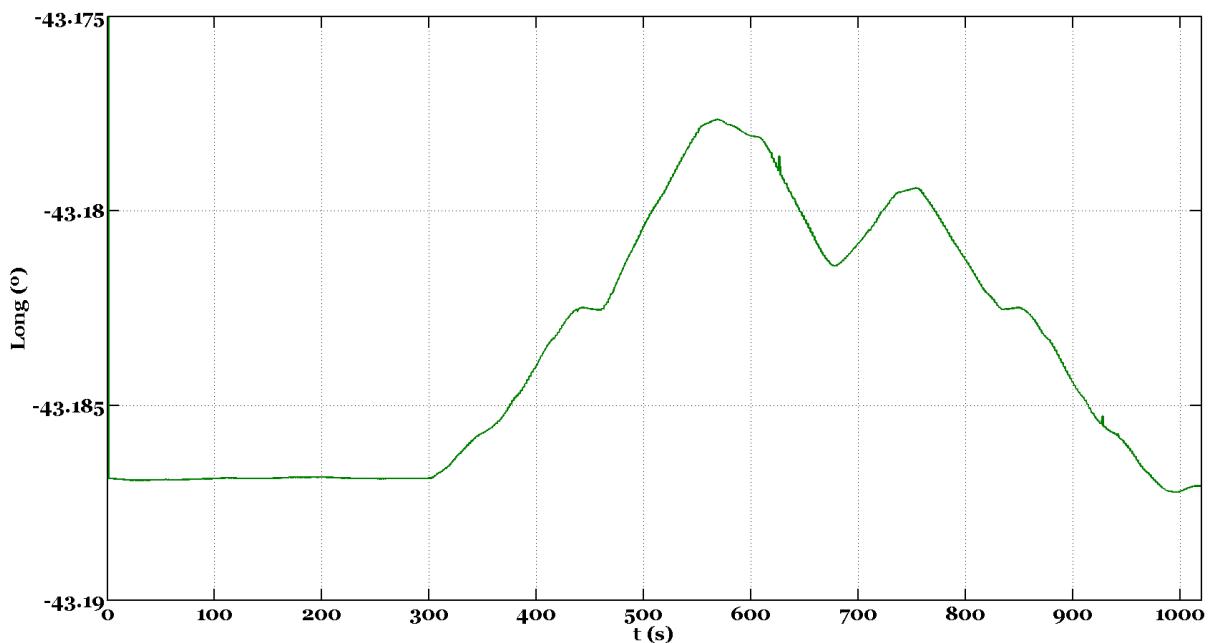


FIG.6.28: Análise da longitude lida pelo GPS 18-5Hz *Garmin*

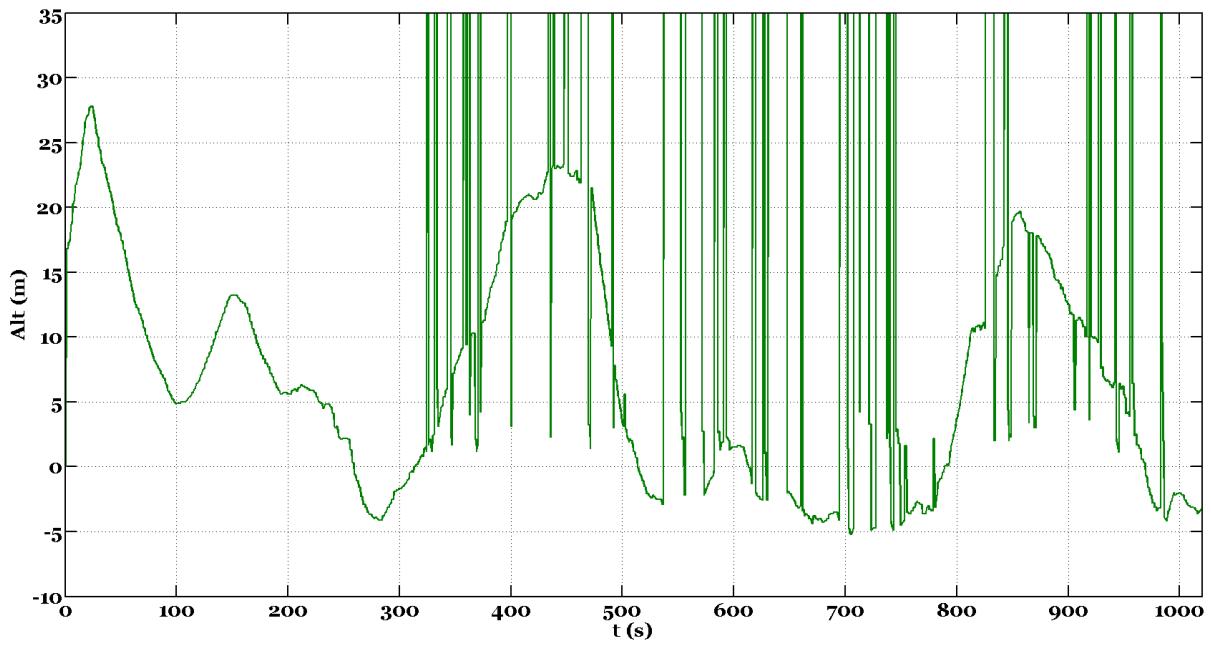


FIG.6.29: Análise da altitude lida pelo GPS 18-5Hz *Garmin*

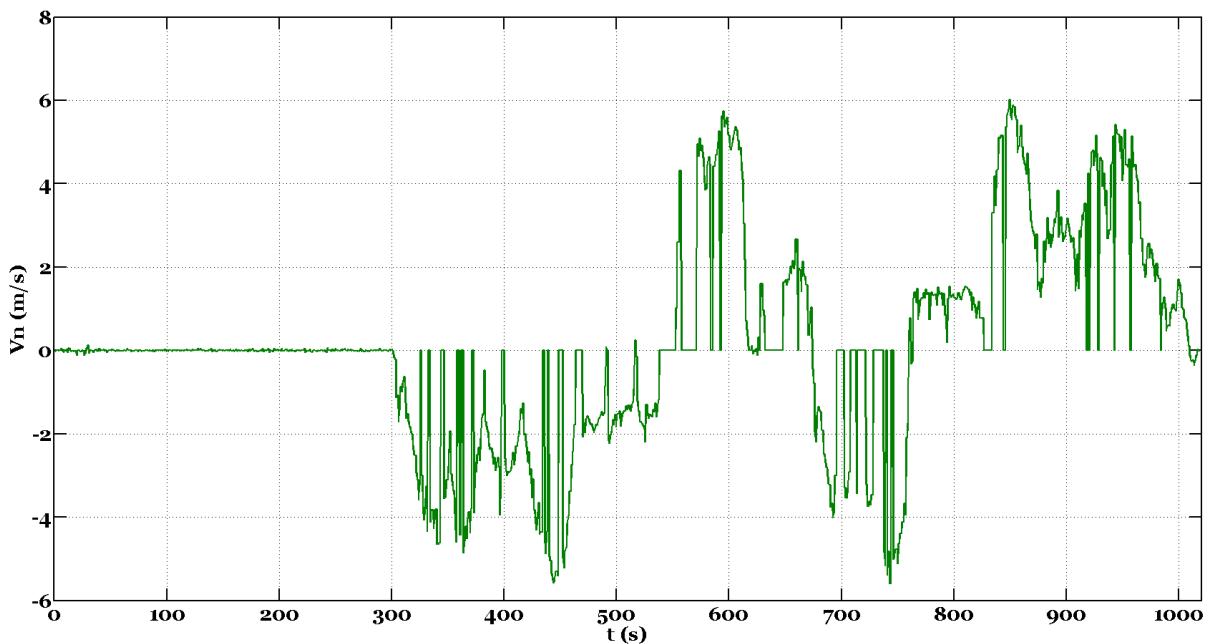


FIG.6.30: Análise da velocidade norte (V_n) lida pelo GPS 18-5Hz *Garmin*

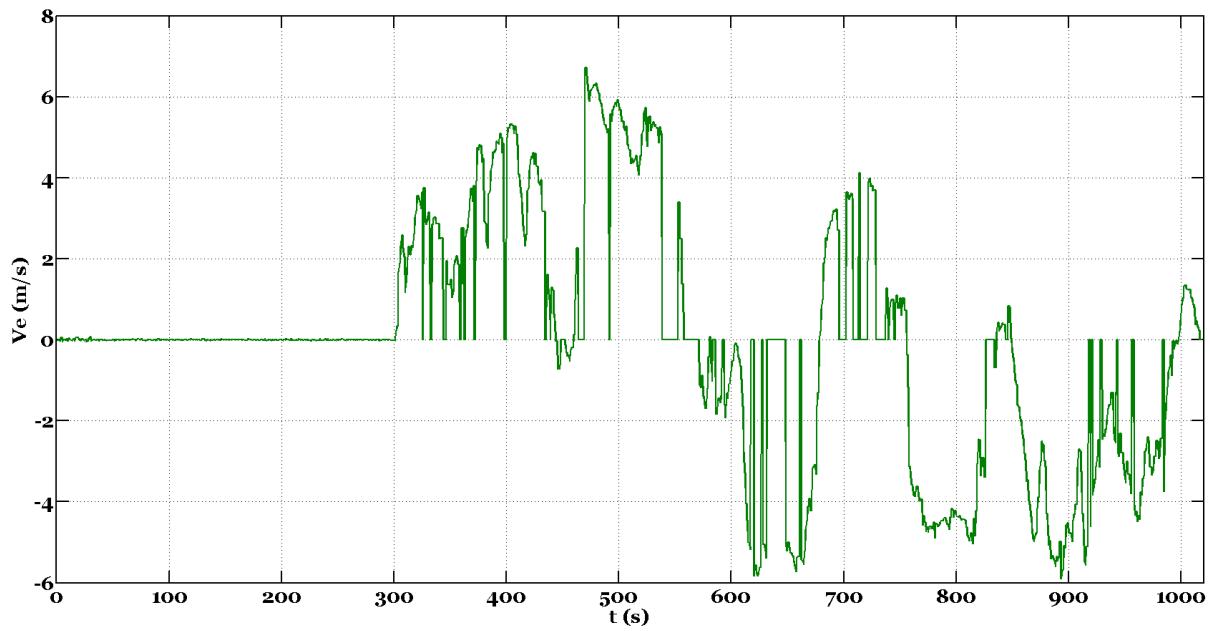


FIG.6.31: Análise da velocidade leste (V_e) lida pelo GPS 18-5Hz *Garmin*

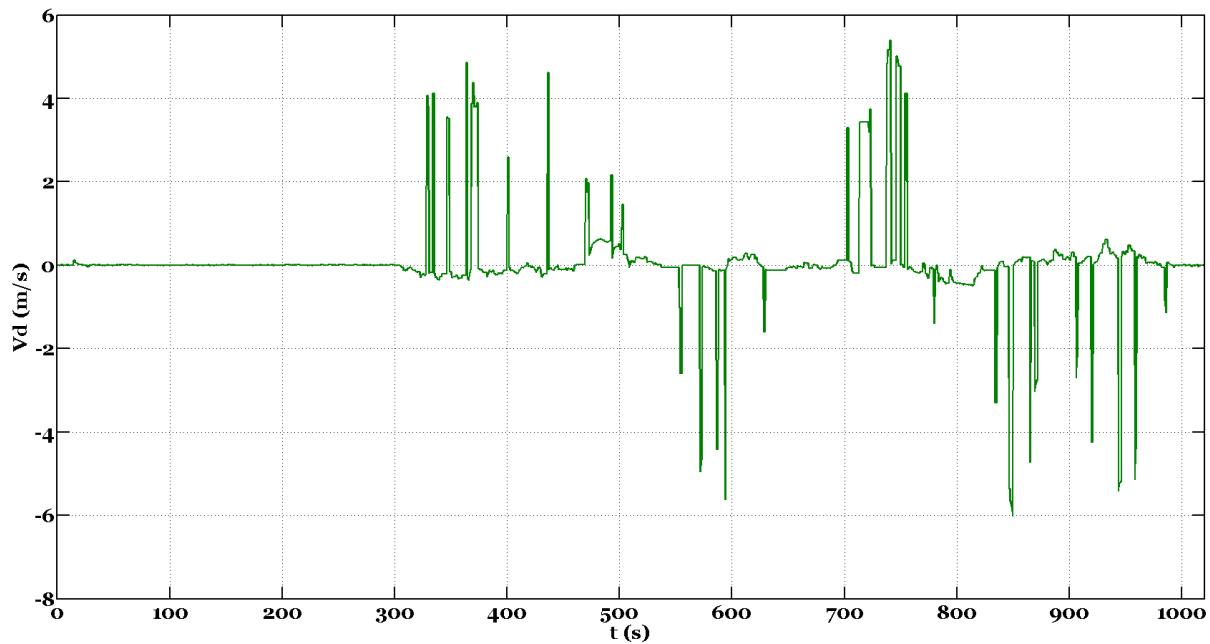


FIG.6.32: Análise da velocidade vertical (V_d) lida pelo GPS 18-5Hz *Garmin*

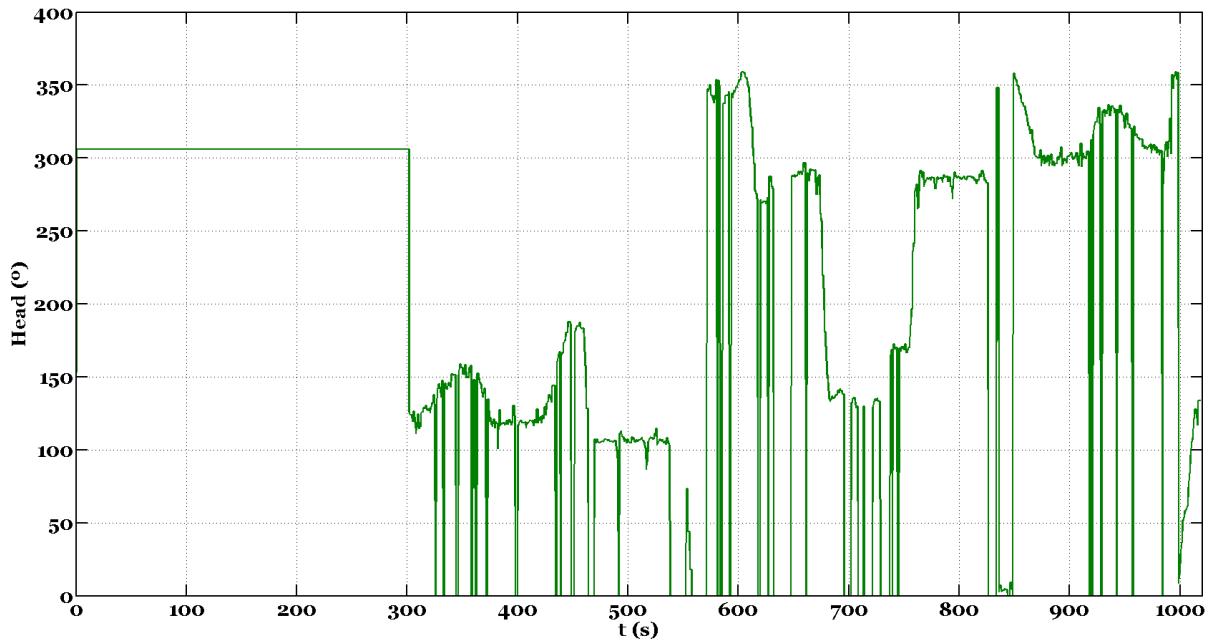


FIG.6.33: Análise do *head* lido pelo GPS 18-5Hz *Garmin*

Durante a trajetória ocorreram muitas perdas momentâneas de sinal do GPS, isto deve-se ao fato da pista ser encoberta por muitas árvores em vários trechos do trajeto. A figura 6.34 é o sinal "Track Status" do GPS que mostra todas estas perdas, onde o valor 0 acusa perda de GPS e o valor 1 acusa sinal disponível. Todas estas perdas de GPS refletem nos dados de posição, velocidade e *head* lidos por este sensor verificados pelos picos e quedas bruscas nos gráficos destes parâmetros.

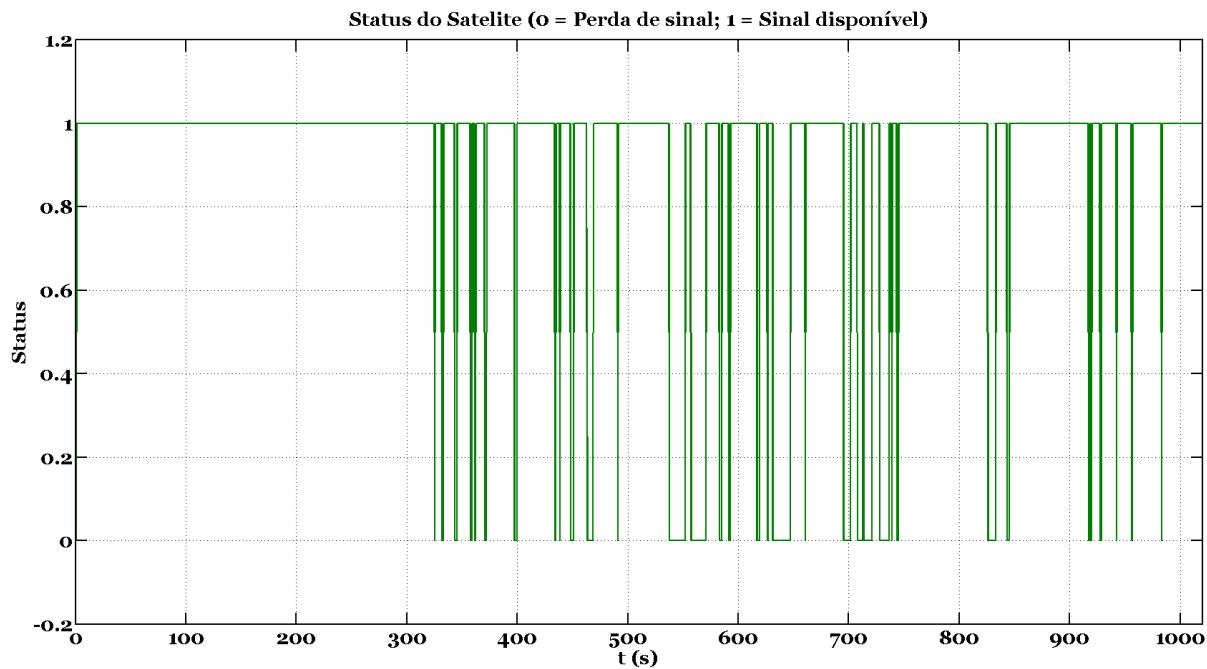


FIG.6.34: Análise do "Track Status" do GPS 18-5Hz *Garmin*

A figura 6.35 representa a quantidade de satélites rastreados pelo GPS durante o trajeto:

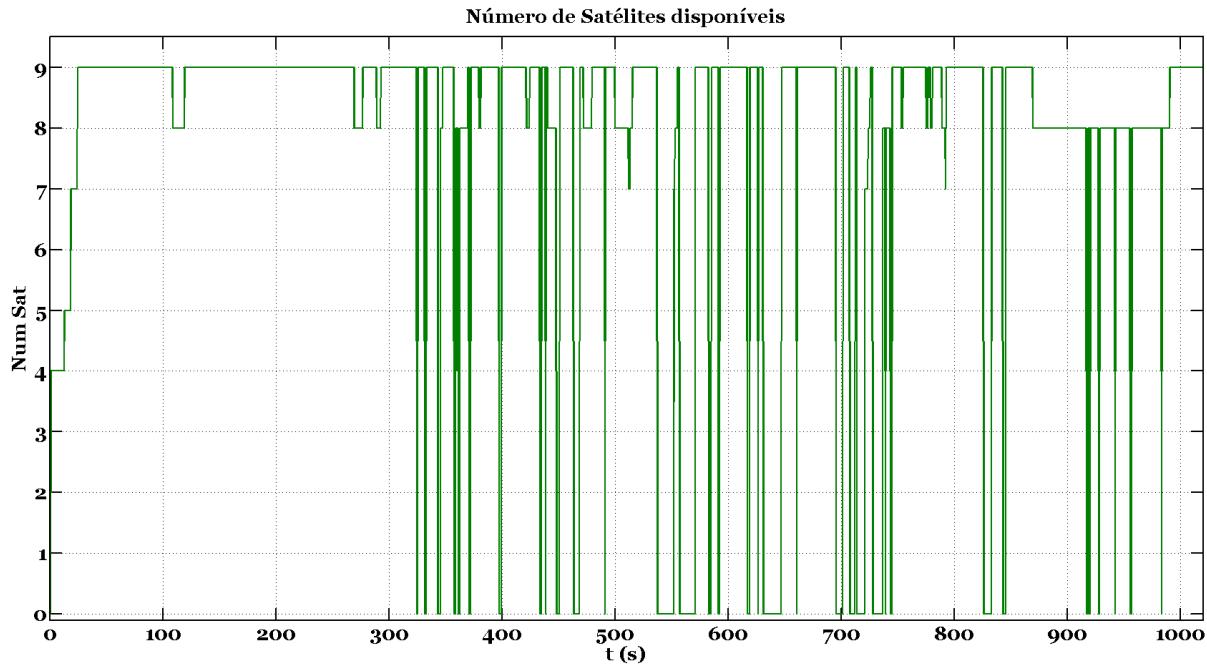


FIG.6.35: Análise do número de satélites rastreados do GPS 18-5Hz *Garmin*

De posse dos dados da IMU e do GPS, foi feito o pós-processamento destes dados no algoritmo INS/GPS e foi analisado o resultado. Vale ressaltar, que foi realizada uma

filtragem em tempo real dos sinais do GPS, para se eliminar os picos e quedas bruscas destes, pois foi verificado que estas variações, que aconteciam mesmo com sinal disponível, denegriam bastante o resultado da navegação.

Uma outra adequação feita no algoritmo foi no modo de processamento quando ocorre a falta de GPS. A maneira tradicional de se contornar este problema é simplesmente navegando com o algoritmo de navegação inercial puro até a volta do sinal de GPS, ou seja, navegando sem o algoritmo INS/GPS integrado. No entanto, este modo só é válido para sensores iniciais de alta exatidão e não é válido para sensores MEMS cuja navegação sem o GPS não permanece tempo suficiente com a precisão necessária. Isto foi verificado neste experimento, e portanto foi usado um artifício bastante simples para minimizar este problema. O artifício foi simplesmente o de nunca navegar no modo INS puro e quando o GPS faltasse seriam lidos os últimos dados fornecidos por ele. Apesar de simples, este método mostrou-se eficaz para este caso específico em que as perdas de GPS não excederam 15 segundos.

Os resultados da navegação podem então ser vistos nas figuras abaixo. Primeiramente mostramos nas figuras 6.36 a 6.44 os parâmetros de navegação calculados pelo algoritmo comparados com os mesmos dados lidos pelo GPS *Garmin*, exceto os sinais de *roll* e *pitch*. Os sinais de GPS mostrados nestes gráficos já se encontram filtrados.

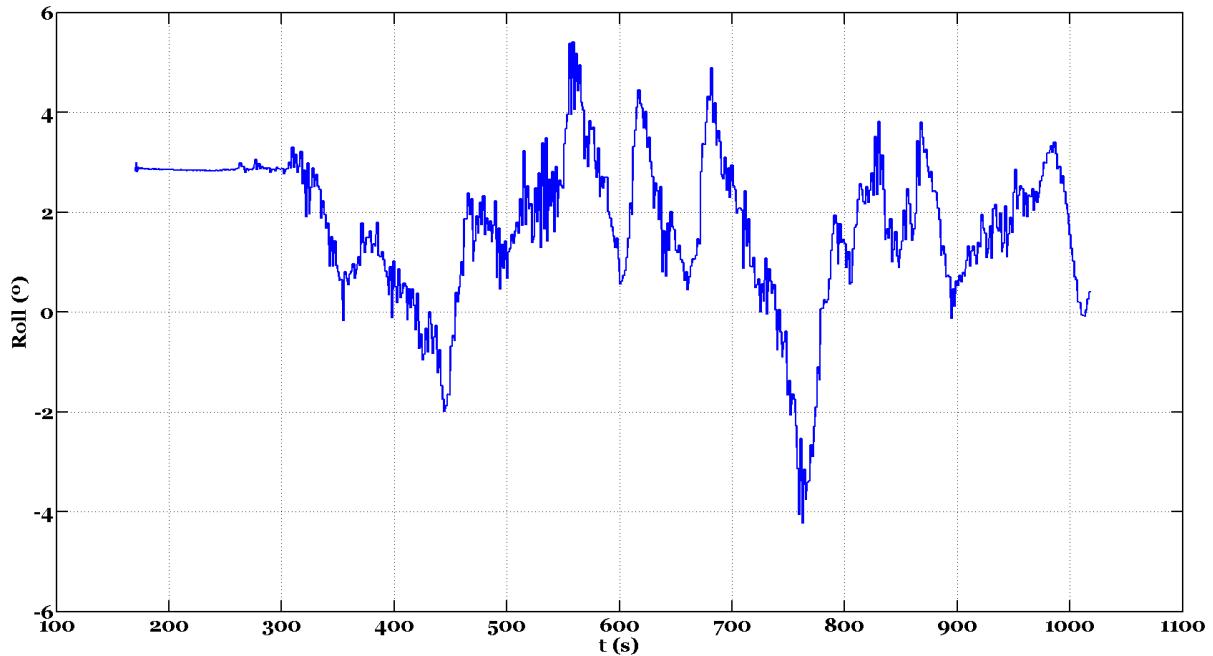


FIG.6.36: *Roll* calculado pelo algoritmo INS/GPS

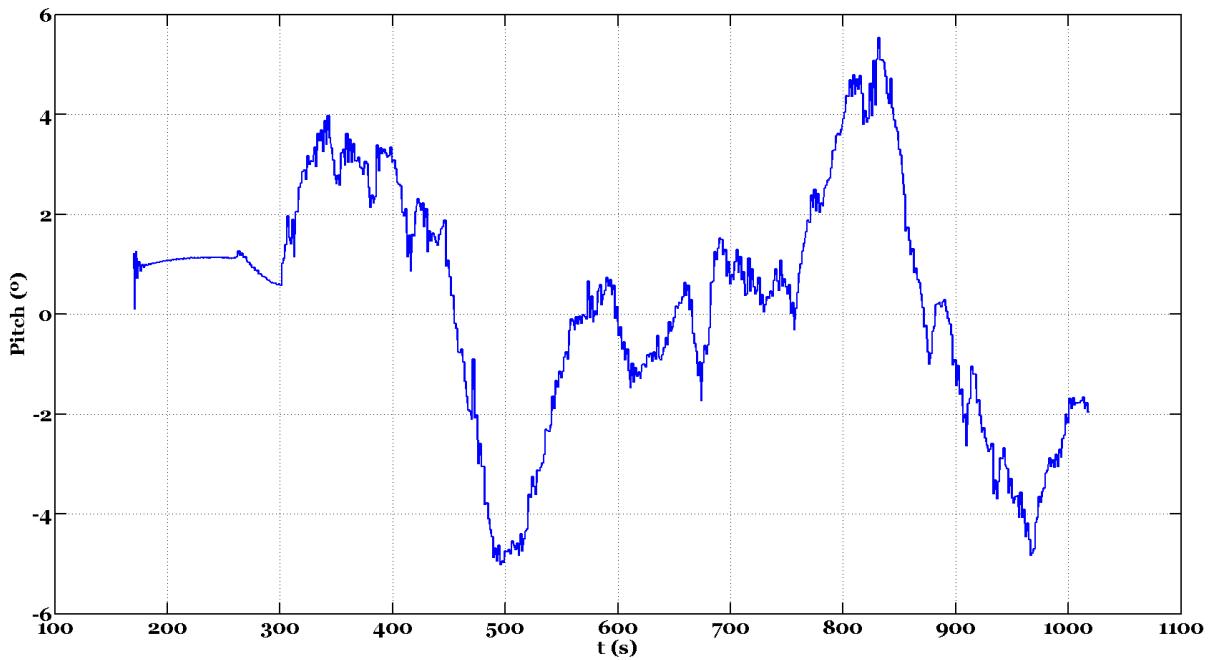


FIG.6.37: *Pitch* calculado pelo algoritmo INS/GPS

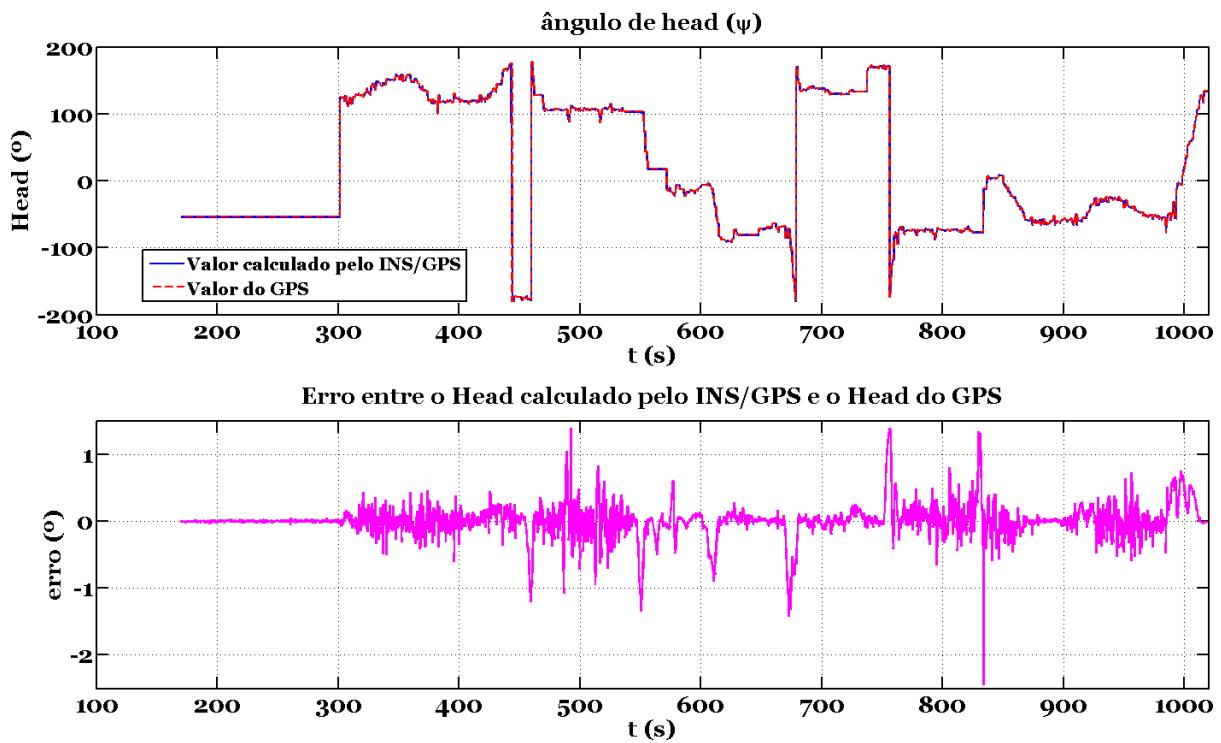


FIG.6.38: Comparação entre o *head* calculado pelo algoritmo e o *head* do GPS *Garmin*

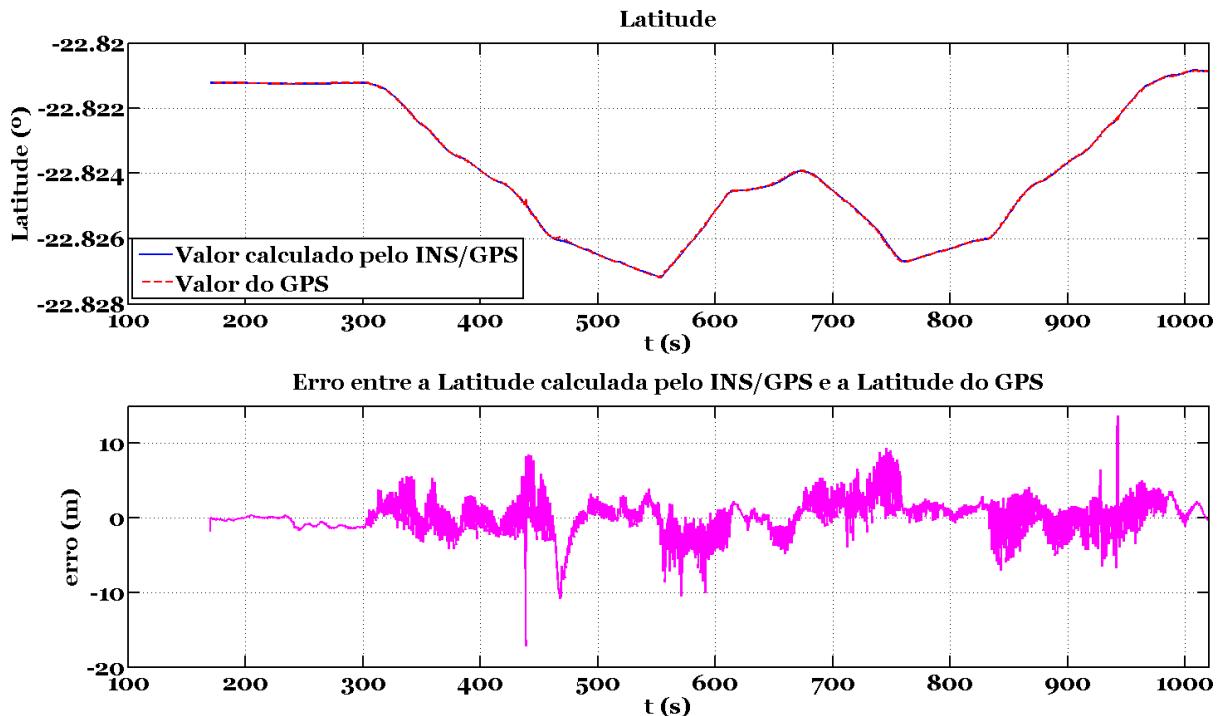


FIG.6.39: Comparação entre a latitude calculada pelo algoritmo e a latitude do GPS
Garmin

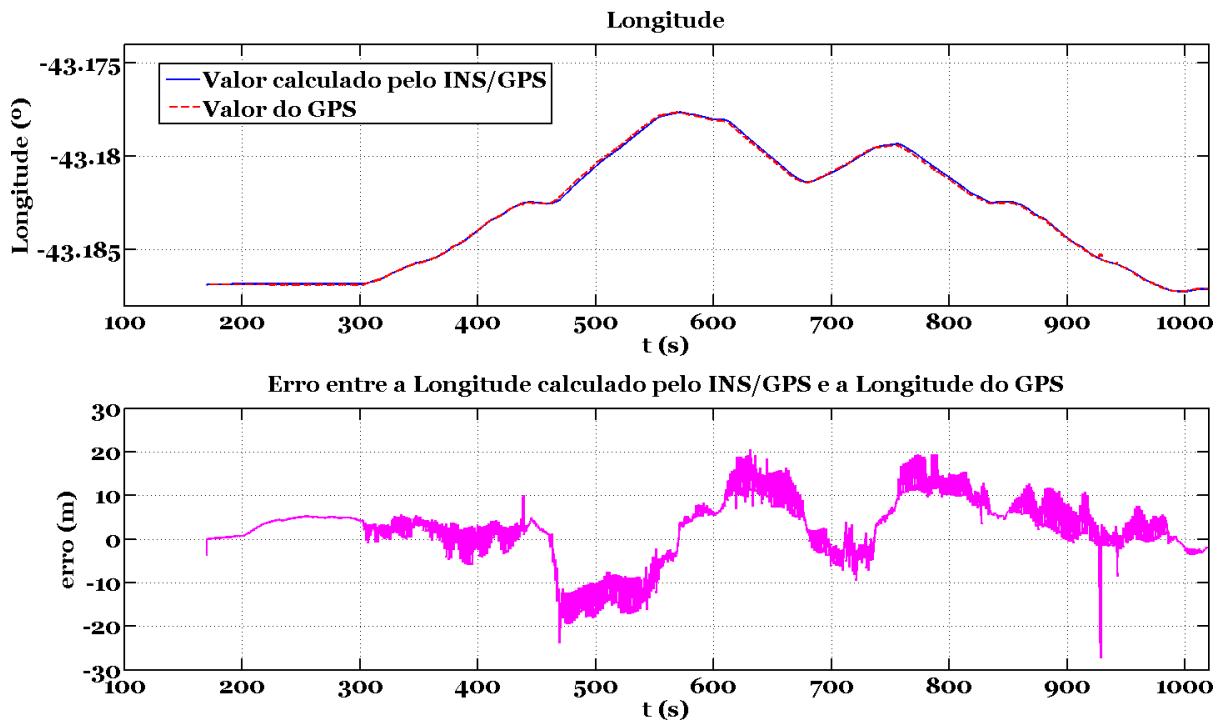


FIG.6.40: Comparação entre a longitude calculada pelo algoritmo e a longitude do GPS
Garmin

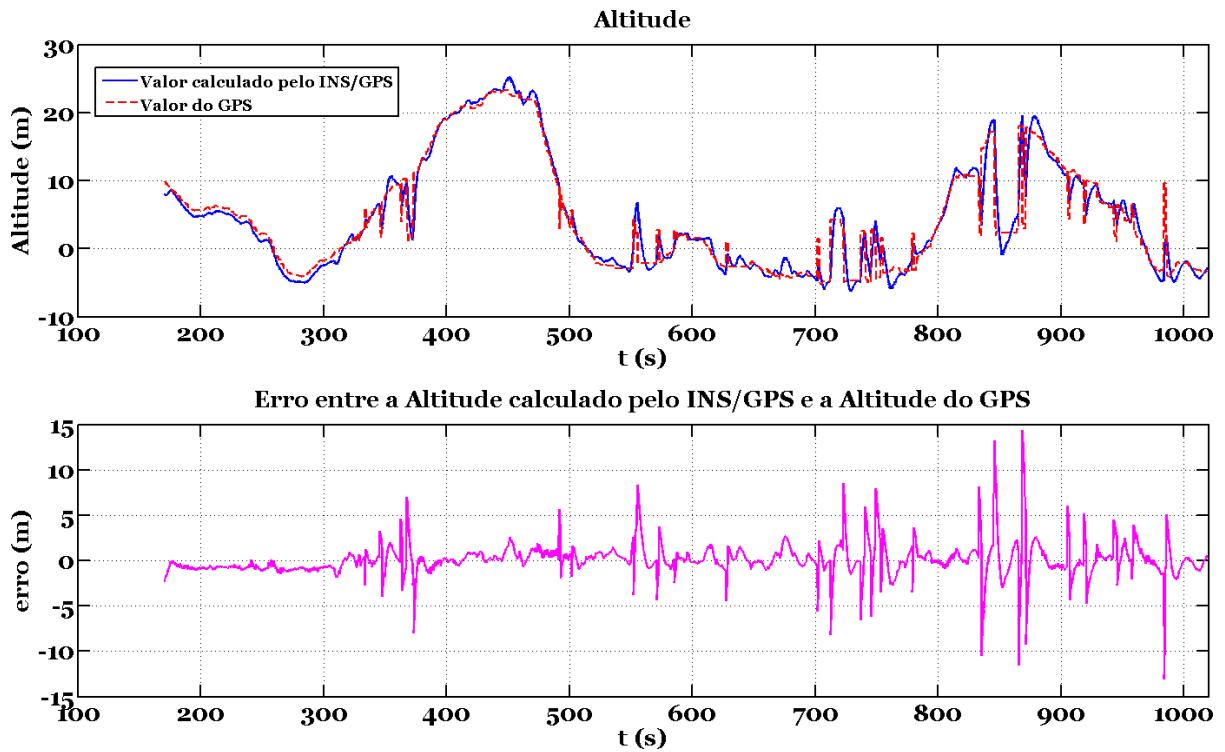


FIG.6.41: Comparação entre a altitude calculada pelo algoritmo e a altitude do GPS
Garmin

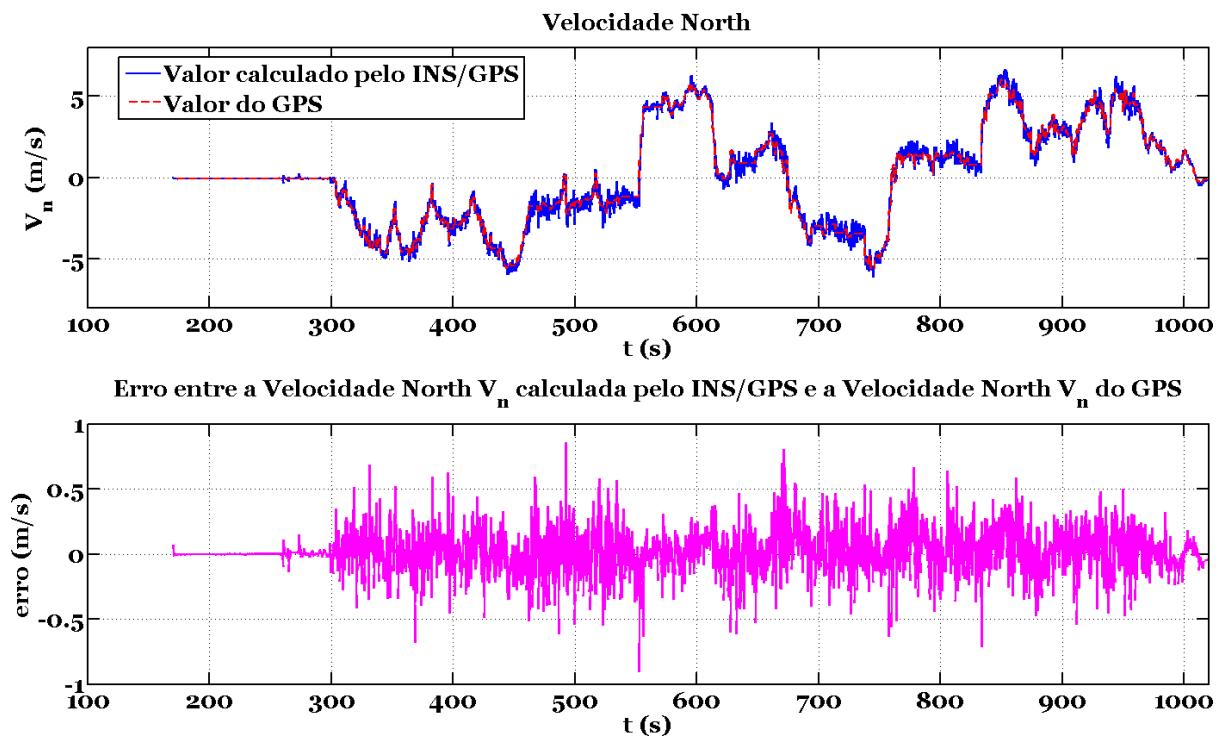


FIG.6.42: Comparação entre o V_n calculado pelo algoritmo e o V_n do GPS *Garmin*

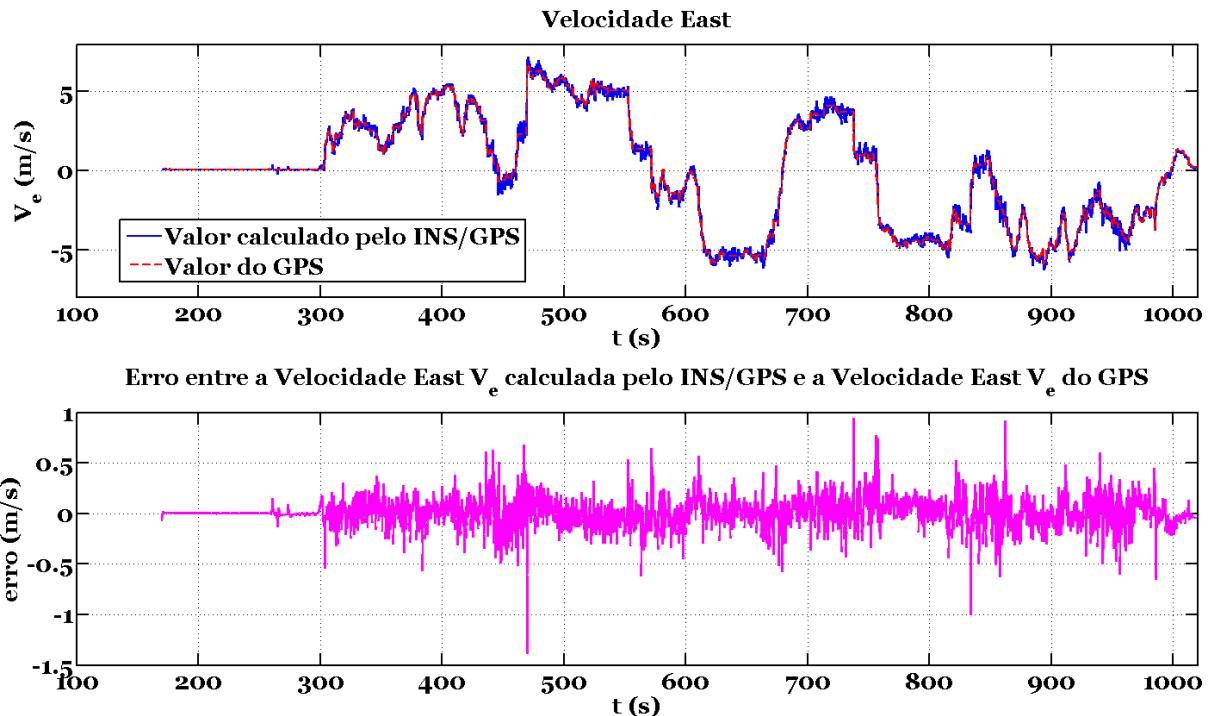


FIG.6.43: Comparação entre o V_e calculado pelo algoritmo e o V_e do GPS *Garmin*

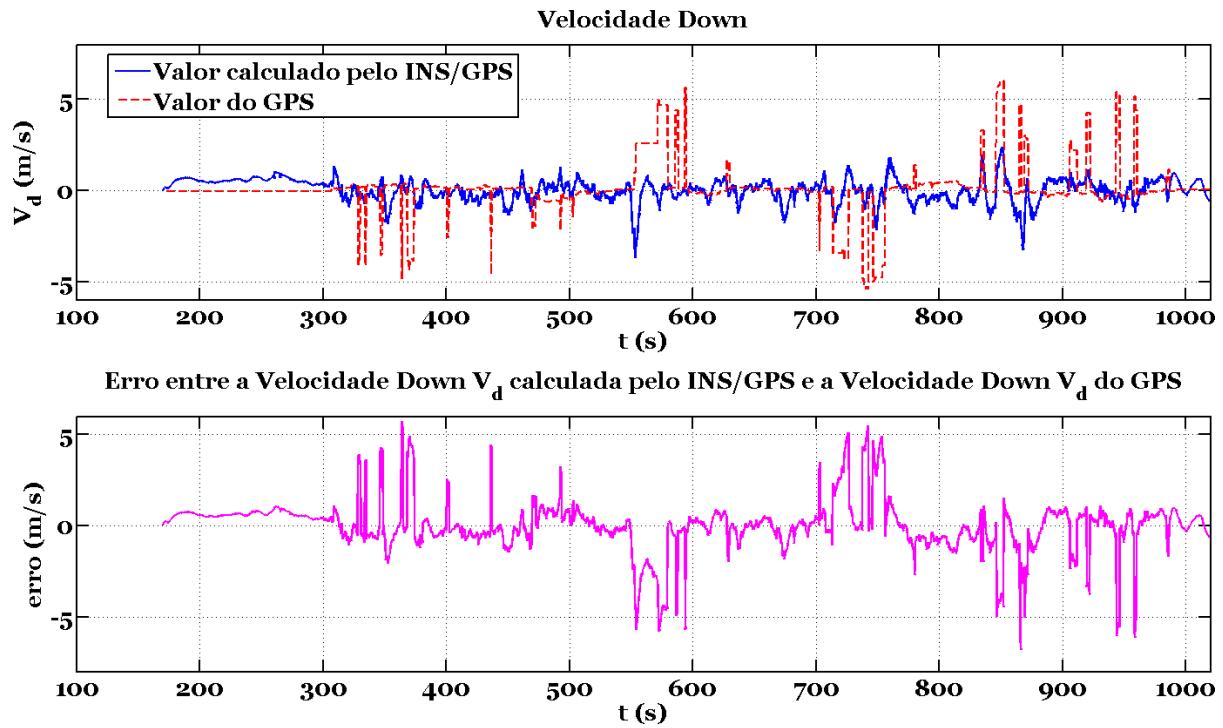


FIG.6.44: Comparação entre o V_d calculado pelo algoritmo e o V_d do GPS *Garmin*

Analizando os gráficos percebe-se que os sinais de GPS e do algoritmo INS/GPS estão bastante próximos um do outro, a análise da diferença entre os sinais mostra que os

maiores erros estão nas medidas de posição (latitude, longitude e altura). Isto se deve ao fato de que os sinais de posição são obtidos após 3 integrações sucessivas (equação 2.34), acumulando assim o erro proveniente dos sensores. Percebe-se também que o sinal de V_d possui um erro maior que os sinais V_n e V_e , isso deve-se ao modelo adotado do cálculo de gravidade. O cálculo da gravidade depende dos sinais de latitude e altitude, portanto o erro associado a estes sinais se propagam afetando principalmente o V_d e a altitude.

As figuras 6.45 a 6.47 mostram os parâmetros de posição calculados pelo algoritmo comparados com os dados de posição dos marcos previamente medidos:

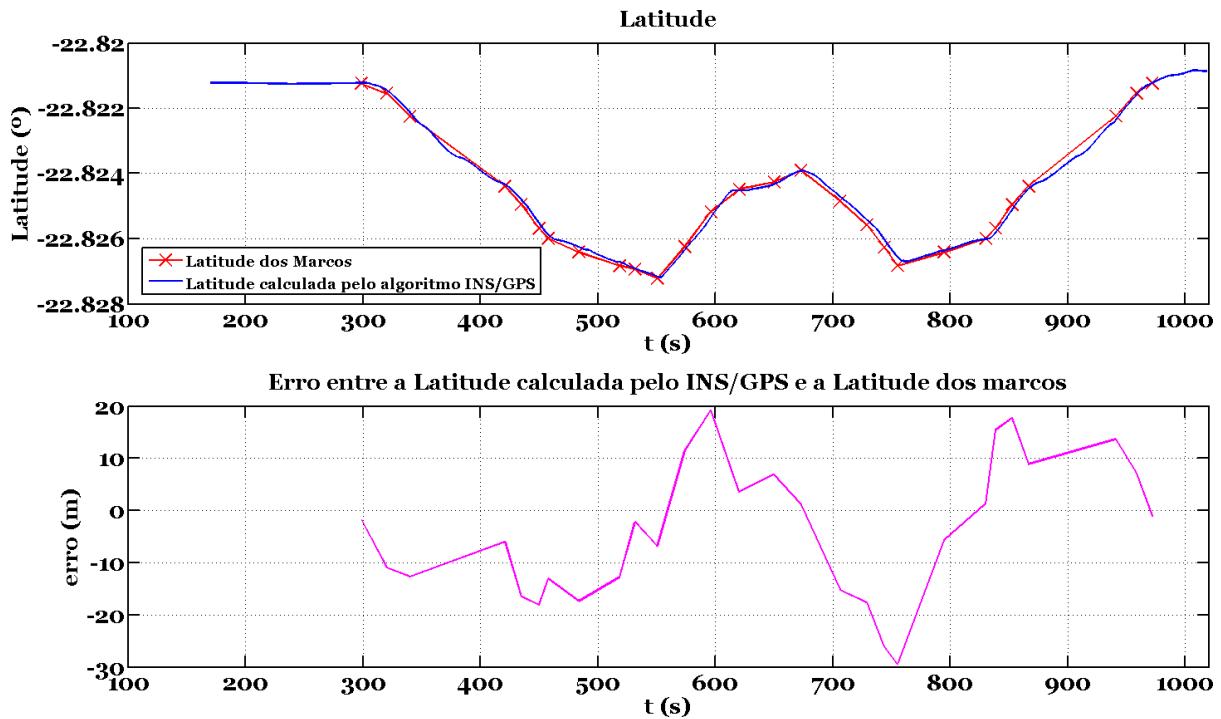


FIG.6.45: Comparaçāo das latitudes calculadas pelo algoritmo com os valores dos marcos

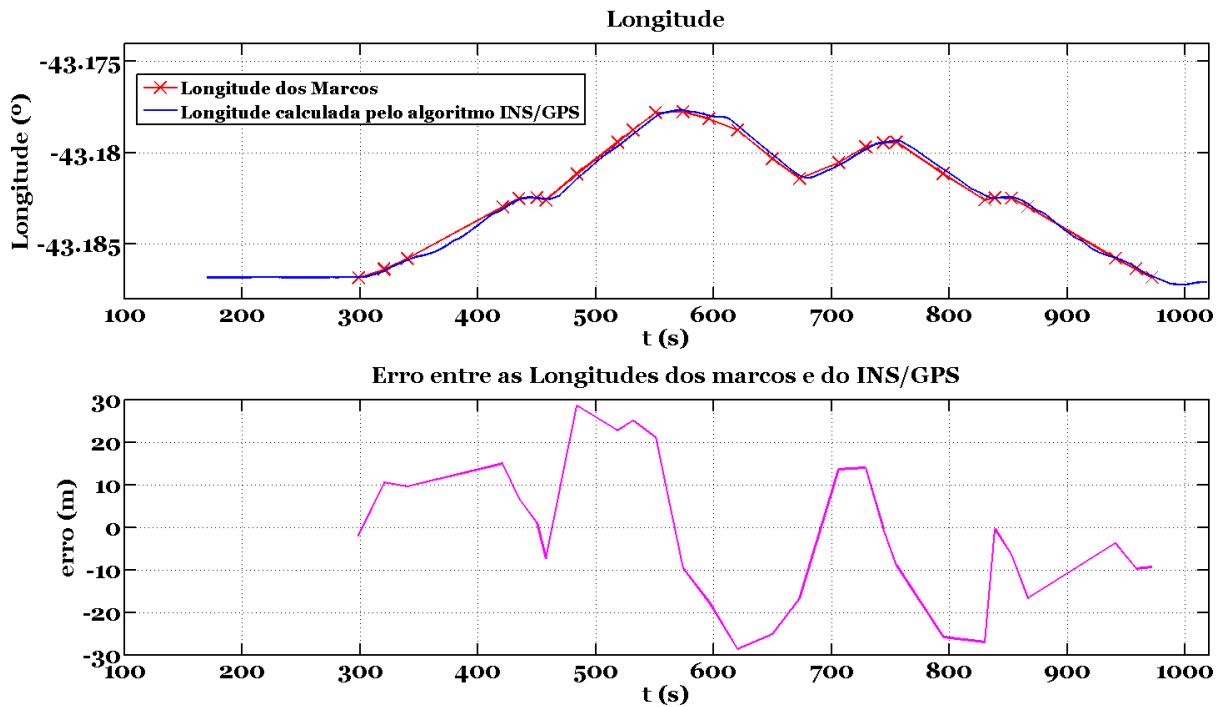


FIG.6.46: Comparação das longitudes calculadas pelo algoritmo com os valores dos marcos

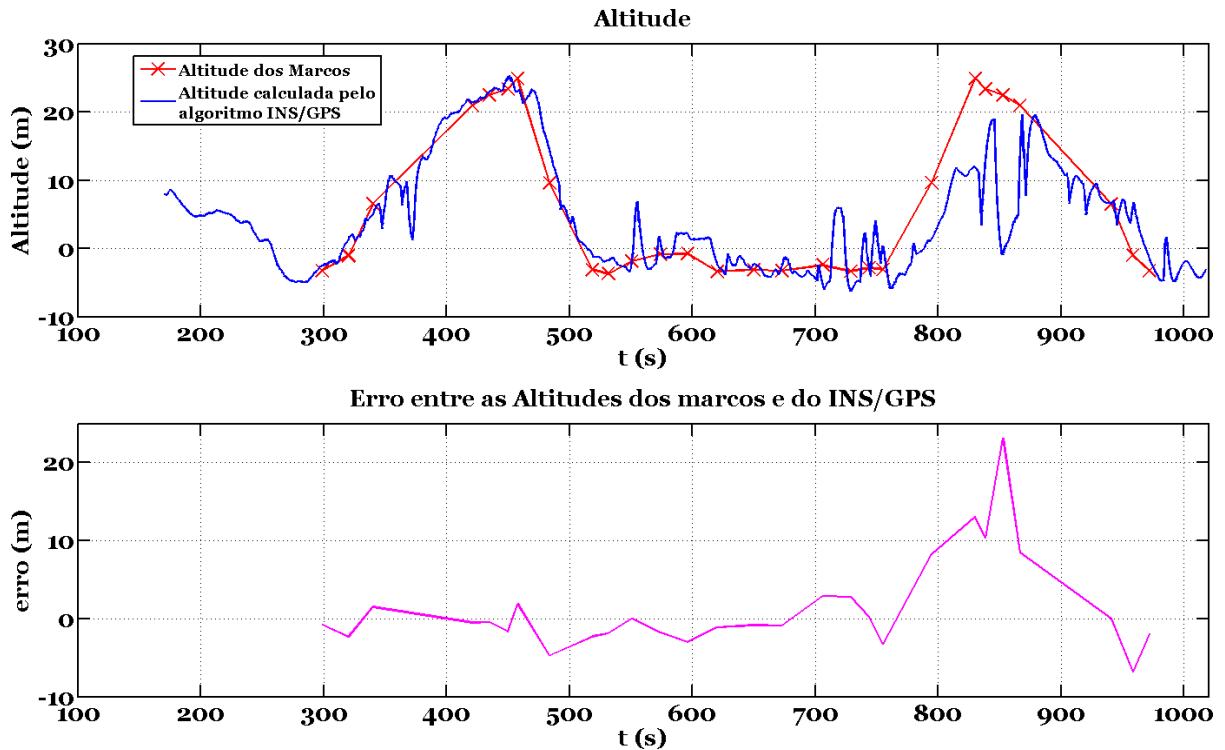


FIG.6.47: Comparação das altitudes calculadas pelo algoritmo com os valores dos marcos

A seguir, será feita a comparação entre os dados calculados pelo NAV420 e os dados

calculados pelo algoritmo INS/GPS apresentado neste trabalho. A comparação será feita apenas em cima dos 28 marcos que se sabe estarem corretos. As figuras 6.48 a 6.53 mostram estes comparativos.

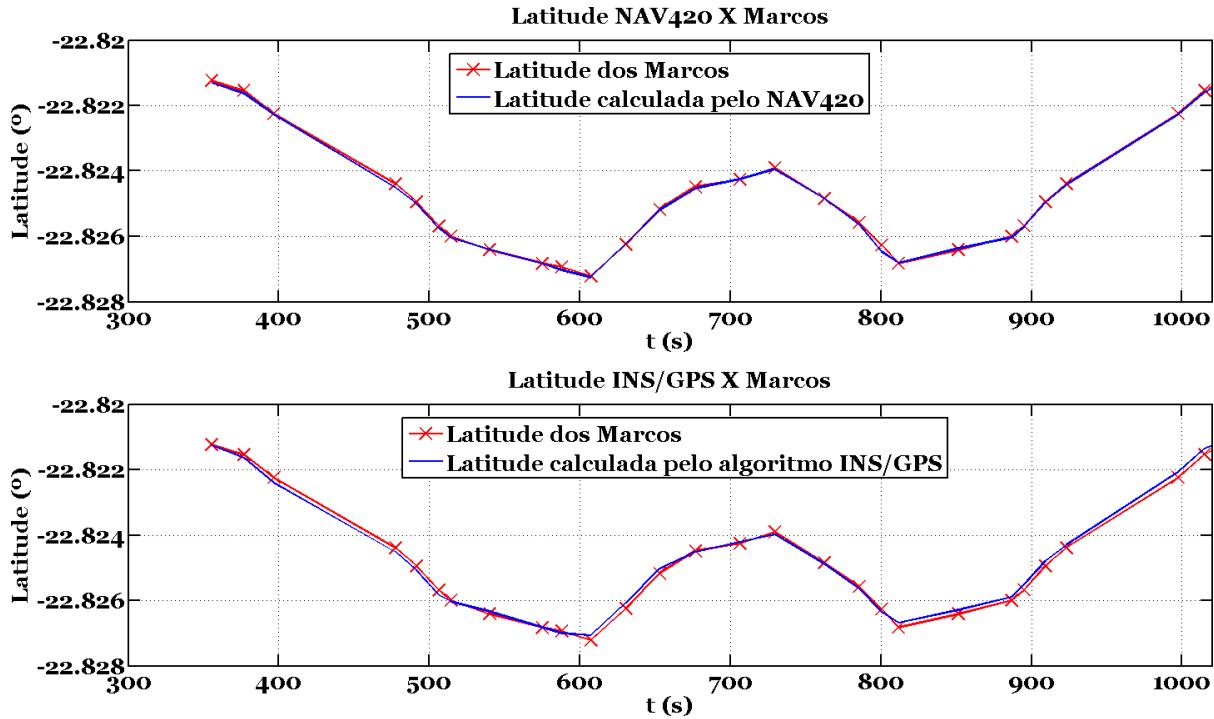


FIG.6.48: Comparação da latitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos

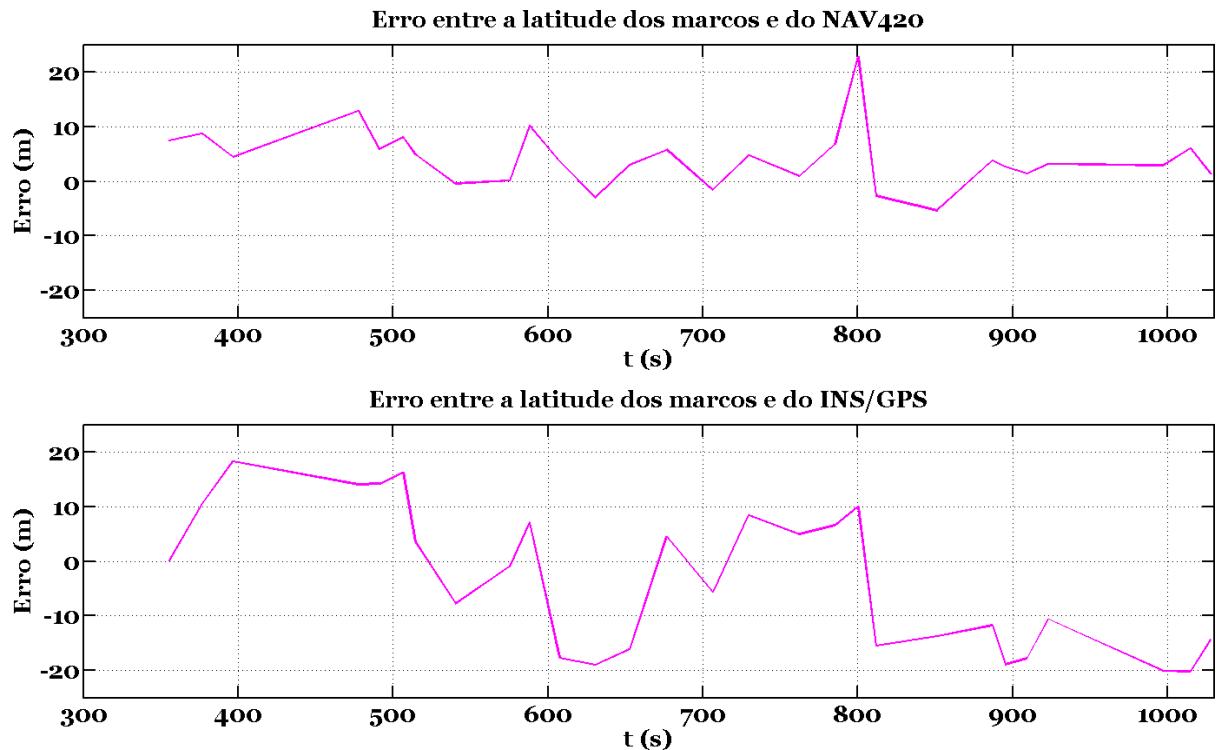


FIG.6.49: Comparação entre o erro da latitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos

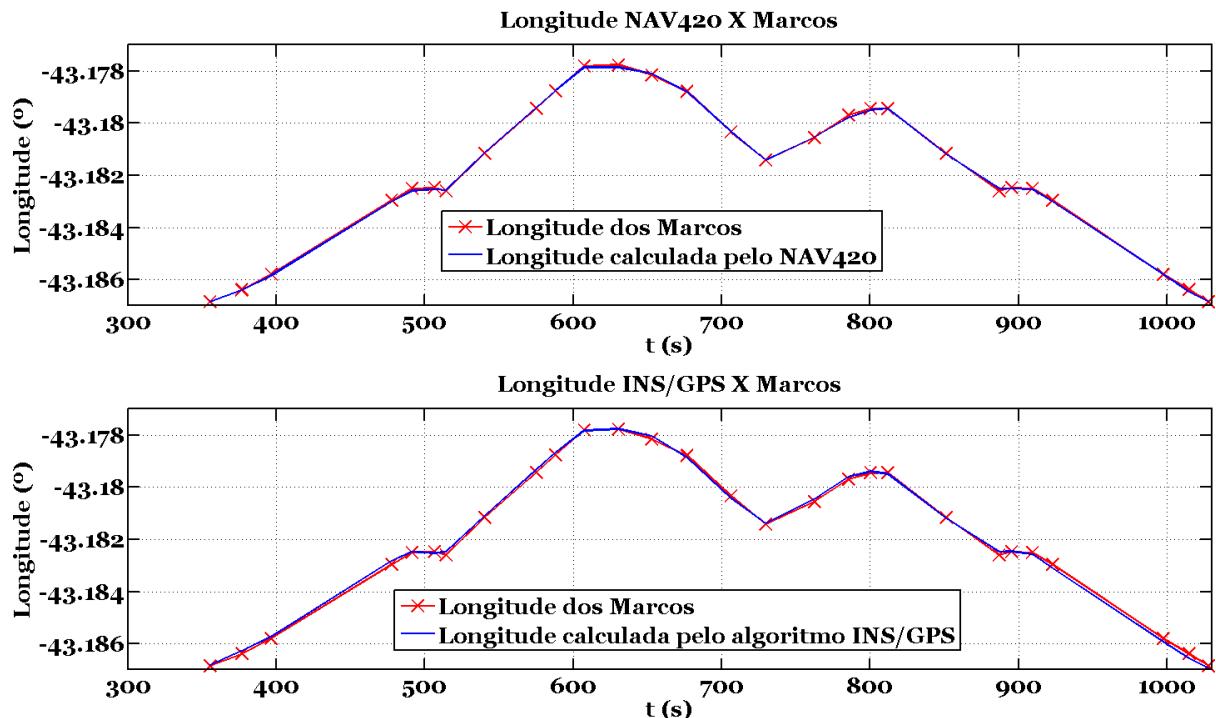


FIG.6.50: Comparação da longitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos

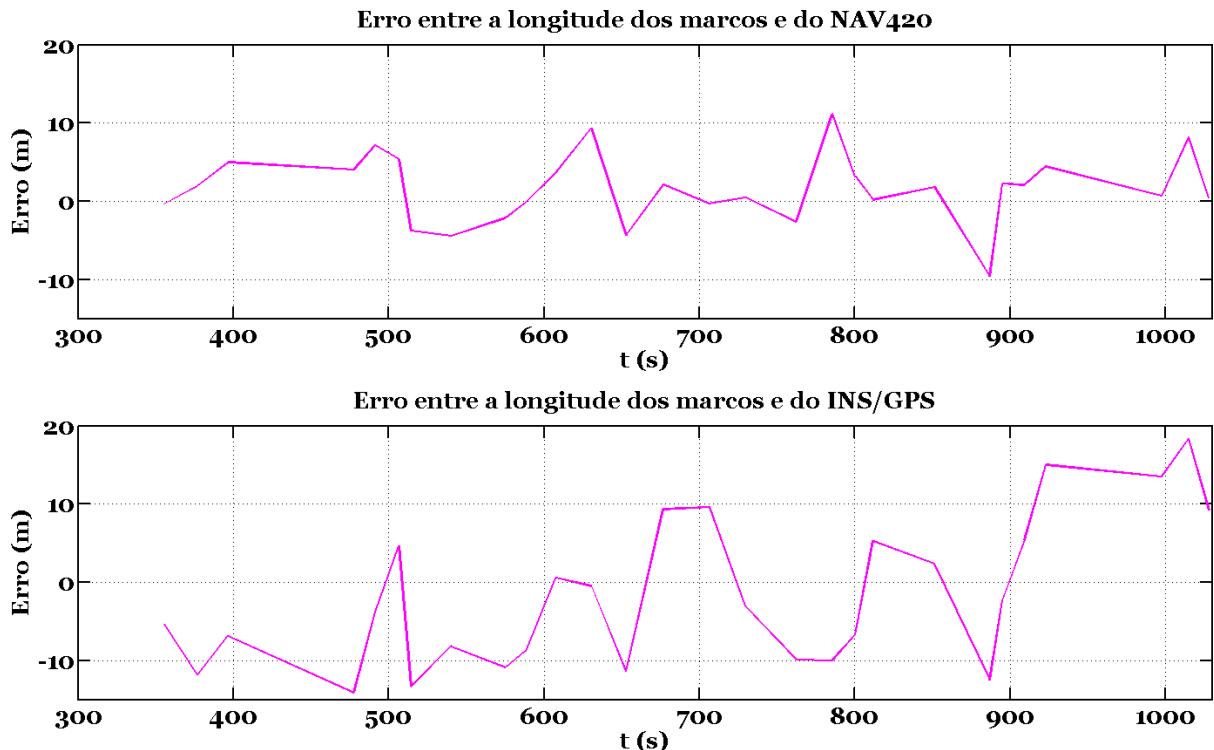


FIG.6.51: Comparação entre o erro da longitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos

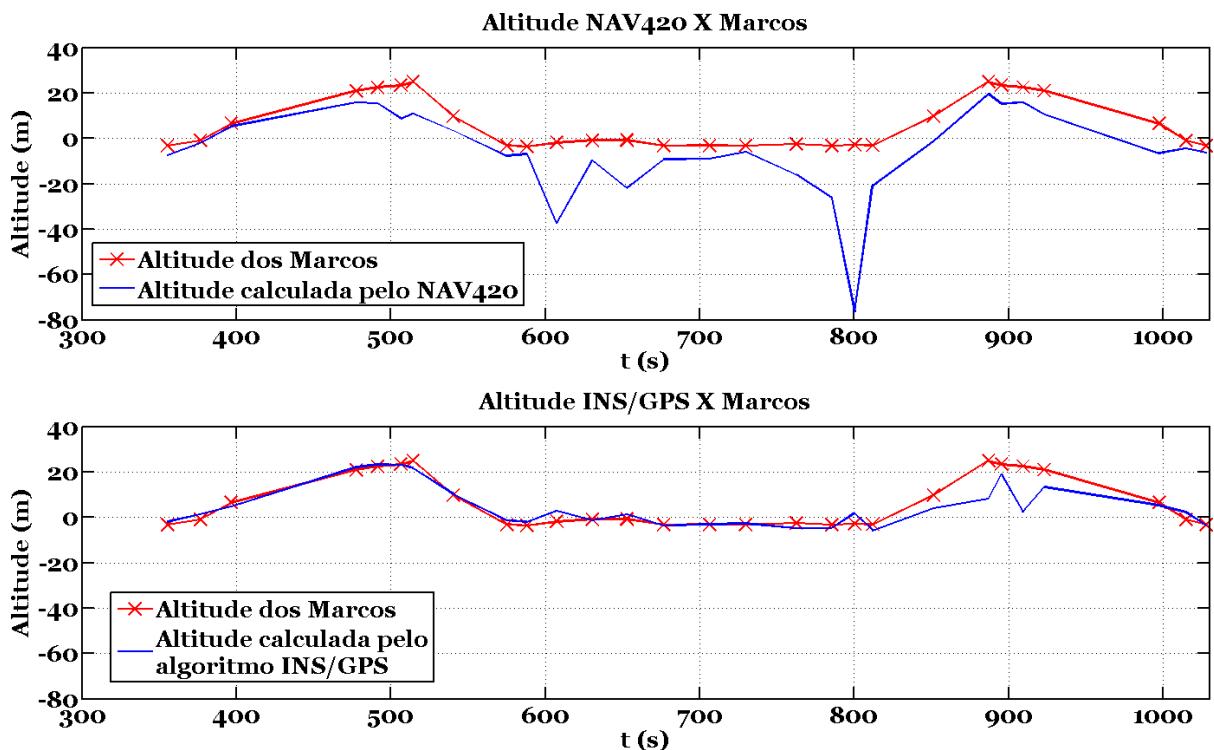


FIG.6.52: Comparação da altitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos

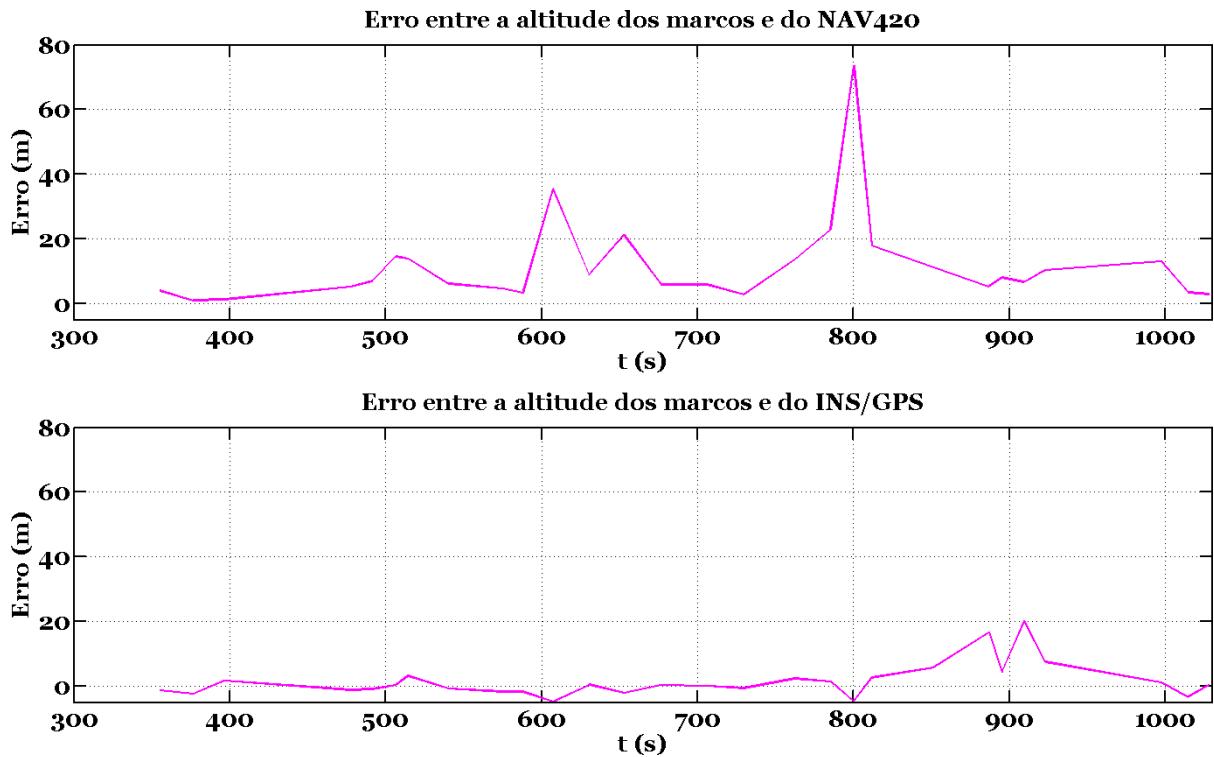


FIG.6.53: Comparação entre o erro da altitude calculada pelo algoritmo INS/GPS e pelo NAV420 com os valores dos marcos

Comparando-se o NAV420 e o algoritmo INS/GPS usado neste trabalho, observa-se que o NAV420 possui um menor erro nos sinais de latitude e longitude, provavelmente devido ao tratamento dado por esta central inercial à falta de seu GPS que deve incluir algoritmos mais sofisticados de predição de trajetória. Para o sinal de altitude entretanto, o erro do NAV420 se mostrou maior. Sendo a altitude a informação de maior erro de um GPS, supõe-se que o GPS do NAV420 apresentou uma maior variação neste sinal.

Em seguida, é mostrado os resultados da navegação no *Google Earth*. A figura 6.54 mostra os dados de latitude e longitude do GPS *Garmin* junto com os marcos ao longo do trajeto. A figura 6.55 mostra os dados de latitude e longitude do algoritmo INS/GPS junto com os marcos enquanto a figura 6.56 mostra tanto os dados do GPS *Garmin* quanto os dados do INS/GPS junto com os marcos.

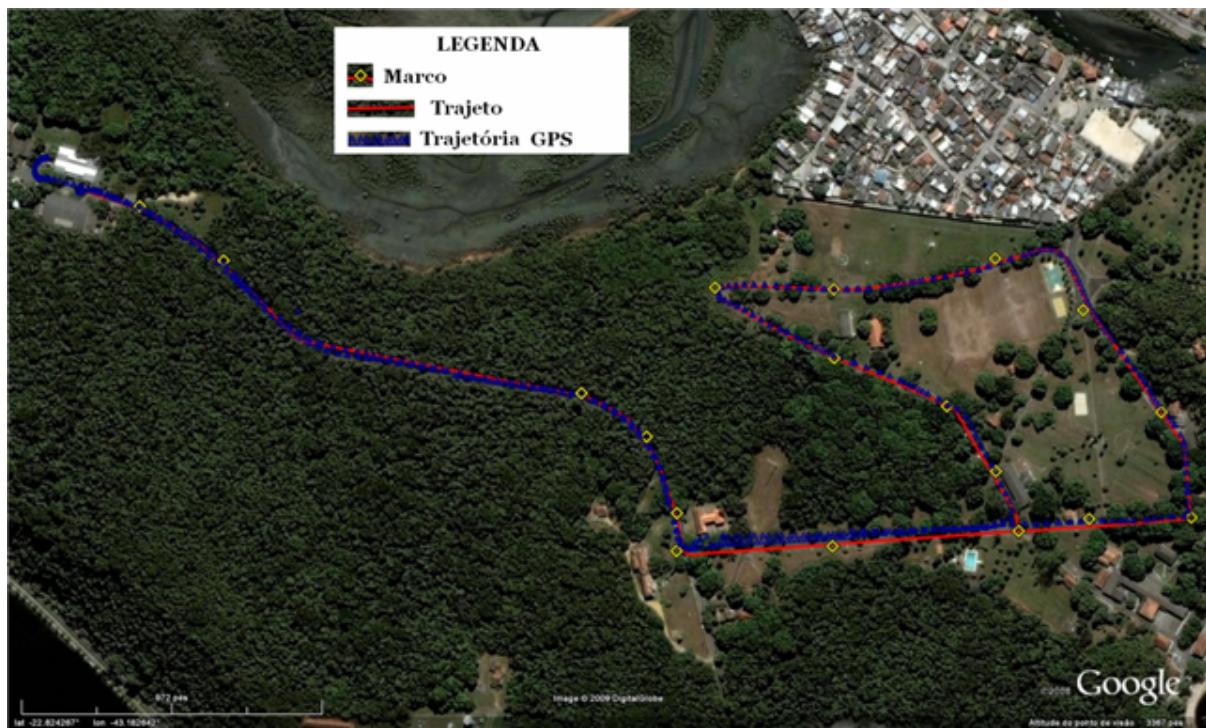


FIG.6.54: dados da trajetória percorrida pelo kombi - GPS Garmin e Marcos

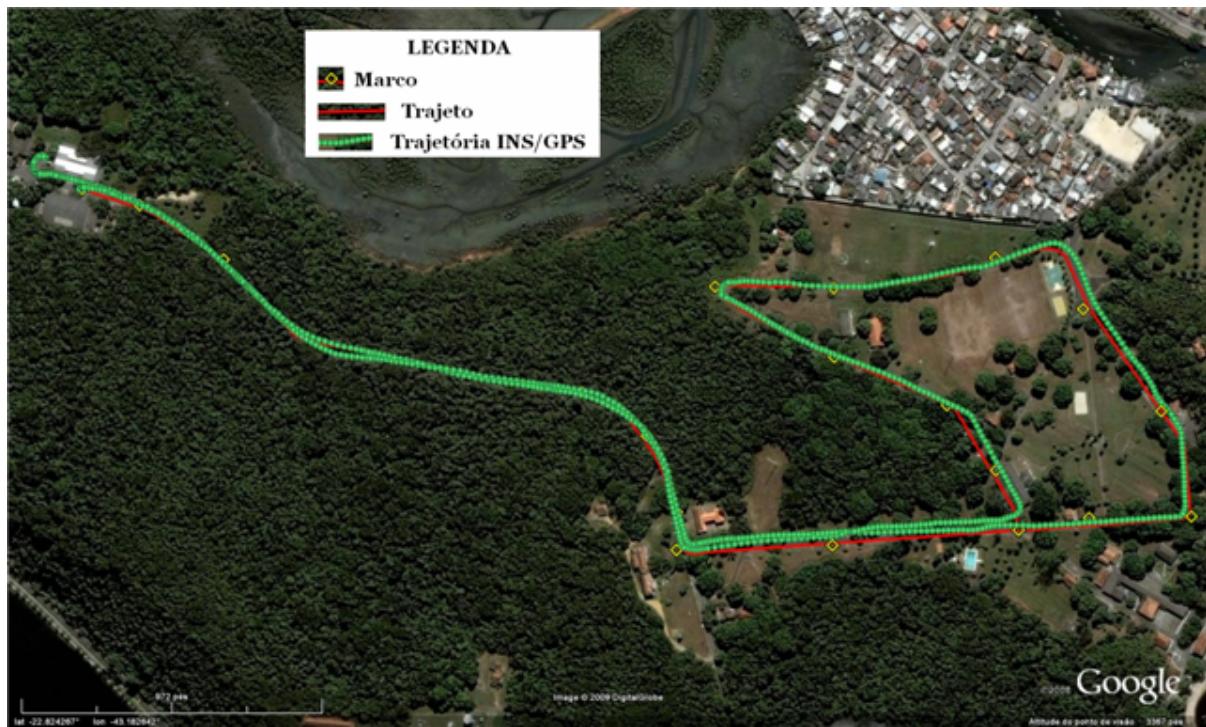


FIG.6.55: dados da trajetória percorrida pelo kombi - INS/GPS e Marcos

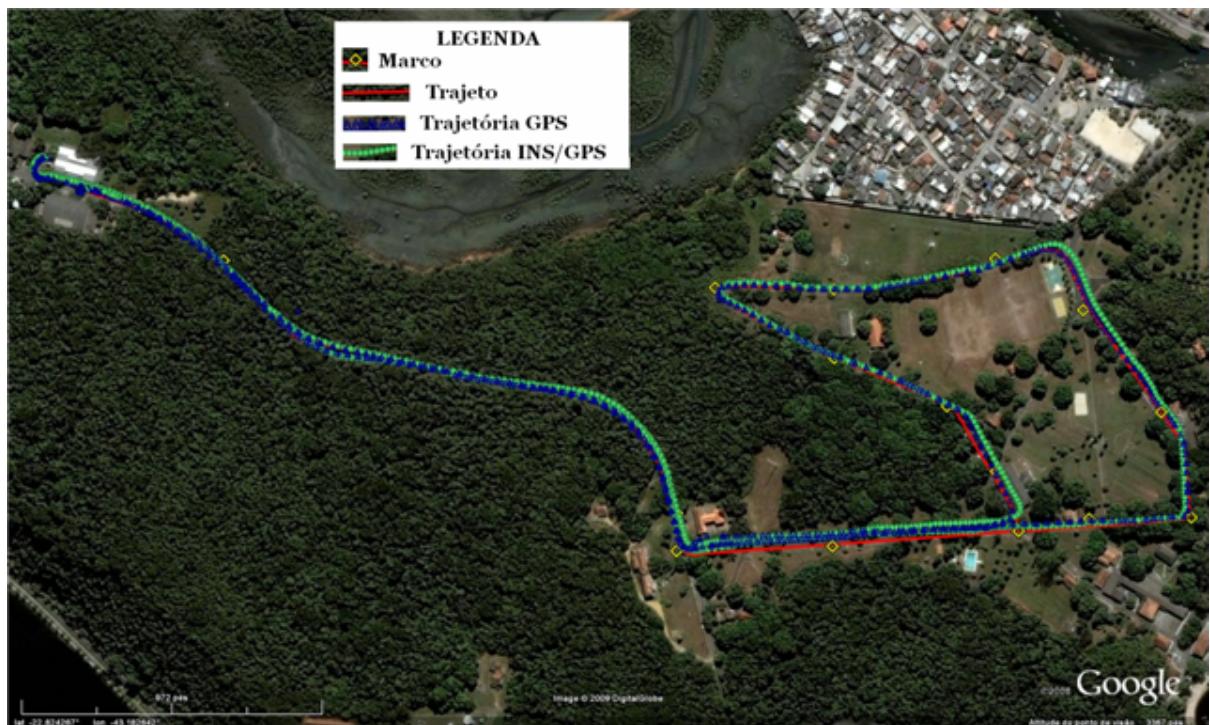


FIG.6.56: dados da trajetória percorrida pelo kombi - GPS *Garmin*, INS/GPS e Marcos

7 CONCLUSÕES E PROPOSTAS PARA FUTUROS TRABALHOS

7.1 CONCLUSÕES

O principal objetivo deste trabalho foi implementar em hardware de tempo real um algoritmo de navegação inercial integrado com o GPS. Em outras palavras, o objetivo traçado e alcançado foi o de desenvolver uma central inercial baseada em IMU e GPS comerciais. Com a análise dos resultados práticos, principalmente com o experimento da kombi, foi verificado que a central inercial baseada na IMU400 da *CrossBow* e no GPS *Garmin* apresentou uma boa exatidão nos resultados se considerarmos as grandes perdas de GPS que ocorreram durante o percurso da viatura.

Em se tratando de programação, foi verificado que o uso do MATLAB/Simulink com as *toolboxes Real Time Workshop* e *xPC Target* são excelentes ferramentas para a implementação de sistemas em tempo real. Pois sendo ferramentas simples, com interfaces amigáveis e intuitivas, economizam um tempo considerável se compararmos com linguagens de baixo nível como o C e o *Fortran* por exemplo. No entanto, a medida que a complexidade do modelo aumenta, também aumenta o nível de conhecimento necessário para o uso destas ferramentas.

No decorrer deste trabalho houve a necessidade de simular o algoritmo antes de testá-lo na prática com sensores reais. O simulador de vôo *FlightGear* foi utilizado então para este fim como gerador de trajetória, e foi de grande utilidade para pequenas mudanças e acertos no algoritmo de forma a aumentar sua exatidão e conhecer seu comportamento de acordo com a dinâmica do avião pilotado pelo *joystick*.

7.2 PROPOSTAS PARA TRABALHOS FUTUROS

7.2.1 GERADOR DE TRAJETÓRIAS

Foi mostrado neste trabalho a utilidade de um gerador de trajetórias na simulação de algoritmos de navegação inercial. O modelo utilizado neste trabalho, mesmo com as modificações feitas para operar como gerador de trajetórias, é um modelo proprietário do MATLAB e possui algumas simplificações. Sendo totalmente modificável, o modelo pode ser enriquecido para se aproximar ao máximo da realidade. A dinâmica das aeronaves também pode ser modificada, tendo em vista que se tem acesso total aos parâmetros

do veículo. Pode-se então comparar o comportamento de uma central inercial em um caça supersônico, ou em um *Boeing 747*. Para engenheiros de controle tem-se então uma ferramenta visual para se testar técnicas de controle a partir dos parâmetros gerados pela central inercial.

7.2.2 IMPLEMENTAÇÃO EM TEMPO REAL COM DIFERENTES RTOS E AVALIAÇÃO DE ROBUSTEZ

A escolha de um bom Sistema Operacional de Tempo Real (RTOS) é muito importante dada a aplicação do sistema de tempo real. O uso do *xPC Target* nos experimentos realizados neste trabalho não apresentou quaisquer anomalias ou problemas, no entanto, isso não pode ser usado como parâmetro para classificar um RTOS quanto a sua robustez. Portanto, um melhor estudo quanto a robustez do *xPC Target* e de outros RTOS como o QNX, RTAI Linux, VXWorks, etc, seria de grande valia para podermos avaliar o emprego destes sistemas operacionais em diferentes aplicações.

7.2.3 REFINAMENTO DO ALGORITMO INS/GPS QUANTO À FALTA DE GPS E TRATAMENTO DE RUÍDOS

Sabe-se que a falta de GPS em sistemas de navegação inercial com sensores MEMS é altamente problemática. Neste trabalho vimos que um simples artifício como o de manter os últimos dados do GPS quando há falta deste se mostrou bastante eficiente. Técnicas muito mais elaboradas que esta podem ser utilizadas na previsão dos sinais do GPS em sua ausência, o que minimizaria ainda mais os erros do INS. Uma técnica que vem sendo bastante utilizada na "filtragem" de sinais digitais é a Transformada *Wavelet*, sendo perfeitamente aplicável a sistemas de navegação, portanto o uso desta ferramenta assim como o uso de outras técnicas de filtragem de sinais digitais em tempo real seria de grande valor para o refinamento do algoritmo.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- BEKIR, E. **Introduction to Modern Navigation Systems.** World Scientific, 2007.
- BRITING, K. R. **Inertial Navigation Sistems Analysis.** 1971.
- DABNEY, J. B. e HARMAN, T. L. **Mastering Simulink.** Prentice Hall, 2003.
- DE SOUZA PINTO, J. L. **Implementação de uma Metodologia para Avaliação do Desempenho em Sistemas Inerciais Puros.** Dissertação de Mestrado, Instituto Militar de Engenharia, 2007.
- FRANKLIN, G. F., POWELL, J. D. e WORKMAN, M. **Digital Control of Dynamic Systems.** Addison-Wesley, 3.ed edition, 1998.
- GARMIN INTERNATIONAL, I. **GPS 18 Technical Specifications.** 2005.
- IEEE. IEEE Std-952. **IEEE Standard Specification Format Guide and Test Procedure for Linear, Single Axis Interferometric Fiber Optical Gyros.** 1977.
- IEEE. IEEE Std-1293. **IEEE Standard Specification Format Guide and Test Procedure for Linear, Single Axis, nom Gyroscopic Accelerometers.** 1998.
- IEEE. IEEE Std-1554. **IEEE Recommended Practice for Inertial Sensors Test Equipment, Instrumentation, Data Acquisition and Analysis.** 2005.
- IMU400-DATASHEET, C. **IMU400 - 6DOF Inertial Measurement Unit.**
- LITMANOVICH, Y. A., LESYUCHEVSKY, V. M. e GUSINSKY, V. Z. Two new classes of strapdown navigation algorithms. **Journal of Guidance, Control, and Dynamics**, 2000.
- MATHWORKS-AEROSPACE, T. **Aerospace Blockset 3 Users Guide.** 2008.
- MATHWORKS-RTW, T. **Real-Time Workshop® 7 Users Guide.** 2007.
- MATHWORKS-SIMULINK, T. **SIMULINK Dynamic System Simulation for MATLAB.** 1999.
- MATHWORKS-XPC, T. **xPC Target 3 Users Guide.** 2008.
- NAV420-DATASHEET, C. **NAV420 - GPS-AIDED MEMS INERTIAL SYSTEM.**
- ROCHA, L. V. **Algoritmo de Nivelamento e Alinhamento de um Sistema de Navegação Inercial do Tipo Solidário ("Strapdown").** Dissertação de Mestrado, Instituto Militar de Engenharia, 2006.
- ROGERS, R. M. **Applied Mathematics in Integrated Navigation Systems.** American Institute of Aeronautics and Astronautics, Inc., 2000.

SAVAGE, P. G. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. **Journal of Guidance, Control and Dynamics**, 1998a.

SAVAGE, P. G. Strapdown inertial navigation integration algorithm design part 2: Velocity and position. **Journal of Guidance, Control and Dynamics**, 1998b.

SCHWARZ, K. P. e WEI, M. **INS/GPS Integration for Geodetic Applications: Lecture Notes ENGO 623**. Dept. of Geomatics Eng., The University of Calgary, Calgary, Canada, 2000.

SHIN, E.-H. **Accuracy Improvement of Low Cost INS/GPS for Land Applications**. Dissertação de Mestrado, University of Calgary, 2001.

TITTERTON, D. H. e WESTON, J. **Strapdown Inertial Navigation Technology**. Peter Peregrinus Ltd, 1997.