



Laboratório de Sistemas de Computação e Robótica - CORO

Programa de Pós Graduação em Engenharia Elétrica

Universidade Federal de Minas Gerais

Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG Brasil

Fone: +55 31 3409-3430

Emulação de uma Unidade de Medições Inerciais para Testes *Hardware-In-the-Loop*

Gabriel Silva Ribeiro

Dissertação submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Luciano Cunha de Araújo Pimenta, Doutor

Coorientador: Prof. Guilherme Augusto Silva Pereira, Doutor

Belo Horizonte, março de 2014

Dedicatória

*Ao meu pai, José Divino
À minha madrasta, Tia Sônia
Meu irmão, Luiz Fernando
E ao meu sobrinho e afilhado Tarcísio*

Agradecimentos

É chegado o momento de agradecer e as lembranças dos momentos difíceis passados ao longo desse período de mestrado me vêm à memória e me levam à emoção. Se esse texto fosse escrito à mão certamente que alguns borrões de lágrimas manchariam essa página pois muitos foram o momentos de superação para que eu chegasse até aqui. Ainda assim, é com muita satisfação e felicidade que agradeço a todos aqueles que me ajudaram. Sei que sozinho eu jamais teria conseguido.

Agradeço aos meus orientadores Luciano Pimenta e Guilherme Pereira. Luciano obrigado pela paciência e compreensão que teve comigo. Guilherme, você é para mim um exemplo de Dedicção e Profissionalismo, que certamente me influenciará pelo resto da minha vida profissional e também pessoal, agradeço pelas cobranças nos momentos necessários e pelas ideias que ajudaram em muito no desenvolvimento desse trabalho.

Rogério, meu Grande Amigo, é impressionante como horas de dúvidas se resolviam com minutos de conversas com você. Não tenho como te agradecer pelos inúmeros momentos que você dedicou a me ajudar nesse trabalho. A única coisa que posso lhe dizer é que você tem a minha eterna gratidão e que pode contar comigo no que precisar.

Agradeço a todos os professores que contribuíram para minha formação ao longo do período de mestrado. Aos Professores Davies William, Felipe Campelo, Antônio Braga, Marcos Severo, Bruno Adorno, João Vasconcelos. Agradeço em especial ao Professor Rodney Saldanha pelos gestos de sabedoria aos quais tive o prazer de contemplar durante nosso tempo de convívio.

Agradeço aos colegas do Laboratório de Sistemas de Computação e Robótica - CORO- pelos momentos de descontração e compartilhamento de ideias que me deram forças para seguir em frente. Obrigado Jullierme, Ernesto, Natália, Reza, Kossar, Diana, Fredy, Breno, Daniel, Lucas, Gilberto e todos os outros colegas. Lembro-me também de todos os colegas que “sofreram” comigo ao longo das disciplinas, um grande abraço a todos vocês.

Obrigado ainda aos meus colegas de serviço lá no SENAI - CETEL em especial ao Osmane, Marlon e Emanuel, meus irmãos, que compartilharam comigo bons e maus momentos e me deram muito apoio nessa caminhada. Obrigado aos colegas Alexandre, Júlio, Rildo, Jackson, Rinaldo, Vitor, Flávio, Edmar, Devanir, Saunaray, Diego, Nilton e

todos os demais.

Não poderia deixar de agradecer também ao PPGEE - UFMG pela mudança e amadurecimento que me proporcionou e às agências CAPES, CNPq, FINEP e FAPEMIG pelo apoio financeiro ao projeto.

Agradeço a Deus por estar presente em cada instante de minha vida, me dando força para superar os momentos difíceis com a sabedoria necessária para encontrar em cada desafio uma forma de crescer e me tornar uma pessoa melhor.

Sumário

Resumo	xi
Abstract	xiii
Lista de Figuras	xvi
Lista de Tabelas	xvii
Lista de Símbolos	xix
Lista de Abreviações	xxi
1 Introdução	1
1.1 Contexto	2
1.2 Motivação	3
1.3 Objetivos	3
1.4 Organização do Texto	4
2 Revisão Bibliográfica	5
2.1 Introdução	5
2.2 <i>Model-Based Design</i>	6
2.3 Simulação Hardware-In-the-Loop	10
2.4 Sensores	11
2.4.1 Acelerômetros MEMS	12
2.4.2 Girômetros MEMS	16
2.4.3 Magnetômetros	22

2.5	Parâmetros dos Sensores	25
2.6	Estimação de Atitude	27
3	Estrutura do Emulador IMU para testes HIL	29
3.1	Introdução	29
3.2	A Unidade de Medições Inerciais Real	30
3.3	A Unidade <i>Host</i>	32
3.4	A Unidade DSP	33
3.4.1	Interface de Comunicação Serial - SCI	35
3.4.2	Comunicação I2C	37
4	Modelagem dos Sensores	41
4.1	Introdução	41
4.2	<i>Embedded Target</i>	41
4.3	Os Modelos dos sensores	42
4.3.1	Acelerômetro	42
4.3.2	Girômetro	44
4.3.3	Magnetômetro	45
4.3.4	<i>Bias</i> e Densidade Espectral do Ruído	48
4.3.5	A IMU/HIL	52
4.3.6	Blocos de Otimização de Código da Texas: IQmath	53
5	Experimentos & Resultados	55
5.1	Introdução	55
5.2	Levantamento de Dados Estáticos dos Sensores	55
5.3	Dados Dinâmicos dos Sensores	58
5.4	Eficiência do Código Gerado	62
6	Conclusões	65
6.1	O Emulador IMU para Testes <i>Hardware-In-the-Loop</i> (IMU/HIL)	65
6.2	Proposta para Trabalhos Futuros	66
	Bibliografia	72

A Comunicação I2C:	
Múltiplos Endereços de escravo	73
B Equações do Girômetro	77

Resumo

Veículos Aéreos Não Tripulados (VANTs) apresentam uma grande quantidade de aplicações civis e militares. Essas aplicações incluem operações de segurança, vigilância de fronteiras, monitoramento de tráfego, inspeção de sistemas distribuição de energia, etc.

O domínio das tecnologias empregadas no desenvolvimento de VANTs pode proporcionar contribuições importantes para a evolução da pesquisa e desenvolvimento tecnológico de um país. Nesse contexto, a UFMG vem desde 2004 apresentando diversos resultados na produção de VANTs incluindo o desenvolvimento de sistemas de instrumentação e controle, de plataformas aéreas e de sistemas de planejamento de missão e operação.

No contexto geral dos projetos de pesquisa da UFMG, esse trabalho corresponde a uma das etapas para a geração de plataformas aéreas autônomas capazes de realizar missões colaborativas e tem o objetivo de desenvolver um emulador de uma Unidade de Medições Inerciais (IMU) para testes *Hardware-In-the-Loop* (HIL).

Uma contribuição importante desse trabalho foi o uso da metodologia *Model-Based Design* para geração automática dos códigos dos modelos dos sensores, a partir da plataforma *Embedded Target* do MATLAB /Simulink® para integração com os DSPs da Texas Instrumens.

Os resultados obtidos mostraram que os códigos gerados são eficientes em termos de tempo de execução e ocupação de memória e que os modelos dos sensores apresentam valores coerentes com o esperado, mostrando assim que o emulador desenvolvido atendeu ao objetivo proposto.

Abstract

Unmanned Aerial Vehicles (UAVs) have wide range of both civilian and military applications. These applications include security operations, border surveillance, traffic monitoring, systems inspection, etc.

The domain of technologies used in the development of UAVs can provide important contributions for the research and technological development of a country. In this context UFMG has, since 2004, presented several results in the production of UAVs including the development of instrumentation and control systems, aerial platforms and mission planning and operation.

In the general context of UFMG's research projects, this work corresponds to one of the steps for generating autonomous aerial platforms able to perform collaborative tasks and aims to develop a emulator of Inertial Measurement Unit (IMU) for Hardware-In-the-Loop (HIL) tests.

An important contribution of this work was the use of *Model-Based Design* methodology for sensor models automatic code generation from the *Embedded Target* platform of MATLAB/Simulink® for integration with the DSPs Texas Instrumens®.

The results showed that the generated codes are efficient in terms of execution time and memory occupancy and that the models of the sensors are consistent with the expected values thus showing that the developed HIL simulator met the objective.

Lista de Figuras

2.1	Modelo “V”	7
2.2	Processo de desenvolvimento MBD	9
2.3	Simuladores HIL comerciais	11
2.4	Sistemas MEMS	12
2.5	Representação do Acelerômetro	13
2.6	Estrutura capacitiva de Acelerômetro	14
2.7	Dispositivo de detecção com amplificador	17
2.8	Representação do Girômetro	17
2.9	Dedução para força de Coriolis	18
2.10	Girômetro MEMS	21
2.11	Geometria básica de um sensor AMR	23
2.12	AMRs em ponte de Wheatstone	25
3.1	HIL para emulação da IMU	29
3.2	Diagrama funcional do acelerômetro ADXL345	31
3.3	Diagrama funcional do girômetro ITG3200	32
3.4	Diagrama funcional do magnetômetro HMC5883L	33
3.5	Formatos de dados na interface SCI	36
3.6	Validação e Sincronização de dados na interface SCI	37
3.7	Barramento I2C para comunicação HIL/Cabeça Sensora	37
3.8	Validação dos Dados na comunicação I2C	38
4.1	Representação do Sentido Positivo para Detecção da Aceleração	43
4.2	Modelo do Acelerômetro no MATLAB/Simulink	44
4.3	Modelo do Girômetro no MATLAB/Simulink	45
4.4	Modelo do Magnetômetro no MATLAB/Simulink	46
4.5	Blocos de dados para cálculo da Variância de Allan	50
4.6	Modelo da Unidade de Medições Inerciais no MATLAB/Simulink	53
4.7	Exemplo de utilização da biblioteca “IQmath” no modelo da IMU/HIL	54
5.1	Dados do Magnetômetro Real para rotação de 270° a 0°	56
5.2	Coleta de Dados dos Sensores sob superfície paramagnética plana	57

5.3	Dados do Acelerômetro Real e Emulado em condições estáticas	58
5.4	Coleta de Dados dos Sensores sob superfície do Manipulador	59
5.5	Esquema para geração de atitude Real	60
5.6	Comparação entre as atitudes	61
5.7	Tempo de Execução dos cálculos do Modelo da IMU/HIL	63
A.1	Periférico de comunicação I2C do DSP 28x	74
A.2	Código do endereço I2C do HIL	74
A.3	Temporização flags I2C	75
A.4	Flag SDIR	76
B.1	Comparação Gyrometros	78

Lista de Tabelas

3.1	Principais características de configuração para sensores da IMU.	35
4.1	Valores dos parâmetros dos sensores.	51
5.1	Comparação das Condições Estáticas dos Sensores da IMU real e Emulada.	58
5.2	Ocupação da Memória pelo Código Gerado.	63

Lista de Símbolos

ϕ	Ângulo de rolamento;
θ	Ângulo de arfagem;
ψ	Ângulo de guinada ou proa;
a_x	Aceleração linear em relação ao eixo-x;
a_y	Aceleração linear em relação ao eixo-y;
a_z	Aceleração linear em relação ao eixo-z;
Ω	Velocidade angular
ρ	Resistividade específica
H	Vetor Campo Magnético
$\sigma^2(\tau)$	Variância de Allan
p	Velocidade angular em relação ao eixo-x no Referencial do Veículo;
q	Velocidade angular em relação ao eixo-y no Referencial do Veículo;
r	Velocidade angular em relação ao eixo-z no Referencial do Veículo;
P	Matriz de covariância do erro de estimação do EKF;
Q	Matriz de covariância de processo;
R	Matriz de covariância de medição;
S	Matriz de fator de escala para os sensores triaxiais;
\mathbf{x}_k	Vetor de estados no instante k ;
\mathbf{u}_k	Vetor de entrada no instante k ;
\mathbf{g}	Vetor da aceleração da gravidade;

Lista de Abreviações

A/D	Analógico para Digital;
ADC	<i>Analog-to-Digital Converter;</i>
AHRS	<i>Attitude and Heading Reference System;</i>
ASIC	<i>Application Specific Integrated Circuit;</i>
bps	bits por segundo;
CI	Circuitos Integrados;
CPU	<i>Central Processing Unit ;</i>
DCM	<i>Direction Cosine Matrix;</i>
EKF	<i>Extended Kalman Filter;</i>
E/S	Entrada/Saída;
FIFO	<i>First-In-First-Out;</i>
GPS	<i>Global Positioning System;</i>
HIL	<i>Hardware-In-the-Loop;</i>
ISR	<i>Interrupt Service Routine;</i>
IMU	<i>Inertial Measurement Unit;</i>
KF	<i>Kalman Filter;</i>
LSB	<i>Least Significant Bit;</i>
MBD	<i>Model-Based Design;</i>
MEMS	<i>Micro-Electro-Mechanical System;</i>
RAM	<i>Random Access Memory;</i>
RMS	<i>Root Mean Square;</i>
SPI	<i>Serial Peripheral Interface;</i>
UAV	<i>Unmanned Aerial Vehicle;</i>
VANT	Veículo Aéreo Não-Tripulado;
UART	<i>Universal Asynchronous Receiver/Transmitter;</i>

Introdução

“People want success but keep running away from problems, and yet it is only in tackling problems that success is achieved.”

Josiah J. Bonire

Veículos Aéreos Não Tripulados (VANTs) têm recebido uma considerável atenção tanto em aplicações civis quanto militares. Essas aplicações incluem operações de reconhecimento, segurança, vigilância de fronteiras, busca e resgate, sensoriamento remoto, monitoramento de tráfego, inspeção de sistemas de distribuição, etc [Adiprawita et al., 2008]. Um VANT pode cumprir suas tarefas de forma exaustiva sem colocar em risco um piloto e geralmente possui um custo operacional menor comparado com aviões tripulados.

O piloto automático consiste de um hardware e algoritmo de navegação de alto nível que opera leis de controle que devem garantir a estabilidade de voo e o funcionamento correto do VANT [Adiprawita et al., 2008]. O desenvolvimento do piloto automático é uma etapa crítica que necessita de uma quantidade de testes suficientes para certificar que os modelos matemáticos utilizados no sistema de controle estão corretos.

A menos que se tenha uma confiança muito grande no sistema de controle desenvolvido, não é recomendado fazer ensaios em campo diretamente na aeronave alvo, pois isso poderia levar a consequências indesejáveis e a perdas de recursos em caso de ocorrências de falhas não previstas no processo de implementação. Ao invés disso, deve-se realizar simulações prévias em um ambiente que ofereça menos riscos para o sistema e ao mesmo tempo reproduza de maneira fiel as características do sistema a ser verificado.

A simulação *Hardware-In-the-Loop* (HIL) é uma das formas mais eficientes de se testar e validar os mecanismos de controle para sistemas embarcados e atualmente é uma prática vastamente empregada no desenvolvimento de VANTs. HIL é uma técnica de simulação em tempo real a muito utilizada na indústria aeroespacial [Maclay, 1997] onde os sinais de entrada e de saída de um simulador mostram os mesmos valores de um processo real [Sanvido, 2002]. Tais simuladores nos permitem testar um sistema de controle embarcado sob diferentes condições de trabalho.

Nesse trabalho, foi desenvolvido um emulador de uma Unidade de Medições Inerciais para testes *Hardware-In-the-Loop* com o objetivo de gerar os dados dos sensores da IMU real de um VANT em condições de voo simulado. A principal aplicação dessa

simulação é a verificação do funcionamento do algoritmo de estimação de atitude implementado no protótipo da Cabeça Sensora que irá compor o sistema de navegação de um conjunto de mini-VANTs reais que irão realizar voos autônomos cumprindo missões colaborativas.

Para o desenvolvimento desse emulador foram utilizados o ambiente MATLAB/SIMULINK® para fornecer dados de voos simulados e uma interface de hardware constituída pelo Processador Digital de Sinais TMS320F28335 da Texas Instrumens®, a qual tem a função de executar os modelos dos sensores que equipam a IMU real e realizar a comunicação, via interface I2C, com o sistema embarcado da Cabeça Sensora (que também é composta pelo mesmo tipo de DSP).

A implementação dos modelos para os sensores foi realizada no SIMULINK e os códigos para o DSP foram gerados automaticamente pela plataforma *Embedded Target*, processo que faz parte de uma metodologia para desenvolvimento de projetos em sistemas embarcados conhecida como *Model-Based Design* (MBD). O código gerado mostrou-se eficiente do ponto de vista de tempo de execução, dados apresentados e ocupação de memória.

Os resultados obtidos mostram que a simulação HIL pode ser uma boa ferramenta no ciclo de desenvolvimento de projetos para sistemas inteligentes modernos, uma vez que possibilita a emulação dos sensores em tempo real e com características muito próximas das dos dispositivos reais.

1.1 Contexto

A UFMG, desde 2004, vem realizando pesquisas para o desenvolvimento de VANTs. Algumas de suas principais atividades contemplam o desenvolvimento de sistemas de instrumentação e controle, de plataformas aéreas e de sistemas de planejamento de missão e operação para VANTs.

Através das pesquisas realizadas pelo grupo PDVA (Pesquisa e Desenvolvimento de Veículos Autônomos) foram apresentados vários resultados importantes dos quais pode-se citar, por exemplo, a geração das plataformas aéreas Watchdog e AqVS.

O Watchdog é um VANT de médio porte com capacidade de carga de 30kg e o AqVS é um pequeno motoplanador com propulsão elétrica, o qual tornou-se o primeiro VANT com tecnologia brasileira capaz de voar completamente autônomo. A principal missão do AqVS é o reconhecimento de solo e o monitoramento de equipamentos em terra.

Com os projetos realizados por seus grupos de pesquisa, a UFMG pretende desenvolver plataformas aéreas autônomas com o domínio de todas as tecnologias empregadas para que, dessa forma, possa contribuir com a pesquisa e desenvolvimento do Brasil nesse campo.

O emulador IMU/HIL desenvolvido neste trabalho também faz parte das etapas para a criação dessa plataforma aérea autônoma e no contexto geral do projeto está relacionada ao desenvolvimento de sistemas de controle e navegação.

1.2 Motivação

A crescente demanda por aplicações civis e militares com a utilização de VANTs levam à necessidade de se propor maneiras mais rápidas para o desenvolvimento de modelos comerciais.

Os riscos inerentes ao processo de ensaios em campo e validação dos sistemas de controle para piloto automático de VANTs fazem com que sejam procuradas alternativas de simulação com as características de comportamento de um modelo real.

Atualmente, encontram-se ainda em desenvolvimento pela UFMG alguns projetos para controle e navegação de VANTs, dos quais uma atividade essencial corresponde ao desenvolvimento de um sistema para estimação de atitude, o qual referenciaremos nesse trabalho como “Cabeça Sensora”. É primordial que esse sistema possua baixo custo, peso e tamanho reduzidos para que o mesmo possa ser empregado em mini-VANTs capazes de realizar missões colaborativas.

Nesse contexto, foi publicado recentemente um trabalho onde alguns algoritmos recursivos para estimação de atitude foram comparados entre si e um novo algoritmo baseado em Filtro de Kalman Extendido (EKF) foi proposto [Lima and Torres, 2012]. Esse algoritmo apresentou bons resultados em simulações computacionais e, por isso, foi implementado no protótipo da Cabeça Sensora que irá compor o sistema de instrumentação para os mini-VANTs reais.

A dificuldade de se realizar testes em voo com a Cabeça Sensora ainda em fase de protótipo é uma das principais motivações para o desenvolvimento do emulador IMU/HIL deste trabalho. A simulação HIL é uma das formas mais eficientes de se testar e validar sistemas de controle para VANTs e vem sendo vastamente empregada nesse campo [Adiprawita et al., 2008; Jung and Tsiotras, 2007; Mueller, 2007]. O sistema HIL é uma ferramenta indispensável para certificação rápida tanto do hardware do veículo aéreo quanto do software de controle, enquanto executa testes de simulação de voo com custos e esforços mínimos [Jung and Tsiotras, 2007].

As aplicações para o sistema HIL podem ser expandidas ainda para fins didáticos, proporcionando uma plataforma de risco nulo em estudos experimentais de metodologias diferentes para controle de sistema embarcados e desenvolvimento de pilotos automáticos para VANTs [Sanvido, 2002].

1.3 Objetivos

O presente trabalho foi realizado com o seguinte objetivo geral: desenvolver um emulador de uma Unidade de Medições Inerciais (IMU) aplicado à realização de testes *Hardware-In-the-Loop*: IMU/HIL.

Como o sistema desenvolvido está equipado com placa de interface DSP e deve possibilitar testes em condições de voo simulado através de modelos desenvolvidos no MATLAB/SIMULINK®, os objetivos específicos da dissertação foram:

1. desenvolver os modelos matemáticos para os sensores da IMU;
2. especificar hardware e implementar software para o sistema;
3. realizar testes e verificações experimentais para validação do sistema.

1.4 Organização do Texto

Esta dissertação está organizada da seguinte maneira: O Capítulo 2 é dedicado à revisão bibliográfica. No Capítulo 3 é apresentada a arquitetura do emulador IMU/HIL desenvolvido, incluindo detalhes sobre as interfaces de comunicação e o DSP utilizado, mostrando as principais funcionalidades e recursos disponíveis no projeto. No Capítulo 4 são apresentados os modelos dos sensores implementados e a metodologia para geração dos algoritmos. O Capítulo 5 apresenta os testes e experimentos realizados no sistema e ainda uma discussão sobre os resultados obtidos. Por fim, no Capítulo 6, são apresentadas as conclusões e as propostas para trabalhos futuros.

Revisão Bibliográfica

“Engineering is not only a learned profession, it is also a learning profession, one whose practitioners first become and then remain students throughout their active careers.”

William L. Everitt

2.1 Introdução

Sistemas de medição são utilizados em veículos terrestres, marítimos, aéreos e espaciais com a intenção de fornecer dados referentes à velocidade, posição e/ou atitude da embarcação para que, a partir dessas informações, um operador ou sistema de controle possa tomar decisões de guiagem do veículo.

Esses sistemas podem extrair os dados de posição e atitude do veículo a partir de informações provenientes de fontes externas ou internas à embarcação. Exemplos de fontes externas de informação podem ser ondas de rádio emitidas de estações terrestres de telemetria ou ainda sinais de satélites. Esses sistemas podem oferecer a vantagem de fornecer informações absolutas de medições, como é o caso do GPS (*Global Positioning System*), mas, em contrapartida, dependem da visada ou da abrangência dos sinais entre os dispositivos emissores e receptores além de sofrerem com interferências externas.

Já a Navegação Inercial independe de qualquer sinal transmitido externamente ao veículo. Ela conta com um conjunto de sensores combinados com computadores embarcados capazes de fornecer dados de navegação continuamente e imunes a interferências externas. A Navegação Inercial consiste em determinar os dados de navegação do veículo (velocidade, posição e atitude) com base na medida das suas acelerações ou velocidades relativas em direções espaciais conhecidas [Rogers, 2003].

Os sensores básicos usados nos sistemas de Navegação Inercial são os girômetros e os acelerômetros que constituem a Unidade de Medições Inerciais (IMU) da embarcação. Além disso, são necessários computadores capazes de executar os cálculos de integração numérica dos sinais de saída desses sensores e fornecer então os dados de navegação.

Como nesse trabalho foi utilizada a metodologia de *Model Based Design* (MBD) no desenvolvimento do emulador de uma Unidade de Medições Inerciais para testes *Hardware-In-the-Loop* HIL, o presente capítulo apresenta os conceitos de MBD e sistemas

HIL, além dos conhecimentos básicos para se entender o princípio de funcionamento dos sensores inerciais, a saber girômetro e acelerômetro, e ainda do magnetômetro, que juntos compõem a IMU emulada (é importante ressaltar que as equações aqui apresentadas são meramente para entendimento do princípio de funcionamento dos sensores, ou seja, essas equações não foram utilizadas na modelagem dos sensores).

Os sensores apresentam ainda erros de medição (bias, ruídos, erros de fator de escala, etc) que interferem no processo de obtenção das informações de navegação, por isso, também são mostradas essas fontes de erros, cujo entendimento faz-se necessário para emular as características dos sensores.

2.2 *Model-Based Design*

Model-Based Design (MBD) é uma metodologia que implementa a modelagem matemática de sistemas de controle de forma gráfica e visual permitindo assim que um projeto seja entendido de forma mais ampla por uma equipe de desenvolvimento. Ao invés de usar estruturas complexas e linhas de códigos, os projetistas podem utilizar MBD para definir modelos com características funcionais avançadas usando blocos de construção.

Essa metodologia surge da necessidade da inserção de modelos e requisitos no processo de desenvolvimento dos sistemas de controle a medida em que crescem suas funcionalidades e algoritmos de controle mais poderosos precisam ser incorporados.

A metodologia MBD inicia através da modelagem do sistema, onde são capturadas as ideias e demandas. O modelo é uma especificação executável que permite uma equipe colaborar desde detalhes de engenharia até os domínios da modelagem. É possível simular o modelo ou realizar a prototipagem rápida do mesmo para agilizar o processo de análise física do projeto. Com as simulações é possível ajustar e aprimorar o modelo para otimizar sua implementação.

Os mecanismos tradicionais baseados em texto são inadequados para a natureza complexa dos sistemas de controle modernos. Devido às limitações e baixa qualidade das ferramentas gráficas antigas, engenheiros de projeto dependiam fortemente de programação baseada em texto para a implementação dos modelos matemáticos do sistema. Contudo, essa implementação era difícil, demorada e altamente propensa a erros. Além disso, a depuração de programas baseados em texto é um processo lento e tedioso, exigindo muita tentativa e erro antes que um modelo final sem falhas possa ser implementado.

Esses desafios são superados com o uso das ferramentas de modelagem gráfica usadas atualmente em todos os aspectos do projeto. Usando as ferramentas de MBD do MATLAB/Simulink, por exemplo, a NASA projetou o sistema de navegação e desembarque para sua pesquisa de exploração em Marte, Alstom estabeleceu o recorde mundial para transporte ferroviário de alta velocidade e a Universidade Jons Hopkins construiu um braço robótico controlado pelo cérebro [MathWorks, 2011]. Várias em-

presas ao redor do mundo também utilizam essas ferramentas como, por exemplo, *Epson Toyocom*, *ABB*, *Océ*, *General Motors* e *Lookheed Martin* [MathWorks, 2014].

A EMBRAER, empresa brasileira que fabrica aeronaves de pequeno e médio porte, utiliza essa metodologia em diversas etapas do desenvolvimento de seus produtos, desde modelos aerodinâmicos [Cavalcanti and Papini, 2004], desenvolvimento de piloto automático [Motta et al., 2009] e várias outras etapas.

A metodologia empregada em MBD se enquadra em um modelo padronizado para Verificação e Validação de sistemas conhecido como “Modelo V”. Esse modelo é vastamente utilizado em Engenharia de Sistemas e segue às etapas apresentadas no esquema da Figura 2.1. A parte central do “V” mostra a definição do projeto, implementação e verificação do processo [ITS, 2007].

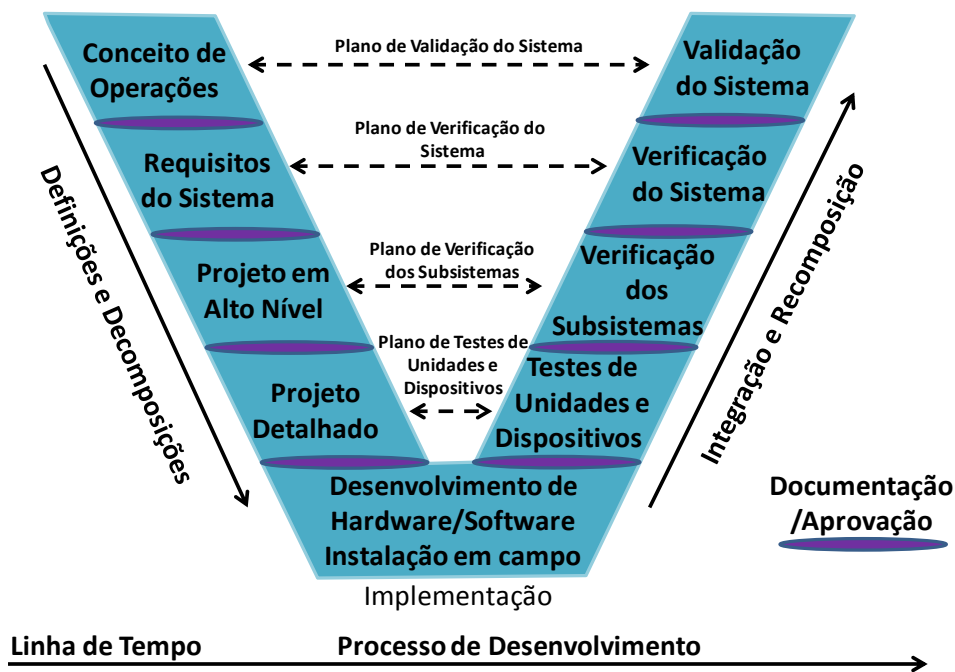


Figura 2.1: Modelo “V” de Verificação e Validação para desenvolvimento de sistemas Elaborada a partir de [ITS, 2007]

Conforme mostrado no “V”, a abordagem de engenharia de sistemas define os requisitos do projeto antes que as escolhas tecnológicas sejam feitas e que o sistema seja implementado. No lado esquerdo do “V”, a definição do sistema progride a partir de uma visão geral do usuário do sistema para uma especificação detalhada de seu projeto. O sistema é decomposto em subsistemas, e os subsistemas são decompostos em componentes - um sistema muito grande pode ser dividido em subsistemas menores através de camadas de decomposição. À medida em que o sistema é decomposto, as exigências também são decompostas em requisitos mais específicos que são atribuídos aos componentes do sistema.

A evolução do processo de desenvolvimento do sistema é apoiada na etapa anterior do modelo "V". Por exemplo, o estabelecimento dos Conceitos de Operação apóia o desenvolvimento requisitos do sistema. Um conjunto básico de requisitos do sistema, em seguida, apóia o projeto do sistema. O hardware e software são executados na parte inferior do "V", e os componentes do sistema são, em seguida, integrados e verificados de forma iterativa à direita. Em última análise, o sistema completo é validado para medir o quão bem ele atende às necessidades do usuário.

A Figura 2.2 mostra que a metodologia MBD põe o modelo no centro do processo de desenvolvimento permitindo que os engenheiros possam criar especificações executáveis, gerar automaticamente o código embarcado, e executar verificação e validação desses modelos [Sandmann and Thompson, 2008]. Sendo assim, MBD permite melhorar a eficiência de um projeto, uma vez que possibilita [Smith et al., 2007]:

- usar um ambiente de projeto comum em equipes de projeto;
- ligar os projetos diretamente aos seu requisitos;
- integrar testes com o projeto para identificar e corrigir erros de forma contínua;
- refinar algoritmos através de simulação de vários domínios;
- gerar códigos de software embarcado automaticamente;
- desenvolver e reutilizar conjuntos de testes;
- gerar documentação automaticamente;
- reutilizar projetos para implantar sistemas em vários processadores e dispositivos de hardware

Apesar de todas as vantagens citadas sobre MBD sua utilização deve ser empregada com critério uma vez que a metodologia não substitui nem descarta a necessidade da análise criteriosa de uma equipe de engenharia, tanto para a geração de modelos quanto para a análise dos resultados obtidos para os códigos gerados. MBD também não torna dispensável a presença de engenheiros de software ou do processo para atingir resultados favoráveis [Gothard, 2006].

Técnicas de teste "*In-the-Loop*" também fazem parte da metodologia MBD e permitem a reutilização das situações e ambientes de testes do modelo para execução com a aplicação durante vários estágios de produção [Erkinen and Conrad, 2008]. No contexto do desenvolvimento de sistemas, os termos mais utilizados são "*Model-In-the-Loop*" (MIL), "*Software-In-the-Loop*" (SIL), "*Processor-In-the-Loop*" (PIL) e "*Hardware-In-the-Loop*" (HIL) [Murphy et al., 2008].

MIL é o caso em que se utiliza um modelo de controle para trabalhar com um modelo da planta. O modelo do controle encontra-se em execução no Simulink, por exemplo, e está ligado diretamente a um modelo da planta física dentro do mesmo



Figura 2.2: Processo de desenvolvimento na metodologia MBD

Fonte: Figura Elaborada com base em [Sandmann and Thompson, 2008]

diagrama no Simulink. O desenvolvimento é extremamente rápido nesta fase uma vez que se pode fazer pequenas alterações no modelo de controle e testar imediatamente seu efeito no sistema.

No teste SIL o sistema de controle não é mais composto por blocos do Simulink, mas pelo código em C ou C++ gerado (automaticamente ou por codificação manual) a partir do modelo. A simulação é um pouco mais “real” nesse contexto pois falhas de codificação começam a se tornar evidentes, porém a interação com o modelo diminui com relação à apresentada em MIL. Além disso, tanto o sistema de controle quanto o modelo da planta executam no mesmo ambiente de simulação.

Na simulação PIL o código gerado para o sistema de controle é embarcado no processador alvo e interage com o modelo da planta que ainda está em execução na plataforma de simulação. Este teste foi concebido para expor os problemas com a execução no ambiente embarcado ou para se observar comportamentos que não podem ser verificados em ambiente de simulação.

No teste HIL o código é gerado para o modelo da planta e executa em um computador de tempo real altamente determinístico e com capacidade para calcular a dinâmica da planta física e emular suas reações. HIL é utilizado quando o sistema de controle já encontra-se instalado no dispositivo alvo e pode interagir apenas com a planta através das interfaces de entrada e saída apropriadas do controlador. A próxima seção trata com maiores detalhes essa técnica de simulação.

2.3 Simulação Hardware-In-the-Loop

Hardware-In-the-Loop (HIL) é uma técnica de simulação em tempo real onde “os sinais de entrada e de saída do simulador mostram os mesmos valores de um processo real” [Sanvido, 2002] criando um sistema que imita as reações da planta física que seria conectada a sua respectiva plataforma de controle. Os testes com simulação HIL tiveram suas primeiras aplicações na indústria aeroespacial devido aos aspectos críticos de segurança envolvidos no desenvolvimento de softwares para os sistemas de controle de voo [Maclay, 1997].

Em uma simulação HIL a plataforma de controle, como por exemplo um microcontrolador, microprocessador ou controlador programável, é conectado a dispositivos que realizam uma simulação computacional do sistema controlado [Sanvido, 2002]. Para tal esses dispositivos devem ser capazes de atender às restrições de tempo real da planta física simulada, ou seja, devem reproduzir fiel e corretamente as características reais do sistema, de modo a entregar o resultado esperado dentro de um prazo específico.

Um sistema HIL é composto por elementos de hardware e software dotados de modelos matemáticos. Esses modelos devem descrever o funcionamento da planta física de forma realística de modo que quando o simulador HIL for conectado ao sistema de controle, este interaja com o HIL da mesma forma que iria interagir com a planta física. Dessa forma, pode-se dizer que um simulador HIL “fecha” o loop de controle com os hardwares dedicados a isso, emulando assim os dados dos sensores e/ou atuadores integrantes desse sistema.

De uma forma geral, várias são as vantagens da utilização da simulação HIL no desenvolvimento de sistemas embarcados ou de controle. A maioria dessas vantagens estão relacionadas com a redução dos custos, riscos, e do tempo de desenvolvimento e testes de dispositivos de controle. Além disso, um simulador HIL possibilita o teste de condições extremas e/ou combinações críticas da planta física levando as unidades de controle aos seus limites e possibilitando o isolamento de erros. Através da simulação HIL pode se testar um subsistema de hardware de forma isolada, sem que o conjunto inteiro esteja pronto, o que torna o processo de desenvolvimento mais eficaz, desde o projeto até a implementação.

Apesar de todas essas vantagens, dentre outras não citadas, uma simulação HIL não tem a intenção de substituir experimentos reais na planta física, mas diminuir a carga de testes nesses sistemas, otimizando assim o tempo e custos no desenvolvimento dos mecanismos de controle do processo.

As aplicações para o sistema HIL podem ser expandidas ainda para fins didáticos proporcionando uma plataforma de risco nulo em estudos experimentais de metodologias diferentes para controle de sistemas embarcados ou desenvolvimento de pilotos automáticos para VANTs.

Atualmente, várias empresas estão investindo no desenvolvimento de plataformas

comerciais para sistemas HIL oferecendo produtos para o desenvolvimento de sistemas embarcados e de controle de plantas industriais como pode ser observado na Figura 2.3.

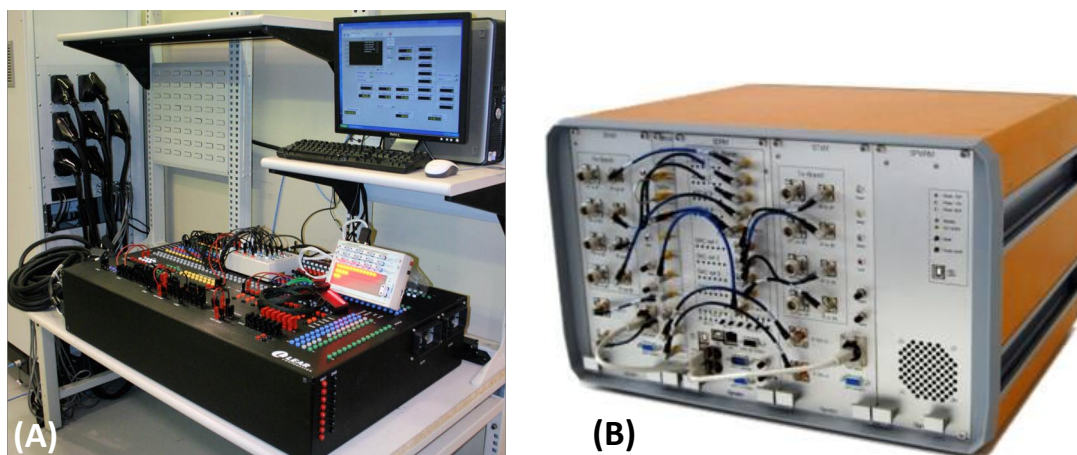


Figura 2.3: Simuladores HIL comerciais: (A) NI VeriStand da National Instruments; (B) HaLo - Plataforma HIL para Matlab e Simulink desenvolvido pela SIGNALION que também pertence à National

2.4 Sensores

A atitude de uma aeronave é determinada pela orientação de seus eixos com relação a um sistema de referências. Ela pode ser obtida através de um conjunto de sensores constituindo um módulo de instrumentação vinculado a uma unidade de processamento e fusão das informações fornecidas por esses sensores.

Durante décadas os sensores mecânicos dominaram a engenharia inercial para realização de estimação da atitude, entretanto, devido a algumas limitações de desempenho e à grande necessidade de manutenções, esses sensores passaram a ser substituídos por sensores ópticos (RLG ou FOG) em aplicações de ponta nas indústrias aeronáutica e naval [Sherry et al., 2003; Rogers, 2003]. Estes sensores, por não possuírem partes móveis, apresentam um MTBF (*Mean Time Between Failures* - tempo médio entre falhas) maior e um custo menor.

Uma outra alternativa é a utilização de sensores MEMS, do inglês *Micro Electro-Mechanical Systems*. Embora menos precisos do que os utilizados em aplicações militares e de voos comerciais, esses sensores possuem os requisitos necessários para viabilizar o desenvolvimento de sistemas para VANTs de pequeno e médio porte [da Paixao et al., 2011], ou seja, tamanho, peso e custo bastante reduzido. Além disso, sensores de tecnologia MEMS possibilitam a integração de outros circuitos em um mesmo chip como, por exemplo, conversores AD, memórias e interfaces de comunicação, dentre outros recursos que facilitam seu emprego e o desenvolvimento de aplicações.

Os dispositivos MEMS ou Sistemas Microeletromecânicos, correspondem à integra-

ção de elementos mecânicos e eletroeletrônicos em um único substrato de silício por meio da utilização de tecnologias de micro-fabricação.

Todos os dispositivos MEMS são produzidos por técnicas de microusinagem, cujas potencialidades oferecidas impressionam no que diz respeito à criação de microsistemas em dimensões e escalas massivamente reduzidas, conforme ilustra a Figura 2.4.

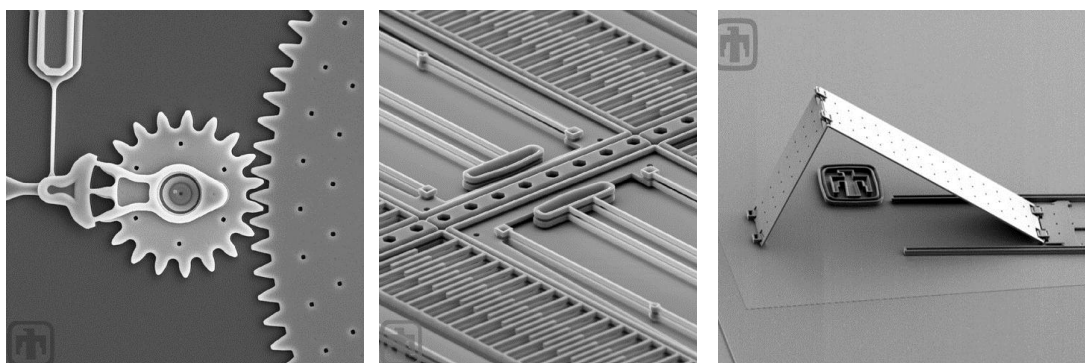


Figura 2.4: Microelementos de sistemas MEMS desenvolvidos por técnicas de microusinagem. Dimensões em dezenas de micrômetros.

O processo de microusinagem consiste em depositar, retirar ou moldar materiais com suas mínimas características na dimensão de micrômetros ou menos, assim os dispositivos mecânicos podem ser criados em escalas miniaturizadas dando origem a peças móveis encapsuladas no interior de um circuito integrado. Essas peças móveis são então acopladas a mecanismos elétricos ou eletroeletrônicos que possibilitam a criação de sensores, atuadores ou sistemas para diversas aplicações, dentre as quais destacamos nesse trabalho a construção dos sensores inerciais para utilização em sistemas de estimação de atitude para VANTs.

Uma das grandes vantagens dos dispositivos MEMS é que, como os mesmos estão construídos no interior de um chip de materiais semicondutores, é possível integrar uma grande quantidade de circuitos de acondicionamento de sinais na própria estrutura do sensor, o que possibilita a extração da grandeza medida sem necessidade de se preocupar com o tratamento do sinal. Nas Subseções 2.4.1 e 2.4.2 são descritos os princípios de funcionamento dos acelerômetros e girômetros MEMS.

2.4.1 Acelerômetros MEMS

Os acelerômetros são dispositivos projetados para medir a aceleração ao longo de um determinado eixo baseando-se, geralmente, no deslocamento provocado por uma força exercida em uma massa de prova ao longo desse eixo de medição.

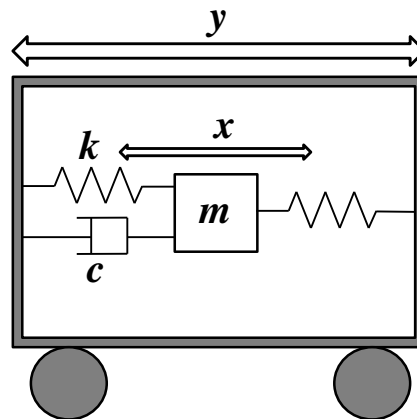


Figura 2.5: Representação do Acelerômetro: sistema Massa-Mola-Amortecedor de 1 grau de liberdade

Um acelerômetro pode ser utilizado para medir acelerações dinâmicas, estáticas (aceleração da gravidade), inclinação, rotação, vibração e colisão, constituindo-se assim um aparelho de elevada utilidade para projetos na área da eletrônica e robótica [Figueiredo et al., 2007].

O princípio básico de funcionamento de um acelerômetro pode ser explicado a partir do esquema massa-mola-amortecedor de um grau de liberdade disposto na Figura 2.5. Esse esquema apresenta uma armação externa que pode se mover livremente para a esquerda ou para a direita na direção y . Internamente existe uma massa de prova suspensa por uma estrutura de molas. Se a armação externa está em repouso, ou em um movimento retilíneo uniforme, a massa de prova mantém-se estacionária na sua posição inicial. Quando a armação é acelerada, a inércia faz com que a massa de prova resista comprimindo uma das molas e estendendo a outra, o que equivale a dizer que existirá um deslocamento relativo na direção x entre a massa suspensa e a armação da estrutura.

Como o deslocamento permitido pela mola é proporcional à força aplicada, a aceleração do corpo é proporcional ao deslocamento da massa, assim essa aceleração pode ser extraída pela aplicação da 2ª Lei de Newton ($F = m \cdot a$). Na realidade, pode-se dizer que para o sistema massa-mola-amortecedor da Figura 2.5 existem basicamente três forças atuantes [Grigorie, 2008], cujos valores absolutos podem ser fornecidos pelas equações:

$$\begin{aligned} F_e &= kx, \\ F_a &= c\dot{x}, \\ F_i &= ma = m\ddot{y}, \end{aligned} \quad (2.1)$$

onde F_e , F_a e F_i são as forças elástica, de amortecimento e a força inercial da armação, respectivamente. k corresponde à constante elástica das molas e c é coeficiente de

amortecimento do amortecedor.

Dessa forma a equação de movimento da massa de prova pode ser expressa como segue

$$m\ddot{y} = m\ddot{x} + c\dot{x} + kx \quad (2.2)$$

Um acelerômetro MEMS corresponde a uma estrutura similar ao sistema exposto, miniaturizado no interior do encapsulamento de um circuito integrado. Dessa forma a aceleração pode ser inferida a partir do deslocamento x no interior o sistema. O amortecimento viscoso é o fornecido pelo ar do interior do encapsulamento [Grigorie, 2008] e pode ser considerado para deslocamentos relativamente grandes (para dimensão de μm) conforme expressão

$$c(x) = \frac{1}{2}\mu A^2 \left[\frac{1}{(d_0 - x)^3} + \frac{1}{(d_0 + x)^3} \right] \quad (2.3)$$

onde μ é a viscosidade do ar, A é a área das placas móveis e $2d_0$ é a distância entre os eletrodos fixos.

A Figura 2.6 ilustra uma construção básica para um acelerômetro MEMS que mede a aceleração em um único eixo. Observa-se nessa ilustração que a massa de prova apresenta a forma de um “pente” cujos dentes correspondem às placas móveis que, juntamente com placas fixas na armação do encapsulamento do sensor, formam a estrutura de um capacitor variável diferencial.

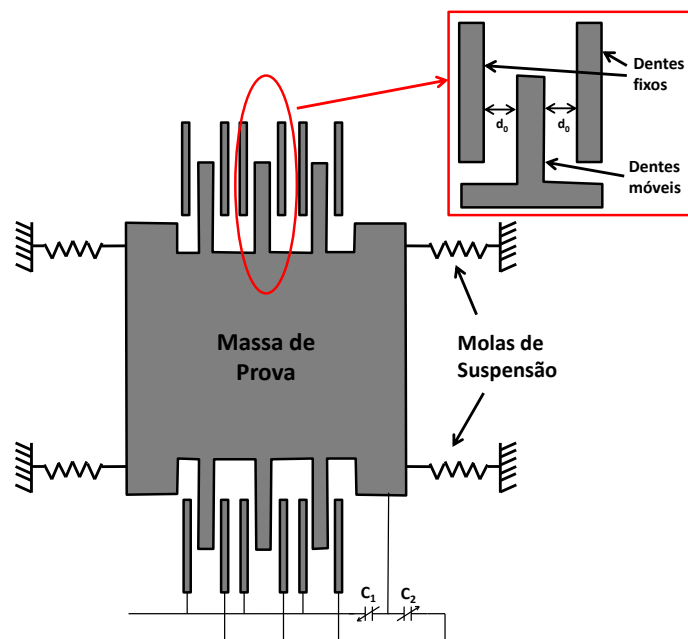


Figura 2.6: Estrutura capacitiva de um Acelerômetro MEMS

Fonte: Figura elaborada com base em [Ang et al., 2004; Boser and Howe, 1996]

Essa estrutura capacitiva do acelerômetro possibilita a obtenção da aceleração a

partir do deslocamento da massa de prova no interior do sensor que se traduz na variação das capacitâncias C_1 e C_2 , denotadas pelas expressões:

$$\begin{aligned} C_1 &= \frac{\varepsilon A}{d_0 - x} = \frac{\varepsilon A d_0}{d_0^2 - x^2} + \frac{\varepsilon A x}{d_0^2 - x^2}, \\ C_2 &= \frac{\varepsilon A}{d_0 + x} = \frac{\varepsilon A d_0}{d_0^2 - x^2} - \frac{\varepsilon A x}{d_0^2 - x^2}, \end{aligned} \quad (2.4)$$

onde $\varepsilon = \varepsilon_0 \varepsilon_r$ corresponde à permissividade elétrica do meio, A à área das placas e d_0 à distância inicial entre elas. x corresponde ao deslocamento da massa de prova.

Se $x \ll d_0$ pode-se considerar a aproximação

$$\begin{aligned} C_1 &= C_0 + \Delta C, \\ C_2 &= C_0 - \Delta C \end{aligned} \quad (2.5)$$

onde

$$\begin{aligned} C_0 &= \frac{\varepsilon A}{d_0}, \\ \Delta C &= \frac{\varepsilon A x}{d_0^2 - x^2} \end{aligned} \quad (2.6)$$

Para que essa variação da capacitância reflita em um sinal elétrico gerado pelo acelerômetro, os eletrodos fixos são alimentados pelas tensões

$$\begin{aligned} v_1 &= V_1 \sin \omega t, \\ v_2 &= -v_1 = -V_1 \sin \omega t, \end{aligned} \quad (2.7)$$

sendo assim as seguintes forças eletrostáticas vão atuar sobre as placas móveis

$$\begin{aligned} F_{el1} &= \varepsilon_0 \frac{A}{2} \frac{v_1^2}{(d_0 - x)^2} = \varepsilon_0 \frac{A}{4} \frac{V_1^2}{(d_0 - x)^2} (1 - \cos 2\omega t), \\ F_{el2} &= \varepsilon_0 \frac{A}{2} \frac{v_2^2}{(d_0 + x)^2} = \varepsilon_0 \frac{A}{4} \frac{V_1^2}{(d_0 + x)^2} (1 - \cos 2\omega t). \end{aligned} \quad (2.8)$$

A componente senoidal presente nas forças eletrostáticas da Equação (2.8) poderia induzir a pensar que a massa de prova seria levada à vibração pela atuação dessas forças. Contudo, isso não ocorre devido ao fato das tensões aplicadas nas placas possuírem uma frequência muito elevada quando comparada com a frequência natural do sistema massa-mola-amortecedor. Nesse caso o sistema mecânico funciona como um filtro passa-baixas de modo que as forças eletrostáticas podem ser consideradas constantes

conforme segue:

$$\begin{aligned}
 F_{el1} &= \varepsilon_0 \frac{A}{4} \frac{V_1^2}{(d_0 - x)^2}, \\
 F_{el2} &= \varepsilon_0 \frac{A}{4} \frac{V_1^2}{(d_0 + x)^2}, \\
 F_{el} &= F_{el1} - F_{el2} = \varepsilon_0 \frac{AV_1^2}{4} \left[\frac{1}{(d_0 - x)^2} - \frac{1}{(d_0 + x)^2} \right].
 \end{aligned} \tag{2.9}$$

Da Equação (2.9) pode-se observar ainda que as forças eletrostáticas são nulas para o caso da não ocorrência de deslocamento x , ou seja,

$$F_{el} = 0$$

se

$$x = 0$$

Considerando-se então a força eletrostática, a Equação (2.2) pode ser reescrita da seguinte forma

$$m\ddot{y} = m\ddot{x} + c\dot{x} + kx - F_{el} \tag{2.10}$$

Usando as Equações (2.3) e (2.9) a Equação (2.10) será

$$\ddot{x} = a - \frac{\mu A^2}{2m} \left[\frac{1}{(d_0 - x)^3} - \frac{1}{(d_0 + x)^3} \right] \dot{x} - \frac{k}{m} x + \varepsilon_0 \frac{AV_1^2}{4m} \left[\frac{1}{(d_0 - x)^2} - \frac{1}{(d_0 + x)^2} \right] \tag{2.11}$$

onde a corresponde à aceleração do sistema. A Equação (2.11) representa o modelo mecânico para o movimento da massa de prova do acelerômetro [Grigorie, 2008].

O sinal elétrico da saída do acelerômetro determina o deslocamento x da massa de prova e é utilizado para a obtenção da aceleração do sistema. Esse sinal é extraído da estrutura capacitiva através do acoplamento do sinal gerado a um circuito amplificador. A Figura 2.7 ilustra uma das possíveis configurações para esse circuito amplificador.

2.4.2 Girômetros MEMS

A atitude é uma informação essencial para qualquer sistema de controle e navegação de veículos aéreos, nesse sentido, o girômetro aplica-se como um dispositivo muito importante, pois o mesmo possibilita extrair as informações de orientação à partir da integração das velocidades angulares medidas no veículo ou da extração do ângulo de orientação propriamente dito.

Embora existam muitas variedades de projetos e princípios de operação para girômetros, nesse trabalho vamos tratar da abordagem que envolve a aplicação das tecnolo-

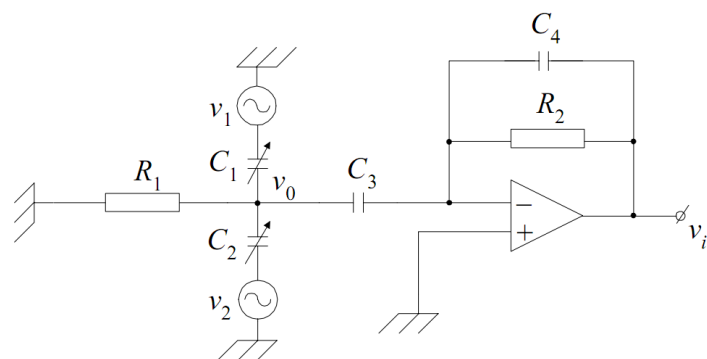


Figura 2.7: Dispositivo de detecção com amplificador de carga

Fonte: Figura retirada de Grigorie [2008] com base em Kraft [1997]

gias MEMS na fabricação desses dispositivos. Nesse caso, pode-se dizer que a maioria das abordagens utilizam de elementos mecânicos vibratórios para detectar variações angulares [Priya and Sushas, 2012].

Os girômetros vibratórios podem ser representados basicamente por um sistema massa-mola-amortecedor de dois graus de liberdade conforme ilustra Figura 2.8. Essa é a primeira diferença importante entre o girômetro e o acelerômetro, cujo sistema básico tem um grau de liberdade. A segunda diferença é que, ao invés de se manter estática e reagir à variação da velocidade linear, a massa de prova no girômetro é conduzida à oscilação ao longo de um eixo (direção de oscilação x) e reage à variação da velocidade angular no eixo perpendicular (direção de detecção y) ainda no plano desse movimento.

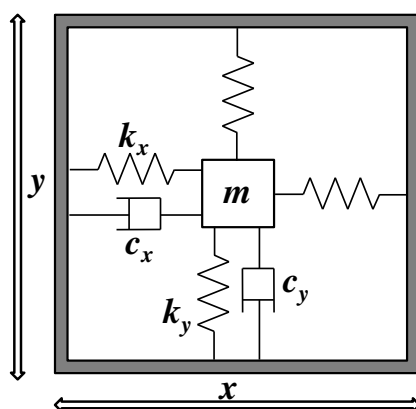


Figura 2.8: Representação do Girômetro: Sistema Massa-Mola-Amortecedor de dois graus de liberdade

Nesse esquema as variações em velocidades lineares, como as provocadas pelos movimentos Retilíneo Uniforme (MRU) e Retilíneo Uniformemente Variado (MRUV), são filtradas pelo sistema mecânico devido à oscilação da massa de prova e às constantes

naturais do sistema, onde k_x, k_y são as constantes elásticas das molas e c_x, c_y são os coeficientes de viscosidade do meio nos eixos de oscilação e detecção, respectivamente.

Girômetros vibratórios micromaquinados podem ser classificados em dois tipos [Shkel et al., 2005], os girômetros de posição angular e os girômetros de variação angular. Os do primeiro tipo medem diretamente o ângulo de orientação, ou seja, a posição angular do sensor, enquanto os do segundo tipo medem a taxa de variação da rotação, ou seja, a velocidade angular. A maioria dos girômetros vibratórios disponibilizados comercialmente são do segundo tipo [Shkel et al., 2005; Priya and Sushas, 2012], por isso, quando mencionado o termo “girômetro” nesse trabalho, a menos que se diga o contrário, estaremos tratando dos girômetros de variação angular. Eles operam baseados nos princípios de Coriolis aplicados em uma massa vibratória suspensa sobre o substrato do sensor.

A força de Coriolis, como a força centrífuga, é um efeito devido ao caráter não inercial do referencial adotado para descrever o movimento. Para compreender o princípio de operação dos girômetros, vamos considerar, primeiramente, a análise da força de Coriolis que aparece quando consideramos o movimento plano de uma partícula em dois referenciais diferentes, um referencial inercial S (eixos X e Y , cujas direções são indicadas pelos vetores unitários \hat{I} e \hat{J} , respectivamente) e um referencial não inercial R (eixos x e y cujos vetores unitários são \hat{i} e \hat{j}) conforme ilustra Figura 2.9.

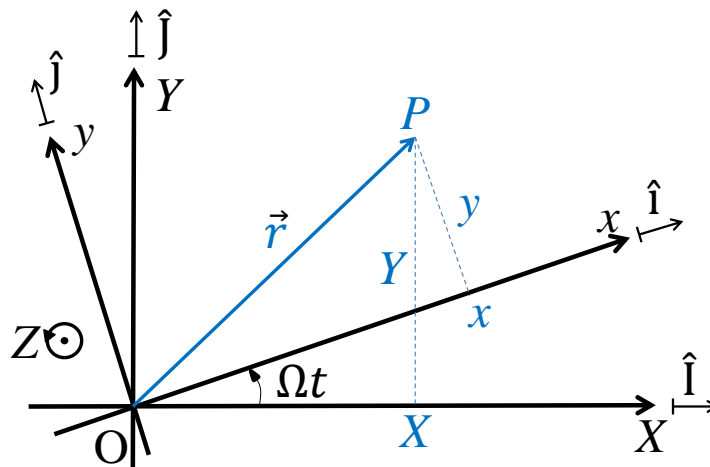


Figura 2.9: Referencial inercial S (eixos X e Y) e referencial não inercial R (eixos x e y) girando à velocidade Ω

O referencial não inercial R gira em sentido anti-horário, com velocidade angular Ω , em relação ao referencial inercial S . Os dois referenciais têm origens no mesmo ponto e o eixo ao redor do qual o referencial R gira (eixo Z) passa por esse ponto e é perpendicular ao plano dos eixos X e Y e também perpendicular ao plano dos eixos x e y .

O vetor posição de uma partícula P no plano dos referenciais S e R pode ser expresso

com relação ao seu respectivo referencial da seguinte forma

$$\begin{aligned}\vec{r}_S &= X\hat{I} + Y\hat{J}, \\ \vec{r}_R &= x\hat{i} + y\hat{j}.\end{aligned}$$

Na realidade $\vec{r}_S \equiv \vec{r}_R \equiv \vec{r}$. A única diferença entre essas duas descrições é que os vetores unitários (\hat{i}, \hat{j}) do referencial em rotação possuem direção e sentido que variam com o tempo, enquanto que os vetores unitários (\hat{I}, \hat{J}) do referencial fixo são constantes em direção e sentido.

Sendo assim, o vetor velocidade da partícula P também pode ser medido com relação ao seu respectivo referencial como se segue

$$\begin{aligned}\vec{v}_S &= \frac{\partial \vec{r}_S}{\partial t} = \frac{\partial X}{\partial t}\hat{I} + \frac{\partial Y}{\partial t}\hat{J} \\ \vec{v}_R &= \frac{\partial \vec{r}_R}{\partial t} = \frac{\partial x}{\partial t}\hat{i} + \frac{\partial y}{\partial t}\hat{j}\end{aligned}$$

A velocidade no eixo inercial S pode ser obtida também pela derivação da posição com relação ao sistema em rotação R conforme segue

$$\begin{aligned}\vec{v}_S &= \frac{\partial}{\partial t}(x\hat{i} + y\hat{j}) \\ &= \frac{\partial x}{\partial t}\hat{i} + \frac{\partial y}{\partial t}\hat{j} + \frac{\partial \hat{i}}{\partial t}x + \frac{\partial \hat{j}}{\partial t}y.\end{aligned}\tag{2.12}$$

Nessa equação observa-se que os dois primeiros termos correspondem à velocidade \vec{v}_R .

Sabe-se que o vetor velocidade linear com relação à velocidade angular em um movimento de rotação é dado por $\vec{v} = \vec{\Omega} \times \vec{r}$. Sendo assim, para os termos da Equação (2.12) relacionados com as derivadas dos vetores unitários podemos inferir que

$$\frac{\partial \hat{i}}{\partial t} = \vec{\Omega} \times \hat{i}, \quad \frac{\partial \hat{j}}{\partial t} = \vec{\Omega} \times \hat{j}$$

Dessa forma a Equação (2.12), pode ser escrita como

$$\begin{aligned}\vec{v}_S &= \vec{v}_R + \vec{\Omega} \times (x\hat{i} + y\hat{j}) \\ &= \vec{v}_R + \vec{\Omega} \times \vec{r}\end{aligned}\tag{2.13}$$

Esta equação pode ainda ser escrita matematicamente, na forma

$$\left(\frac{\partial}{\partial t}\right)_S \vec{r} = \left(\frac{\partial}{\partial t} + \vec{\Omega} \times\right)_R \vec{r}$$

Com isso podemos concluir que a taxa de variação de um certo vetor no referencial

fixo S é igual à taxa de variação do mesmo vetor no referencial em rotação R somada ao produto vetorial entre a velocidade angular do referencial em rotação e o referido vetor.

Essa formulação é geral e vale para qualquer vetor e não apenas para o vetor posição \vec{r} . Assim, se este vetor arbitrário for a velocidade linear \vec{v}_S com relação ao referencial fixo, teremos:

$$\left(\frac{\partial}{\partial t}\right)_S \vec{v}_S = \left(\frac{\partial}{\partial t} + \vec{\Omega} \times\right)_R \vec{v}_S$$

onde o termo da esquerda é a definição do vetor aceleração a_S no sistema fixo. À direita, fazemos a substituição de \vec{v}_S pelo resultado obtido na Equação (2.13). Assim,

$$\begin{aligned} a_S &= \left(\frac{\partial}{\partial t} + \vec{\Omega} \times\right)_R (\vec{v}_R + \vec{\Omega} \times \vec{r}) \\ &= \frac{\partial \vec{v}_R}{\partial t} + \frac{\partial}{\partial t}(\vec{\Omega} \times \vec{r}) + \vec{\Omega} \times \vec{v}_R + \vec{\Omega} \times \vec{\Omega} \times \vec{r} \end{aligned} \quad (2.14)$$

Esta expressão relaciona a aceleração a_S com as acelerações medidas no sistema em rotação onde o termo $\frac{\partial \vec{v}_R}{\partial t}$ é a aceleração medida no sistema em rotação, a_R ; usando a derivada do produto para o termo $\frac{\partial}{\partial t}(\vec{\Omega} \times \vec{r})$ encontramos

$$\frac{\partial \vec{\Omega}}{\partial t} \times \vec{r} + \vec{\Omega} \times \frac{\partial \vec{r}}{\partial t}$$

Supondo que a velocidade angular $\vec{\Omega}$ é constante então

$$\frac{\partial}{\partial t} \vec{\Omega} \times \vec{r} = \vec{\Omega} \times \frac{\partial \vec{r}}{\partial t} = \vec{\Omega} \times \vec{v}_R$$

com isso o segundo e o terceiro termos da Equação (2.14) se somam o que resulta em $2\vec{\Omega} \times \vec{v}_R$. Esse é o termo que dá origem à aceleração de Coriolis. Dessa forma a Equação (2.14) pode ser reescrita como se segue

$$a_S = \frac{\partial \vec{v}_R}{\partial t} + 2\vec{\Omega} \times \vec{v}_R + \vec{\Omega} \times \vec{\Omega} \times \vec{r} \quad (2.15)$$

onde o terceiro termo $\vec{\Omega} \times \vec{\Omega} \times \vec{r}$ corresponde à aceleração centrípeta.

Admitindo-se então o movimento de rotação para um corpo de massa m e aplicando a 2ª Lei de Newton na Equação (2.15), obtém-se então a seguinte expressão para força resultante no referencial inercial

$$\vec{F}_S = \vec{F}_R + 2m\vec{\Omega} \times \vec{v}_R + \vec{F}_{centripeta}$$

Já para o sistema em rotação, tudo se passa como se a partícula se movesse sob a ação de uma força efetiva dada por:

$$\vec{F}_R = \vec{F}_S - 2m\vec{\Omega} \times \vec{v}_R - \vec{F}_{centripeta}$$

onde 1º termo corresponde à força resultante e o 2º e 3º termos são forças que só existem porque o sistema gira com velocidade angular $\vec{\Omega}$. O 3º termo $-\vec{F}_{centripeta}$ é a força conhecida como força centrífuga, cujo sentido é oposto ao da força centrípeta. O 2º termo é a chamada força de Coriolis, ou seja

$$F_{Coriolis} = -2m\vec{\Omega} \times \vec{v}_R \quad (2.16)$$

No caso dos girômetros MEMS a massa de prova é suspensa por hastes flexíveis fixadas na estrutura do substrato de modo a possibilitar a oscilação da mesma livremente em duas direções perpendiculares conforme ilustra Figura 2.10.

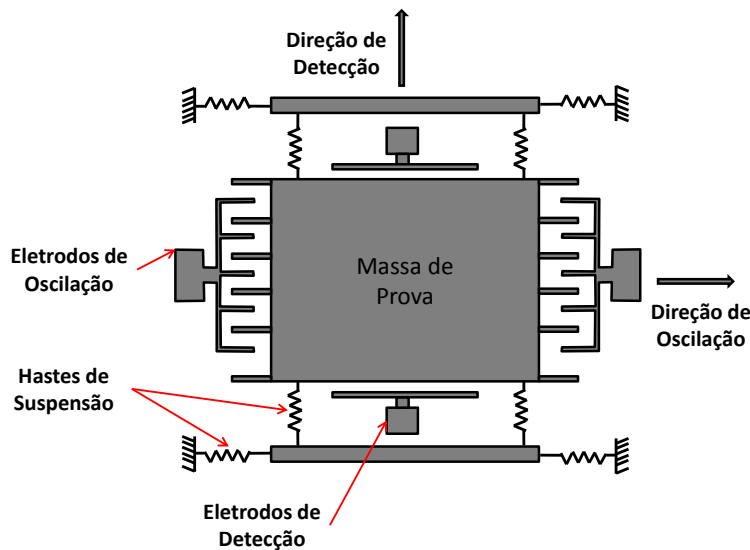


Figura 2.10: Girômetro MEMS

O movimento de oscilação da massa de prova é causado por uma tensão aplicada entre os Eletrodos de Oscilação e a massa de prova. Essa tensão dá origem a uma força eletrostática $F_x(t)$ que provoca o deslocamento da massa de prova ao longo do eixo x.

$$F_x(t) = F_0 \sin \omega_x t \quad (2.17)$$

sendo ω_x a frequência de oscilação da massa de prova e F_0 uma força constante de oscilação.

Nos girômetros é desejável fazer com que o sistema trabalhe em ressonância ou

próximo a ela, ou seja, tanto a frequência elétrica do sinal de tensão aplicado quanto as frequências naturais do sistema mecânico encontrem-se na mesma faixa. Isso possibilita uma grande sensibilidade na detecção de mudanças nos parâmetros do sistema [Shkel et al., 2005].

Se o sistema gira em torno de Z com uma velocidade angular Ω , uma força é induzida pelo efeito Coriolis causando a oscilação da massa de prova na direção y .

A amplitude dessa oscilação é proporcional à variação da rotação do dispositivo. Essa oscilação é detectada por um esquema capacitivo formado entre a massa de prova e placas fixas na direção y .

Para um sistema de coordenadas não inercial, a equação dinâmica simplificada do movimento de um girômetros pode ser descrita como

$$\begin{aligned} m\ddot{x} + c_x\dot{x} + k_x x &= F_0 \sin \omega_x t, \\ m\ddot{y} + c_y\dot{y} + k_y y - 2m\Omega\dot{x} &= 0 \end{aligned} \quad (2.18)$$

Mais uma vez a medição é extraída através do acoplamento do sistema capacitivo a dispositivos eletrônicos que detectam o deslocamento da massa de prova no eixo de detecção, condudo, no caso do girômetro essa medição só apresenta variações quando o sensor é submetido a deslocamentos angulares permanecendo constante para movimentos lineares do girômetro

2.4.3 Magnetômetros

Conforme mostrado nas Seções 2.4.1 e 2.4.2, os acelerômetros e girômetros são sensores cujo funcionamento se baseia nos princípios da 2ª Lei de Newton aplicados a uma massa de prova que pode se deslocar no interior do sensor. Sendo assim, esses sensores são inerciais devido às características de funcionamento e MEMS devido à tecnologia empregada na sua fabricação para possibilitar a mecanização da Lei de Newton no interior de um circuito integrado.

O magnetômetro é um instrumento usado para medir a intensidade, direção e sentido de campos magnéticos em sua proximidade, por isso, pode-se perceber que o mesmo não é um sensor inercial. No entanto, quando utilizado para medir o campo magnético da terra, os magnetômetros podem servir como bússolas eletrônicas, o que revela sua grande importância para utilização em sistemas de navegação e para o auxílio na IMU.

Embora existam técnicas para a fabricação de magnetômetros MEMS [Ciudad et al., 2004; Du and Chen, 2012; Izham and Ward, 2004], nesse trabalho não trataremos dessas abordagens uma vez que o magnetômetro emulado nesse trabalho emprega a tecnologia de Magnetoresistência Anisotrópica (AMR).

O efeito magnetorresistivo ocorre em determinados materiais ferromagnéticos, na

maioria das vezes o Permalói (liga de níquel e ferro), podendo ser usado na fabricação de sensores. Esse efeito consiste na modificação da resistividade do material em função de um campo magnético aplicado externamente ao mesmo.

Se tomarmos uma pequena tira desses materiais, provocarmos a circulação de corrente entre suas extremidades e submetermos os mesmos a um campo magnético, surge uma força que atua sobre os portadores de carga desviando-os de sua trajetória. O resultado é uma alteração na mobilidade dos portadores de carga que se reflete na resistência elétrica do material. Dessa forma, a resistência apresentada depende da ação do campo magnético externo.

A Figura 2.11 mostra a geometria básica de um filme de permalói retangular. Para atuação como um sensor, o filme é submetido a uma magnetização interna M que forma um ângulo ϕ com relação ao eixo do sensor. A corrente I que circula entre as extremidades do filme forma um ângulo ϑ ao longo desse eixo. A direção apontada por M sem a interferência de campos magnéticos externos é denominada “Eixo Fácil”.

Nessa figura as dimensões do filme são: sua espessura (t), comprimento (l) e largura (w - que às vezes é tomada como altura h). Normalmente $t \ll w < l$.

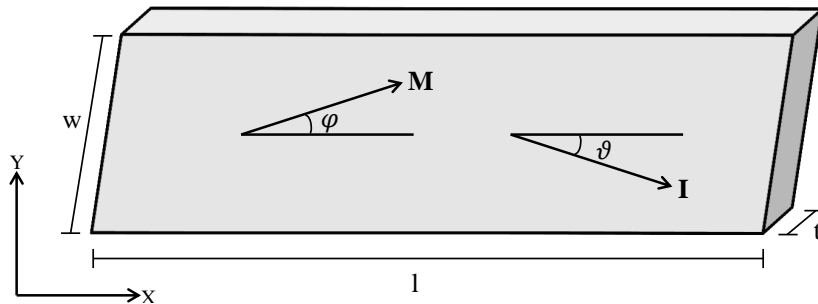


Figura 2.11: Geometria básica de um sensor AMR

A resistência específica ρ dessa estrutura depende do ângulo θ (que corresponde à soma $\phi + \vartheta$) formado entre a magnetização interna M e a corrente I [Dibbern, 1986].

$$\rho(\theta) = \rho_{\perp} + (\rho_{\parallel} - \rho_{\perp}) \cos^2 \theta \quad (2.19)$$

sendo que ρ_{\parallel} e ρ_{\perp} são respectivamente as resistividades paralela e perpendicular a M . A maior resistividade do filme ocorre quando a corrente flui paralela à magnetização M [Caruso and Smith, 1998].

O quociente $(\rho_{\parallel} - \rho_{\perp})/\rho_{\perp} = \Delta\rho/\rho$ é chamado efeito magnetorresistivo [Dibbern, 1986]. Adotando que a corrente I está fluindo paralela ao eixo X ($\theta = \phi$), e pela aplicação da Primeira e Segunda Leis de Ohm, podemos inferir que a queda de tensão ao longo desse eixo será

$$U_x = \rho_{\perp} I \frac{l}{wt} \left(1 + \frac{\Delta\rho}{\rho} \cos^2 \theta \right) \quad (2.20)$$

sendo que wt corresponde à seção transversal do filme.

Quando campos magnéticos externos são aplicados à estrutura do sensor, os momentos magnéticos internos são rotacionados de modo a modificar o ângulo ϕ . Nesse caso o campo

$$H_0 = H_k + \frac{t}{w} \frac{M_s}{\mu_0} \quad (2.21)$$

determina a faixa de medição do sensor AMR onde H_k corresponde ao campo de anisotropia o qual depende das características do material e $H_d \approx \frac{t}{w} \frac{M_s}{\mu_0}$ representa uma aproximação simplificada para o campo de desmagnetização que depende da geometria de construção do material.

Podemos relacionar o ângulo θ a partir da seguinte notação

$$\sin \theta = \frac{H_y}{H_0 + H_x / \cos \theta} \quad (2.22)$$

onde H_y e H_x são as componentes do campo externo aplicado. Assim, podemos reescrever a Equação (2.20) da seguinte forma

$$U_x = \rho_{\perp} I \frac{l}{wt} \left\{ 1 + \frac{\Delta\rho}{\rho} \left[1 - \left(\frac{H_y}{H_0 + H_x / \cos \theta} \right)^2 \right] \right\} \quad (2.23)$$

As Equações (2.19) e (2.22) permitem o desenvolvimento de uma relação entre a magnetorresistência e a magnitude e direção do campo aplicado, embora deva-se deixar claro que a teoria magnética é bem mais complexa do que a apresentada na Equação (2.21).

Esta é a base do funcionamento do sensor magnetorresistivo, contudo, na fabricação desses sensores deve-se utilizar de algum artifício que possibilite a linearização da dependência quadrática apresentada na Equação (2.23). Isso pode ser conseguido se M e I não forem paralelos no estado em que não há campo externo sendo aplicado [Dibbern, 1986], ou seja, $\theta = \phi + \vartheta$ deve ser diferente de zero, preferencialmente 45° . Uma das soluções mais adotadas em aplicações comerciais é a construção de “*Barber Poles*”¹ ao longo do eixo do sensor. Os *barber poles* são pequenas barras de material condutor colocadas ao longo do eixo do sensor em uma inclinação de 45° , o que força a corrente elétrica fluir com essa inclinação.

Além disso, para que se possa conseguir uma representação das características resistência-campo do sensor em valores positivos e negativos são utilizados quatro AMRs conectados numa configuração de Ponte de Wheatstone, conforme ilustra Figura 2.12. Dessa forma, a intensidade e direção do campo magnético externo podem ser medidas ao longo de um eixo através de variações de tensão na saída da ponte.

¹Expressão empregada como alusão ao símbolo utilizado por barbeiros para indicar local onde praticavam seu ofício

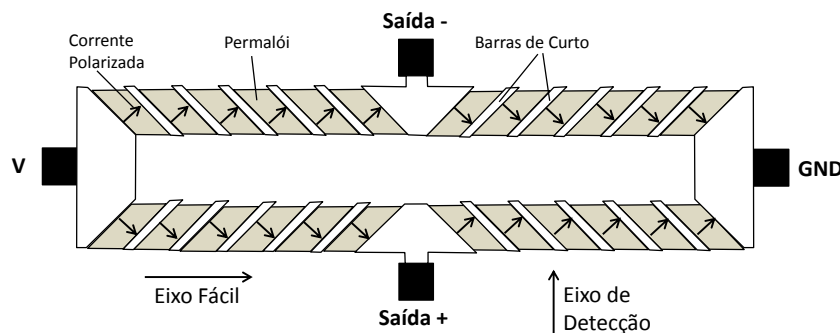


Figura 2.12: AMRs com *Barber Poles* configurados em ponte de Wheatstone

2.5 Parâmetros dos Sensores

A maioria das informações necessárias sobre um sensor, e qualquer dispositivo eletrônico, podem ser encontradas na folha de dados fornecida pelo fabricante desses dispositivos. Outras informações, no entanto, devem ser levantadas experimentalmente e estimadas para utilização em aplicações específicas.

Os parâmetros de erros de natureza determinística e estocástica precisam ser aplicados na modelagem dos sensores de modo que o sensor emulado apresente características semelhantes às do sensor real.

Nessa seção é apresentada uma pequena descrição dos principais parâmetros necessários à utilização, entendimento e modelagem dos sensores que compõe a IMU emulada pelo HIL.

Fundo de escala (*full-scale range*)

Este parâmetro define a faixa de medição do sensor em sua respectiva unidade de medida. O fundo de escala representa apenas um limite de leitura, ou seja, não é possível que o sensor realize a medição de valores acima desse limite.

Quando a grandeza aplicada possui um valor superior ao do fundo de escala, o sinal de saída do sensor estará saturado à quantidade especificada pelo fundo de escala. A partir desse parâmetro e do número de bits do conversor analógico digital é possível determinar a máxima resolução que se pode conseguir com o sensor.

Saída de nível-zero (*zero-rate level*)

Este parâmetro define o sinal de saída (nível de tensão para sensores analógicos ou valor numérico para sensores digitais) quando nenhuma variação é aplicada ao eixo de detecção do sensor. Idealmente, a não existência da grandeza física aplicada ao eixo de medição do sensor deveria resultar em um sinal nulo, contudo, devido às imperfeições

de fabricação e às polarizações da alimentação e temperatura, existe um deslocamento do sinal de saída do sensor em relação à origem.

Em termos práticos, a saída de nível zero equivale ao *bias* estático do sensor, que representa a média dos valores de saída quando nenhum sinal é aplicado ao eixo de detecção do sensor.

Sensibilidade ou Fator de Escala (*scale-factor*)

Define a relação entre a grandeza física e sua respectiva representação em forma de sinal na saída do sensor.

Supondo que o sinal de saída do sensor esteja representado em um plano cartesiano no eixo das abscissas e que a grandeza física medida esteja no eixo das ordenadas; assumindo ainda que exista uma relação linear entre esses elementos, então, pode-se dizer que o fator de escala seria a inclinação da reta que determina a relação entre esses pontos. De posse do fator de escala é possível converter o sinal de saída do sensor em sua respectiva grandeza física equivalente.

Para sensores analógicos geralmente o fator de escala é relacionado a níveis de tensão e para os digitais ao bit menos significativo (LSB).

Não-linearidade

Conforme exposto, a conversão do sinal de saída do sensor para sua grandeza física referente é obtida através do fator de escala do sensor que, na folha de dados disponibilizada pelo fabricante aparece como um valor constante. Contudo, o fator de escala não é completamente constante para toda extensão de medição do sensor e sofre algumas variações.

Não-linearidade é definida como o erro ou desvio máximo que o fator de escala pode apresentar ao longo da faixa de medição do sensor. A não linearidade está geralmente associada à dependência térmica.

Largura de banda do sistema (BW)

Também denominado como resposta em frequência, este parâmetro define como o sensor responde a frequências diferentes. Sensores com maior largura de banda são capazes de medir maiores frequências de movimento do sensor, contudo, também carregam maior quantidade de ruído.

Densidade espectral do ruído

O ruído é muitas vezes considerado como a variação aleatória de pico-a-pico ou o desvio padrão da saída do sensor enquanto o mesmo está em repouso.

A densidade espectral do ruído pode ser especificada em função da largura de banda do sistema utilizando a densidade espectral de potência (PSD), ou a Transformada Rápida de Fourier (FFT). Sendo assim, sua unidade de medida pode aparecer basicamente em duas formas:

- PSD → $\text{unid}/\sqrt{\text{Hz}}$
- FFT → unid^2/Hz

onde “**unid**” corresponde à unidade de medida do sensor (podendo ser g para o acelerômetro, $(^\circ/s)$ para o girômetro e $Guass$ para o magnetômetros).

Para os girômetros uma especificação mais interessante para a densidade espectral do ruído seria o Passeio Aleatório Angular (*ARW - Angle Random-Walk*) que mostra diretamente como o ruído afeta nos cálculos de ângulos. Nesse caso a unidade de medida do ruído é o $(^\circ/\sqrt{hr})$.

A Largura de banda e a densidade espectral do ruído definem a resolução padrão que se pode obter a partir de saídas do sensor. Sensores inerciais geralmente possuem um ruído branco Gaussiano e não correlacionado.

2.6 Estimação de Atitude

A Cabeça Sensora, comumente denominada pelo termo em inglês *Attitude and Heading Reference System* (AHRS), é o sistema responsável pela fusão sensorial e pela estimação de Atitude do sistema dos dados fornecidos pelos acelerômetros e girômetros da Unidade de Medições Inerciais (IMU) e, eventualmente, por magnetômetros, GPS dentre outros sensores.

O termo fusão sensorial refere-se à extração dos sinais de vários sensores diferentes para que, a partir desses, se possa obter um número maior de informações complementares que permitem melhorar as estimativas dos estados do sistema.

No caso da navegação inercial, a estimação da atitude refere-se à obtenção dos ângulos de orientação da aeronave em relação a um sistema de referência. Uma possível abordagem para a estimação da atitude é a que se utiliza dos sensores inerciais (girômetro e acelerômetro) e do magnetômetro.

O girômetro fornece informações de velocidades angulares de modo que, a partir da integração dessas informações ao longo do tempo, pode-se inferir a atitude da aeronave. No entanto, dependendo da quantidade de ruído e de desvios de polarização do girômetro, as integrações das informações fornecidas pelo mesmo levariam rapidamente a estimativas erradas da atitude.

Uma outra estratégia seria utilizar as informações do magnetômetro, que através das projeções do vetor campo magnético pode servir como uma bússola eletrônica tridimensional e fornecer diretamente a atitude da embarcação, contudo, a estimação da atitude a partir dessas projeções depende do magnetômetro estar nivelado em

relação ao plano da terra e, além disso, esse sensor sofre bastante com a interferências de campos e dispositivos magnéticos na sua vizinhança.

A questão de correção da inclinação do magnetômetro pode ser resolvida pela utilização de um acelerômetro caso este não esteja acometido por acelerações lineares. A partir das projeções do campo gravitacional em seus eixos de medição o acelerômetro pode fornecer os ângulos de rolamento e arfagem, ângulos esses que seriam aplicados às medidas do magnetômetro para compensar sua inclinação e extrair as informações de proa.

A leitura dos girômetros e, principalmente, dos acelerômetros sofrem influências das acelerações lineares aplicadas, sendo assim, outros sensores como, por exemplo, GPS, barômetros e sensores de pressão diferencial podem ser adicionados ao sistema para compensar as interferências dessas acelerações e tornar o sistema ainda mais robusto.

Sendo assim, a Cabeça Sensora tem a função de realizar a estimação da atitude da aeronave, implementando algoritmos de fusão sensorial e compensando ainda os erros de medições provenientes dos sensores integrantes do módulo de instrumentação. Existem vários algoritmos que implementam essas técnicas como, por exemplo, a filtragem complementar linear [Brown, 1997], filtragem complementar não linear [Mahony and Hamel, 2008] e o filtro de Kalman.

A Cabeça Sensora desenvolvida por Lima [2013] implementa o algoritmo de filtro de Kalman para realizar a estimação de atitude, por isso, nesse trabalho focaremos nessa classe de algoritmos, uma vez que essa Cabeça Sensora será alvo de testes pelo emulador desenvolvido nesse trabalho na fase de experimentos.

Estrutura do Emulador IMU para testes HIL

“Scientists study the world as it is, engineers create the world that never has been.”

Theodore von Karman

3.1 Introdução

A Figura 3.1 representa um sistema de simulação HIL para emulação de uma IMU. Uma simulação completa contempla a dinâmica de voo da aeronave e os modelos dos sensores e atuadores que a compõem. No caso específico desse trabalho foram implementados apenas os modelos para os sensores da IMU, a saber, acelerômetro, girômetro e também do magnetômetro.

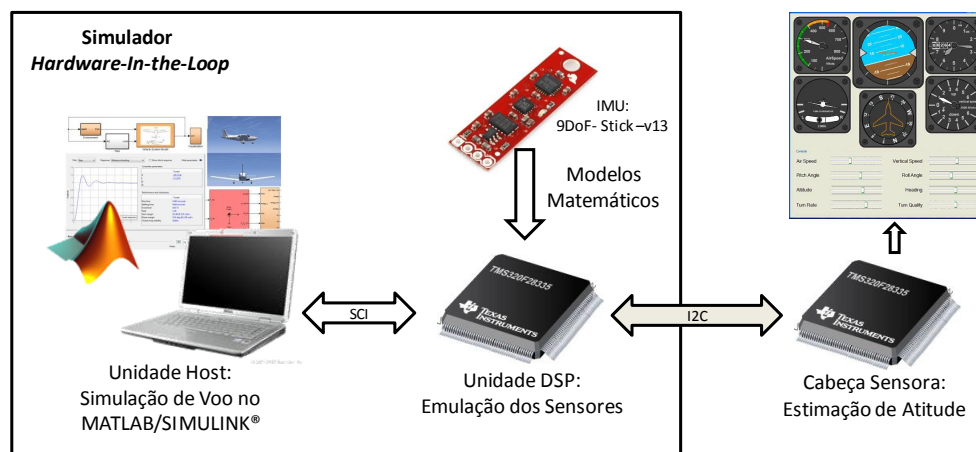


Figura 3.1: HIL para emulação de IMU de 9DoF composta por acelerômetro, girômetro e magnetômetro

Conforme pode ser observado, o emulador desenvolvido neste trabalho é constituído, basicamente, pela integração do MATLAB/Simulink com um DSP que realiza as interfaces de comunicação e executa os modelos dos sensores da IMU real.

No Simulink é possível gerar sinais de teste ou dados de voo à partir de modelos que simulem a dinâmica de um VANT e ainda realizar interfaces gráficas e de visualização como, por exemplo, a interface com o simulador de voo *FlightGear*, dentre outros. Esses sinais gerados no Simulink correspondem às informações ideais de atitude e acelerações lineares do VANT.

O DSP, por sua vez, recebe os sinais através da interface de comunicação serial (SCI) e os converte em dados de sensor, a partir dos algoritmos que executam os modelos matemáticos do acelerômetro, girômetro e magnetômetro, emulando então as características reais da IMU. Esses dados emulados pelo DSP devem ainda ser disponibilizados para leitura através do mesmo mecanismo que os sensores reais utilizam; que nesse caso é a interface de comunicação I2C.

As próximas seções detalham essas funcionalidades do sistema HIL.

3.2 A Unidade de Medições Inerciais Real

Conforme mencionado na Seção 1.1, a emulação IMU para testes *Hardware-in-the-loop* (HIL) desenvolvida neste trabalho faz parte de um projeto da UFMG para a criação de uma plataforma aérea autônoma para mini-VANTs capazes de realizar missões colaborativas.

Ainda no contexto desse projeto, o trabalho desenvolvido por Lima [2013], dentre outras coisas, levantou os requisitos mínimos para o módulo de instrumentação da Cabeça Sensora e estabeleceu quais seriam os sensores utilizados para a estimação da atitude dos VANTs. Esses sensores, digitais e com interface de comunicação I2C, são o acelerômetro ADXL345 da *Analog Devices*, o girômetro ITG3200 da *InvenSense* e o magnetômetro HMC5883L da *Honeywell*, os quais estão detalhados nas subseções seguintes e cujas principais características de configuração estão resumidos na Tabela 3.1.

Acelerômetro

A Figura 3.2 apresenta o diagrama de blocos do acelerômetro ADXL345 da *Analog Devices Inc* [AnalogDevices, 2012]. Esse é o acelerômetro proposto para ser emulado pelo sistema HIL dado que o mesmo compõe o sistema de instrumentação real que integrará os VANTs do projeto. Ele é um dispositivo digital que realiza a medição das acelerações em três eixos ortogonais e implementa o acondicionamento dos sinais gerados a partir de amplificadores e filtros. Ele realiza medições de acelerações nas faixas de ± 2 , ± 4 , ± 8 e $\pm 16g$, configuráveis através de seus registradores internos. O conversor AD, também configurável, pode ser ajustado para 10 ou 13-bits.

As informações medidas podem ser disponibilizadas através das interfaces de comunicação I2C ou SPI em palavras de 16-bits a complemento de dois. Quando configurado para interface I2C a comunicação ocorre nas taxas de transferência de 100kHz

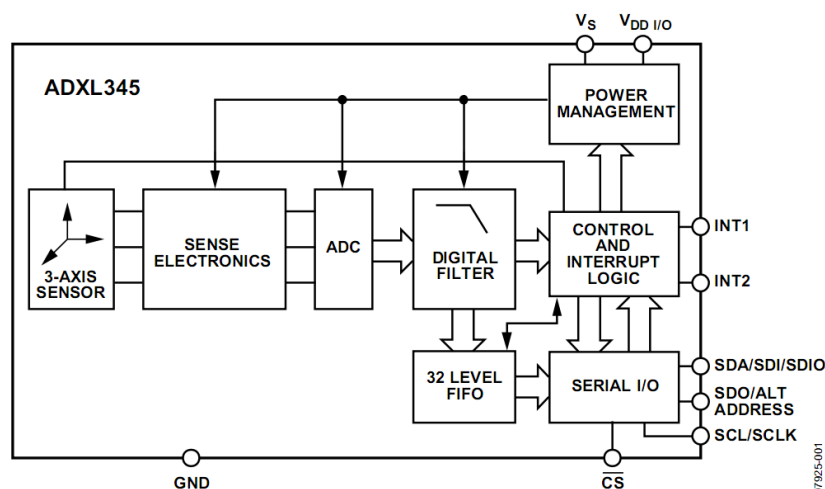


Figura 3.2: Diagrama funcional do acelerômetro ADXL345

Fonte: Figura retirada da folha de dados do ADXL345 [[AnalogDevices, 2012](#)]

ou 400kHz, casos em que a máxima taxa de saída de dados é de 200Hz e 800Hz, respectivamente.

As medições de aceleração podem apresentar erros relativos a não-linearidades, desalinhamento entre os eixos e sensibilidade de eixo cruzado, além de ruídos.

Girômetro

Outro sensor emulado pelo sistema HIL desenvolvido nesse trabalho é o girômetro ITG-3200 produzido pela InvenSense [[InvenSense, 2012](#)]. A Figura 3.3 apresenta o diagrama de blocos desse girômetro que realiza as medições das velocidades angulares em três eixos ortogonais e o acondicionamento dos sinais lidos.

O ITG-3200 é um girômetro digital de três eixos cujos conversores AD são de 16-bits e o fundo de escala é pré-estabelecido em $\pm 2000^\circ/s$. As velocidades angulares são amostrados internamente com frequências de 1kHz ou 8kHz, determinadas pela configuração de um dos seus registradores internos. O valor medido é filtrado digitalmente e entregue nos registros de saída do sensor.

Na configuração da Cabeça Sensora, a leitura do girômetro ocorre via I2C à taxa de transferência de 100kHz com a frequência de amostragem interna de 1kHz e a taxa de saída de dados a 50Hz.

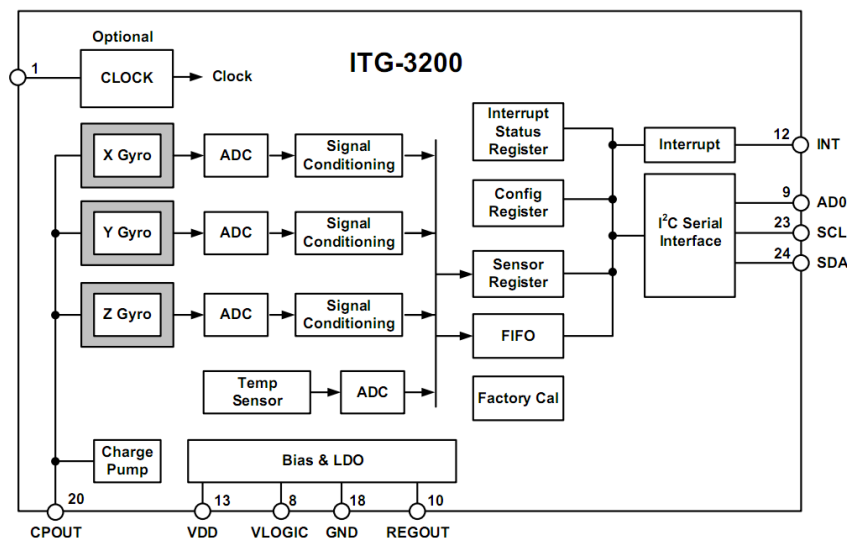


Figura 3.3: Diagrama funcional do girômetro ITG3200

Fonte: Figura retirada da folha de dados do Girômetro ITG3200 [InvenSense, 2012]

Magnetômetro

A Figura 3.4 apresenta a estrutura do sensor HMC5883L da Honeywell [Honeywell, 2012]. Esse sensor utiliza a tecnologia de magnetorresistência anisotrópica (AMR), um efeito de estado sólido de certas ligas ferromagnéticas que, quando utilizado para a construção de sensores, possibilita a detecção de campos magnéticos, conforme explicado na Seção 2.4.3.

O HMC5883L foi projetado para medir tanto a direção quanto a intensidade do campo magnético da terra sendo adequada sua utilização como uma bússola eletrônica. Ele é um sensor digital de três eixos cujos conversores AD são de 12-bits, o que permite uma precisão de 1° a 2° em medidas de posição, segundo fabricante. A medição de campo magnético é configurável para as faixas de $\pm 0,88\text{gauss}$ a $\pm 8,1\text{gauss}$. Os dados de medição são entregues via interface I2C em palavras de 16-bits a complemento de dois.

Na cabeça sensora o HMC5883L foi configurado para atuar na faixa de medição de $\pm 2,5\text{gauss}$ a uma resolução digital (desconsiderando largura de banda e ruído) de $1,52\text{mgauss/LSB}$. Os dados são amostrados a uma frequência de 50Hz.

A Tabela 3.1 contém os principais parâmetros de configuração para os sensores da IMU. Os nomes dos parâmetros nesta tabela foram mantidos em inglês para facilitar a identificação dos mesmos junto à folha de dados dos fabricantes.

3.3 A Unidade Host

A Unidade *Host* é composta basicamente pelo software MATLAB/Simulink e sua principal função nesse trabalho é gerar os dados de atitude (ângulos ϕ , θ e ψ) e

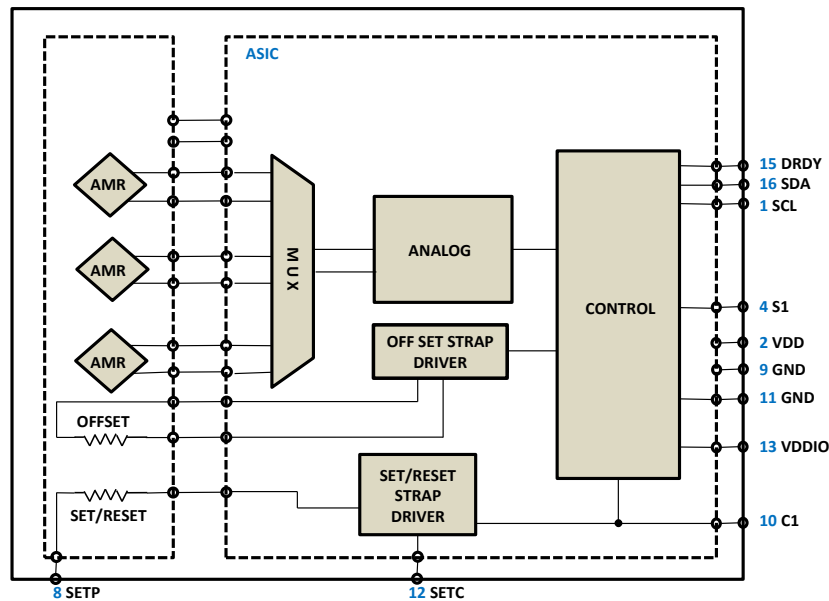


Figura 3.4: Diagrama funcional do magnetômetro HMC5883L

acelerações lineares (a_x , a_y e a_z) ideais, disponibilizando esses dados via interface SCI para processamento na unidade DSP.

Dadas as potencialidades do MATLAB/Simulink, os dados gerados podem ser de fontes arbitrárias como sinais senoidais, pulso, degrau, arquivos com dados ideais de vôo ou ainda podem ser provenientes de modelos que simulem a aerodinâmica do VANT executando paralelamente à comunicação. O MATLAB/Simulink possibilita ainda a integração com simuladores de vôo e a interface gráfica com usuário, o que facilita a visualização dos estados da aeronave em vôo e a configuração da simulação.

Esse recurso não foi disponibilizado no emulador desenvolvido, mas pode ser implementado uma vez que os sinais de entrada do DSP (que seguem via SCI) podem ser enviados simultaneamente para um visualizador de vôo como, por exemplo, o *FlightGear*, haja vista que as entradas desse simulador são basicamente as mesmas do DSP.

3.4 A Unidade DSP

O emulador IMU/HIL deve executar a função realizada pela IMU real no barramento de comunicação com a Cabeça Sensora. Uma idéia preliminar para desempenhar essa função é executar os modelos matemáticos dos sensores na própria Unidade *Host* e enviar os dados gerados para uma interface de hardware intermediária, a qual seria responsável unicamente pela conversão das informações do protocolo SCI para o I2C (utilizado pelos sensores reais da IMU) [Ribeiro et al., 2013].

O problema dessa abordagem é que a velocidade de requisição de dados pela

interface I2C é maior que a taxa de atualização pela interface SCI. Além disso, o período de amostragem determinada pelo EKF implementado na Cabeça Sensora é constante e bastante preciso, característica essa improvável de se obter da comunicação serial com o MATLAB dadas as limitações e características assíncronas da interface SCI.

Dessa forma, no caso da Cabeça Sensora fazer uma requisição, e dados novos não existirem na interface intermediária, os dados antigos devem ser repetidos, o que não corresponde ao comportamento esperado de um sensor, devido à natureza do sistema real e da existência de imperfeições e ruído.

Diante do exposto, os modelos dos sensores devem ser executados na interface intermediária, que nesse trabalho denominamos Unidade DSP, pois é composta por um processador digital de sinais.

A Unidade DSP consiste de um dispositivo de hardware com capacidade de processamento suficiente para executar os modelos dos sensores e ainda realizar as interfaces de comunicação com Unidade *Host* e Cabeça Sensora, emulando dessa forma as características do sistema em tempo real. Assim, mesmo que não existam dados de entrada atualizados, os dados da saída representarão o comportamento esperado do sensor uma vez que suas imperfeições estarão modeladas nessa unidade.

O DSP utilizado nesse trabalho foi o TMS320F28335 da Texas Instruments. Em termos mais exatos esse dispositivo é um DSC (*Digital Signal Controller*). Isto porque, além de possuir um processador dedicado à execução de cálculos (DSP - *Digital Signal Processor*), ele conta internamente com alguns periféricos de armazenamento, temporização, comunicação, conversão e interface, de modo a caracterizá-lo como um controlador completo, ou microcontrolador, para aplicações dedicadas. Esse dispositivo possui um DSP de 32-bits e trabalha em frequência de até 150MHz, executa cálculos em ponto-fixe e ponto-flutuante com ciclos de clock de 6,7ns e possui as interfaces de comunicação necessárias à implementação do simulador HIL.

Como a princípio não é de interesse desenvolver um hardware dedicado exclusivamente ao simulador HIL, nesse trabalho o DSP encontra-se no kit de desenvolvimento TMDSPREX28335 da Texas. Esse kit possui uma grande quantidade de recursos, porém, nesse trabalho foram exploradas apenas as facilidades de acesso às interfaces de comunicação e aos pinos de entrada e saída digitais.

Outros dispositivos ou microprocessadores poderiam ser adequados ao desenvolvimento do emulador IMU/HIL, contudo, o principal motivo para a escolha do TMS320F28335 é a sua disponibilidade de uma vasta gama de recursos e periféricos internos vinculados a um poderoso DSP, características essas que satisfazem às necessidades do emulador IMU/HIL proposto e também possibilitam sua futura expansão com a inserção de novas funções sem a necessidade de substituir a unidade de processamento. A versatilidade para execução de cálculos tanto em ponto-fixe como em ponto-flutuante também é uma característica interessante desse dispositivo. Além disso, o mesmo fornece as interfaces de comunicação SCI e I2C, indispensáveis para um simulador HIL.

Tabela 3.1: Principais características de configuração para sensores da IMU.

Sensor	Parâmetros
Acelerômetro {ADXL345}	Measurement range $\pm 2g$ Output resolution 10-bits Sensitivity 3.9 mg/LSB output data rate 50 Hz
Girômetro {ITG3200}	Full-Scale Range $\pm 2000^\circ/s$ ADC Word Length 16-bis Sensitivity 0.07($^\circ/s$)/LSB Temperature Range -40 to $85^\circ C$
Magnetômetro {HMC5883}	Full Range $\pm 8\text{gauss}$ ADC Word Length 12-bits Sensitivity 1.5mili-gauss Field Range $\pm 2.5\text{gauss}$

Conforme exposto, a unidade DSP toma o lugar dos sensores no barramento I2C e opera na configuração escravo quando comunicando com a Cabeça Sensora. Ela fornece os dados dos sensores necessários à estimação de atitude pela Cabeça Sensora.

Resumidamente pode-se dizer que a interface DSP executa em seu loop principal as seguintes operações:

1. recebe os sinais de atitude ideal e acelerações lineares provenientes do MATLAB/Simulink através da interface SCI;
2. executa os modelos matemáticos dos sensores com os sinais recebidos;
3. autoconfigura-se com o endereço de escravo do sensor que será requisitado pelo DSP da Cabeça Sensora;
4. fornece os dados referentes ao sensor requisitado através do barramento I2C.

3.4.1 Interface de Comunicação Serial - SCI

A Interface de Comunicação Serial (SCI) é uma porta de entrada/saída serial que permite a comunicação assíncrona entre o F28335 e outros dispositivos. É normalmente conhecido como UART (Universal Asynchronous Receiver Transmitter) e é usado de acordo com o padrão RS232.

A interface de recepção SCI do DSP dispõe de uma FIFO de 16 Bytes de profundidade. Na aplicação IMU/HIL essa FIFO é implementada com uma profundidade de 12 Bytes, o que possibilita a geração de uma interrupção apenas quando o MATLAB/SIMULIK enviou todos os dados de atitude e acelerações lineares. Isso reduz a sobrecarga na comunicação entre os dispositivos e possibilita que o DSP direcione

a sua capacidade de processamento para execução de sua tarefa principal, que é a de executar os cálculos dos modelos dos sensores.

A taxa de transmissão SCI é programável através de registradores específicos do DSP e foi configurada para 115200bps, uma vez que essa é a maior taxa possível de se configurar o conversor USB/Serial utilizado na interface física de comunicação.

A interface SCI tanto transmite quanto recebe dados no formato Não-Retorno-a-Zero (NRZ) onde simplesmente representa-se um 1 por meio de um sinal alto e um 0 por meio de um sinal baixo. O formato de dados NRZ mostrado na Figura 3.5, consiste de:

- um bit de inicialização (*start*),
- um a oito bits de dados (padrão 8 para caracteres),
- um bit de paridade (opcional),
- um ou dois bits de finalização (*stop*),
- um bit extra para distinguir endereços e dados (modo multiprocessador).

Na comunicação SCI devem existir oito bits de clock para cada bit de dado presente no barramento. O receptor começa o procedimento de recepção de dados pela procura de um bit de inicialização válido. A validação desse bit é alcançada pela detecção de um valor de nível baixo durante os quatro primeiros períodos consecutivos do clock, conforme mostrado na Figura 3.6. Se qualquer bit da linha de dados é diferente de zero para essa sequência de quatro períodos de clock, então o processador começa a procurar novamente por um bit de inicialização.

Uma vez finalizada a primeira sequência de oito períodos de clock para identificação do bit de inicialização, o valor do próximo bit da sequência de transmissão é determinado a partir do quarto, quinto e sexto períodos de clock da SCI, que correspondem ao centro do próximo bit. A confirmação do valor desse bit é realizada para as amostras coletadas nesses três ciclos de clock a partir do “voto de maioria”, ou seja, dois dos três determinam o valor do bit na sequência de transmissão. Esse procedimento também pode ser visualizado na Figura 3.6.

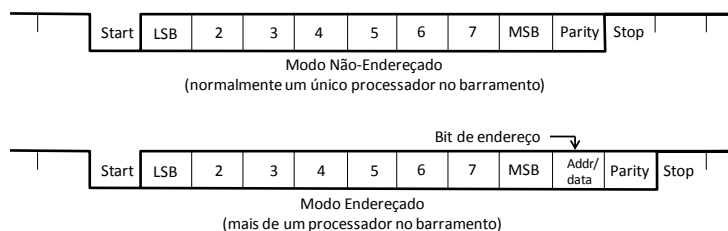


Figura 3.5: Formatos de dados na interface SCI

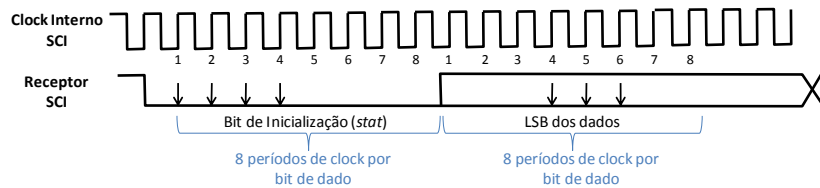


Figura 3.6: Validação e Sincronização de dados na interface SCI

Como pode ser percebido a transmissão é sincronizada automaticamente pela identificação do bit de inicialização, o que significa que os dispositivos de transmissão e recepção não precisam de uma linha de clock adicional. O clock pode ser gerado internamente nos dispositivos de comunicação.

3.4.2 Comunicação I2C

Uma das grandes vantagens da utilização de sensores MEMS no desenvolvimento de sistemas para navegação inercial é a sua possibilidade de integração com interfaces de comunicação digital, o que permite a abstração de detalhes como conversões AD e tratamento do sinal medido, uma vez que tudo isso já foi realizado internamente no sensor.

Os sensores da IMU emulada nesse trabalho disponibilizam os dados das medições através da interface de comunicação I2C. Sendo assim, os valores calculados pelos modelos dos sensores devem ser disponibilizados para medição através dessa interface de comunicação.

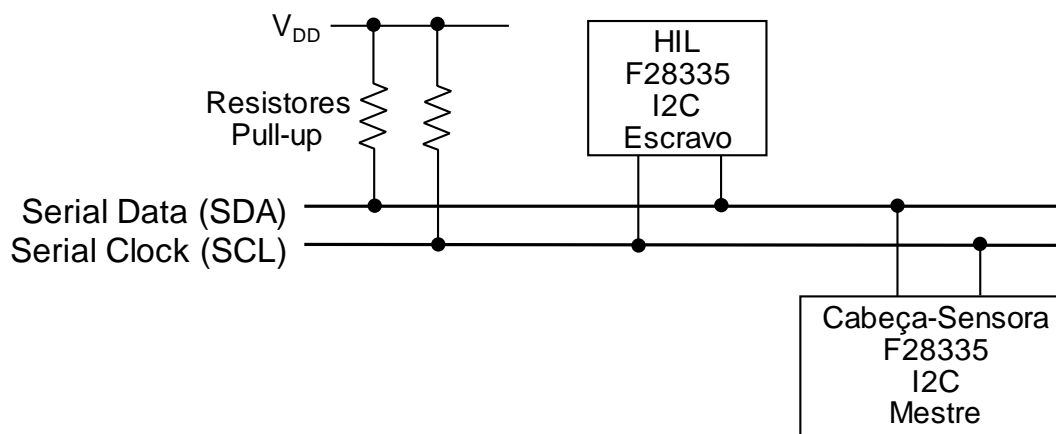


Figura 3.7: Barramento I2C para comunicação HIL/Cabeça Sensora

A interface de comunicação I2C, desenvolvida pela Philips para conexão de periféricos à unidade processamento, é composta por um barramento de apenas dois fios,

as vias SDA e SCL, que correspondem ao sinal de dados e de clock, respectivamente. Dispositivos conectados ao barramento I2C podem ser mestres ou escravos. O mestre é o dispositivo que controla o sinal de clock SCL durante a comunicação e realiza as requisições de leitura e/ou escrita. Sensores geralmente atuam apenas como escravos no barramento I2C, ou seja, quando um dispositivo mestre envia seu respectivo endereço ele atende à requisição do mestre. No contexto desse trabalho o dispositivo mestre é a Cabeça Sensora e o escravo o emulador IMU/HIL, conforme ilustra a Figura 3.7.

É possível observar que as linhas SDA e SCL são conectadas à tensão positiva da fonte através de resistores de *pull-up*. Quando o barramento está livre, ambas as linhas estão em nível lógico alto.

A linha SDA deve se manter estável enquanto a linha SCL estiver em nível alto. A comunicação no barramento I2C inicia quando o mestre aplica uma condição de *START* (S) no barramento, que é definida como uma transição do nível alto-para-baixo na linha SDA enquanto a linha SCL está em nível alto. O barramento é considerado como ocupado até que o mestre aplique uma condição de *STOP* (P) no barramento, que é definida como uma transição do nível baixo-para-alto na linha SDA enquanto a linha SCL está em nível alto. A Figura 3.8 ilustra essas condições.

Na transmissão I2C padrão os bytes de dados possuem 8-bits e não existe restrição com relação ao número de bytes que serão enviados por transmissão. Cada byte transferido deve ser seguido por um sinal de confirmação (ACK). Os dados na linha SDA devem ser alterados enquanto a linha SCL se mantém em nível baixo e são confirmados quando esta retorna para o nível alto.

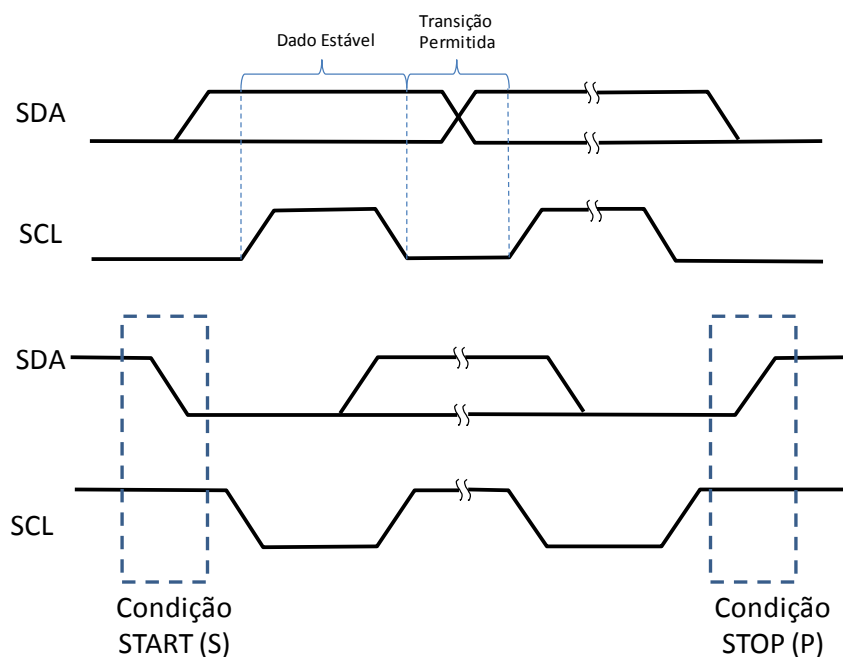


Figura 3.8: Validação dos Dados na comunicação I2C

Depois de começar a comunicação com a condição (S), o mestre envia um endereço escravo de 7-bits seguido por um oitavo bit (R/W) que indica as operações de leitura ou escrita. Em seguida, o mestre libera a linha SDA e aguarda o sinal de confirmação (ACK) do dispositivo escravo. Cada byte transferido deve ser seguido por um bit de confirmação. A transmissão de dados é sempre terminada pelo mestre com uma condição de parada (P), liberando assim a linha de comunicações. No entanto, o mestre pode gerar uma condição de *START* repetido (Sr), e abordar um outro escravo, sem a necessidade de gerar primeiro uma condição (P).

Modelagem dos Sensores

“It’s engineering that changes the world”

Isaac Asimov

4.1 Introdução

Neste capítulo são apresentados os modelos matemáticos dos sensores e seus respectivos diagramas de blocos que deram origem ao código executado na Unidade DSP para emulação da IMU.

O modelo de cada sensor é discutido separadamente, fornecendo detalhes de implementação e evidenciando as equivalências entre o modelo matemático e o blocos que implementam suas funções. São modelos baseados em representações matemáticas já existentes e difundidas na literatura e implementam características básicas dos sensores bem como suas fontes de erros mais simples: bias e ruído.

Em destaque apresenta-se também a ferramenta utilizada para implementação dos códigos embarcados no DSP: a plataforma *Embedded Target* do MATLAB/Simulink que possibilita a geração automática de códigos em C para compilação no *Code Composer Studio*® e execução no DSP. Essa plataforma contempla a metodologia de *Model-Based Design* que permite a abstração de detalhes de programação, torna o modelo de fácil interpretação, evita a geração de erros introduzidos pelo processo de programação manual dentre outras vantagens.

4.2 *Embedded Target*

Os códigos implementados na Unidade DSP do emulador IMU/HIL foram gerados automaticamente através da plataforma *Embedded Target* do MATLAB/Simulink. Esse processo faz parte da metodologia *Model-Based Design* (MBD) para desenvolvimento de projetos em sistemas embarcados.

A plataforma *Embedded Target* para os DSPs da família TI TMS320C2000™ [[MathWorks, 2003](#)] realiza a integração entre o MATLAB/Simulink e as ferramentas *eXpressDSP*™

da Texas Instruments e gera códigos customizados para os DSP's da família c28x da Texas. O *Real-Time Workshop*® cria um projeto para o *Code Composer Studio*®, Ambiente de Desenvolvimento Integrado da Texas, permitindo compilar, fazer links, baixar e executar o código C gerado para uma placa eZdsp™ DSP da *Spectrum Digital*.

Outra grande vantagem dessa plataforma é a disponibilidade de blocos para cálculos em ponto-fixado empregando as bibliotecas *IQmath* da Texas. Esses blocos possibilitam a geração de códigos otimizados em Assembly nos DSP's para operações em ponto-fixado. Esse recurso foi utilizado na maioria dos cálculos empregados nos modelos dos sensores. [Fainguelernt \[2011\]](#) apresenta vários exemplos de utilização da plataforma *Embedded Target* para integração do MATLAB/Simulink com os DSP's da Texas empregando desde a criação do modelo, simulação e a geração dos códigos executáveis no dispositivo alvo.

4.3 Os Modelos dos sensores

4.3.1 Acelerômetro

O acelerômetro detecta a diferença entre as acelerações lineares e o campo gravitacional local aplicados ao sensor [[Stevens and Lewis, 2003](#)]. A folha de dados de qualquer acelerômetro irá designar o sentido positivo dos eixos x , y e z no encapsulamento do sensor e, por convenção, estes são definidos de modo que uma aceleração linear alinhada na direção desses eixos vai resultar em uma saída positiva no acelerômetro [[Pedley, 2013](#)].

Observe a Figura 4.1. O sentido positivo de detecção da aceleração no eixo z é indicado conforme seta apresentada para esse eixo na figura. Se a estrutura externa do acelerômetro sofrer a ação de uma aceleração linear ao longo desse eixo, a massa de prova m se deslocará para o sentido indicado pela seta a_z , o que leva a valores positivos sendo entregues ao conversor AD. Uma componente do campo gravitacional alinhada ao longo desse mesmo eixo resultaria, no entanto, no deslocamento da massa de prova para o sentido indicado pela seta g , o que reflete em uma leitura negativa sendo entregue ao ADC.

Sendo assim, a aceleração medida pelo sistema de detecção do acelerômetro \bar{A}_{sys} corresponde a:

$$\bar{A}_{sys} = \bar{A}_{CG} - \bar{A}_{pig}, \quad (4.1)$$

onde $\bar{A}_{CG} \in \mathbb{R}^{3 \times 1}$ são os valores de aceleração linear detectados nos eixos do sensor considerando o mesmo no centro de gravidade do corpo. O vetor $\bar{A}_{pig} \in \mathbb{R}^{3 \times 1}$ representa as projeções da gravidade nos eixos de detecção do acelerômetro e pode ser determinado em função apenas dos ângulos de rolamento (ϕ) e arfagem (θ) [[Pedley, 2013](#)], sendo:

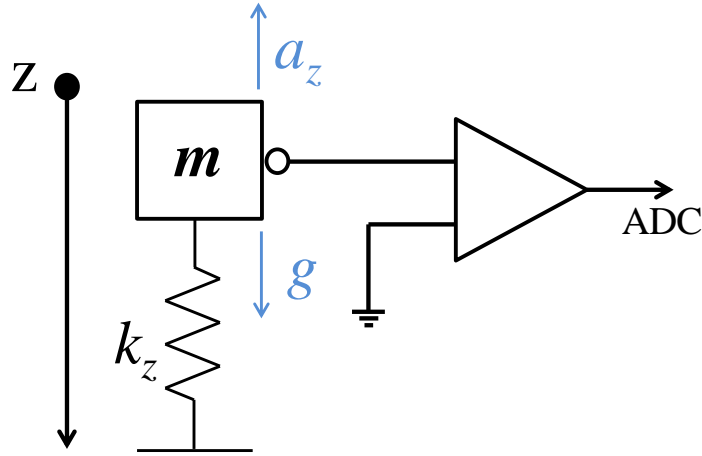


Figura 4.1: Representação do Sentido Positivo para Detecção da Aceleração

$$\bar{A}_{pig} = \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix}. \quad (4.2)$$

As acelerações medidas \bar{A}_{med} incluem ainda os efeitos de alavanca causados no acelerômetro caso o mesmo não esteja no centro de gravidade e as influências provocadas por velocidades e acelerações angulares conforme formulado por [Stevens and Lewis \[2003\]](#):

$$\bar{A}_{med} = \bar{A}_{sys} + \Omega_{gyro} \times (\Omega_{gyro} \times \hat{d}) + \dot{\Omega}_{gyro} \times \hat{d}, \quad (4.3)$$

onde Ω_{gyro} são as velocidade angulares do corpo, $\dot{\Omega}_{gyro}$ são as acelerações angulares, \hat{d} representa os efeitos de alavanca, que são definidos como as distâncias em x , y e z que o acelerômetro se encontra com relação ao centro de gravidade da aeronave

$$\hat{d} = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} -(x_{ac} - x_{CG}) \\ (y_{ac} - y_{CG}) \\ -(z_{ac} - z_{CG}) \end{bmatrix}$$

Os termos de x e de z aparecem negativos devido ao fato do sentido positivo de seus eixos serem opostos àqueles considerados como positivos para orientação do centro de gravidade no corpo do acelerômetro.

As medidas do acelerômetro são ainda expostas a fontes de erros que são somadas ao vetor de medições e, por fim, é realizada a conversão da soma de sua unidade de medida padrão “g”, para LSB’s através da multiplicação desse resultado pela matriz de fator de escala e eixo cruzado.

$$\bar{A}_{saida} = \mathbf{S}_{accel}(\bar{A}_{med} + A_{bias} + A_{noise}) \quad (4.4)$$

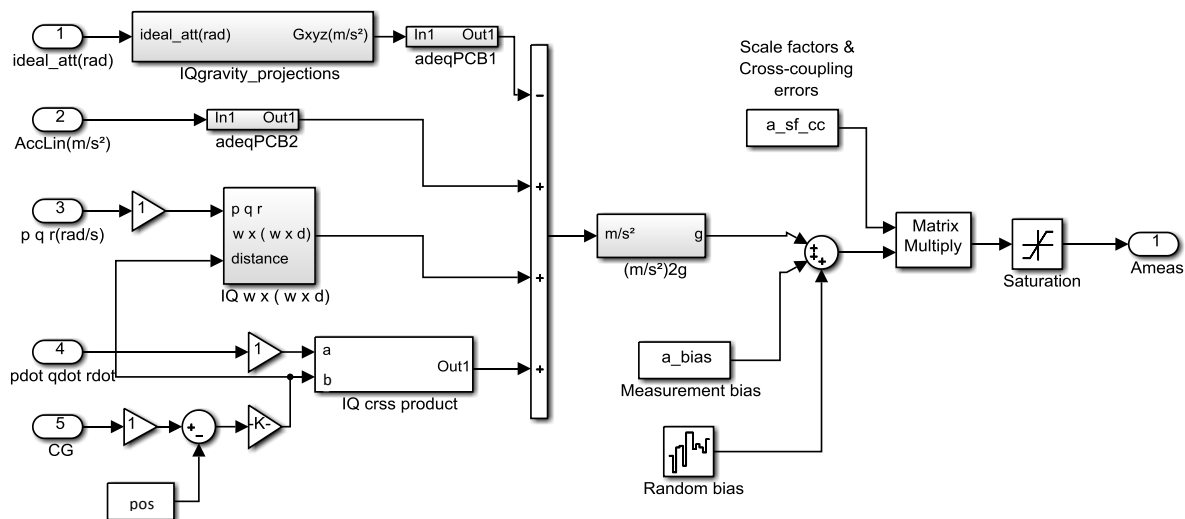


Figura 4.2: Modelo do Acelerômetro no MATLAB/Simulink

Os vetores A_{bias} e A_{noise} correspondem ao *bias* e ao ruído. O valor do *bias* e da densidade espectral do ruído foram obtidos por procedimentos experimentais usando a metodologia explicada na Seção 4.3.4.

A Figura 4.2 mostra como essa formulação foi implementada no Simulink. Na etapa final do modelo é colocado o bloco de saturação para adequar os valores da saída do acelerômetro aos limites impostos pelo conversor AD.

4.3.2 Girômetro

Conforme estabelecido na Seção 2.4.2, o girômetro mede as velocidades angulares do corpo baseado nas forças de Coriolis induzidas do movimento de rotação. Os sinais de velocidades medidas $\bar{\Omega}_{saida}$, além da velocidade angular do corpo Ω_{gyro} , incluem erros de medição. A Equação (4.5) exibe a representação do sinal de saída do girômetro adotada para modelagem desse dispositivo:

$$\bar{\Omega}_{saida} = \mathbf{S}_{gyro} [\Omega_{gyro} + \Omega_{bias} + \Omega_{noise} + (G_{lin} \Omega_{gsens})] \quad (4.5)$$

A Figura 4.3 mostra como essa equação foi implementada no Simulink. O cálculo principal para o girômetro é o da derivada discreta dos ângulos de atitude ideais informados

$$y(k) = \frac{u(k) - u(k-1)}{T_s},$$

onde $y(k)$ e $u(k)$ correspondem à saída e à entrada no passo atual do cálculo da derivada, $u(k-1)$ são as entradas no passo anterior e T_s representa o período de amostragem, que é adotado como constante. Essa equação dá como resultado $\dot{\phi}$, $\dot{\theta}$ e $\dot{\psi}$, contudo, esses valores não correspondem exatamente às velocidades angulares medidas pelo

girômetro conforme é explicado no Apêndice B.

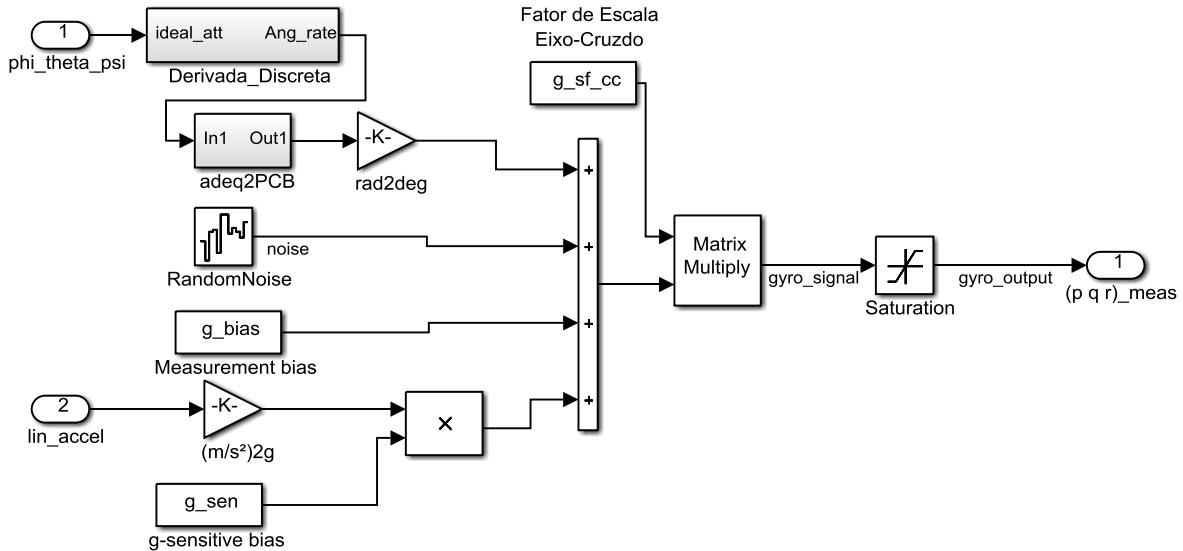


Figura 4.3: Modelo do Girômetro no MATLAB/Simulink

Os valores obtidos da derivada são adaptados à posição de montagem dos eixos do girômetro na Placa de Circuito Impresso (PCB) e em seguida é realizada a conversão desse valor de rad/s para $^{\circ}/s$. Com isso é obtido o vetor $\Omega_{gyro} \in \mathbb{R}^{3 \times 1}$, que corresponde às velocidades angulares p , q e r ideais, medidas pelo girômetro em seus três eixos de detecção.

Os vetores Ω_{bias} e $\Omega_{noise} \in \mathbb{R}^{3 \times 1}$ correspondem, respectivamente, ao *bias* e ruído do girômetro. O vetor G_{lin} corresponde às acelerações lineares nos três eixos do girômetro dadas em “g” ($9,81m/s^2$). Sua multiplicação pelo escalar Ω_{gsens} , cujo valor aparece na folha de dados do sensor em $(^{\circ}/s)/g$, resulta na medida das interferências provocadas por acelerações lineares nos valores medidos pelo girômetro.

Todos esses vetores são somados e o resultado dessa soma aparece em $^{\circ}/s$. Sendo assim, para se obter o valor de saída do girômetro em LSB’s é aplicada a matriz $S_{gyro} \in \mathbb{R}^{3 \times 3}$ cuja diagonal principal corresponde aos fatores de escala do sensor e os demais termos são as interferências de eixo cruzado. Os limites impostos pelo conversor AD são representados pelo bloco de saturação colocado na etapa final do modelo.

4.3.3 Magnetômetro

A Figura 4.4 mostra o modelo do magnetômetro implementado no Simulink. Esse modelo foi elaborado com base no trabalho exposto por Jensen and Vinther [2010] e recebe como entradas os ângulos ideais de atitude e as intensidades do vetor campo magnético no referencial da terra. Para que o modelo do magnetômetro emulado pudesse ter informações compatíveis com as do magnetômetro real disponível, o vetor

campo magnético foi considerado como constante e seus valores foram estimados a partir de procedimentos experimentais, conforme apresentado no Capítulo 5.

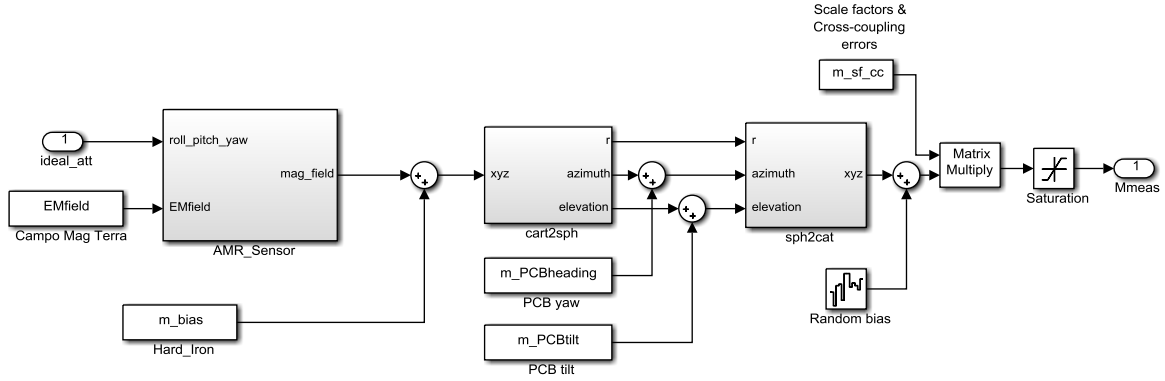


Figura 4.4: Modelo do Magnetômetro no MATLAB/Simulink

De posse do vetor campo magnético terrestre e dos ângulos de atitude é então realizada a rotação do sistema, através da operação com uma matriz de cossenos diretores, de forma a se obter as projeções do campo magnético no referencial do sensor, o que corresponde às leituras “ideais” obtidas em seus eixos de medição.

As medidas do magnetômetro são altamente dependentes dos campos magnéticos em seu interior e nas proximidades da placa de circuito impresso (PCB) em que o mesmo foi montado. Por convenção, esses campos são divididos entre os que são fixos (denominado efeitos de *Hard-Iron*) e aqueles que são induzidos pelo campo geomagnético (efeitos de *Soft-Iron*) [Ozyagcilar, 2012, 2011]. Sendo assim, os efeitos correspondentes às interferências de campos magnéticos fixos como os provocados pelos circuitos eletrônicos próximos ao magnetômetro na PCB são entendidos como efeitos de *Hard-Iron* e inserido no modelo do magnetômetro como um vetor de polarização (*bias*).

A Equação (4.6) descreve esse procedimento:

$$\mathbf{H}_{\{rm\}} = \mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)\mathbf{H}_{\{rt\}} + \mathbf{V}. \quad (4.6)$$

Os vetores $\mathbf{H}_{\{rm\}} \in \mathbb{R}^{3 \times 1}$ e $\mathbf{H}_{\{rt\}} \in \mathbb{R}^{3 \times 1}$ correspondem aos campos magnéticos nos referenciais do magnetômetro e da terra, respectivamente. As matrizes $\mathbf{R}_x(\phi)$, $\mathbf{R}_y(\theta)$ e $\mathbf{R}_z(\psi) \in \mathbb{R}^{3 \times 3}$ equivalem às rotações

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}$$

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

e

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

O vetor $\mathbf{V} \in \mathbb{R}^{3 \times 1}$ é definido como:

$$\mathbf{V} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

representa os efeitos de *Hard-Iron* nos três eixos de medição e seus valores foram estimados através de procedimentos experimentais. A metodologia para obtenção desses valores segue na Seção 4.3.4. Os efeitos de *Soft-Iron* não foram considerados para essa modelagem.

A próxima etapa da modelagem equivale à inserção dos desalinhamentos referentes às imperfeições na PCB. Para tal, o valor $\mathbf{H}_{\{rm\}}$ foi convertido para coordenadas esféricas, somados os ângulos de inclinação na placa e convertidos novamente para coordenadas cartesianas. A Equação (4.7) apresenta essa formulação:

$$\mathbf{H}_{\{rp\}} = sph2cart[cart2sph(\mathbf{H}_{\{rm\}}) + \mathbf{Incl}_{PCB}] \quad (4.7)$$

A operação *cart2sph* equivale à conversão de coordenadas cartesianas para esféricas e gera o vetor

$$\begin{bmatrix} \rho \\ azimuth \\ elev \end{bmatrix} = \begin{bmatrix} \sqrt{H_{x\{rm\}}^2 + H_{y\{rm\}}^2 + H_{z\{rm\}}^2} \\ atan\left(\frac{H_{y\{rm\}}}{H_{x\{rm\}}}\right) \\ atan\left(\frac{H_{z\{rm\}}}{\sqrt{H_{x\{rm\}}^2 + H_{y\{rm\}}^2}}\right) \end{bmatrix}$$

Esse vetor que contém os ângulos de azimute e de elevação no referencial do sensor é somado ao vetor de inclinação no referencial da placa

$$\mathbf{Incl}_{PCB} = \begin{bmatrix} 0 \\ azimuth_{PCB} \\ elev_{PCB} \end{bmatrix}$$

e convertido para coordenadas cartesianas através da operação *sph2cart* gerando, dessa

forma, o vetor $\mathbf{H}_{\{rp\}}$ que representa o campo magnético medido no referencial da placa.

$$\mathbf{H}_{\{rp\}} = \begin{bmatrix} \rho \cos elev_{tot} \cos azim_{tot} \\ \rho \cos elev_{tot} \sin azim_{tot} \\ \rho \sin elev_{tot} \end{bmatrix}$$

Feito isso, o vetor $\mathbf{H}_{\{rp\}}$ é somado ao ruído de medição e multiplicado pela matriz de fator de escala e eixo cruzado, conforme apresenta Equação (4.8). Assim obtém-se o valor de saída do magnetômetro em LSB's.

$$\mathbf{H}_{saída} = \mathbf{S}(\mathbf{H}_{\{rp\}} + \mathbf{H}_{noise}). \quad (4.8)$$

A matriz $\mathbf{S} \in \mathbb{R}^{3 \times 3}$ apresenta em sua diagonal principal os valores de fator de escala e os fatores de eixo cruzado nas demais posições. Esses valores foram extraídos da folha de dados do HMC5883L.

O vetor $\mathbf{N} \in \mathbb{R}^{3 \times 1}$ é o ruído de medição, considerado como Branco e Gaussiano e cujo valor da densidade espectral foi obtido através da aplicação da Variância de Allan em dados reais obtidos do sensor.

Na última etapa do modelo foi inserido um bloco de saturação para enquadrar os valores de saída do sensor aos limites do conversor AD.

4.3.4 Bias e Densidade Espectral do Ruído

O *bias* e a densidade espectral do ruído são parâmetros muito importantes para criação de modelos que emulem as características reais dos sensores da IMU. Seus valores correspondentes geralmente aparecem na folha de dados dos fabricantes, contudo, para que a saída dos modelos pudesse se equiparar com as medidas reais dos sensores disponíveis, esses valores foram estimados através de procedimentos experimentais, apresentados em detalhes no Capítulo 5. A seguir é apresentada a metodologia para obtenção desses valores.

Obtenção do Bias dos sensores

Conforme apresentado na Seção 2.5, *bias* é uma média dos valores de saída quando nenhum sinal de entrada é aplicado ao sistema de detecção do sensor. Sendo assim, em uma definição mais geral, o *bias* pode ser considerado como a diferença entre essa média e o valor esperado para o sensor quando o mesmo encontra-se estático:

$$Bias_{sensor} = \bar{V}_{med} - V_{esp}, \quad (4.9)$$

em que \bar{V}_{med} equivale à média dos valores medidos e V_{esp} ao valor esperado para aquela condição.

Considerando que a IMU está nivelada e com inclinação zero, e que não existem acelerações lineares sendo aplicadas à mesma, o valor esperado na saída do acelerômetro é:

$$\bar{A}_{saida} = \begin{bmatrix} 0 \\ 0 \\ -1g \end{bmatrix}$$

A projeção da gravidade deve aparecer integralmente no eixo z , sendo equivalente à aceleração da gravidade ($1g = 9,81m/s^2$) e com valor negativo, conforme explicado na Seção 4.3.1.

Sendo assim o *bias* do acelerômetro nos eixos x e y equivale à própria média dos valores medidos para esses eixos. No caso do eixo z é aplicada a Equação (4.9).

No girômetro é esperado que todas as velocidades angulares medidas sejam iguais a zero quando o sensor estiver imóvel sob a estrutura de coleta de dados. Nesse caso o *bias* corresponde à própria média dos valores medidos para cada eixo. É importante lembrar que nesse trabalho a instabilidade do *bias* do girômetro não foi implementado no modelo, sendo assim, o valor calculado por esse procedimento corresponde ao *bias* estático.

No caso do magnetômetro, o *bias* deve ser calculado por um procedimento um pouco diferente, pois não se conhece o valor esperado para as projeções do campo magnético no local de testes. Dessa forma, deve-se adotar o valor esperado como sendo a média dos valores medidos sem interferências de outros campos magnéticos e com o eixo x do magnetômetro alinhado na direção Norte, para possibilitar que a projeção do campo magnético alcance seu valor máximo nessa medição. A obtenção do valor do *bias* é realizada como segue:

$$Bias_{mag} = \bar{V}_{B2} - \bar{V}_{B1} \quad (4.10)$$

onde \bar{V}_{B2} corresponde à média dos valores medidos sob a interferência de outros materiais ou campos magnéticos e \bar{V}_{B1} é a média dos valores medidos quando não existe esse efeito. O resultado obtido da Equação (4.10) é provocado pelos efeitos de Hard-Iron provocados pela presença dos campos magnéticos nas proximidades do magnetômetro.

Obtenção da Densidade Espectral do Ruído

O método de Variância Allan é frequentemente usado para identificar e quantificar ruídos aleatórios com diferentes propriedades de autocorrelação e expor o seu efeito quando se tira a média ou se integra o sinal ao longo do tempo.

A Variância de Allan (AVAR) é um método de análise de uma série temporal para se determinar o nível de ruído no sistema como uma função do tempo médio. Originalmente foi desenvolvido para caracterização de ruídos e analisar a estabilidade de clocks, contudo, esse método pode ser adaptado para outras aplicações como é o caso

da determinação da instabilidade do *bias* e da densidade espectral de ruído em sensores inerciais [El-Sheimy et al., 2008].

A ideia básica consiste em se extrair uma longa sequência de dados, dividir esses dados em blocos com um tempo médio τ conforme ilustra a Figura 4.5, e calcular a média dos dados em cada um dos blocos. Em seguida pega-se a diferença da média entre blocos sucessivos, eleva-se esse número ao quadrado, soma-se todos, e divide-se por um fator de escalonamento. Por fim, extrai-se a raiz quadrada do resultado. Dessa forma tem-se uma medida quantitativa de como o valor médio mudou naquele determinado intervalo de tempo. Esse procedimento é repetido incrementando-se o valor de τ até um determinado número mínimo de blocos.

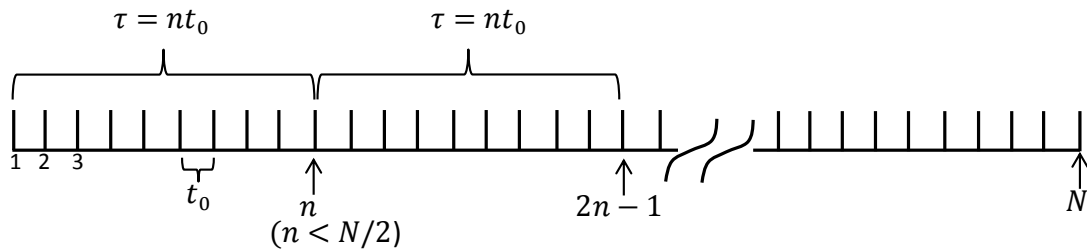


Figura 4.5: Blocos de dados para cálculo da Variância de Allan
Fonte: Figura adaptada a partir de [Hou, 2004]

Conforme estabelecido por Hou [2004], N é o número de pontos consecutivos na sequência de dados, cada ponto com um período de amostragem igual a t_0 . Cada bloco é formado por n pontos sendo $n < N/2$. Dessa forma $\tau = n \cdot t_0$.

A média dos dados extraídos do sensor, onde a leitura inicia no k -ésimo instante de tempo e possui n pontos é:

$$\overline{\mathfrak{M}}_k(\tau) = \frac{1}{\tau} \sum_{t_k}^{t_k+\tau} \mathfrak{M}(t) \partial t \quad (4.11)$$

onde $\mathfrak{M}(t)$ é uma função que representa o sinal de saída do sensor. A média do bloco subsequente é:

$$\overline{\mathfrak{M}}_{k+1}(\tau) = \frac{1}{\tau} \sum_{t_{k+1}}^{t_{k+1}+\tau} \mathfrak{M}(t) \partial t \quad (4.12)$$

em que t_{k+1} equivale a $t_k + \tau$.

A diferença entre dois blocos adjacentes é dada por:

$$\xi_{k+1,k} = \overline{\mathfrak{M}}_{k+1}(\tau) - \overline{\mathfrak{M}}_k(\tau) \quad (4.13)$$

Para cada bloco de intervalo τ o conjunto de ξ 's forma uma série de variáveis aleatórias. A quantidade de interesse é a Variância de ξ sobre os blocos de mesmo tamanho, formados a partir dos dados extraídos dos sensores.

Sendo assim a Variância de Allan é definida por [IEEE, 1998] como:

$$\sigma^2(\tau) = \frac{1}{2} \langle \xi_{k+1,k}^2 \rangle \quad (4.14)$$

em que a expressão $\langle \rangle$ denota valor esperado.

De acordo com as definições de [Hou, 2004], para qualquer número finito de N pontos, um número finito de blocos de duração fixa τ pode ser formado. A Equação (4.15) representa uma estimativa da Variância de Allan, cuja qualidade depende do número de grupos independentes que pode ser formado com tamanho fixo τ [Hou, 2004; Tehrani, 1983].

$$\sigma^2(\tau) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} [\overline{\mathfrak{M}}_{k+1}(\tau) - \overline{\mathfrak{M}}_k(\tau)]^2, \quad (4.15)$$

A curva da Variância de Allan contém essencialmente a mesma informação que a Densidade Espectral de Potência (PSD) do sinal, contudo apresenta essas informações de uma forma mais conveniente. A PSD e a raiz da Variância de Allan são relacionadas entre si da seguinte forma:

$$PSD(f) \propto f^\alpha \equiv \sigma(\tau) \propto \tau^\beta \quad (4.16)$$

onde $\beta = -(\alpha + 1)/2$ e $f = 1/\tau$.

O ruído branco é um sinal aleatório com PSD constante e independente da frequência, isto é, $PSD(f) \propto f^\alpha$ com $\alpha = 0$. Assim, em um gráfico log-log da raiz da Variância de Allan ($\sigma(\tau)$), o ruído branco é identificado por uma reta de inclinação $\beta = -1/2$. O coeficiente para o ruído branco pode ser obtido diretamente para $\tau = 1$ e a densidade espectral do ruído utilizada nos modelos dos sensores equivale ao quadrado desse coeficiente.

A Tabela 4.1 apresenta alguns dos principais parâmetros utilizados nas configurações dos sensores modelados. Conforme exposto, o *Bias* e densidade de ruído espectral foram obtidos experimentalmente para cada sensor, já os demais parâmetros foram retirados da folha de dados de seus respectivos fabricantes.

Tabela 4.1: Valores dos parâmetros dos sensores.

Parâmetros	Acelerômetro	Girômetro	Magnetômetro
Bias	[-0,02 0,02 -0,00]	[-3,10 3,60 0,04]	[-0,16 0,17 -0,02]
Densidade Espectral do Ruído	1e-4*[0,16 0,16 0,15]	1e-3*[0,05 0,12 0,04]	1e-6*[0,58 0,77 0,20]
Fator de Escala	256	14,375	660
Acoplamento Cruzado	1%	2%	0,2%
Resolução do ADC	10-bits	16-bits	12-bits

4.3.5 A IMU/HIL

Os três modelos dos sensores, os quais foram implementados separadamente, são agrupados para dar origem ao modelo da Unidade de Medições Inerciais, conforme ilustra a Figura 4.6.

Algumas modificações foram realizadas para melhorar a eficiência do código gerado, como, por exemplo, o código da derivada discreta foi extraído do interior do modelo do girômetro e colocado externamente para que pudesse servir de entrada também para o modelo do acelerômetro. Além disso, foram implementadas algumas *flags* para controlar o fluxo do código e variáveis de usuário pra bufferizar dados no modelo (como, por exemplo, aqueles provenientes da interface SCI).

O bloco “TMS320F28335” no canto superior esquerdo do modelo é responsável pelas definições do processador utilizado e as configurações de seus periféricos, além de outros ajustes para compilação do código gerado. Esse bloco faz parte da plataforma *Embedded Target* para os DSPs da família TI TMS320C2000™ [MathWorks, 2003].

Os blocos “SCI RCV” na entrada do modelo e “I2C XMT” na saída, são responsáveis pelas configurações de seus respectivos periféricos de comunicação nos DSPs da família C2000 da Texas.

O bloco “SCI RCV” está configurado para o modo FIFO com uma profundidade de 12-Bytes de forma a gerar uma interrupção apenas quando novos dados de atitude e acelerações lineares já estiverem disponíveis para a leitura. Isso possibilita que o DSP ocupe sua capacidade de processamento com a função principal de cálculo dos modelos dos sensores.

O bloco “I2C XMT” foi configurado para o modo “*pooling*”, ou seja, sem a geração de interrupção, e envia os dados de girômetro, acelerômetro e magnetômetro para o dispositivo de aquisição de dados, que nesse caso é a Cabeça Sensora.

Uma ressalva sobre esse bloco é que o mesmo teve que ser adaptado para o emulador IMU/HIL uma vez que o código de comunicação com a Cabeça Sensora foge um pouco dos padrões usuais. Isso se explica porque, uma vez que na comunicação I2C o endereço do dispositivo escravo deve ser único, e existe a necessidade de se emular três sensores com endereços diferentes, então teve que se fazer uma adaptação para que a interface I2C sempre modificasse seu endereço de escravo para o do próximo sensor que será requisitado de modo que, quando requisitado pela Cabeça Sensora, o DSP pudesse atender a essa requisição. O Apêndice A apresenta maiores detalhes sobre as modificações realizadas.

Outra solução teria sido implementar um código que emulasse em software a interface I2C com endereços múltiplos ou ainda ter gerado um código com um único endereço I2C que enviasse os dados dos três sensores em uma única requisição da Cabeça Sensora. A primeira opção é pouco eficiente pois ocupa tempo de CPU e desperdiça o periférico interno existente para essa função; a segunda implicaria na modificação do código original da Cabeça Sensora.

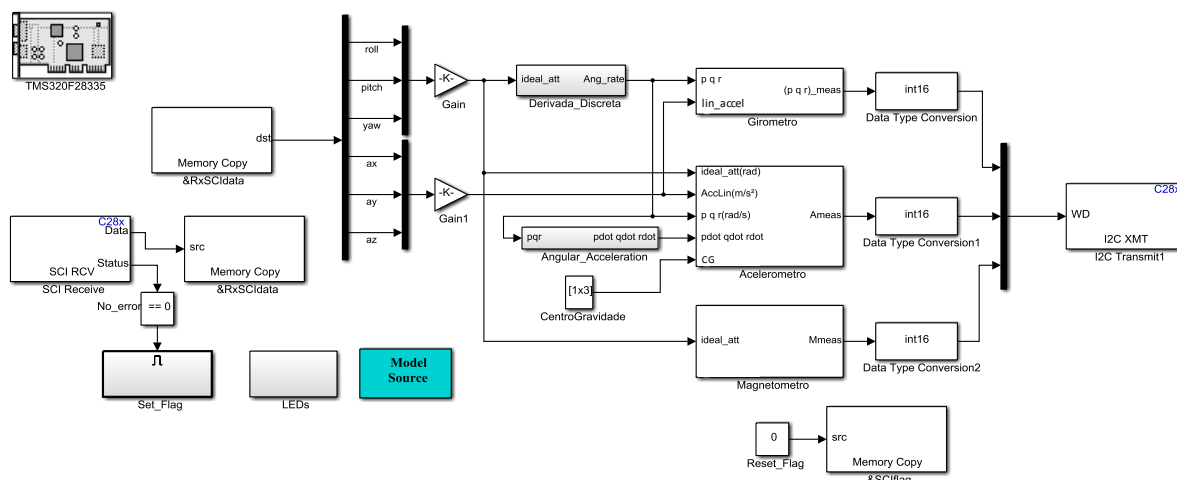


Figura 4.6: Modelo da Unidade de Medições Inerciais no MATLAB/Simulink

4.3.6 Blocos de Otimização de Código da Texas: IQmath

A *Texas Instruments* fornece uma biblioteca que usa o hardware interno dos DSPs da família C28x da forma mais eficiente para operar com números de 32-bits de ponto fixo. A biblioteca “*IQmath*” é uma coleção de funções matemáticas altamente otimizadas e precisas para se transpor algoritmos de ponto-flutuante em códigos de ponto-fixos [Texas, 2010].

Todas as rotinas executadas com o uso da biblioteca “*IQmath*” não realizam chamadas de função para a realização dos cálculos. Em vez disso, um conjunto otimizado de algumas linhas de código de máquina é colocado diretamente na sequência de compilação. A vantagem é um tempo de execução muito curto, porque não há chamada nem retorno de função, poupando a necessidade de salvamento e restauração de contexto. Naturalmente, o tamanho do código aumenta ligeiramente, no entanto, todas as funções da “*IQmath*” são otimizadas e consistem em apenas algumas palavras de código [Texas, 2010].

Uma vez que a maioria dos dados de aplicações práticas não costumam ultrapassar uma resolução de 16 bits, caso dos sensores da IMU, por exemplo, a biblioteca “*IQmath*” oferece espaço suficiente para cálculos numéricos avançados. Estas rotinas são normalmente utilizados em aplicações em tempo real, onde a velocidade de execução ideal e de alta precisão é fundamental.

A maioria dos cálculos dos subsistemas internos no modelo da IMU foram realizados através da utilização dos blocos “*IQmath*”. O formato utilizado para as frações binárias foi o IQ15 que, para os 32-bits do processador, corresponde a 1-bit de sinal, 15-bits sendo utilizados para a parte inteira (I - “*Integer*”) e 16-bits para a parte fracionária (Q - “*Quotient*”). A Figura 4.7 mostra, por exemplo, o subsistema de conversão de coordenadas cartesianas para esféricas no modelo do magnetômetro.

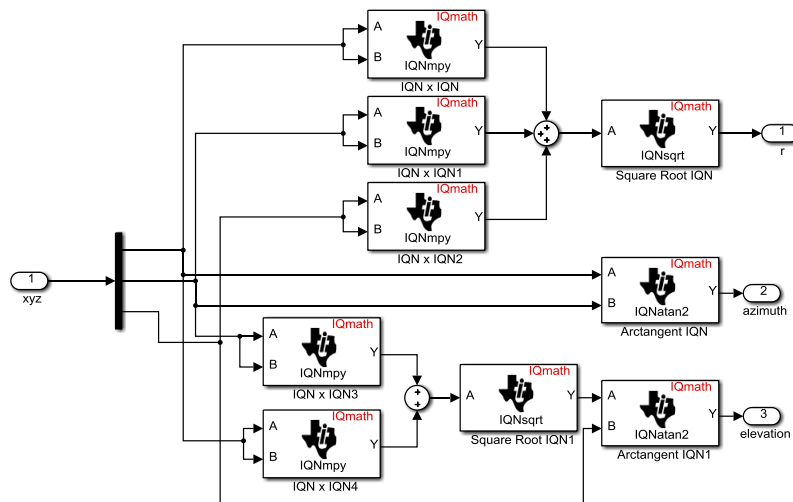


Figura 4.7: Exemplo de utilização da biblioteca “IQmath” no modelo da IMU/HIL

Experimentos & Resultados

“Research is to see what everybody else has seen, and think what nobody has thought.”

Albert Szent-Gyorgyi

5.1 Introdução

Nesse capítulo são apresentados os experimentos realizados para a implementação e teste do sistema desenvolvido.

A primeira etapa dos experimentos teve o objetivo de realizar o levantamento de parâmetros a partir de dados extraídos dos sensores reais disponíveis. Em seguida os modelos foram simulados no emulador IMU/HIL e seus sinais foram comparados aos dos dispositivos reais.

Numa segunda etapa são apresentados os experimentos para verificação do comportamento dinâmico dos modelos confrontando os dados gerados pelo emulador IMU/HIL com os dados dos sensores reais para um mesmo estímulo aplicado. Também são apresentadas as atitudes estimadas pela Cabeça Sensora comparando novamente os dados do emulador IMU/HIL com os da IMU real.

Por fim, são discutidas as características dos códigos gerados como tempo de execução e o tamanho ocupado em memória.

5.2 Levantamento de Dados Estáticos dos Sensores

O primeiro experimento, executado ainda na fase de modelagem, foi realizado com o objetivo de coletar dados estáticos dos sensores e, a partir destes, extrair os parâmetros de *bias* e densidade espectral de ruído conforme metodologia apresentada na Seção 4.3.4.

Esses dados foram coletados em duas etapas:

- primeiramente os sensores foram mantidos estáticos sobre uma superfície plana afastada de qualquer material ferromagnético que pudesse causar interferências

ou dispersões nos campos magnéticos (Figura 5.2). Foram coletadas 10000 amostras de dados na taxa de 50Hz (mesma taxa utilizada pela Cabeça Sensora);

- em seguida o mesmo procedimento foi repetido, porém com a IMU fixa sobre o suporte do manipulador usado nos testes dinâmicos, conforme ilustra Figura 5.4.

Uma bússola militar profissional dotada de clinômetro de pêndulo e bolha de nível foi utilizada para garantir que a IMU se encontrava nivelada e com inclinação igual a zero durante a coleta de dados. O suporte da IMU foi fixado à superfície da bússola com seu eixo x apontando na mesma direção e sentido da linha norte indicada pela flecha da bússola. A medida apresentada na lente de aumento para medição de ângulo é de 0° , o que indica que a bússola aponta exatamente para o norte durante a execução dos experimentos. Essa premissa faz-se necessária para garantir que o magnetômetro apresente um valor máximo de projeção do vetor campo magnético para esse eixo de medição no momento da coleta de dados. Os detalhes apresentados na parte superior da Figura 5.2 expõem essas configurações da bússola. Os eixos desenhados próximos à IMU indicam a orientação da mesma na estrutura. O símbolo \otimes indica que o eixo z penetra no plano da IMU apontando para baixo, enquanto x aponta para frente (proa) e y para a direita.

O alinhamento do suporte da IMU com a bússola foi realizado de forma visual e sem a utilização de instrumentos de medida, no entanto, é possível ter uma noção da precisão desse alinhamento com base na visualização da Figura 5.1. Essa figura foi extraída a partir de um experimento onde realizou-se a rotação em torno do eixo z da IMU do ângulo 270° até o ângulo de 0° , parando brevemente pelos ângulos de 180° e 90° .

Dada a afirmação de que a maior projeção do campo ocorre quando o magnetômetro está alinhado com as linhas de força que apontam para o norte, então, com base nas projeções para os eixos x e y para cada uma dessas posições é possível assumir que a IMU estava razoavelmente alinhada com a bússola. Quando alinhado com os

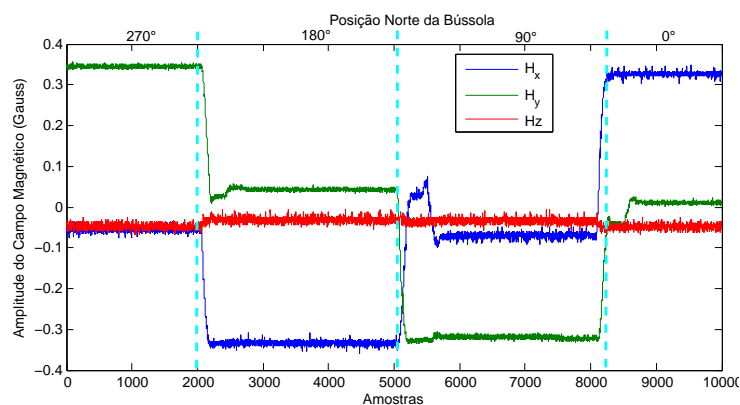


Figura 5.1: Dados do Magnetômetro Real para rotação de 270° a 0°

ângulos de 0° e 180° , por exemplo, o eixo x estava em seus valores máximos positivo e negativo, respectivamente. Nessas mesmas posições os valores apresentados pelo eixo y eram aproximadamente iguais ao *bias*. Os pequenos desvios no eixo z indicam que o campo magnético era praticamente horizontal e que os desalinhamentos da IMU eram mínimos.

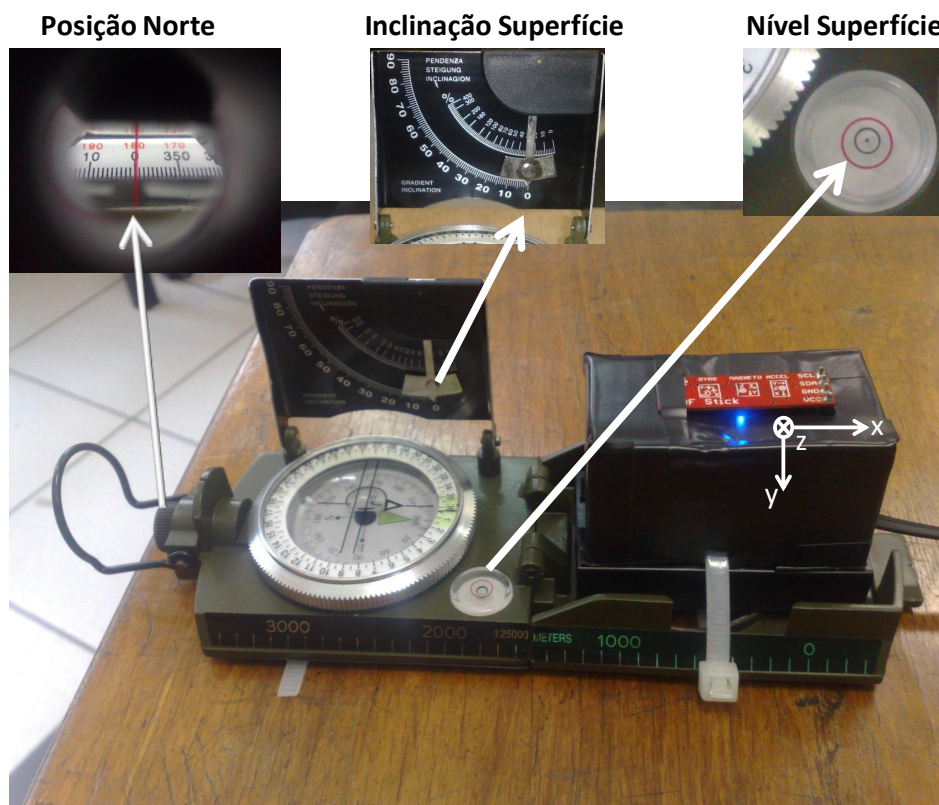


Figura 5.2: Coleta de Dados dos Sensores sob superfície paramagnética plana

Os dados extraídos desse experimento foram utilizados para os cálculos de *bias* e densidade espectral de ruído os quais em seguida foram introduzidos nos modelos desenvolvidos para os sensores da IMU. Para que se tenha uma idéia em aspecto qualitativo dos resultados obtidos, apresenta-se na Figura 5.3 os dados extraídos para o acelerômetro da IMU real em comparação com os dados emulados. Os sinais de entrada no emulador IMU/HIL para essa simulação foram todos iguais a zero para simular as condições estáticas dos sensores. É possível observar que os dois sinais apresentam características bem próximas.

A Tabela 5.1 sumariza as relações entre os sinais da IMU real e do emulador para os três sensores modelados e dá uma noção mais apropriada para comparação entre esses sinais. Pode-se observar que são bastante pequenas as diferenças entre a média,

três desvios padrão (3σ) e variância (σ^2) para os sinais reais da IMU e os emulados pelo sistema desenvolvido. Em aspectos práticos, essas diferenças poderiam ocorrer inclusive entre sensores de um mesmo lote de fabricação.

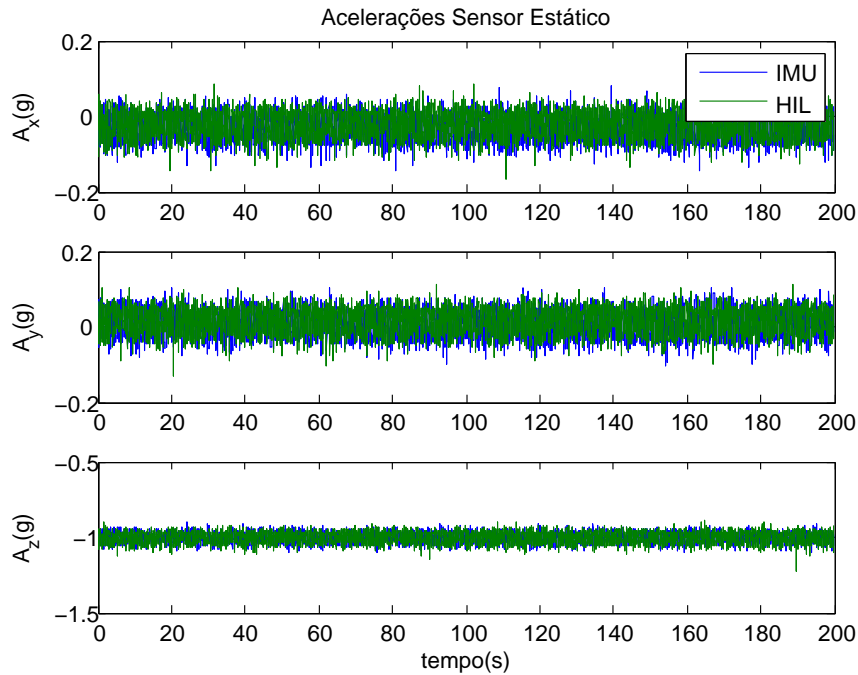


Figura 5.3: Dados do Acelerômetro Real e Emulado em condições estáticas

Tabela 5.1: Comparação das Condições Estáticas dos Sensores da IMU real e Emulada.

Sensor	Média			3σ			σ^2		
	x	y	z	x	y	z	x	y	z
Accel IMU	-0.0188	0.0196	-1.0007	0.0831	0.0801	0.0887	0.0008	0.0007	0.0009
Accel HIL	-0.0301	0.0076	-1.0035	0.0857	0.0845	0.0839	0.0008	0.0008	0.0008
Gyro IMU	-3.1041	3.5997	0.0438	0.1312	0.1475	0.1338	0.0019	0.0024	0.0020
Gyro HIL	-3.0313	3.5380	0.0535	0.1452	0.2370	0.1356	0.0023	0.0062	0.0020
Mag IMU	0.3239	-0.0447	0.0060	0.0149	0.0190	0.0082	0.0000	0.0000	0.0000
Mag HIL	0.3231	-0.0447	0.0056	0.0164	0.0187	0.0097	0.0000	0.0000	0.0000

5.3 Dados Dinâmicos dos Sensores

Os resultados mostrados na seção anterior realizaram uma comparação dos dados estáticos da IMU real com relação aos gerados pelo emulador, contudo, o principal interesse com relação aos modelos implementados é saber se os mesmos conseguem reproduzir as características dos sensores da IMU em movimento.

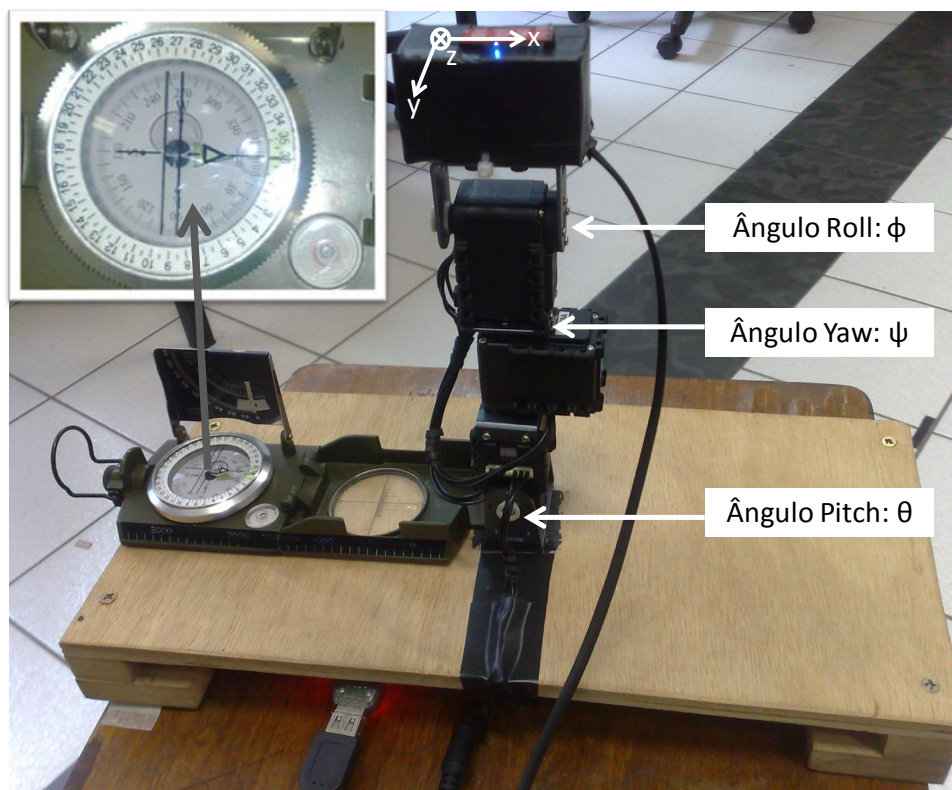


Figura 5.4: Coleta de Dados dos Sensores sob superfície do Manipulador

Sendo assim, foi necessária a montagem de um mecanismo que permitisse coletar simultaneamente as informações de posição real e os dados dos sensores da IMU. Esse mecanismo segue ilustrado na Figura 5.4 e corresponde a um pequeno manipulador robótico sobre o qual foi afixado o suporte da IMU.

O Manipulador é constituído por servomotores AX-18A da ROBOTIS [Robotis, 2014], cujas principais características para essa aplicação seguem resumidas abaixo:

- Resolução: $0,29^\circ$
- Velocidade Máxima a vazio: 97rpm
- Ângulos de atuação: 0° a 300°
- Material: plástico de engenharia

Esses servomotores fazem parte da linha de atuadores de alta performance desse fabricante para a montagem de robôs. Nos experimentos realizados os servos foram arranjados de modo a formar três juntas que possibilitam movimentar o suporte da IMU nos três ângulos de Euler: *roll* (rolagem), *pitch* (arfagem) e *yaw* (guinada).

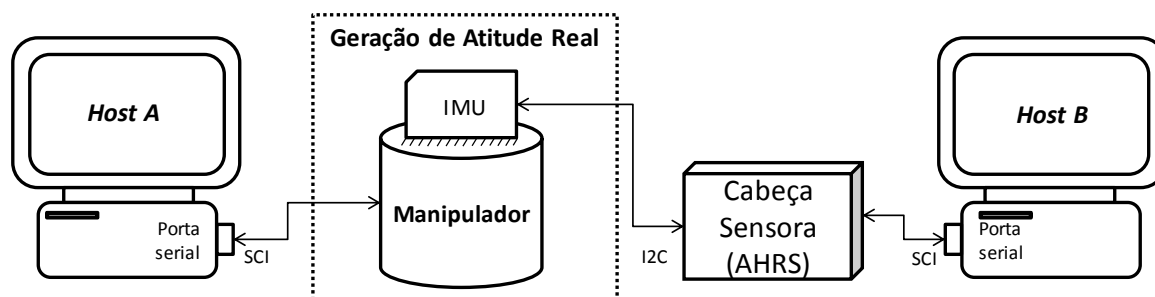


Figura 5.5: Esquema para geração de atitude Real

Os servomotores do manipulador podem ser configurados em rede, cada um com seu endereço específico, e possibilitam a leitura das posições de seus eixos através de uma comunicação serial assíncrona *half-duplex*. Essas posições medidas correspondem aos ângulos de rotação e foram consideradas como a “Atitude real” gerada. Em uma segunda etapa essas posições foram introduzidas como entrada para o emulador IMU.

O diagrama apresentado na Figura 5.5 esquematiza o procedimento utilizado para realização desse experimento. A Unidade **Host A** envia instruções de velocidade e trajetória para os servomotores e recebe a posição angular de seus eixos. A IMU, que está afixada sobre o manipulador, fornece os sinais de medição para a Cabeça Sensora que, por fim, envia os dados dos sensores e a atitude estimada para a Unidade **Host B**.

Com todos os servos na posição 0° o manipulador encontra-se vertical formando um ângulo de 90° com a base da estrutura. Nessa posição, o suporte da IMU apresenta as mesmas características empregadas para coleta de dados estáticos, ou seja, está nivelada com inclinação zero e apontando para o norte. A Figura 5.4 ilustra exatamente esse estado. As juntas responsáveis pelos movimentos de cada ângulo também estão destacadas na figura, assim como o sentido positivo para orientação dos eixos. Inicia-se a coleta de dados na posição 0° e envia-se uma instrução de deslocamento rápido (velocidade constante e igual a 52,2rpm) para o primeiro motor (ângulo θ) em direção à posição 50° . O manipulador é mantido nessa posição por um pequeno intervalo de tempo e em seguida envia-se uma instrução de deslocamento lento para a posição -50° à velocidade de 2,9rpm. Após algumas repetições do movimento de vai-vem do motor a estrutura é reposicionada a 0° e permanece estática por um pequeno intervalo de tempo. Uma instrução de deslocamento rápido é enviada para o segundo motor (ângulo ψ) e por fim para o terceiro (ângulo ϕ) repetindo o procedimento utilizado para o motor 1, contudo, para um deslocamento até a posição 80° .

A Figura 5.6 mostra os gráficos da atitude real coletada dos servomotores (parte superior da figura), a atitude estimada a partir dos sensores reais da IMU (parte central da figura) e a atitude estimada a partir dos sensores emulados (parte inferior da figura).

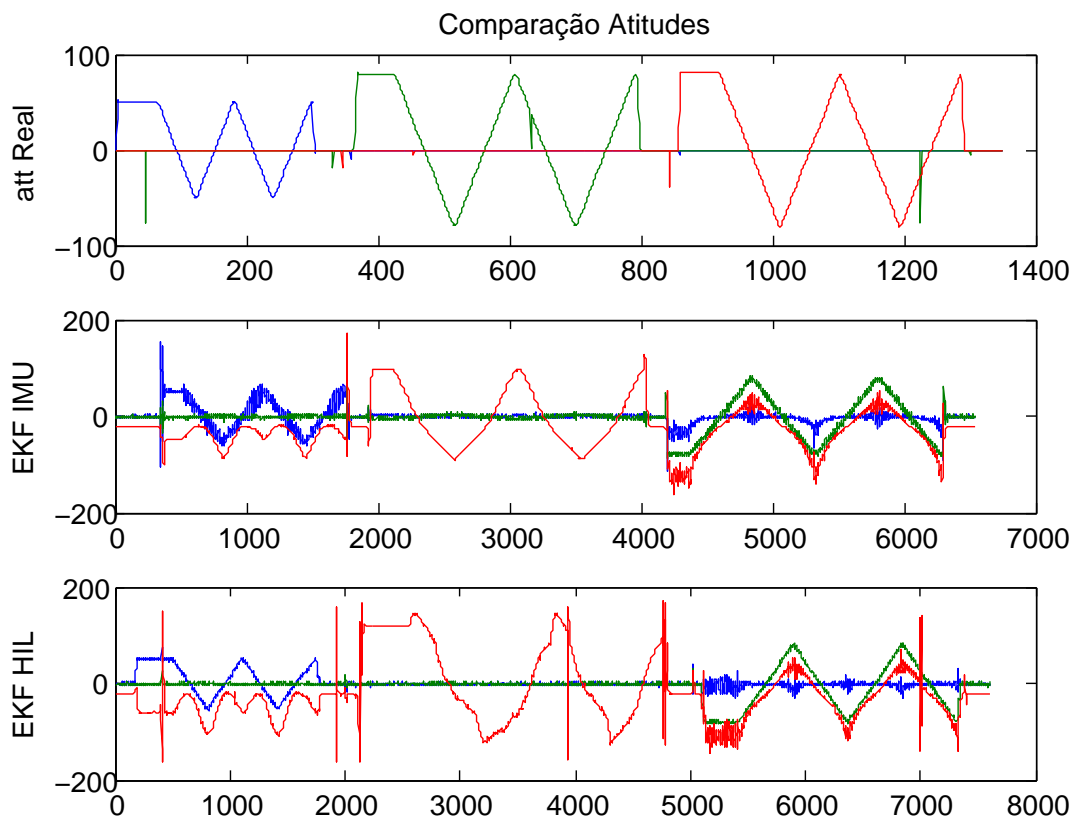


Figura 5.6: Comparação entre as atitudes Real, Estimada para IMU e Estimada para IMU/HIL

Apesar dos dados dos servomotores e os da IMU terem sido coletados ao mesmo tempo é possível observar que a quantidade de amostras existentes é muito maior para a IMU. Isso se deve ao fato de que a interface de comunicação serial com os servos é bem mais lenta, acarretando assim em uma menor quantidade de amostras. Assim, antes de se enviar os dados de posição dos servos para a entrada do emulador IMU teve que se fazer a interpolação dos mesmos para possibilitar uma quantidade de amostras mais próxima daquela coletada para a IMU.

O procedimento para se obter a atitude estimada a partir dos sensores emulados foi basicamente o mesmo apresentado no diagrama da Figura 5.5. A diferença é que no lugar do bloco "Gerador de Atitude Real" foi colocado o kit de desenvolvimento com o DSP TMS28335 e ao invés de instruções de velocidade e trajetória foram enviados os dados coletados dos servomotores.

Comparando os gráficos de atitude estimada para a IMU real e para o emulador pode-se observar que o sinal estimado para o IMU/HIL encontra-se mais espalhado do que os da IMU real. Isso se deve ao fato de que a existência de ruídos nos dados provenientes da Cabeça Sensora dificulta a determinação exata do período do sinal

para que a partir deste se proceda o cálculo do coeficiente de interpolação. Sendo assim, a interpolação realizada nos dados de atitude real resultou em uma quantidade de amostras maior do que a dos dados reais.

Ainda assim é possível perceber que a atitude gerada para os dados da IMU real e do IMU/HIL é muito semelhante, o que significa que os modelos realmente conseguem reproduzir as características dinâmicas dos sensores reais de forma coerente. Além disso, observando visualmente o sistema em simulação foi possível perceber que as variações no sinal da entrada eram instantaneamente reproduzidas no sinal de saída, o que transmite uma ideia de que o sistema esteja funcionando em tempo real. Essa suposição é confirmada na próxima seção a partir da avaliação do tempo de execução do código.

Com relação a atitude estimada é possível observar que a versão do EKF utilizada na Cabeça Sensora não estava devidamente calibrada, mas, uma vez que essa calibração foge do escopo desse trabalho é o suficiente saber que o IMU/HIL pode ser utilizado para facilitar a realização da mesma, uma vez que consegue reproduzir as características dos sensores em tempo real.

5.4 Eficiência do Código Gerado

A utilização da plataforma *Embedded Target* do MATLAB/Simulink possibilitou a geração automática dos códigos em C para os modelos dos sensores implementados no Simulink.

Conforme apresentado na seção anterior, esses códigos funcionam corretamente, uma vez que os sensores modelados apresentaram respostas coerentes e resultaram em uma atitude estimada bastante similar à gerada a partir da IMU. Contudo, para garantir que as informações de saída estejam disponíveis no barramento I2C no momento em que são requisitadas, é necessário que o DSP seja capaz de realizar todos os cálculos do modelo em um intervalo de tempo menor do que o período de amostragem determinado pela Cabeça Sensora.

Isto pode ser confirmado a partir da Figura 5.7. Essa figura foi extraída com o auxílio do osciloscópio TDS2000 da Tektronix e corresponde ao tempo total de execução das rotinas de cálculos do modelo. Esse sinal foi gerado pela medição de um dos pinos de saída digital do DSP o qual foi configurado para comutar para o nível alto no início da execução da rotina de cálculos e para nível baixo logo após a finalização da mesma.

O DSP leva em média $250\mu s$ para realizar todos os procedimentos de cálculo do modelo, o que supera em muito aos requisitos da Cabeça Sensora, já que o período de amostragem estipulado pela mesma é de $20ms$. Na realidade, com esse tempo de execução o DSP conseguiria trabalhar até nas maiores frequências de amostragem impostas para cada sensor e ainda com bastante folga. Isso comprova que o código gerado é eficiente com relação ao atendimento das condições de tempo real do sistema.

É importante observar que o código gerado foi executado na memória RAM do

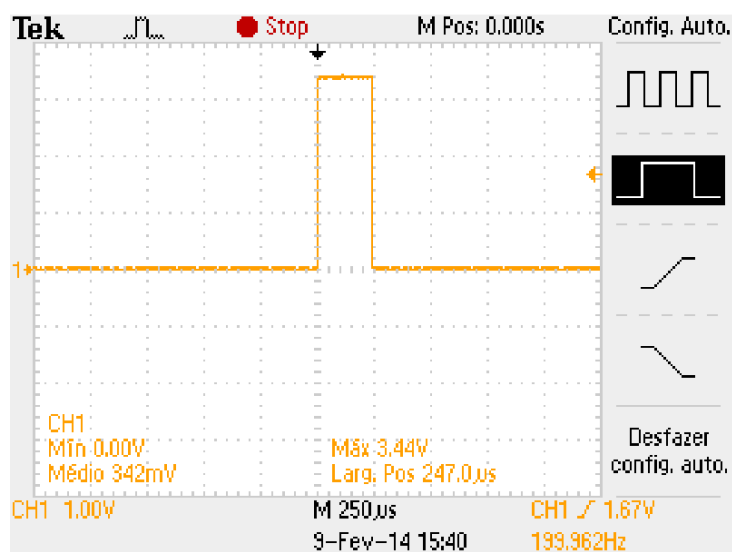


Figura 5.7: Tempo de Execução dos cálculos do Modelo da IMU/HIL

DSP. A transferência desse código para a FLASH acarretaria em um aumento no tempo de execução, mesmo assim esse aumento é bastante sensível e não prejudicaria o desempenho do sistema.

Outro fator também muito importante para sistemas embarcados é a quantidade de espaço ocupado pelo código na memória de programa. A Tabela 5.2 mostra a ocupação de memória para as principais rotinas geradas. É possível observar que a quantidade de memória ocupada é relativamente pequena quando comparada com a capacidade total apresentada pelo DSP, que é de $256k \times 16$ para memória FLASH e $34k \times 16$ para memória RAM.

Tabela 5.2: Ocupação da Memória pelo Código Gerado.

Código Gerado	Palavras de Memória	Espaço em kB	%RAM	%FLASH
Model_IMU.obj	6396	12,49	18,81%	2,50%
Model_IMU_main.obj	130	0,25	0,38%	0,05%
MW_c28xx_csl.obj	183	0,36	0,54%	0,07%
MW_c28xx_board.obj	119	0,23	0,35%	0,05%
MW_c28xx_i2c.obj	36	0,07	0,11%	0,01%
Demais Rotinas	3839	7,50	11,29%	1,50%
Total	10703	20,90	31,48%	4,18%

Conclusões

“The important thing about a problem is not its solution, but the strength we gain in finding the solution.”

Anonymous

6.1 O Emulador IMU para Testes *Hardware-In-the-Loop* (IMU/HIL)

O presente trabalho foi realizado com o objetivo de desenvolver um sistema *Hardware-In-the-Loop* para emular os dados de sensores de uma Unidade de Medições Inerciais.

Para a realização desse objetivo foram implementados os modelos dos sensores Acelerômetro, Girômetro e Magnetômetro no MATLAB/Simulink e os códigos para o DSP foram gerados automaticamente pela plataforma *Embedded Target* para os DSPs da família TI TMS320C2000™ da Texas, fazendo uso da metodologia *Model-Based Design*.

O código gerado mostrou-se eficiente e, de um ponto de vista geral, o emulador IMU/HIL atendeu ao objetivo proposto uma vez que realiza a comunicação I2C em tempo real com a Cabeça Sensora e entrega informações coerentes com as que se espera dos sensores reais com relação aos dados ideais fornecidos. A atitude estimada a partir dos dados provenientes do IMU/HIL mostrou-se bastante similar àquela obtida com os dados dos sensores reais, confirmando novamente que o objetivo do trabalho foi alcançado. Conforme apresentado nas propostas para trabalhos futuros, ainda devem ser realizadas melhorias nos modelos dos sensores para que suas respostas sejam mais próximas daquelas que seriam enviadas pelos sensores reais.

Uma contribuição importante desse trabalho foi o uso da metodologia *Model-Based Design*. MBD e HIL são duas ferramentas bastante convenientes para o desenvolvimento de projetos em sistemas inteligentes modernos e seu uso contribui consideravelmente para obtenção de um produto final projetado com maior eficiência, qualidade e menor custo e tempo de desenvolvimento.

Como parte dos resultados obtidos ao longo do desenvolvimento desse trabalho foi realizada a publicação do artigo *“Simulação Hardware-In-the-Loop Aplicada à Avaliação*

de Algoritmos de Estimação de Atitude para VANTs” [Ribeiro et al., 2013] no XI Simpósio Brasileiro de Automação Inteligente (SBAI 2013).

6.2 Proposta para Trabalhos Futuros

Começando pelo refinamento dos modelos dos sensores trabalhados, pelo desenvolvimento de modelos de outros sensores e implementação de novas estruturas, o prosseguimento desse trabalho pode seguir à seguinte estrutura:

Finalização do Módulo de Instrumentação: As implementações realizadas nesse trabalho foram baseadas em modelos matemáticos já existentes e difundidos na literatura e, apesar de exibirem resultados coerentes, são muito simples e precisam ser aprimoradas para apresentar reações ainda mais próximas das dos sensores reais. Maiores estudos em identificação de sistemas e processos estocásticos devem ser realizados para incrementar os modelos existentes e melhorar seus resultados.

Além disso, devem ser desenvolvidos os modelos de outros sensores que também podem ser utilizados em sistemas de navegação, tais como, GPS, barômetro, sensor de pressão diferencial, dentre outros, para que se tenha um Módulo de Instrumentação completo emulado pelo HIL.

Melhoria da interface de Comunicação: No presente trabalho a comunicação só flui devido ao fato que se conhece previamente a sequência de requisição da Cabeça Sensora. O IMU/HIL configura-se com o endereço de escravo do próximo sensor que será requisitado e quando essa requisição acontece, então procede-se a comunicação. Esse modelo deve ser modificado para ser robusto a qualquer sequência de requisição. Possíveis soluções são a implementação em software de um emulador de interface I2C ou então a inserção de dispositivos de hardware externos para cumprir com essa função.

Implementação do modelo dinâmico do VANT na Unidade DSP: Os modelos dos sensores devem estar na Unidade DSP para contornar o fato da comunicação serial com o MATLAB não atender à velocidade necessária para comunicação com a Cabeça Sensora.

A implementação de modelo dinâmico do VANT também na Unidade DSP seria interessante para possibilitar o funcionamento do sistema em modo automático (quando recebendo dados da Unidade *Host*) e manual através de comandos enviados por *joysticks* e botões. Isso simularia uma situação real e comum de voo com VANTs, além de possibilitar a integração com dispositivos atuadores e de monitoramento.

Implementação de interfaces para dispositivos atuadores e de monitoramento: Uma vez implementado o modelo dinâmico do VANT na Unidade DSP seria possível fazer com que a mesma gerasse dados para dispositivos atuadores tais como motores e servos e ainda disponibilizasse uma interface de transmissão via rádio para comunicar com a estação rádio base de modo que o simulador HIL pudesse emular completamente as características de um VANT real.

Desenvolvimento de um sistema dedicado para o HIL: A execução dos passos descritos anteriormente encaminham para a concepção de um sistema completo para emular todas as características de um VANT real. Sendo assim, convém o desenvolvimento (hardware e software) de um sistema HIL dedicado, com Placa de Circuito Impresso própria e outros recursos disponíveis como, por exemplo, uma interface de visualização .

Referências Bibliográficas

- Adiprawita, W., Ahmad, A. S., and Semibiring, J. (2008). Hardware in the loop simulator in uav rapid development life cycle. *CoRR*, abs/0804.3874.
- AnalogDevices (2012). 3-Axis Digital Accelerometer ADXL345. <http://www.analog.com>. Acessado em 12/fev/2013.
- Ang, W.-T., Khoo, S., Khosla, P., and Riviere, C. (2004). Physical model of a mems accelerometer for low-g motion tracking applications. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1345–1351 Vol.2.
- Boser, B. and Howe, R. (1996). Surface micromachined accelerometers. *Solid-State Circuits, IEEE Journal of*, 31(3):366–375.
- Brown, Robert Grover. e Patrick, Y. H. (1997). *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions, 3rd ed.* John Wiley & Sons.
- Caruso, M. J. and Smith, C. H. (1998). A new perspective on magnetic field sensing. Technical report, Honeywell.
- Cavalcanti, S. and Papini, M. (2004). Preliminary model matching of the embraer 170 jet. *Journal of Aircraft*, 41(4).
- Ciudad, D., Aroca, C., Sánchez, M., Lopez, E., and Sánchez, P. (2004). Modeling and fabrication of a mems magnetostatic magnetic sensor. *Sensors and Actuators A: Physical*, 115:408–416.
- da Paixao, R., Rosa, P., and Neto, J. (2011). An attitude heading and reference system: Basic concepts and prototype. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, pages 2225–2230.
- Dibbern, U. (1986). Magnetic field sensors using the magnetoresistive effect. *Sensors and Actuators*, 10:127–140.
- Du, G. and Chen, X. (2012). Mems magnetometer based on magnetorheological elastomer. *Measurement*, 45:54–58.

- El-Sheimy, N., Hou, H., and Niu, X. (2008). Analysis and modeling of inertial sensors using allan variance. *Instrumentation and Measurement, IEEE Transactions on*, 57(1):140–149.
- Erkkinen, T. and Conrad, M. (2008). Verification, validation, and test with model-based design. *The MathWorks, Inc.* Acesso em 20/fev/2014.
- Fainguelernt, J. (2011). From matlab and simulink to real-time with ti dsp's. In *CONNECTIONS*. Cathy Wicks.
- Figueiredo, L. J., Gafaniz, A. R., Lopes, G. S., and Rúben, P. (2007). Aplicações de acelerómetros. *Instrumentação e Aquisição de Sinais*.
- Gothard, A. (2006). Model-based design - model misbehaviour. *Electronics Systems and Software*, 4(6):40–44.
- Grigorie, T. (2008). The matlab/simulink modeling and numerical simulation of an analogue capacitive micro-accelerometer. part 1: Open loop. In *Perspective Technologies and Methods in MEMS Design, 2008. MEMSTECH 2008. International Conference on*, pages 105–114.
- Honeywell (2012). 3-Axis Digital Compass IC HMC5883L. <http://www.honeywell.com>. Acessado em 12/fev/2014.
- Hou, H. (2004). Modeling inertial sensors errors using allan variance. Master's thesis, University of Calgary.
- IEEE (1998). Ieee standard specification format guide and test procedure for single-axis interferometric fiber optic gyros. *IEEE Std 952-1997*, pages i–.
- InvenSense (2012). 3-Axis MEMS Digital Gyrometer ITG-3200. <http://www.invensense.com>. Acessado em 12/fev/2012.
- ITS, N. A. T. (2007). *System Engineering for Intelligent Transportation Systems*. Department of Transportation, Department of Transportation, Office of Operations : HOTM Room 3404, 400 Seventh Street SW, Washington, DC 20590.
- Izham, Z. and Ward, M. C. (2004). Dynamic simulation of a resonant mems magnetometer in simulink®. *Sensors and Actuators A: Physical*, 115:392–400.
- Jensen, K. F. and Vinther, K. (2010). Attitude determination and control system for aausat3. Master's thesis, Aalborg University - Department of Electronic Systems.
- Jung, D. and Tsiotras, P. (2007). Modelling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. In *Aerial Vehicle, AIAA Infotech at Aerospace, Rohnert Park, CA, May 2007, AIAA*, pages 07–2763.

- Kraft, M. (1997). *Closed-loop accelerometer employing oversampling*. PhD thesis, Coventry University, UK.
- Lima, R. and Torres, L. A. B. (2012). Performance evaluation of attitude estimation algorithms in the design of an ahrs for fixed wing uavs. In *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, pages 255–260.
- Lima, R. R. (2013). Desenvolvimento de uma cabeça sensora para veículos aéreos não-tripulados. Master's thesis, Universidade Federal de Minas Gerais.
- Maclay, D. (1997). Simulation gets into the loop. *IEE Review*, 43(3):109–112.
- Mahony, R. and Hamel, T. (2008). Nonlinear Complementary Filters on the Special Orthogonal Group. *Automatic Control, IEEE*, 53(5):1203–1218.
- MathWorks, I. (2003). *Embedded Target for the TI TMS320C2000™ DSP Platform For Use with Simulink®*. 3 Apple Hill Drive Natick, MA 01760-2098.
- MathWorks, I. (2011). Matlab and simulink in the world - model-based design. <http://www.mathworks.com/company/newsletters/articles/matlab-and-simulink-in-the-world-model-based-design.html>. Acessado em 6/fev/2014.
- MathWorks, I. (2014). Matlab and simulink in the world - model-based design. <http://www.mathworks.com/company/newsletters/articles/matlab-and-simulink-in-the-world-model-based-design.html>. Acessado em 6/fev/2014.
- Motta, D. d., Polati, A. V., and Kawakami, R. (2009). Model-based-design of an aircraft auto-pilot controller. *CTA-DLR Workshop on Data Analysis & Flight Control*.
- Mueller, E. R. (2007). Hardware-in-the-loop simulation design for evaluation of unmanned aerial vehicle control system. *AIAA Modeling and Simulation Technologies Conference and Exhibit*.
- Murphy, B., Wakefield, A., and Friedman, J. (2008). Best practices for verification, validation, and test in modelbased design. *The MathWorks, Inc.* Acesso em 20/fev/2014.
- Ozyagcilar, T. (2011). Accuracy of angle estimation in ecompass and 3d pointer applications. Technical Report AN4249, Freescale Semiconductor, Inc.
- Ozyagcilar, T. (2012). Implementing a tilt-compensated ecompass using accelerometer and magnetometer sensors. Technical Report AN4248, Freescale Semiconductor, Inc.
- Pedley, M. (2013). Tilt sensing using a three-axis accelerometer. Technical Report AN3461, Freescale Semiconductor, Inc.

- Priya, P. and Sushas, S. (2012). Design and analysis of rotation rate sensor: Modeling of vibratory mems gyroscope. *International Journal of Applied Research in Mechanical Engineering (IJARME)*.
- Ribeiro, G. S., Lima, R. R., Pereira, G. A. S., Tôrres, L. A. B., and Pimenta, L. C. d. A. (2013). Simulação hardware-in-the-loop aplicada à avaliação de algoritmos de estimação de atitude para vants. *XI Simpósio Brasileiro de Automação Inteligente (SBAI 2013)*.
- Robotis (2014). Ax-18a. <http://support.robotis.com/en/>. Acesso em 31/jan/2014.
- Rogers, R. M. (2003). *Applied mathematics in integrated navigation systems*. American Institute of Aeronautics and Astronautics, Reston, Virginia, 2nd edition.
- Sandmann, G. and Thompson, R. (2008). Development of autosar software components within model-based design. *The MathWorks, Inc.* Acesso em 20/fev/2014.
- Sanvido, M. A. A. (2002). *Hardware-in-the-loop Simulation Framework*. PhD thesis, Swiss Federal Institute of Technology (ETH).
- Sherry, L., Brown, C., Motazed, B., and Vos, D. (2003). Performance of automotive-grade mems sensors in low cost ahrs for general aviation. In *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd*, volume 2, pages 12.C.2–12.1–5 vol.2.
- Shkel, A., Acar, C., and Painter, C. (2005). Two types of micromachined vibratory gyroscopes. In *Sensors, 2005 IEEE*, pages 6 pp.–.
- Smith, P. F., Prabhu, S. M., and Friedman, J. (2007). Best practices for establishing a model-based design culture. *The MathWorks, Inc.* Acesso em 20/fev/2014.
- Stevens, B. L. and Lewis, F. L. (2003). *Aircraft Control and Simulation*. Wiley-Interscience.
- Tehrani, M. M. (1983). Ring laser gyro data analysis with cluster sampling technique. *Proc. SPIE*, 0412:207–220.
- Texas, I. (2010). C28x iqmath library: A virtual floating point engine. www.ti.com. Acesso em 01/dez/2010.

Comunicação I2C: Múltiplos Endereços de escravo

A Seção 3.4.2 do Capítulo 3 apresenta explicações básicas sobre o funcionamento do protocolo de comunicação I2C. Cada dispositivo conectado ao barramento I2C deve possuir um endereço único e é reconhecido no barramento por esse endereço. Os dispositivos podem ser ainda transmissores ou receptores, mestres ou escravos, dependendo de sua função e da forma como foram configurados na interface de comunicação. O dispositivo mestre inicia e finaliza a comunicação e ainda controla o sinal de clock que permite a transferência de dados. Durante essa transferência, qualquer dispositivo endereçado pelo mestre é considerado como escravo.

Para a comunicação de dados, o módulo I2C tem um pino de dados (SDA) e um pino de clock (SCL). Estes dois pinos transferem informação entre o dispositivo 28x e outros dispositivos ligados ao barramento I2C. Ambos os pinos SDA e SCL são bidirecionais. Cada um deles tem de ser ligado a uma tensão de alimentação positiva através de resistores de pull-up. Quando o barramento está livre, ambos os pinos estão em nível lógico alto.

A Figura A.1 mostra os quatro registros utilizados para a transmissão e recepção no DSP 28x em modo padrão. A CPU escreve dados para transmissão ao I2CDXR e lê os dados recebidos de I2CDRR. Quando o módulo I2C é configurado como um transmissor, os dados gravados para I2CDXR são copiados para I2CXSR e deslocados bit a bit para fora no pino SDA. Quando o módulo I2C é configurado como um receptor, os dados recebidos são deslocado para I2CRSR e depois copiados para I2CDRR.

Conforme mencionado na Seção 4.3.5 do Capítulo 4 foi necessária uma adaptação no código de comunicação I2C do HIL para atender à condição de múltiplos endereços do dispositivo escravo, uma vez que o mesmo deve emular três sensores com endereços diferentes. A solução implementada foi a modificação sequencial do endereço escravo do DSP de acordo com o sensor que seria requisitado pela Cabeça Sensora. Basicamente, o DSP do HIL é configurado inicialmente com o endereço de escravo do primeiro sensor a ser requisitado pela Cabeça Sensora e em seguida seu endereço de escravo é modificado para o do segundo sensor e assim sucessivamente. A Figura A.2 apresenta o código para a modificação do endereço de escravo no DSP 28x implementado no HIL.

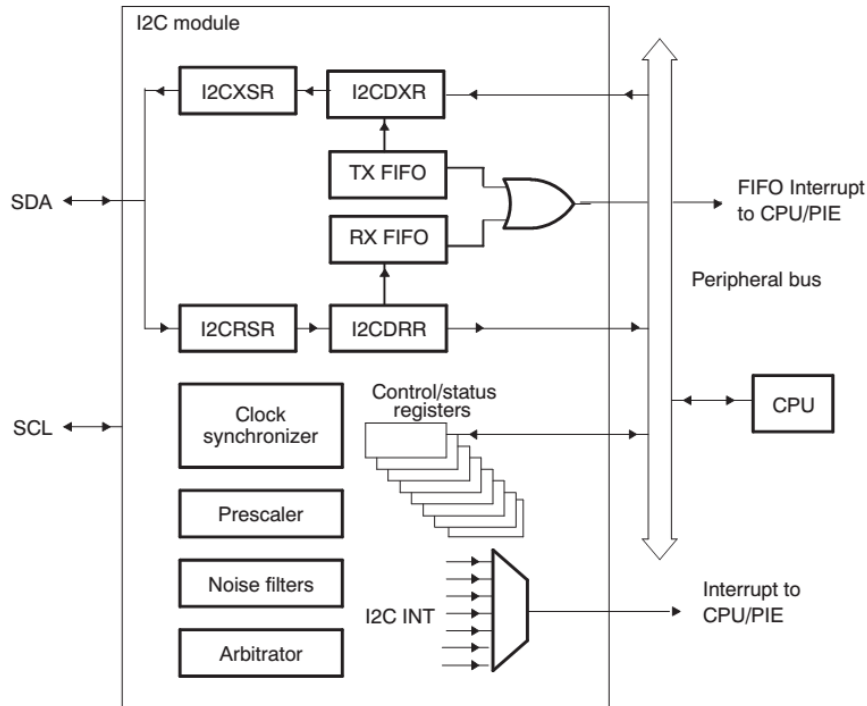


Figura A.1: Periférico de comunicação I2C do DSP 28x.

```

32
33 void I2C_SLAVE_CONFIG(uint8_t sensorAddress)
34 //recebe endereço do sensor e configura DSP com esse endereço
35 I2caRegs.I2CMDR.bit.IRS = 0; // Reseta o módulo I2C
36 I2caRegs.I2CMDR.bit.MST = 0;
37 I2caRegs.I2COAR = sensorAddress; // configura endereço de escravo atual do DSP
38 I2caRegs.I2CMDR.bit.IRS = 1; // Tira módulo I2C da condição de reset
39 }
40 void I2C_Read( I2C_RawSensorData* i2c_data )
41 {
42     I2C_SLAVE_CONFIG(ITG_SLAVE_ADDRESS7); // config DSP com endereço do girômetro
43     getITGData(); // fornece dados do girômetro
44     I2C_SLAVE_CONFIG(ADXL_SLAVE_ADDRESS7); // config DSP com endereço do acelerômetro
45     getADXLData( ); // fornece dados do acelerômetro
46     I2C_SLAVE_CONFIG(MAG_SLAVE_ADDRESS7); // config DSP com endereço do magnetômetro
47     getHMCDData( ); // fornece dados do magnetômetro
48 }

```

Figura A.2: Funções para modificação do endereço I2C do DSP 28x utilizado no HIL.

Uma outra peculiaridade dessa transmissão é que o DSP do HIL sempre atua como um dispositivo escravo e, no recebimento de uma instrução de leitura da Cabeça Sensora, ele é um escravo transmissor. Sendo assim, conforme especificações do manual SPRU603B, o bit BCM do registrador I2CEMDR da CPU do DSP deve ser configurado com o valor “0” para garantir um atraso na temporização das flags de status (XRDY e XSMT) no registro I2CSTR do DSP e proporcionar a coordenação entre os dados do registrador I2CDXR e I2CXSR conforme ilustra a Figura A.3.

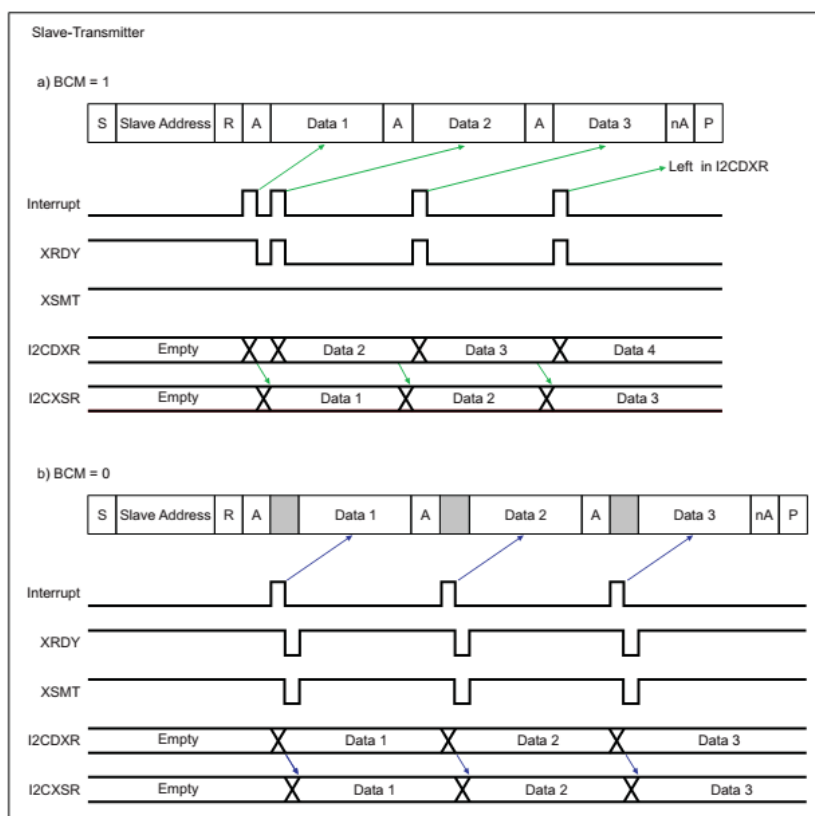


Figura A.3: Temporização das flags e registradores I2C na configuração “Escravo-Transmissor”.

O problema é que, ao contrário do que aparece no manual SPRU603B, o registrador I2CEMDR não está implementado no TMS320F28335, assim não se tem acesso ao bit BCM, impossibilitando a implementação dessa solução. O fato desse registrador não ter sido implementado no 28335 pode ser comprovado no manual do DSP (sob nº SPRS438M).

A solução implementada então foi a utilização de outras flags do registro de status I2CSTR para causar um efeito similar ao bit BCM. A flag utilizada foi a SDIR que indica a direção de comunicação do escravo, se recepção ou transmissão. Fez-se com que a CPU do DSP aguardasse enquanto não havia sido assinalada como escravo-transmissor. Dessa forma, foi provocado um atraso do envio dos primeiros 8 bits de

dados de transmissão. Com essa estratégia conseguiu-se com que todos os conjuntos de bits fossem transmitidos corretamente. A Figura A.4 apresenta o fragmento de código utilizado para implementar essa solução.

```
82 *****/
83 uint8_t i2cBatchRead( uint8_t device_register_address, uint8_t* rxBuffer, uint8_t bytesToRead )
84 {
85     int i =1;
86     uint8_t returnval;
87     while (I2caRegs.I2CSTR.bit.AAS != 1); // espera ser endereçado como escravo
88     I2caRegs.I2CCNT = 1; // recebe um byte
89     while(I2caRegs.I2CSTR.bit.RRDY == 0); // aguarda os dados estarem prontos
90     I2caRegs.I2CMDR.bit.TRX = 1;
91     I2caRegs.I2CCNT = bytesToRead;
92     I2caRegs.I2CSTR.bit.SCD = 1; // reestabelece condição de parada
93     while(I2caRegs.I2CSTR.bit.SDIR == 0); // aguarda ser configurado como escravo transmissor
94     while(i) // loop para estabelecer envio dos dados
95     {
96         I2caRegs.I2CDXR = *rxBuffer++;
97         while(I2caRegs.I2CSTR.bit.XRDY==0)
98         {
99             if(I2caRegs.I2CSTR.bit.SCD==1)
100             {
101                 i=0;
102                 break;
103             }
104         }
105     }
106     I2caRegs.I2CMDR.bit.IRS = 1;
107     returnval = I2caRegs.I2CSTR.bit.NACK; // 0: ACK, 1: NACK
108     return returnval;
109 }
```

Figura A.4: Solução com a utilização da flag SDIR.

Equações do Girômetro

Conforme apresentado no Capítulo 3 a estrutura do emulador HIL é composta por uma Unidade *Host* que envia dados de atitude e acelerações lineares ideais e uma Unidade DSP que processa essas informações e emula os dados dos sensores da IMU. A estratégia de se enviar apenas os ângulos de atitude (ϕ , θ e ψ) e as acelerações lineares (a_x , a_y e a_z) foi utilizada para diminuir a quantidade de dados enviados pela Unidade *Host* através da comunicação SCI com o intuito de minimizar os problemas inerentes a essa comunicação, conforme explicado na Seção 3.4. Sendo assim todos os modelos matemáticos da Unidade DSP devem receber essas informações como entradas e processá-las para poder emular em suas saídas os dados reais de seus respectivos sensores.

A modelagem do Girômetro apresentada na Seção 4.3.2 considerou a derivada discreta dos ângulos de atitude (ϕ , θ e ψ) como sendo as velocidades angulares ideais lidas pelo girômetro, contudo essa solução não é geral, pois não contempla as transformações de coordenadas que ocorrem no sistema devido às rotações executadas.

A expressão abaixo determina a relação correta entre as variações angulares dos ângulos de atitude com as velocidades angulares medidas pelo girômetro.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_x(\phi)\mathbf{R}_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix},$$

sendo p , q e r as velocidades angulares no referencial do girômetro em relação aos eixos x , y e z respectivamente e as matrizes $\mathbf{R}_x(\phi)$ e $\mathbf{R}_y(\theta)$ equivalem às matrizes de rotação

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}$$

e

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}.$$

Estendendo-se essa expressão pode-se chegar às equações individuais para cada velocidade angular:

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi} \sin \theta, \\ q &= \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta, \\ r &= \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi. \end{aligned}$$

de onde pode-se concluir que o erro na modelagem do girômetro não comprometeu os resultados obtidos nessa dissertação, uma vez que os testes realizados atendem a um caso especial da equação geral onde cada ângulo é girado individualmente e partindo da posição de referência zero com todos os eixos de giro alinhados com esse eixo de referência. Então pode-se dizer que a velocidade p é a derivada do ângulo de rolagem ($p = \dot{\phi}$) caso $\theta = 0$ pois a componente $\sin \theta$ se anula na expressão. O mesmo se conclui no caso da velocidade $q = \dot{\theta}$ para $\phi = 0$ e $r = \dot{\psi}$ para $\phi = 0$ e $\theta = 0$.

A Figura B.1 mostra os gráficos para o girômetro real (parte superior da figura) e para o girômetro emulado (parte inferior da figura) para o mesmo conjunto de dados do experimento ilustrado pela Figura 5.6. Levando em conta as considerações realizadas na Seção 5.3 para interpolação realizada na geração dos gráficos de comparação das atitudes, pode-se dizer mais uma vez que os dados emulados são coerentes.

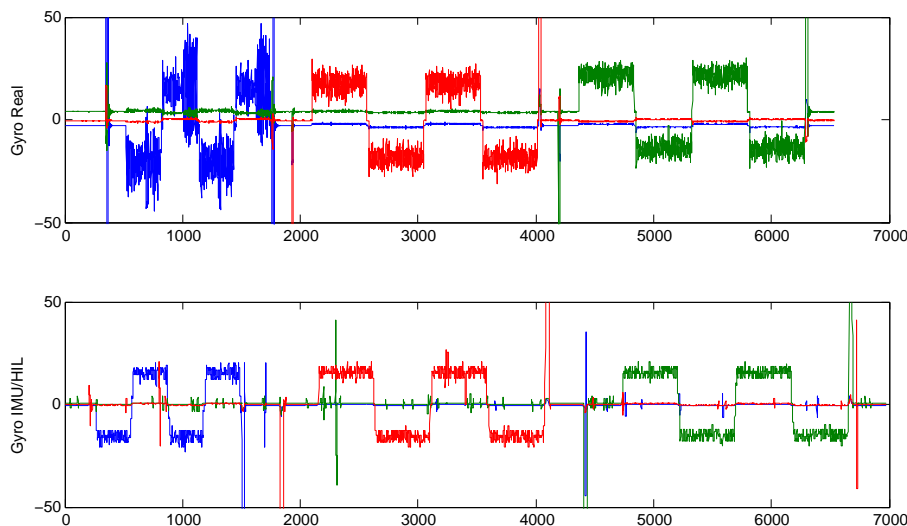


Figura B.1: Comparação entre Girômetro real e Girômetro emulado