



Universidade Federal
do Rio de Janeiro

Escola Politécnica

ESTABILIZAÇÃO INERCIAL DE CÂMERA PAN-TILT DE UM ROV

Victor Frangipani de Oliveira Lima

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Liu Hsu

Rio de Janeiro

Agosto de 2013

Frangipani de Oliveira Lima, Victor

Estabilização Inercial de Câmera Pan-Tilt de um ROV/Victor Frangipani de Oliveira Lima. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2013.

I, 74 p.: il.; 29,7cm.

Orientador: Liu Hsu

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia de Controle e Automação, 2013.

Referências Bibliográficas: p. 60 – 61.

1. Plataformas Inercialmente Estabilizadas. 2. Estabilização de Linha de Visada. 3. Sensor Inercial. 4. Câmera Pan-Tilt. I. Hsu, Liu. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação. III. Título.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Automação.

Estabilização Inercial de Câmera Pan-Tilt de um ROV

Victor Frangipani de Oliveira Lima

Agosto/2013

Orientador: Liu Hsu

Curso: Engenharia de Controle e Automação

O robô submarino de operação remota (ROV) LUMA (*Light Underwater Mobile Asset*) realiza pesquisas da fauna e da flora marinha na Baía do Almirantado predominantemente por meio de sistemas de câmeras. A aquisição de imagens de boa qualidade é degradada pelo movimento do veículo, que também pode causar a perda do rastreamento de um ponto de interesse. Um método de compensação para este tipo de problema é por meio da aplicação de uma plataforma inercialmente estabilizada, permitindo o controle da direção da linha de visada de uma câmera, de forma a isolar de movimentos do veículo. Neste trabalho duas estratégias de controle são discutidas por meio do uso de um mecanismo com dois graus de liberdade (estrutura Pan-Tilt) e um sensor de atitude e rumo (AHRS). Em uma dessas estratégias não é necessário o uso de sensores de posição para os eixos dos motores.

Palavras-chave: Plataformas Inercialmente Estabilizadas, Estabilização de Linha de Visada, Câmera Pan-Tilt, Sensor Inercial.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

INERTIAL STABILIZATION OF AN ROV PAN-TILT CAMERA

Victor Frangipani de Oliveira Lima

August/2013

Advisor: Liu Hsu

Course: Automation and Control Engineering

The remotely operated submarine robot (ROV) LUMA (Light Underwater Mobile Asset) research the marine flora and fauna in Admiralty Bay primarily by the use of camera systems. The acquisition of good quality images is degraded by the movement of the vehicle, which can also cause loss of tracking a point of interest. A method of compensation for this kind of problem is through the use of an inertially stabilized platform, allowing to control the direction of the line of sight of the camera, isolating vehicle motion. In this work two control strategies are discussed by using a mechanism with two degrees of freedom (Pan-Tilt) and an attitude heading reference system (AHRS). In one of these strategies it's not necessary to use a motor shaft position sensor.

Keywords: Inertially Stabilized Platforms, Line of Sight Stabilization, Pan-Tilt Camera, Inertial Sensor.

ESTABILIZAÇÃO INERCIAL DE CÂMERA PAN-TILT DE UM ROV

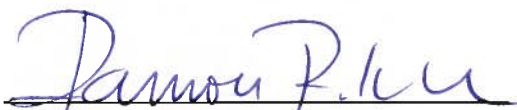
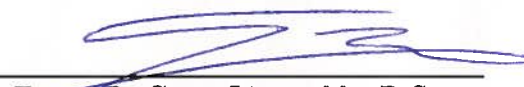
VICTOR FRANGIPANI DE OLIVEIRA LIMA

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA POLITÉCNICA/-COPPE/ESCOLA DE QUÍMICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE CONTROLE E AUTOMAÇÃO.

Aprovada por:



Liu Hsu , Dr. d'État
(Orientador)


Ramon Romankevicius Costa, D.Sc.
Fernando Cesar Lizarralde, D.Sc.

Rio de Janeiro, RJ - Brasil

Agosto de 2013

Sumário

| | |
|---|-------------|
| Lista de Figuras | viii |
| Lista de Tabelas | xi |
| Lista de Abreviaturas | xii |
| 1 Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Organização do Trabalho | 3 |
| 2 Sistemas de Estabilização Inercial | 4 |
| 3 Descrição do Problema e Metodologia Proposta | 7 |
| 3.1 Descrição do Problema | 7 |
| 3.1.1 Sensor inercial fixado no veículo | 8 |
| 3.1.2 Sensor inercial fixado na ISP | 11 |
| 3.1.3 ISP com dois graus de liberdade | 11 |
| 3.2 Metodologia Proposta | 16 |
| 3.2.1 Sensor inercial | 16 |
| 3.2.2 Motor DC | 19 |
| 3.2.3 Projeto de um controlador PI | 23 |
| 4 Desenvolvimento do Sistema Experimental | 26 |
| 4.1 Comunicação com o Sensor Inercial | 27 |

| | | |
|----------|---|-----------|
| 4.1.1 | Comunicação serial | 29 |
| 4.1.2 | Comunicação por memória compartilhada | 32 |
| 4.1.3 | Parâmetros de configuração do sensor inercial | 35 |
| 4.2 | Controle do Motor DC pela placa Q8 | 37 |
| 4.2.1 | Por meio de um canal analógico | 38 |
| 4.2.2 | Por meio de um sinal PWM | 41 |
| 4.2.3 | Medida da orientação dos motores | 45 |
| 5 | Resultados | 48 |
| 5.1 | Resultados com base em dados do sensor inercial | 48 |
| 5.2 | Resultados com base em imagens da câmera | 53 |
| 6 | Conclusão e Trabalhos Futuros | 58 |
| | Referências Bibliográficas | 60 |
| A | Folhas de Dados | 62 |
| A.1 | Motor DC: Faulhaber 2342S024CR | 62 |
| A.2 | Encoder: Avago HEDM-5500J14 | 64 |
| A.3 | Sensor Inercial: MicroStrain GX3-25-OEM | 66 |
| B | Códigos Fonte | 69 |
| B.1 | Comunicação entre o Sensor 3DM-GX3-25 e o QUARC | 69 |
| B.2 | Rastreamento de Alvo por Cor | 73 |

Lista de Figuras

| | | |
|------|---|----|
| 1.1 | ROV LUMA | 1 |
| 2.1 | a) Câmera do VANT <i>ScanEagle</i> [14] b) Telescópio espacial <i>Hubble</i> [6] c) Câmera televisiva inercialmente estabilizada[15] | 5 |
| 2.2 | ISP com um grau de liberdade | 5 |
| 3.1 | Ilustração do sistema de um grau de liberdade com o sensor inercial fixado na base do veículo | 9 |
| 3.2 | Diagrama de blocos do sistema com o sensor inercial fixado no veículo . . . | 9 |
| 3.3 | Diagrama de blocos do sistema simplificado com o sensor inercial fixado no veículo | 10 |
| 3.4 | Ilustração do sistema de um grau de liberdade com o sensor inercial fixado na ISP | 10 |
| 3.5 | Diagrama de blocos do sistema com o sensor inercial fixado na ISP | 12 |
| 3.6 | Diagrama de blocos do sistema simplificado com o sensor inercial fixado na ISP | 12 |
| 3.7 | Representação dos seis graus de liberdade do veículo [5] | 12 |
| 3.8 | Modelo de uma ISP com dois graus de liberdade em uma estrutura Pan/Tilt | 14 |
| 3.9 | Sistemas de coordenadas utilizado para a ISP com dois graus de liberdade | 14 |
| 3.10 | Sensor <i>MicroStrain 3DM-GX3-25</i> | 16 |
| 3.11 | Sistema de coordenadas NED representado no globo terrestre | 18 |
| 3.12 | Representação do Campo Magnético Terrestre | 19 |
| 3.13 | Motor DC - <i>Faulhaber 2342S024CR</i> sem a caixa de redução | 19 |

| | | |
|------|--|----|
| 3.14 | Circuito elétrico do Motor DC | 20 |
| 3.15 | Diagrama em blocos do Motor DC | 22 |
| 3.16 | Ganho dos sistemas 3.28 (a) e 3.27 (b) com $\xi = 1/\sqrt{2}$ e $\omega_n = 31,42\text{rad/s}$ | 25 |
| 4.1 | Foto do sistema experimental montado em uma base de madeira | 27 |
| 4.2 | Diagrama de blocos do sistema experimental com o sensor inercial fixado no veículo | 28 |
| 4.3 | Diagrama de blocos do sistema experimental com o sensor inercial fixado na ISP | 28 |
| 4.4 | Diagrama da comunicação do sensor <i>3DM-GX3-25</i> com o <i>QUARC</i> | 29 |
| 4.5 | Fluxograma do algoritmo de comunicação do sensor <i>3DM-GX3-25</i> com o <i>QUARC</i> | 30 |
| 4.6 | Configuração do <i>Visual Studio</i> para a comunicação com o <i>QUARC</i> | 33 |
| 4.7 | Configuração do <i>Visual Studio</i> para a comunicação com o <i>QUARC</i> | 34 |
| 4.8 | Configuração do <i>Visual Studio</i> para a comunicação com o <i>QUARC</i> | 34 |
| 4.9 | Configurações do bloco de comunicação do <i>Simulink/QUARC</i> | 36 |
| 4.10 | Bloco de servidor do <i>Simulink/QUARC</i> para a comunicação por memória compartilhada | 37 |
| 4.11 | Configuração do sensor inercial por meio do software <i>3DM-GX3 Monitor</i> | 37 |
| 4.12 | Calibração do sensor inercial por meio do software <i>3DM-GX3 Iron Calibration</i> | 38 |
| 4.13 | Placa de controle <i>Q8</i> | 39 |
| 4.14 | Universal Power Module 1503 | 39 |
| 4.15 | Bloco de saída analógica do <i>Simulink/Quarc</i> | 40 |
| 4.16 | Sinal modulado por largura de pulso (PWM) | 41 |
| 4.17 | Estrutura de uma Ponte H | 42 |
| 4.18 | Foto do circuito de Ponte H utilizado | 42 |
| 4.19 | Configuração uma saída de <i>clock</i> da placa <i>Q8</i> para um sinal PWM | 43 |
| 4.20 | Configuração do sinal de PWM da placa <i>Q8</i> | 44 |
| 4.21 | Diagrama de estados para a decodificação do encoder | 45 |
| 4.22 | Configuração do bloco de leitura de <i>encoders</i> da <i>QUARC Target</i> | 47 |

| | | |
|-----|---|----|
| 5.1 | Leitura do sensor no canal de Caturro (<i>pitch</i>) e sinal de controle enviado para o motor de <i>Tilt</i> no primeiro experimento, sem movimento na plataforma | 49 |
| 5.2 | Leitura do sensor no canal de Guinada (<i>yaw</i>) e sinal de controle enviado para o motor de <i>Pan</i> no primeiro experimento, sem movimento na plataforma | 50 |
| 5.3 | Leitura do sensor no canal de Caturro (<i>pitch</i>) e sinal de controle enviado para o motor de <i>Tilt</i> no segundo experimento, sem movimento da plataforma | 51 |
| 5.4 | Leitura do sensor no canal de Guinada (<i>yaw</i>) e sinal de controle enviado para o motor de <i>Pan</i> no segundo experimento, sem movimento da plataforma | 52 |
| 5.5 | Leitura do sensor no canal de Caturro (<i>pitch</i>) e sinal de controle enviado para o motor de <i>Tilt</i> no terceiro experimento, com movimento da plataforma | 53 |
| 5.6 | Leitura do sensor no canal de Guinada (<i>yaw</i>) e sinal de controle enviado para o motor de <i>Pan</i> no terceiro experimento, com movimento da plataforma | 54 |
| 5.7 | Processamento da imagem para o rastreamento de um alvo: a) Imagem original b) Extração do Canal Vermelho c) Filtragem mediana d) Conversão em imagem binária | 55 |
| 5.8 | a) Rastreamento visual de um alvo com estabilização da LOS b) sem estabilização | 57 |

Lista de Tabelas

| | | |
|-----|---|----|
| 1.1 | Lista de dispositivos Pan/Tilt marinizados | 2 |
| 4.1 | Estrutura da resposta do sensor <i>3DM-GX3-25</i> para uma requisição dos ângulos de Balanço (<i>roll</i>), Caturro (<i>pitch</i>) e Guinada (<i>yaw</i>) | 32 |

Lista de Abreviaturas

| | |
|------|--|
| AHRS | Sistema de Referenciamento de Atitude e Rumo (<i>Attitude and Heading Reference System</i>), p. 16 |
| DC | Corrente Contínua (<i>Direct Current</i>), p. 3 |
| IMU | Sistema de Medição Inercial (<i>Inertial Measurement Unit</i>), p. 16 |
| ISP | Plataforma Inercialmente Estabilizada (<i>Inertially Stabilized Platform</i>), p. 2 |
| LOS | Linha de Visada (<i>Line of Sight</i>), p. 2 |
| MEMS | Sistema Microeletromecânico (<i>Microelectromechanical Systems</i>), p. 16 |
| NED | Sistema de coordenadas <i>North-East-Down</i> , p. 16 |
| PWM | Modulação por largura de pulso (<i>Pulse Width Modulation</i>), p. 25 |
| ROV | Veículo de Operação Remota (<i>Remotely Operated Vehicle</i>), p. 1 |

Capítulo 1

Introdução

1.1 Motivação

Este trabalho surgiu como parte do desenvolvimento do robô submarino de operação remota (ROV, *Remotely Operated Vehicle*) LUMA (*Light Underwater Mobile Asset*), figura 1.1, desenvolvido pelo Laboratório de Controle do PEE/COPPE. Seu principal objetivo é o estudo da fauna e da flora da Baía do Almirantado, na Antártica, por meio de coleta de imagens, vídeos de alta definição e amostras ambientais (organismos, rochas e sedimentos).

A filmagem de vídeos de alta qualidade é comprometida pelo movimento do veículo, causado tanto pela operação remota por um piloto do ROV, quanto por distúrbios do ambiente (correntes oceânicas e ondas, por exemplo). Um determinado alvo ou uma



Figura 1.1: ROV LUMA

Tabela 1.1: Lista de dispositivos Pan/Tilt marinizados

| Modelo | Velocidade Angular Máx. [graus/s] | Profundidade Máxima [m] |
|-----------------------|-----------------------------------|-------------------------|
| Sidus SS252 | 35 | 6000 |
| Kongsberg OE10-102 | 30 | 6000 |
| Kongsberg OE10-100 | 25 | 6000 |
| Rosys PT-10-FB RS-485 | 20 | 3000 |
| Sidus SS109HT | 20 | 6000 |
| Novasub P&T-C-01 | 15 | 4000 |

região de interesse pode sair do campo de visão da câmera, caso a orientação desta seja muito alterada pela movimentação do ROV. Vibrações de pequena amplitude também degradam a qualidade da imagem captada. Em tais casos é preciso desenvolver um sistema que compense movimentos e distúrbios do veículo.

1.2 Objetivos

O objetivo geral deste trabalho é o desenvolvimento de uma plataforma inercialmente estabilizada (ISP, *Inertially Stabilized Platform*) com o objetivo de manter a orientação da linha de visada (LOS, *Line of Sight*) de uma câmera, isto é, do vetor determinado por seu centro de visão, constante em relação a um referencial inercial. Isto permite reduzir variações no campo de visão da câmera, de forma que tarefas de pesquisa e rastreamento de um determinado alvo ou região de interesse sejam facilitadas.

Os trabalhos [7] e [3] propõem e desenvolvem, como estratégia para estabilização do campo de visão de uma câmera, a aplicação de um mecanismo de dois graus de liberdade, na estrutura Pan/Tilt. Para o caso do ROV LUMA é preciso que este aparato seja marinizado. Posto isso, foi feita uma pesquisa com os dispositivos Pan/Tilt presentes no mercado que possam operar em condições subaquáticas. Um resultado parcial é apresentado na tabela 1.1, com os mecanismos de maior velocidade angular encontrados. A constatação da falta de sistemas que operem em maiores velocidades levam a necessidade de construir um mecanismo próprio para as condições do ROV LUMA.

Este trabalho procura cobrir os seguintes pontos:

- Descrever a tecnologia básica que compreende sistemas de estabilização inercial

- Apresentar duas estratégias para a estabilização inercial que podem ser empregadas
- Construir um mecanismo Pan/Tilt que possa ser utilizado como parte de uma plataforma inercialmente estabilizada
- Implementar a metodologia proposta e obter os resultados com base na imagem da câmera estabilizada

1.3 Organização do Trabalho

O trabalho possui cinco capítulos além deste. O próximo capítulo discute de forma geral a tecnologia envolvida em sistemas de estabilização inercial, apresenta exemplos de aplicações, as estruturas e os componentes básicos que são adotados, bem como a nomenclatura comumente utilizada.

O terceiro capítulo apresenta o princípio teórico que será aplicado neste trabalho. Isto envolve duas estratégias possíveis para a estabilização inercial, o princípio de funcionamento dos componentes utilizados, basicamente motores de corrente contínua (DC, *Direct Current*) e um sensor inercial, e propõe um tipo de controle a ser implementado.

O quarto capítulo descreve o processo de elaboração do sistema experimental e seus detalhes técnicos. É relatado como foi feita a integração entre todos os componentes em um sistema de controle, a comunicação com o sensor inercial, o sistema de alimentação dos motores e a leitura de sensores de posição angular dos motores.

O penúltimo apresenta os resultados obtidos, a partir de dados do sensor inercial e de uma câmera que foi estabilizada.

Por último uma conclusão é feita em cima do que foi desenvolvido, com uma análise de trabalhos futuros, o que pode estender e desenvolver o que foi realizado neste.

Capítulo 2

Sistemas de Estabilização Inercial

Plataformas inercialmente estabilizadas (ISP, *Inertially Stabilized Platform*) são aplicadas para apontar e estabilizar a linha de visada (LOS, *Line of Sight*) de diversos tipos de sensores, câmeras, telescópios, antenas de comunicação e armamentos. Uma ISP permite orientar uma carga com base em um sistema de coordenadas inercial, de forma a compensar movimentos da base em que esteja fixada. Na imagem 2.1 temos três exemplos de aplicações em sistemas de visão:

- a) Uma ISP de dois graus de liberdade (Pan/Tilt) aplicado para estabilizar a câmera do *ScanEagle*, veículo aéreo não tripulado (VANT) desenvolvido pela *Boeing* e pela *Insitu*. Este VANT foi criado com o objetivo inicial de rastrear cardumes, auxiliando barcos de pesca. Atualmente é bastante utilizado para mapear e monitorar regiões de acesso remoto em operações militares.
- b) O telescópio aeroespacial *Hubble*, lançado em 1990 para capturar imagens de altíssima resolução do espaço, também se trata de uma ISP. Neste caso a orientação do telescópio é controlada afim de estabilizar a linha de visada de sua câmera.
- c) A *Scorpio Stabilized Head* da *Servicevision* de uso comum em filmagem de eventos em estádios esportivos, consiste de uma ISP com 3 graus de liberdade, que por meio de cabos se desloca por cima do campo.

Embora não seja utilizada na prática, uma ISP com apenas um grau de liberdade pode servir para descrever o funcionamento básico, como está ilustrado na figura 2.2.

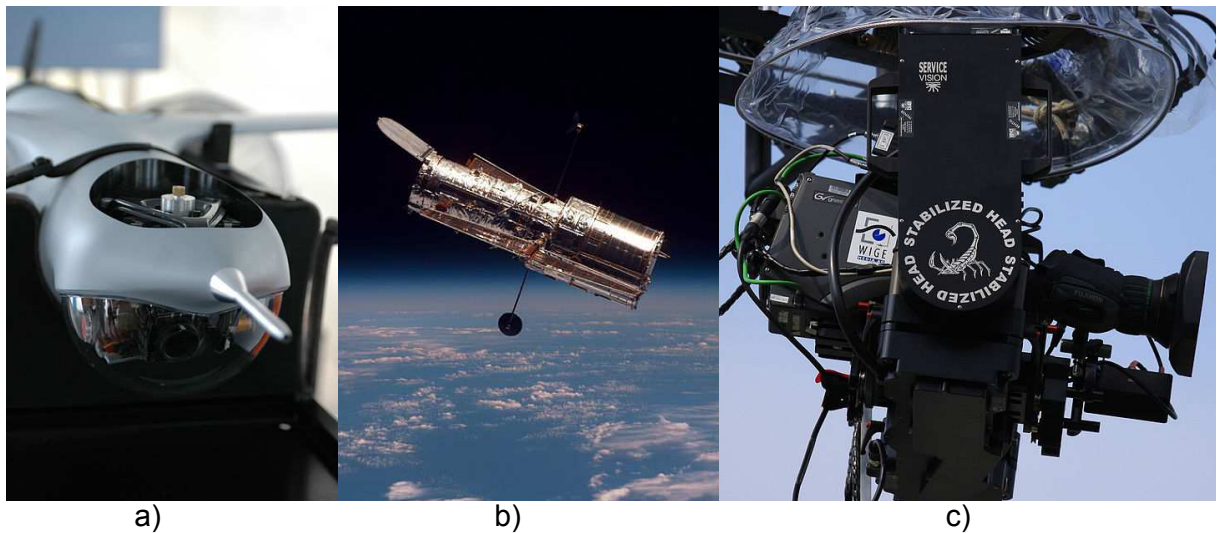


Figura 2.1: a) Câmera do VANT *ScanEagle*[14] b) Telescópio espacial *Hubble*[6] c) Câmera televisiva inercialmente estabilizada[15]

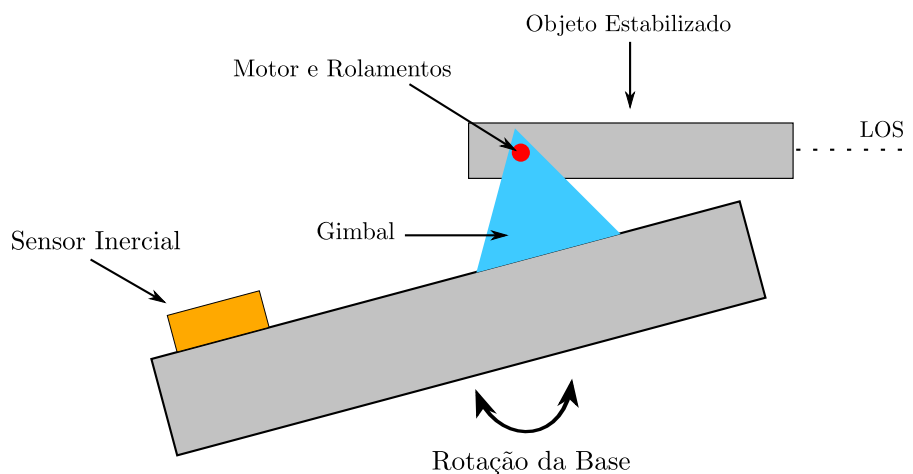


Figura 2.2: ISP com um grau de liberdade

Nesta configuração o sensor mede a rotação da base, esta é compensada por um motor, que aplica a rotação inversa em sua carga, de forma a mantê-la em uma direção constante.

ISPs envolvem basicamente o uso de uma estrutura eletromecânica chamada de gimbal e de um sistema de medição inercial. O termo gimbal é comumente aplicado para descrever um conjunto de anéis concêntricos, estruturas que são usadas em giroscópios mecânicos, por exemplo. Para ISPs a palavra tem um significado mais aberto e designa qualquer estrutura que permita que a LOS de uma carga seja orientada. Tipicamente são estruturas com dois ou mais graus de liberdade (DOF, *Degrees of Freedom*).

O sensor inercial permite que se meça a orientação do corpo com base em um sistema de coordenadas inercial. Geralmente giroscópios, sensores que medem a velocidade angular ao redor de um eixo em relação a um sistema de coordenadas inercial, são usados.[8] É também possível utilizar um sensor de atitude e rumo, que por meio das medidas de um conjunto de acelerômetros, magnetômetros e giroscópios, calcula a orientação inercial.

Neste trabalho o objetivo é estabilizar inercialmente a linha de visada (LOS, *Line of Sight*) de uma câmera, isto é, do vetor definido pela direção que a câmera aponta. Na particularidade de um sistema óptico a câmera pode ser colocada em duas posições diferentes. Na primeira, *Platform Stabilization*, a câmera é fixada diretamente em cima da plataforma. Na segunda, *Steering Stabilization*, elementos ópticos, como prismas e espelhos, são montados em uma estrutura similar a um periscópio, que direciona a luz para a câmera, fixada no veículo. Esta abordagem é mais complexa pois deve levar em conta fenômenos ópticos como a magnificação da imagem, por exemplo. Pode ser vantajosa se a câmera em questão for muito pesada ou tiver dimensões que impossibilitam sua livre orientação. O primeiro caso foi abordado neste trabalho.

Geralmente ISPs são empregadas como componentes de um sistema maior, com o objetivo de rastrear um alvo, por exemplo. O sistema de controle da ISP é uma malha rápida, para remover distúrbios de alta frequência e controlar a LOS, dentro de uma malha mais lenta, responsável pelo rastreamento do alvo. O controle da linha de visada de uma câmera envolve dois requisitos principais, *tracking* e *jitter*. O primeiro se refere ao problema de apontar a LOS em um determinado alvo ou em alguma região de interesse. O segundo diz da necessidade de manter a LOS estabilizada na orientação especificada. A precisão varia com a aplicação desejada. Câmeras portáteis ou antenas de comunicação requerem uma estabilização na ordem de miliradianos. Enquanto que telescópios aeroespaciais como o *Hubble*, requerem a estabilização na ordem de frações de micro-radianos.

Capítulo 3

Descrição do Problema e Metodologia Proposta

O problema considerado neste capítulo envolve a estabilização da orientação da linha de visada de uma câmera por meio de uma plataforma inercialmente estabilizada (ISP). Como gimbal um mecanismo de dois graus de liberdade (Pan/Tilt) é utilizado, com cada eixo acionado por um motor DC. O sensor inercial consiste de um sistema de referenciamento de atitude e rumo (AHRS), que permite medir a orientação em relação a um referencial inercial.

A alocação do sensor inercial pode ser feita basicamente de duas formas: fixado no corpo do veículo ou na plataforma que se visa estabilizar. Isto será tratado de forma simplificada, com apenas um grau de liberdade, nas seções 3.1.1 e 3.1.2. Em seguida é feita uma generalização para o mecanismo Pan/Tilt na seção 3.1.3.

A descrição do funcionamento dos componentes básicos do sistema, sensor inercial e motor DC, é feito nas seções 3.2.1 e 3.2.2. O projeto de controle é desenvolvido no final, na seção 3.2.3.

3.1 Descrição do Problema

Embora o comportamento de um sistema real seja mais complicado, a simplificação para um grau de liberdade ajuda a entender a natureza do problema. O sistema é representado na figura 3.1, com os seguintes sistemas de coordenadas:

L sistema de coordenadas inercial (*local level*)

b sistemas de coordenadas do veículo (*body*)

a sistema de coordenadas da câmera

E as seguintes variáveis do sistema são definidas:

ψ_{bL} elevação do veículo em relação ao referencial inercial

ψ_{ab} elevação da câmera em relação ao veículo

ψ_{aL} elevação da câmera em relação ao referencial inercial

Temos por objetivo estabilizar a direção da linha de visada da câmera, ψ_{aL} , em um valor de referência ψ_r :

$$\psi_{aL} = \psi_{ab} + \psi_{bL} = \psi_r \quad (3.1)$$

Duas estratégias são definidas, a primeira fixa o sensor inercial no corpo do veículo, de forma a medir a orientação deste em relação ao referencial inercial, que no caso simplificado é apenas o ângulo de elevação ψ_{bL} . O segundo método fixa o sensor inercial na ISP, ao lado da câmera, possibilitando a medida da orientação da linha de visada da câmera diretamente. Em ambas o gimbal, constituído de um motor DC, é usado para rotacionar a câmera.

3.1.1 Sensor inercial fixado no veículo

Nesta configuração o sensor inercial mede a orientação ψ_{bL} . O ângulo ψ_{ab} , controlado por um motor DC, é medido por meio um sensor formado por um conjunto de *encoder* e *decoder*. Para atingir o objetivo, conforme a equação 3.1, devemos controlar o sinal de saída $y(t) = \psi_{ab}(t)$ para seguir a referência $r(t) = \psi_r(t) - \psi_{bL}(t)$. Se trata de um problema de rastreamento. O sistema está representado em diagrama de blocos na figura 3.2.

Uma outra simplificação pode ser feita considerando sensores ideais, sem dinâmica e atrasos de medida, resultando no sistema da figura 3.3. Em que $C(s)$ e $P(s)$ são a função de transferência do Controlador e da Planta, respectivamente.

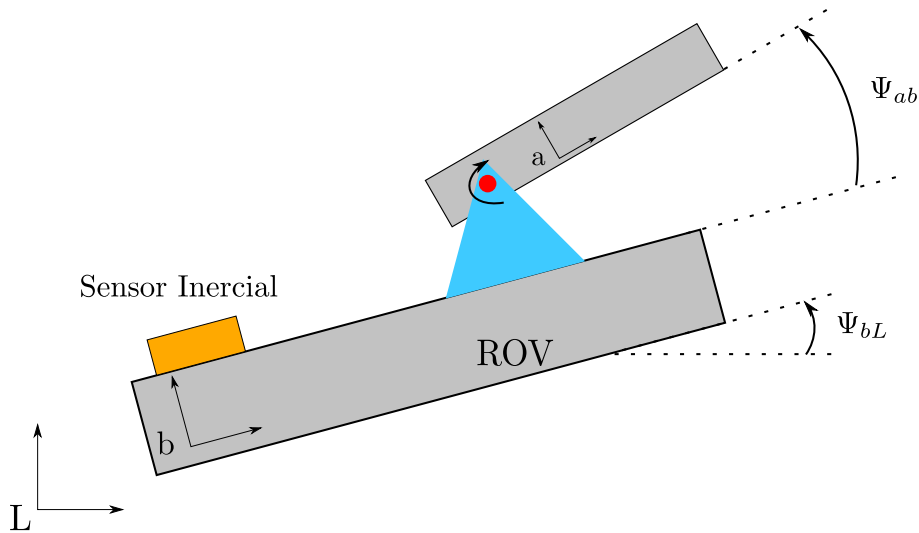


Figura 3.1: Ilustração do sistema de um grau de liberdade com o sensor inercial fixado na base do veículo

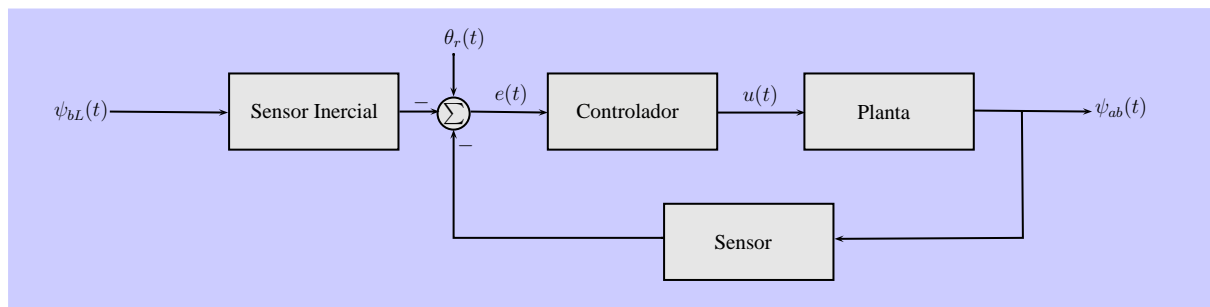


Figura 3.2: Diagrama de blocos do sistema com o sensor inercial fixado no veículo

A relação entre o sinal de referência, $R(s)$, e a saída, $Y(s)$, é dada por:

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \quad (3.2)$$

E a função de transferência entre o sinal de referência e o sinal de erro, $E(s) = R(s) - Y(s)$, é dada por:

$$\frac{E(s)}{R(s)} = \frac{1}{1 + C(s)P(s)} \quad (3.3)$$

Posto isso passamos para o segundo caso, que se mostrará análogo, de tal modo que as duas estratégias poderão ser tratadas de forma unificada.

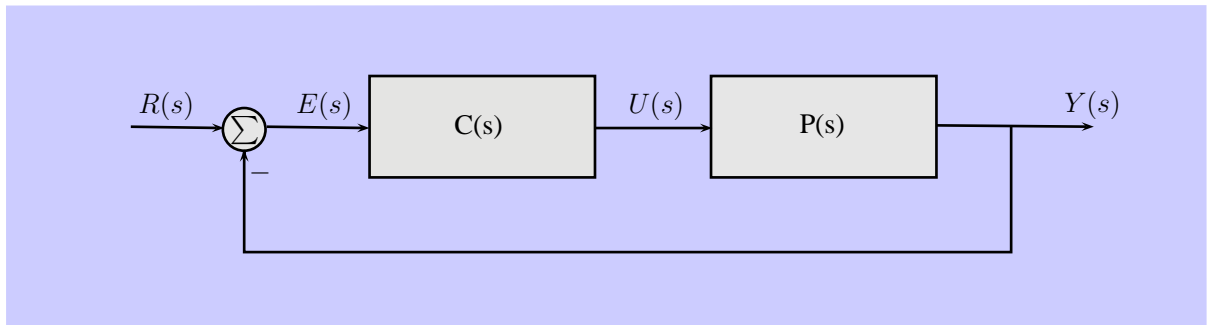


Figura 3.3: Diagrama de blocos do sistema simplificado com o sensor inercial fixado no veículo

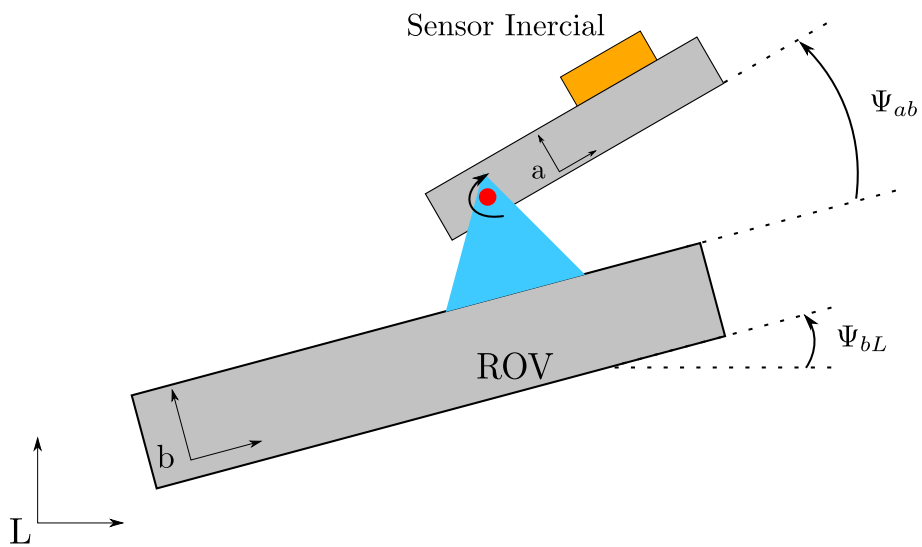


Figura 3.4: Ilustração do sistema de um grau de liberdade com o sensor inercial fixado na ISP

3.1.2 Sensor inercial fixado na ISP

Nessa estratégia o sensor é fixado na ISP, ao lado da câmera, de forma que agora mede diretamente o ângulo que se quer estabilizar, $\psi_{aL}(t)$. A planta considerada é a mesma do sistema anterior. Conforme é mostrado na figura 3.5, controlamos a variável $\psi_{ab}(t)$ por meio do gimbal e o movimento do veículo, $\psi_{bL}(t)$, é visto como uma perturbação na saída do sistema. Em tal configuração não se faz necessário o uso de Encoders/Decoders, uma vez que não precisamos medir o sinal $\psi_{ab}(t)$. Como o Sensor Inercial mede diretamente a direção da linha de visada da câmera, é possível a correção de distúrbios que atuem em cima da plataforma, que não seriam tratados pelo primeiro caso.

A saída, $y(t) = \psi_{aL}(t)$ deve seguir um sinal de referência $r(t) = \psi_r(t)$, rejeitando um distúrbio $w(t) = \psi_{bL}$. Se a referência for nula, se trata de um problema de regulação.

Novamente, para simplificar o estudo do problema, apenas a dinâmica da planta e do controlador são considerados, tal como se apresenta na figura 3.6. Temos então uma função de transferência relacionando o sinal de referência, $R(s)$, e de saída, $Y(s)$, igual ao caso anterior:

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \quad (3.4)$$

A função de transferência entre a saída e o distúrbio é dada por:

$$\frac{Y(s)}{W(s)} = \frac{1}{1 + C(s)P(s)} \quad (3.5)$$

Esta é igual a função de transferência do caso anterior, definida por 3.3, em que é manter o sinal de erro pequeno, independente de variações do sinal de referência causados pela rotação do veículo. Tais variações agora são vistas como um distúrbio, que precisa ser rejeitado na saída. A comparação implica que o mesmo controlador projetado para um caso vale para o outro, isto considerando as idealizações feitas.

3.1.3 ISP com dois graus de liberdade

Para ser aplicável no ROV LUMA, a ISP deve ser projetado considerando os seis graus de liberdade do ROV. A notação náutica é utilizada para nomear os movimentos

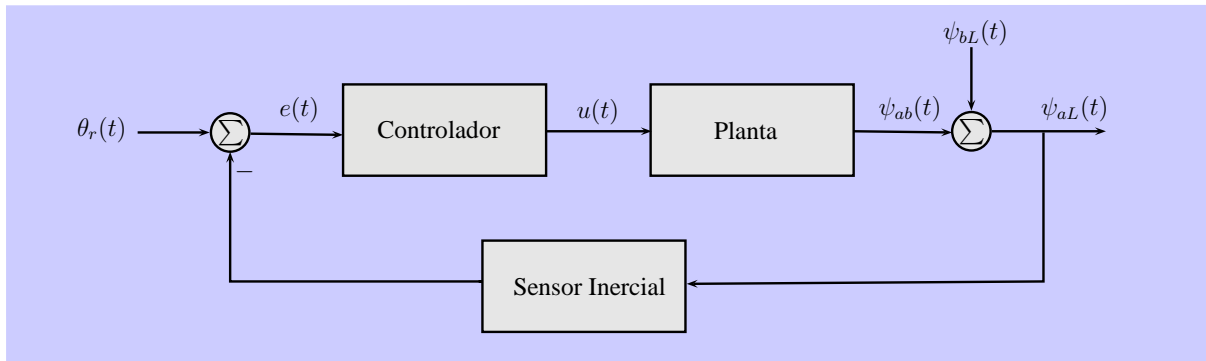


Figura 3.5: Diagrama de blocos do sistema com o sensor inercial fixado na ISP

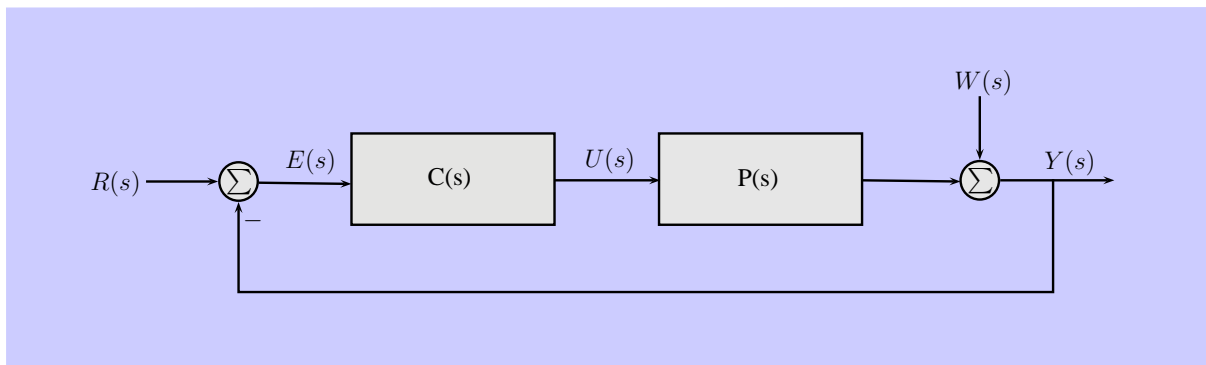


Figura 3.6: Diagrama de blocos do sistema simplificado com o sensor inercial fixado na ISP

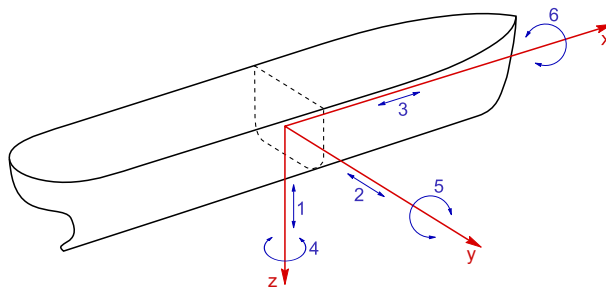


Figura 3.7: Representação dos seis graus de liberdade do veículo [5]

do veículo, de acordo com a figura 3.7:

- 1 Arfagem (translação paralela ao eixo z, *Heave*)
- 2 Deriva (translação paralela ao eixo y, *Sway*)
- 3 Avanço (translação paralela ao eixo x, *Surge*)
- 4 Guinada (rotação em torno do eixo z, *Yaw*)
- 5 Caturro (rotação em torno do eixo y, *Pitch*)
- 6 Balanço (rotação em torno do eixo x, *Roll*)

Um mecanismo com apenas um grau de liberdade não é usado na prática. Para ser aplicável, o gimbal deve possuir dois ou mais graus de liberdade. Uma estrutura do tipo Pan/Tilt é adotada, Fig. 3.8, em que um motor DC no eixo externo compensa rotações na guinada do veículo e outro motor DC, no eixo interno, compensa rotações no caturro.

Quando o sensor inercial é fixado na plataforma, a orientação da linha de visada da câmera é diretamente medida. Para o caso em que o sensor mede a orientação do veículo, como foi visto em [3] e [7], o problema consiste em calcular os ângulos de conteira (γ) e elevação (α), indicados na figura 3.8, que permitam orientar a LOS na direção desejada.

Os sistemas de coordenadas da figura 3.9 são definidos:

L: sistema de coordenadas inercial (ou *local level*)

b: sistema de coordenadas do veículo

B: sistema de coordenadas do eixo exterior do gimbal

a: sistema de coordenadas da extremidade, onde se localiza a câmera

R_{ij} : $R_{ij} \in SO(3)$ é a orientação do sistema de coordenadas i com respeito ao sistema de coordenadas j

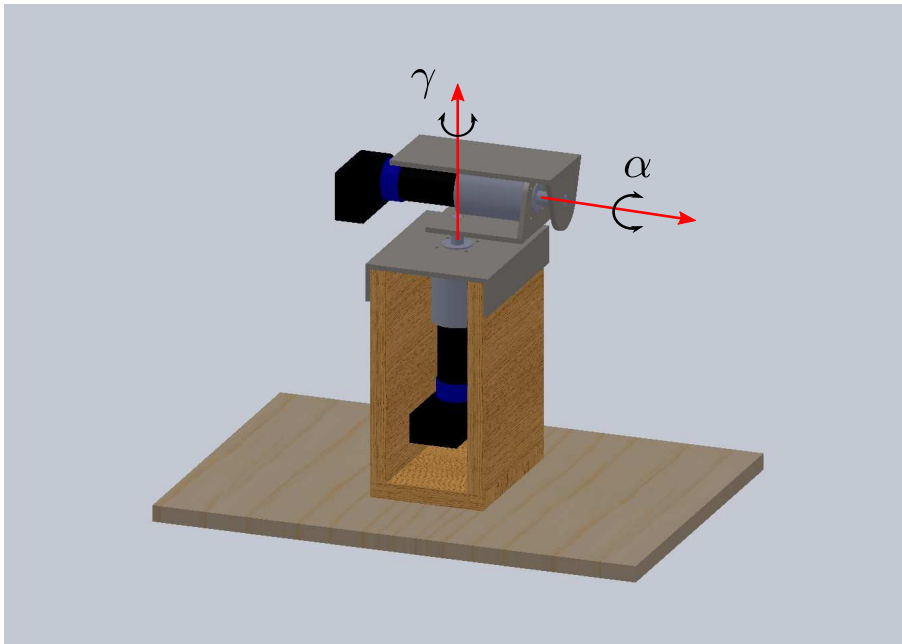


Figura 3.8: Modelo de uma ISP com dois graus de liberdade em uma estrutura Pan/Tilt

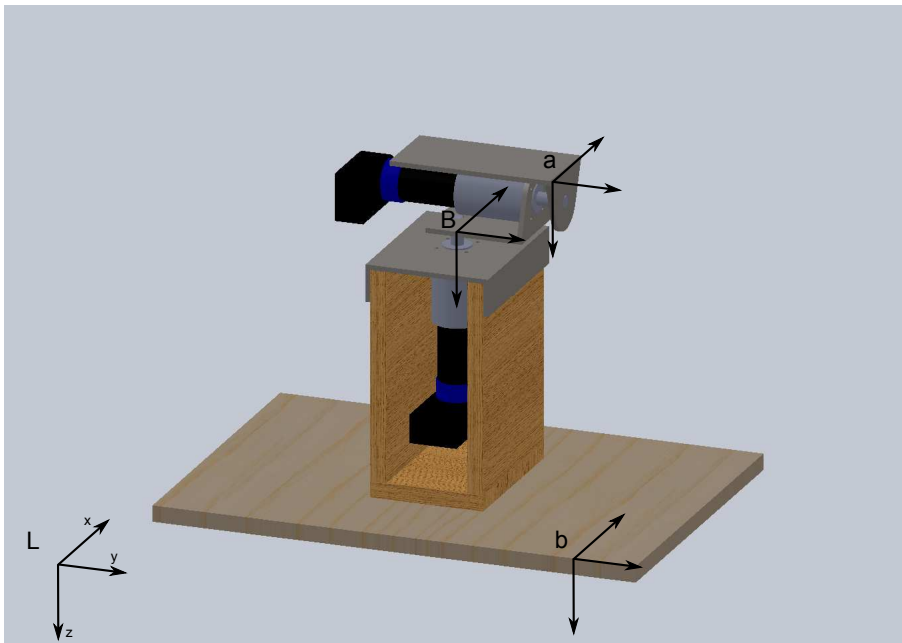


Figura 3.9: Sistemas de coordenadas utilizado para a ISP com dois graus de liberdade

Desta forma, seguindo a representação em termo de ângulos de balanço (ψ , *roll*), caturro (θ , *pitch*) e guinada (ϕ , *yaw*), a orientação do veículo em relação ao referencial inercial L, é dado por

$$R_{bL} = R_z(\phi)R_y(\theta)R_x(\psi) \quad (3.6)$$

Em que $R_i(k)$ é a rotação elementar ao redor do eixo i com um ângulo k . Este tipo de representação pode ser utilizado, uma vez que seus pontos de singularidade, $\theta = \pm\pi/2$, não fazem parte das condições de operação do ROV.

A linha de visada da câmera é considerada paralela ao eixo x do sistema de coordenadas a . A orientação da LOS pode ser obtida pela primeira coluna de R_{aL} , que por sua vez é obtida em termos da orientação do veículo e dos ângulos de compensação:

$$U_L = R_{bL}(\psi, \theta, \phi)R_z(\gamma)R_x(\alpha)e_x \quad (3.7)$$

Com base em uma orientação da LOS desejada, definida por U_L^* , e da orientação do veículo, é preciso calcular os ângulos de elevação. Isto é

$$R_z(\gamma)R_x(\alpha)e_x = R_{bL}^T(\psi, \theta, \phi)U_L^* \quad (3.8)$$

O lado direita da equação é conhecido, calculado a partir da medição do sensor inercial e da direção desejada. O lado esquerda é definido por

$$R_z(\gamma)R_x(\alpha)e_x = \begin{bmatrix} \cos(\gamma) \cos(\alpha) \\ \sin(\gamma) \cos(\alpha) \\ -\sin(\alpha) \end{bmatrix} \quad (3.9)$$

A igualdade permite definir os ângulos de correção de elevação e conteira. Note que apesar de possibilitar a orientação da linha de visada, este tipo de sistema não compensa movimentos de translação e rotações no balanço (*roll*) do veículo.

Em relação a dinâmica, este sistema possui propriedades que não são vistas no caso planar. Mas se efeitos giroscópicos forem desprezíveis, se não houver acoplamento cinemático entre os eixos e se estes tiverem sistemas idênticos de servomotores, a simplificação pode ser adotada para um gimbal com mais de um grau de liberdade.[13] Esta



Figura 3.10: Sensor *MicroStrain 3DM-GX3-25*

consideração será adotada para o projeto do sistema.

3.2 Metodologia Proposta

Esta seção visa explicar o princípio de funcionamento dos componentes da ISP que foi desenvolvida, que é constituída basicamente um sensor inercial, seção 3.2.1, e um motor DC, seção 3.2.2. Em seguida um controle é proposto na seção 3.2.3.

3.2.1 Sensor inercial

O sensor *MicroStrain 3DM-GX3-25* é um sistema de referenciamento de atitude e rumo (AHRS). O dispositivo é constituído de uma unidade de medição inercial (IMU, *Inertial Measurement Unit*), sensores de temperatura e um processador embarcado. A IMU é um conjunto de sensores MEMS (*Microelectromechanical systems*): um acelerômetro triaxial (mede a aceleração linear), um magnetômetro triaxial (mede o campo magnético local) e um giroscópio triaxial (mede a velocidade angular com relação a um referencial inercial). Como algumas propriedades dos sensores são alteradas com variações na temperatura, cinco termômetros são empregados para possibilitar a aplicação de fatores de correção. A folha de dados com suas especificações técnicas está no apêndice A.3.

Princípio de Funcionamento

Basicamente os sensores MEMS possuem uma carga de prova e medem uma força aplicada sobre ela em um determinado eixo. Com base na força resultante é possível medir a aceleração linear. A velocidade angular do corpo em relação a um referencial inercial é derivado da força inercial de Coriolis. E o campo magnético local é medido por meio da força de Lorentz.

O sensor utiliza como referência inercial o sistemas de coordenadas NED (*North-East-Down*), também conhecido como LTP (*Local Tangent Plane*), que é comumente usado na navegação de veículos. Neste sistema, representado na figura 3.11, a origem é fixada arbitrariamente em um ponto da superfície da terra, o eixo x aponta para o norte geográfico e tangência o meridiano que passa pela origem, o eixo z aponta para o centro da terra, e por se tratar de um sistema de mão direita, y conseqüentemente aponta para o leste.

Logo que o sensor começa a funcionar é preciso calcular a orientação do sistema de coordenadas do sensor em relação ao sistema de coordenadas inercial. Para isso o acelerômetro e o magnetômetro são lidos. O primeiro, desprezando a presença de outras forças, mede a aceleração da gravidade, o que define o eixo z . O segundo, considerando apenas a presença do campo magnético terrestre, mede o vetor que aponta para o norte magnético, que é usado como uma aproximação para o eixo x .

É possível compensar o erro entre o norte magnético e o norte geográfico por meio de uma medida de latitude e longitude do sensor, dada por um GPS por exemplo, e de um modelo do campo magnético da terra. No caso do ROV LUMA, isto só será possível enquanto estiver na superfície.

Sabendo a orientação do sistema de coordenadas do sensor em relação ao sistema de coordenadas inercial e se considerarmos giroscópios ideais, a integração das medidas destes permite saber a orientação do sensor em qualquer instante de tempo. Imperfeições no sistema, como por exemplo deriva térmica (*drift*), levam a um erro que cresce com o tempo, no caso do sensor *3DM-GX3-25* em uma taxa em torno de meio grau por minuto em cada eixo. Para compensar tais erros o sensor deve medir continuamente as referências

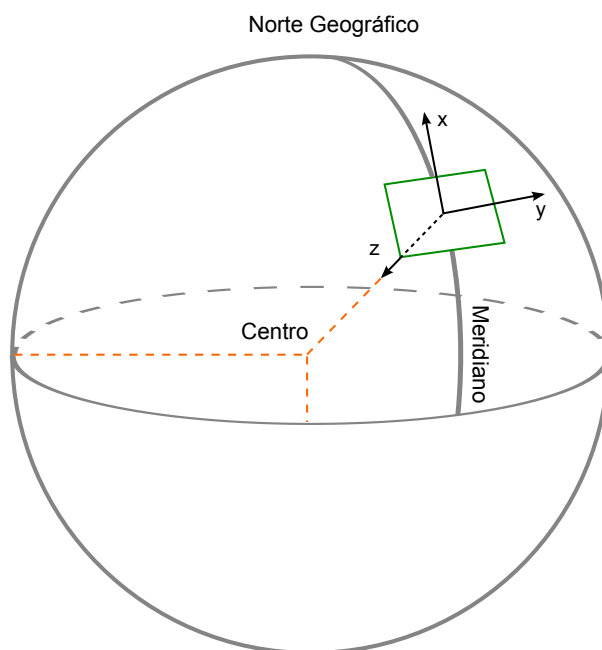


Figura 3.11: Sistema de coordenadas NED representado no globo terrestre

de norte e de gravidade. O ganho de compensação pode ser ajustado para definir quão rápido o sistemas de coordenadas de referência é realinhado com o sistemas de coordenadas NED. Este parâmetro é dado em unidade de segundos e representa o tempo que leva para o sistema de coordenadas se aproximar em torno de 90% dos vetores de norte e gravidade. [12]

Outro problema decorre do uso de magnetômetros quando o sensor está próximo de um dos polos magnéticos, onde vetor do campo magnético terrestre é aproximadamente perpendicular a superfície da terra, como é representado na figura 3.12. O problema é relatado em [10], em um trabalho que envolveu a navegação de um AUV (*Autonomous Underwater Vehicle*) em uma região ártica, em um local que o campo magnético possuía uma inclinação de $83,5^\circ$ em relação a superfície terrestre. No caso da Baía do Almirantado, a inclinação do campo magnético é de $55,5^\circ$ de acordo com o modelo IGRF11 (*International Geomagnetic Reference Field - 11*). Portanto, para o ROV LUMA, este problema não precisa ser considerado.

O cálculo do norte magnético é particularmente sensível em ambientes fechados, cujo campo magnético local é distorcido por materiais ferromagnéticos, como ferro e níquel, por exemplo. Se esta distorção for estática é possível calibrar os magnetômetros por meio

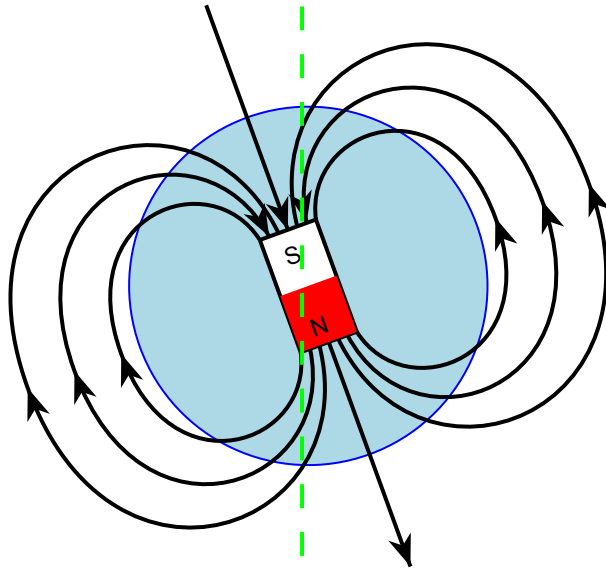


Figura 3.12: Representação do Campo Magnético Terrestre

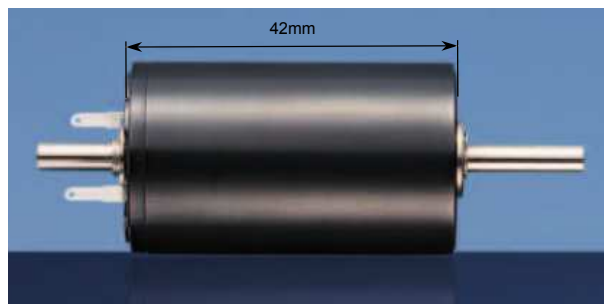


Figura 3.13: Motor DC - *Faulhaber 2342S024CR* sem a caixa de redução

de *Soft Iron Calibration*, para materiais que distorcem o campo magnético local, e *Hard Iron Calibration*, para materiais que são fontes de campo magnético. Tais métodos não funcionam para distorções que variam com o tempo, como é o caso deste trabalho, em que o sensor se movimenta em relação a motores DC. [12] Nesta ocasião é possível desativar o magnetômetro e considerar apenas a leitura do giroscópio, cujo desvio não é compensado.

3.2.2 Motor DC

O motor DC utilizado, *Faulhaber 2342S024CR*, é constituído de um estator de ímã permanente e de um rotor ligado a um comutador de grafite. O circuito elétrico equivalente é mostrado na figura 3.14. Como o fluxo magnético é gerado pelo ímã permanente, V_f é constante e o motor é controlado pela tensão de armadura, $u(t)$. No circuito de armadura,

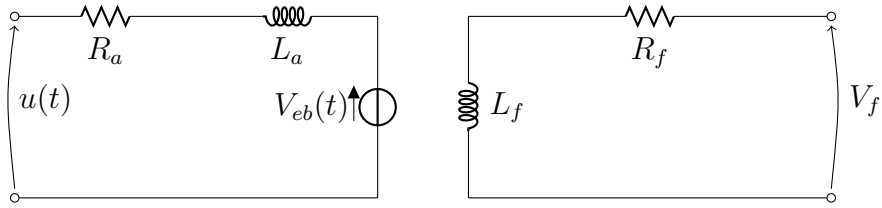


Figura 3.14: Circuito elétrico do Motor DC

R_a e L_a representam a resistência e a indutância medidas nos terminais do rotor.

É provado pela lei de Faraday-Neumann-Lenz que a força contra-eletromotriz gerada pelo movimento do rotor é proporcional a velocidade angular do mesmo, $V_{eb}(t) = K_e \omega_1(t)$.

Temos então pela Lei das Malhas de Kirchhoff aplicada ao circuito de armadura:

$$L_a \frac{di_a(t)}{dt} = u(t) - R_a i_a(t) - K_e \omega_1(t) \quad (3.10)$$

Para a parte mecânica, temos que considerar a presença de uma caixa de redução, de forma que a velocidade angular do eixo de saída do motor é uma fração da velocidade angular do rotor:

$$\omega_2(t) = \frac{\omega_1(t)}{N} \quad (3.11)$$

Em uma idealização, mostrada em [4], desprezando perdas por atrito e folgas entre os dentes das engrenagens da redução, a energia se conserva, de forma que:

$$T_1(t) \omega_1(t) = T_2(t) \omega_2(t) \quad (3.12)$$

Sendo $T_1(t)$ e $T_2(t)$ os torques exercidos na entrada e na saída da caixa de redução.

Das equações 3.11 e 3.12, temos:

$$T_2(t) = N T_1(t) \quad (3.13)$$

Fazendo o somatório dos torques exercidos no eixo interno do motor:

$$J_1 \frac{d\omega_1(t)}{dt} = T_m(t) - T_v(t) - T_1(t) \quad (3.14)$$

Sendo que J_1 é o momento de inércia do rotor, J_m , adicionado do momento de inércia do eixo interno e da engrenagem de entrada da caixa de redução.

Pela equação da força de Lorentz aplicada ao motor DC, o torque gerado pelo rotor é proporcional à corrente de armadura, $T_m = K_t i_a(t)$. O torque gerado pelo atrito viscoso é considerado proporcional à velocidade angular do sistema, $T_v = F_1 \omega_1(t)$, e T_1 é o torque exercido para girar a engrenagem de entrada do conjunto de redução.

$$J_1 \frac{d\omega_1(t)}{dt} = K_t i_a(t) - F_1 \omega_1(t) - T_1(t) \quad (3.15)$$

O torque da saída da caixa de redução, T_2 , pode ser modelado por:

$$T_2(t) = J_2 \frac{d\omega_2(t)}{dt} + F_2 \omega_2(t) + T_l(t) \quad (3.16)$$

Sendo J_2 o momento de inércia das engrenagens de saída da caixa de redução, F_2 o coeficiente de atrito viscoso e T_l o torque exercido por uma carga.

Juntando as equações 3.11, 3.13 e 3.16:

$$T_1(t) = \frac{J_2}{N^2} \frac{d\omega_1(t)}{dt} + \frac{F_2}{N^2} \omega_1(t) + \frac{T_l(t)}{N} \quad (3.17)$$

Juntando as equações 3.15 e 3.17, temos:

$$J_{eq} \frac{d\omega_1(t)}{dt} = K_t i_a(t) - F_{eq} \omega_1(t) - T_{Leq}(t) \quad (3.18)$$

Em que

$$J_{eq} = J_1 + \frac{J_2}{N^2} \quad F_{eq} = F_1 + \frac{F_2}{N^2} \quad T_{Leq} = \frac{T_l}{N} \quad (3.19)$$

A fusão das equações diferenciais 3.10 e 3.18 é mostrada, na forma de diagrama de blocos, na figura 3.15.

Se considerarmos que a dinâmica do acionamento elétrico é muito mais rápida que

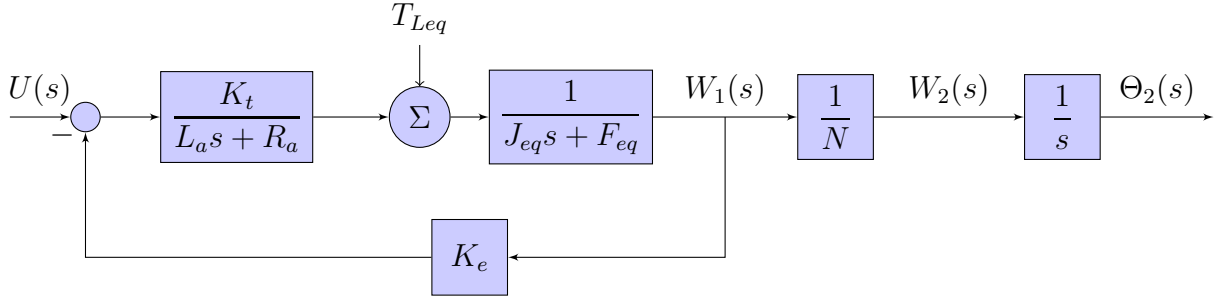


Figura 3.15: Diagrama em blocos do Motor DC

a dinâmica associada ao sistema mecânico, a indutância L_a é desprezada e podemos simplificar o sistema para que o modelo da função de transferência entre a tensão de entrada e a velocidade angular do sistema seja de primeira ordem:

$$W_2(s) = \frac{k_n/N}{\tau_m s + 1} U(s) \quad (3.20)$$

Onde $k_n = \frac{K_t}{\gamma}$ e $\tau_m = \frac{R_a J_{eq}}{\gamma}$, com $\gamma = K_t K_e + F_{eq} R_a$

Fazendo o mesmo para o torque de carga:

$$W_2(s) = \frac{k_{tl}}{\tau_{tl} + 1} T_l(s) \quad (3.21)$$

Em que $k_{tl} = -\frac{1}{\psi N^2}$ e $\tau_{tl} = \frac{J_{eq}}{\psi}$, com $\psi = \frac{K_t K_e}{R_a} + F_{eq}$

A maioria dos parâmetros do modelo podem ser pegos na folha de dados do motor, ver apêndice A.1, no entanto os parâmetros referentes aos momentos de inércia e aos coeficientes de atrito viscoso são desconhecido.

O cálculo de τ_m revela um valor na ordem de alguns milissegundos. A dinâmica do motor é bem mais rápida do que será exigido pelo gimbal no problema da estabilização, de modo que a relação entre a posição angular do eixo e a tensão de entrada pode ser modelada por um ganho seguido de um integrador:

$$\frac{\Theta_2(s)}{U(s)} = \frac{K_m}{s} \quad (3.22)$$

Em que $K_m = K_n/N$.

3.2.3 Projeto de um controlador PI

Para uma ISP o parâmetro mais importante para avaliar a performance do controle é a largura de banda da malha fechada do sistema. Há diferentes definições para esse parâmetro. Em [2] isto é definido como a faixa de frequência em que o ganho do sistema é maior que um fator de $1/\sqrt{2}$ de seu valor de referência, o que equivale a uma queda de $3dB$. Este valor depende do tipo de sistema considerado. É o ganho de regime, quando este é finito e diferente de zero, em sistemas passa-baixas. Para sistemas passa-altas, o valor de referência é o ganho de alta frequência.

O movimento da base, no caso do ROV LUMA, está restrito a uma banda de aproximadamente 5 Hz, valor aproximado definido a partir de medições coletadas em uma das expedições realizadas. Também consideramos que o sinal que comanda o posicionamento da LOS está restrito a essa mesma faixa. Portanto, no caso da estrutura discutida em 3.1.1 é preciso que a banda do sistema de malha fechada passe sinais de até 5 Hz. No segundo caso, 3.1.2, o sistema deve bloquear distúrbios na saída de até 5 Hz.

Consideramos um controlador do tipo PI (Proporcional e Integral), que segundo [8] é bastante utilizado no controle de ISPs, mesmo alguns de alta performance. O mesmo autor nota que um desempenho melhor pode ser obtido com um controle mais complexo, com métodos adaptativos e observadores de distúrbios, por exemplo.

A seguinte notação para um controlador PI é adotada,

$$C(s) = K_p \left(1 + \frac{1}{T_i s}\right) \quad (3.23)$$

em que K_p é o ganho proporcional e T_i o tempo integral.

Se considerarmos o modelo da planta como um ganho seguido de um integrador, o que foi visto na seção anterior, temos que

$$P(s) = \frac{K_m}{s} \quad (3.24)$$

Substituindo 3.23 e 3.24 nas funções de transferência 3.2 e 3.4, temos

$$\frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{K_m K_p s + (K_m K_p / T_i)}{s^2 + K_m K_p s + (K_m K_p / T_i)} \quad (3.25)$$

E para as funções de transferência 3.3 e 3.5, temos

$$\frac{1}{1 + C(s)P(s)} = \frac{s^2}{s^2 + K_m K_p s + (K_m K_p / T_i)} \quad (3.26)$$

Como se trata de um sistema de segunda ordem, a parametrização da equação característica com base no fator de amortecimento, ξ , e na frequência natural do sistema, ω_n , facilitará na interpretação do problema.

$$\frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.27)$$

$$\frac{1}{1 + C(s)P(s)} = \frac{s^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.28)$$

O ganho para a frequência natural da função 3.28 é dado por $1/(2\xi)$. Portanto, quando $\xi = 1/\sqrt{2}$, ω_n define a largura de banda do sistema, que será escolhida igual a $5Hz$, ou aproximadamente $31,42 \text{ rad/s}$. Com estes parâmetros, o gráfico do ganho com relação a frequência de entrada é mostrado na figura 3.16.

Os parâmetros de controle podem ser então definidos:

$$K_p := \frac{2\xi\omega_n}{K_m} \quad T_i := \frac{K_m K_p}{\omega_n^2} \quad (3.29)$$

Este modelo teórico será usado para a definição do controlador do sistema experimental que será descrito no capítulo seguinte. Como pode ser visto no capítulo 4, a largura de banda será definida a partir de resultados experimentais.

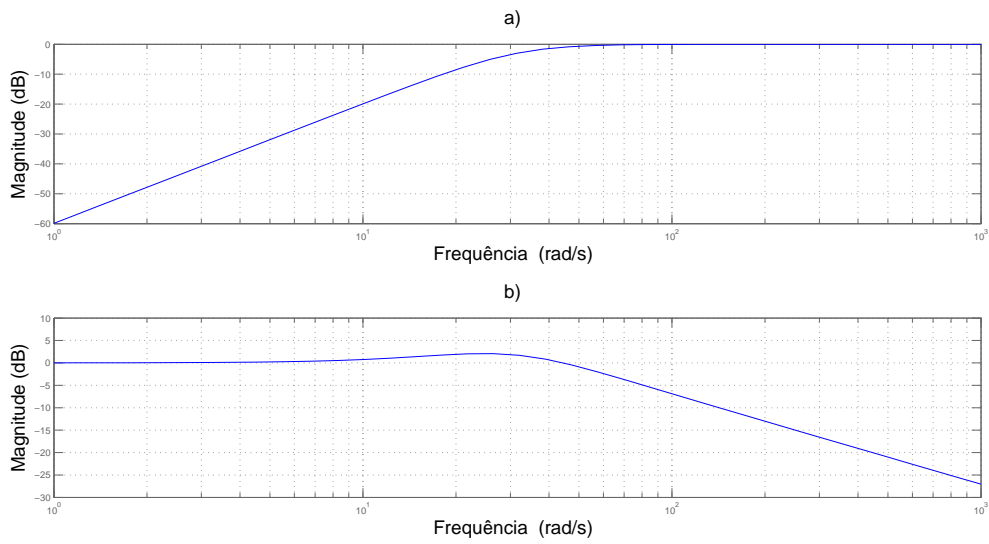


Figura 3.16: Ganho dos sistemas 3.28 (a) e 3.27 (b) com $\xi = 1/\sqrt{2}$ e $\omega_n = 31,42\text{rad/s}$

Capítulo 4

Desenvolvimento do Sistema Experimental

O mecanismo Pan/Tilt foi montado com dois motores DC, como mostra a figura 4.1. Nesta imagem o sensor inercial está fixado em cima da ISP, ao lado da câmera. O sistema é colocado em uma prancha de madeira, que servirá para simular movimentos do ROV LUMA.

O desenvolvimento começou com a implementação de um circuito de controle com o dispositivo *Arduino*, uma plataforma baseada em um microcontrolador *Atmel AVR*. Esta primeira tentativa acabou não obtendo sucesso. Uma vez que o microcontrolador não possui uma unidade aritmética de ponto flutuante, cálculos com este tipo de variável consomem muito tempo de processamento, o que acarreta em baixas frequências de amostragem do ciclo de controle. O circuito de potência, que alimenta o motor, possuía uma zona morta razoavelmente grande, o que criava dificuldades no controle. Apesar de serem problemas contornáveis, por meio de aritmética de ponto fixo, para o primeiro caso, e compensação de zona morta, para o segundo, a soma destes e outros fatores estimulou o uso de uma plataforma já consagrada para a prototipagem de sistemas de controle.

A placa Q8, de controle *Hardware in the Loop* (HIL), da *Quanser*, foi utilizada. Ela possui canais de saída analógicas e sinais de PWM (*Pulse Width Modulation*), que permitem por meio de um circuito de potência, alimentar o motor, e canais de entrada para a leitura de *encoders*, sensores que servem para a leitura da posição angular do eixo dos motores. Na parte de *software* foi utilizada a plataforma *QUARC*, também da *Quanser*.

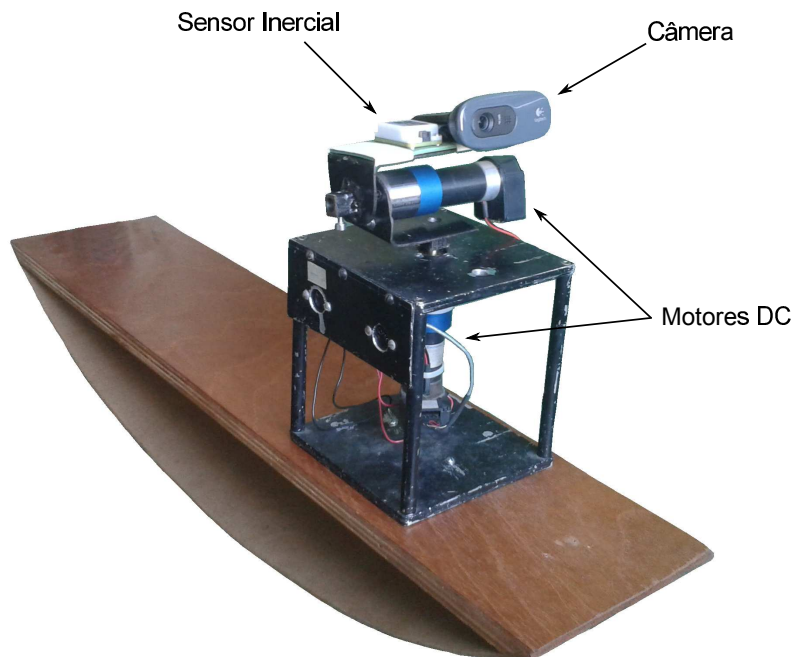


Figura 4.1: Foto do sistema experimental montado em uma base de madeira

O que permite integrar a placa Q8 com modelos do *Simulink*, gerando um código que possa ser executado em tempo real no *Real-Time Windows Target*. O *QUARC* também possui bibliotecas que possibilitam diversos protocolos de comunicação, o que será usado para requisitar dados do sensor inercial, conforme será visto na seção 4.1.

De forma simplificada o diagrama do sistema experimental está representado nas figuras 4.2 e 4.3, para as duas estratégias discutidas no capítulo anterior. A próxima seção, 4.1, explicará a comunicação entre o *QUARC* e o sensor inercial, bem como sua configuração. Na seção 4.2 a forma como foi feita o controle dos motores DC será apresentada.

4.1 Comunicação com o Sensor Inercial

O *QUARC* possibilita a comunicação em diversos protocolos: TCP/IP, UDP, SPI, Memória Compartilhada, Serial, ARCNET, *Pipe* e *File Protocol*. Estes protocolos são implementados pelo *QUARC* de maneira unificada, de forma que a comunicação é definida por um URI (*Universal Resource Identificador*), de modo a tornar a configuração o mais simples possível. Em contrapartida há perda de flexibilidade para fazer operações de

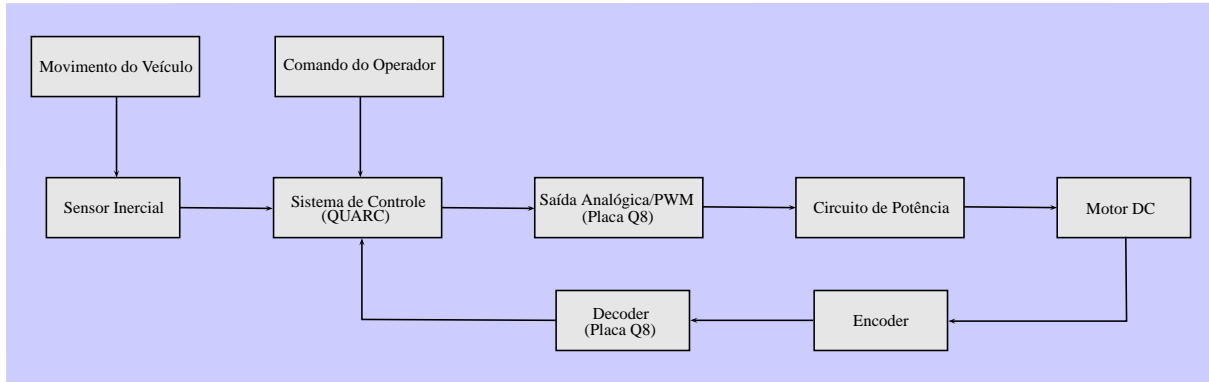


Figura 4.2: Diagrama de blocos do sistema experimental com o sensor inercial fixado no veículo

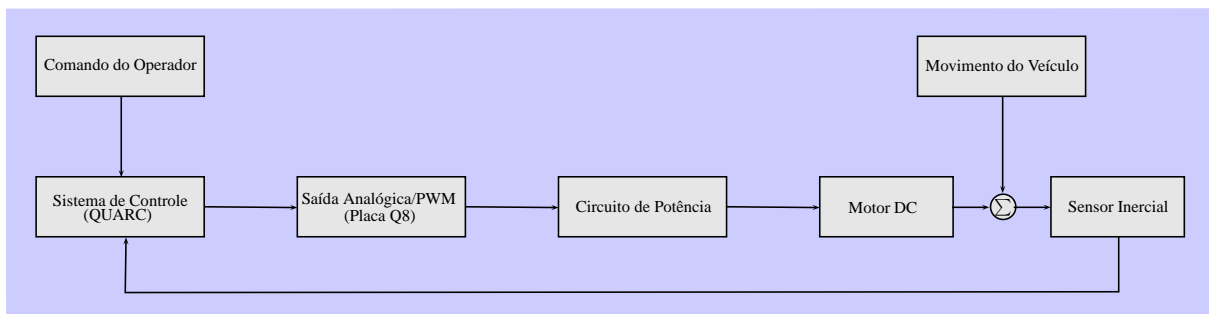


Figura 4.3: Diagrama de blocos do sistema experimental com o sensor inercial fixado na ISP

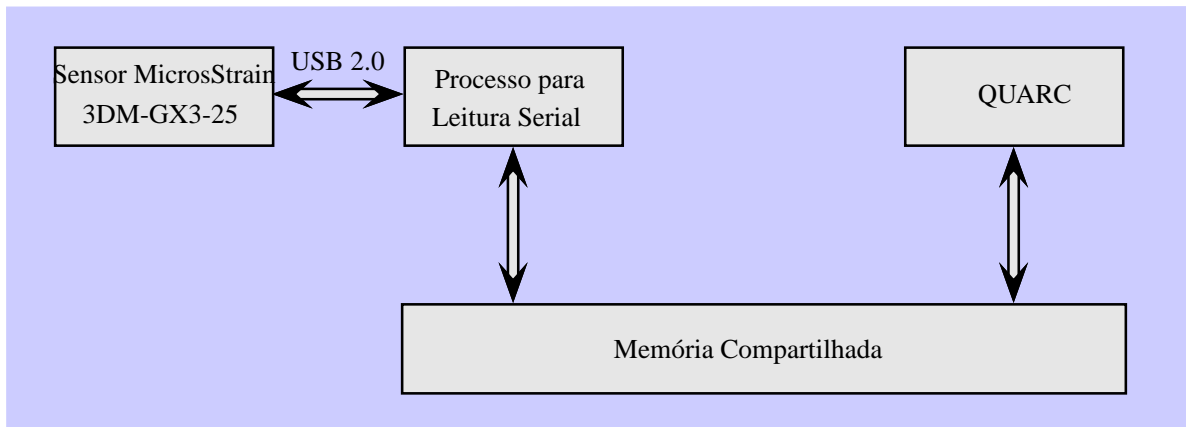


Figura 4.4: Diagrama da comunicação do sensor *3DM-GX3-25* com o *QUARC*

baixo nível com dados recebidos.

No caso é preciso processar e verificar a integridade dos dados antes de usá-los. Para tal a comunicação foi feita em duas etapas, como está representado na figura 4.4. Um processo é responsável pela comunicação serial com o sensor, bem como pela verificação dos dados. Se estes forem válidos, são escritos em um endereço da memória RAM compartilhada com um segundo processo, que os insere na malha de controle. A comunicação por memória compartilhada foi escolhida por ser, geralmente, a mais rápida para transmissão de dados entre dois processos.

O algoritmo do processo que estabelece a interface entre o sensor e o *QUARC* obedece a estrutura mostrada no fluxograma 4.5. Após configurações de inicialização, a comunicação via memória compartilhada é estabelecida com o sistema desenvolvido no *QUARC*. Se houver sucesso o programa entra em um ciclo, que requisita, verifica e transmite os dados do sensor para o *QUARC*. Em caso de falha no algoritmo que verifica a integridade dos dados, o último passo não é feito e o programa volta para o começo do ciclo. Em qualquer outra falha o programa é encerrado.

4.1.1 Comunicação serial

A comunicação serial foi realizada com o sistema operacional *Microsoft Windows XP*, no ambiente de desenvolvimento *Microsoft Visual Studio 2005*, compatível com o *QUARC*. A implementação se baseou em bibliotecas do sistema, com base em configurações vistas

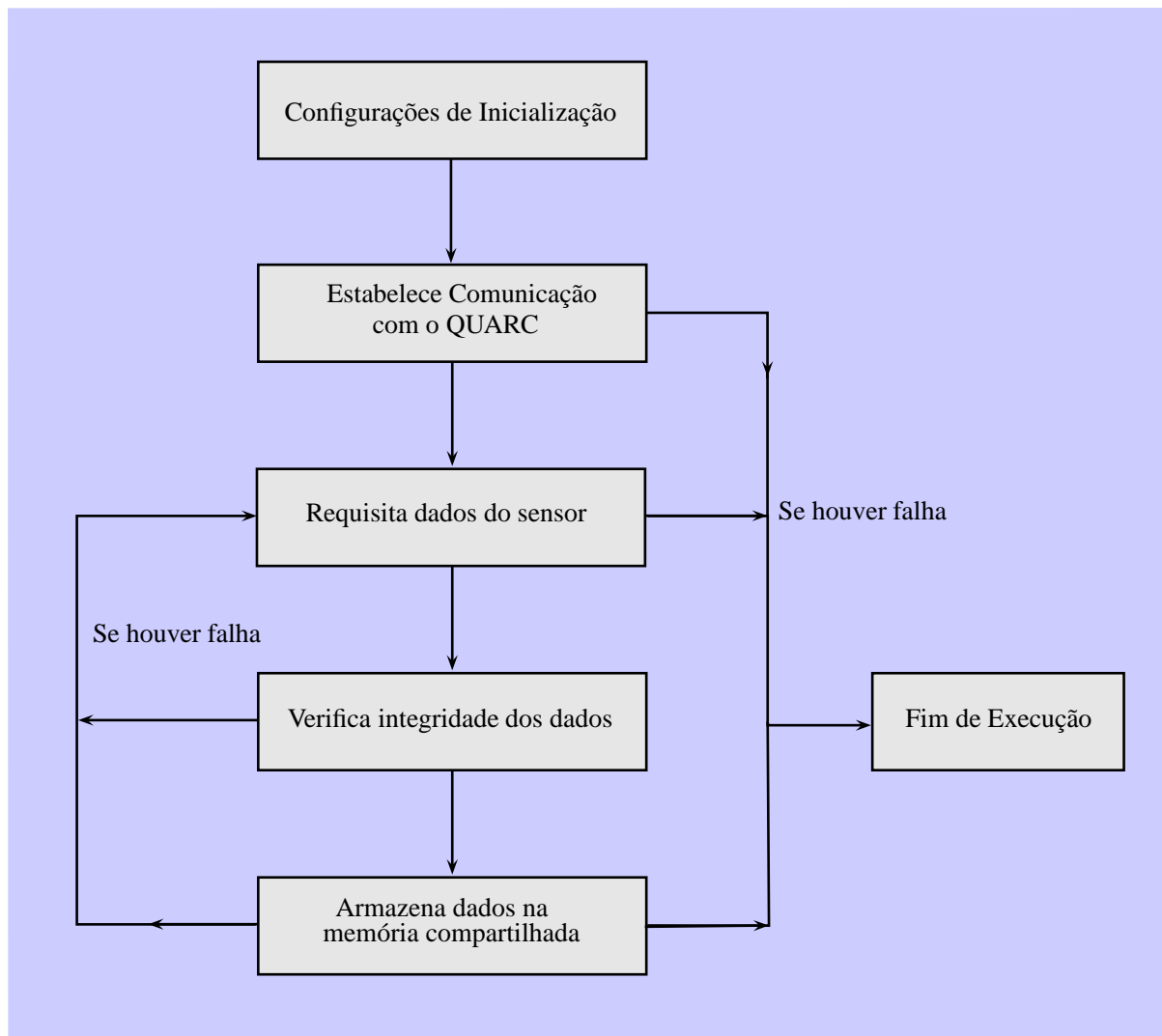


Figura 4.5: Fluxograma do algoritmo de comunicação do sensor *3DM-GX3-25* com o *QUARC*

em [1]. O código, na linguagem *C*, está completo no apêndice B.1. Nesta seção apenas os pontos mais importantes serão comentados.

A biblioteca do sistema, *windows.h*, define uma estrutura chamada *DCB*, que armazena parâmetros de controle para dispositivos de comunicação serial. Neste caso os seguintes parâmetros são definidos:

```
72 dcb.BaudRate = 921600;           //BaudRate (ignorado para USB)
dcb.ByteSize = 8;                //Dados com 8 bits
dcb.Parity = NOPARITY;          //Sem bit de paridade
74 dcb.StopBits = ONESTOPBIT;     //1 bit de stop
fSuccess = SetCommState(hCom, &dcb);
```

O valor de *Baud Rate* corresponde ao número de símbolos que são transmitidos por segundo. Isto compreende tanto os bits de controle, quanto aos dados propriamente ditos. Este valor não é considerado para uma interface USB, como é o caso. Quando a comunicação é feita pela saída Serial TTL do sensor, há apenas quatro valores possíveis de *Baud Rate*: 115200, 230400, 460800 e 921600. Este parâmetro restringe a frequência máxima de saída de dados do sensor, que é de 1000 amostras por segundos para o maior valor de *Baud Rate*. Em casos de dispositivos mais limitados, como microcontroladores, o *Baud Rate* não pode ser escolhido arbitrariamente. Um fator de erro de comunicação, que depende do dispositivo, deve ser considerado na escolha.

A comunicação se dá requisitando um dado específico. A descrição completa com todos os comandos e respostas possíveis do sensor está em [11]. Aqui apenas os ângulos de orientação de *roll*, *pitch* e *yaw* são de interesse. Para receber tais dados um *byte* de comando deve ser enviado com o valor *0xCE*.

```
154 WriteFile (hCom,                // handle
      &getEuler,                  // Array que sera enviado
      1,                          // Bytes que devem ser enviados
156 &dwNumBytesWritten,          // Bytes efetivamente enviados
      NULL);
```

A resposta deve vir em um pacote de 19 *bytes*, de acordo com a tabela 4.1, que são armazenado no *array SensData*.

```
160 ReadFile (hCom,                // handle
      &SensData,                  // Array que sera escrito
      19,                          // Numero de bytes lidos
162 &dwBytesTransferred,        // Bytes efetivamente recebidos
      NULL);
```

| | |
|--------------------|--|
| <i>Byte 1</i> | 0xCE |
| <i>Bytes 2-5</i> | Balanço ou <i>Roll</i> (IEEE-754 <i>Float</i>) |
| <i>Bytes 6-9</i> | Caturro ou <i>Pitch</i> (IEEE-754 <i>Float</i>) |
| <i>Bytes 10-13</i> | Guinada ou <i>Yaw</i> (IEEE-754 <i>Float</i>) |
| <i>Bytes 14-17</i> | <i>Timer</i> |
| <i>Bytes 18-19</i> | <i>Checksum</i> |

Tabela 4.1: Estrutura da resposta do sensor *3DM-GX3-25* para uma requisição dos ângulos de Balanço (*roll*), Caturro (*pitch*) e Guinada (*yaw*)

O primeiro *byte* é um eco do comando enviado para o sensor. Cada ângulo é uma variável do tipo, *Float* IEEE-754, com quatro *bytes*, sendo que os mais significativos são transmitidos primeiro. O *Checksum* é uma variável de dois *bytes*, do tipo *Unsigned Short*, e deve corresponder a soma de todos os *bytes* anteriores. Se a soma for diferente do valor de *Checksum*, significa que houve algum tipo de erro na transmissão dos dados e será mais seguro não utilizá-los.

O ambiente em que o sistema foi desenvolvido possui um processador que armazena os *bytes* menos significativos por último (*little-endian*). Como os *bytes* mais significativos são enviados primeiro, o *array SensData* tem a ordem revertida. Desta forma a verificação de *Checksum* e a leitura dos dados fica simplificado:

```

170 //Define um ponteiro apontado para o CheckSum recebido
    unsigned short* imuChecksum = (unsigned short*) revSensData;
    //Subtrai a soma de todos os bytes recebidos (com execucao dos de CheckSum) do dado
    de CheckSum
172 for(int i = 2; i < 19; i++)
    *imuChecksum -= revSensData[i];
174
    if ( *imuChecksum == 0 ){//Se o CheckSum esta correto
176 float yaw = *((float*) (revSensData + 6));
    float pitch = *((float*) (revSensData + 10));
178 float roll = *((float*) (revSensData + 14));

```

Os dados então são transmitidos via memória compartilhada, o que será descrito a seguir.

4.1.2 Comunicação por memória compartilhada

O *QUARC* fornece uma API para estabelecer um canal de comunicação entre um aplicativo e o código gerado para o modelo do *Simulink*. Antes é preciso configurar a IDE para utilizar as bibliotecas fornecidas. No caso da IDE *Microsoft Visual Studio 2005*, nas

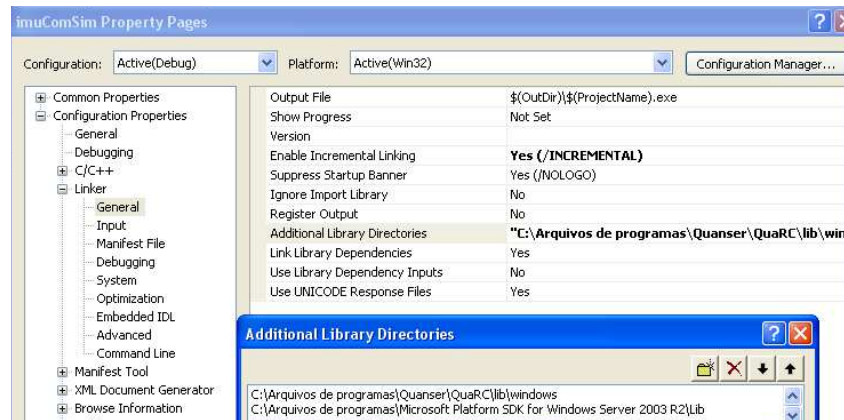


Figura 4.6: Configuração do *Visual Studio* para a comunicação com o *QUARC*

propriedades do projeto é preciso adicionar o local das bibliotecas estáticas, conforme é visto na figura 4.6, bem como o nome delas, na figura 4.7.

Dois arquivos de cabeçalho serão utilizados, *quanser_messages.h* e *quanser_stream.h*. O local destes arquivos deve ser incluído no campo mostrado pela figura 4.8.

Os parâmetros de configuração da comunicação por meio das bibliotecas do *QUARC* são definidas a seguir:

```

122  const char    uri []           = "shmem://imuData:1"; //URI
123  const t_boolean nonblocking   = false;
124  const t_int    send_buffer_size = 8000; //Pilha de
125  const t_int    receive_buffer_size = 8000; //Pilha de entrada
126  const char *   locale         = NULL;
127  t_stream client;
128  t_error result;
129  char message [512];

```

O endereço URI define o protocolo usado, *shmem*, e uma identificação, *imuData:1*, para estabelecer a comunicação com um servidor. Pilhas de entrada e saída também são definidas, para armazenar os dados recebidos e transmitidos. A conexão é feita chamando a função *stream_connect*:

```

134  result = stream_connect(uri, nonblocking, send_buffer_size, receive_buffer_size, &
      client);

```

Após as configurações de inicialização, quando dados válidos são recebidos do sensor, esses são transmitidos para o *QUARC*:

```

190  //array com os dados a serem enviados
191  t_single dataEuler [3] = {roll-roll10, pitch, yaw-yaw0};
192  //envia o array dataEuler para a comunicacao com o Quarc

```

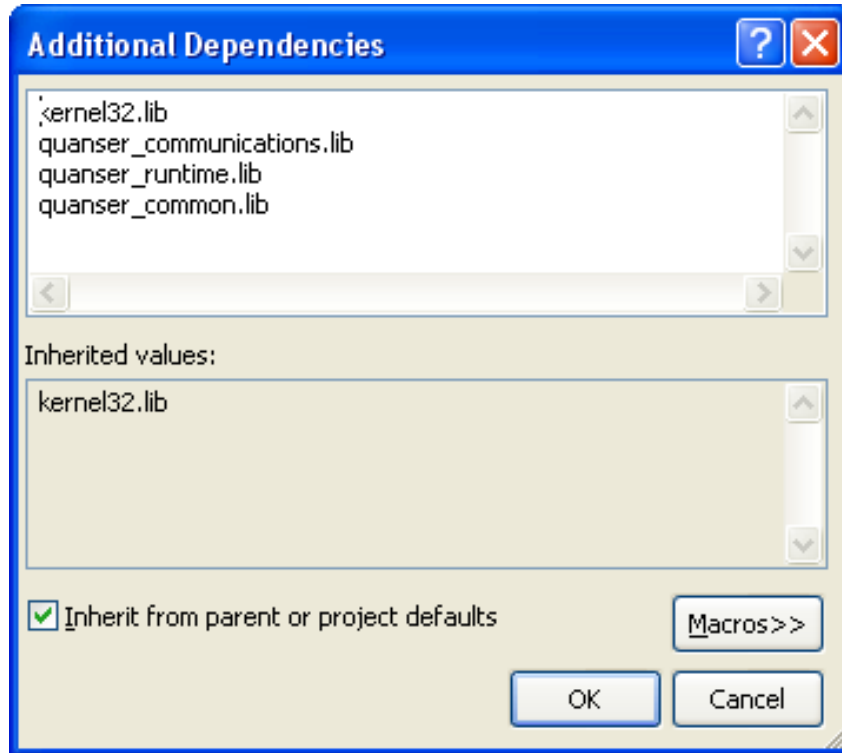


Figura 4.7: Configuração do *Visual Studio* para a comunicação com o *QUARC*

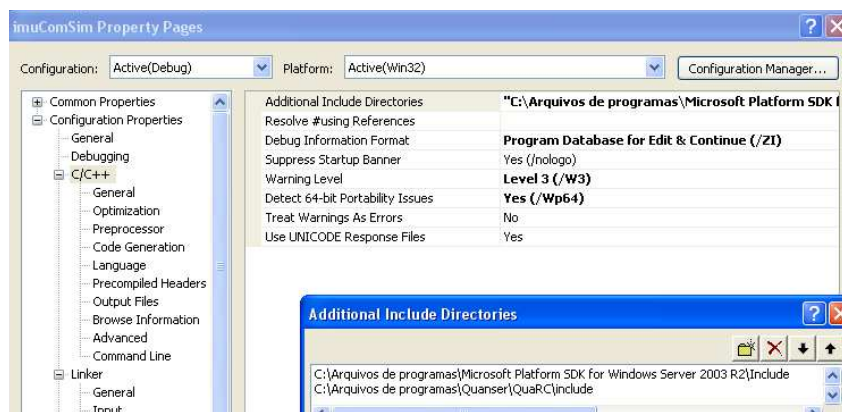


Figura 4.8: Configuração do *Visual Studio* para a comunicação com o *QUARC*

```

result = stream_send_single_array(client, dataEuler, 3);
if (result < 0)
    break;//Se houve erro na transmissao, sai do loop principal

```

Por parte do modelo no *Simulink*, há um bloco de servidor, figura 4.10, que procura e estabelece comunicação com um cliente, local ou remoto, que tente se conectar na URI especificada, no caso *shmem://imuData:1*. Os parâmetros desse bloco são exibidos na figura 4.9. Os dados são recebidos como uma tripla de variáveis do tipo *single*, que corresponde ao tipo *IEEE-754 Float* enviado pelo sensor.

4.1.3 Parâmetros de configuração do sensor inercial

É possível configurar e calibrar o sensor por meio de comandos, de forma parecido com que é feito para requisitar os ângulos de orientação. A *MicroStrain* também fornece um software que facilita esse tipo de tarefa, o *3DM-GX3 Monitor*. A janela de configuração, figura 4.11, permite modificar parâmetros como: taxa de amostragem, taxa de comunicação, compensação de norte e largura da janela de filtros dos sensores.

Como foi visto no capítulo anterior, o magnetômetro é particularmente problemático, o que levou a necessidade de desativá-lo. Desta forma apenas os acelerômetros e giroscópios são usados para o cálculo da orientação. Sem um referencial de norte para o cálculo do ângulo de *yaw*, o erro dos giroscópios são acumulados, o que leva a um *drift* em torno de $0,5^\circ/min$.

A filtragem digital dos dados dos sensores é feita em duas etapas. A primeira remove ruídos de alta frequência, acima de 30kHz, dos MEMS. A segunda etapa pode ser configurada de acordo com as necessidades do usuário para remover outros tipos de distúrbios, como por exemplo ruído de fonte de alimentação em cima dos magnetômetros. Neste trabalho a frequência de corte foi definida em seu valor mínimo, de 31,25Hz, uma vez que a dinâmica do veículo é restrita a uma banda de 5Hz.

Além disto, a taxa de saída do sensor foi configurada em seu valor máximo, de 1000 amostras por segundo. A comunicação se deu por meio de uma porta USB 2.0, o que torna o parâmetro de *Baud Rate* irrelevante.

No caso em que os magnetômetros são utilizados e há distúrbios no campo magnético

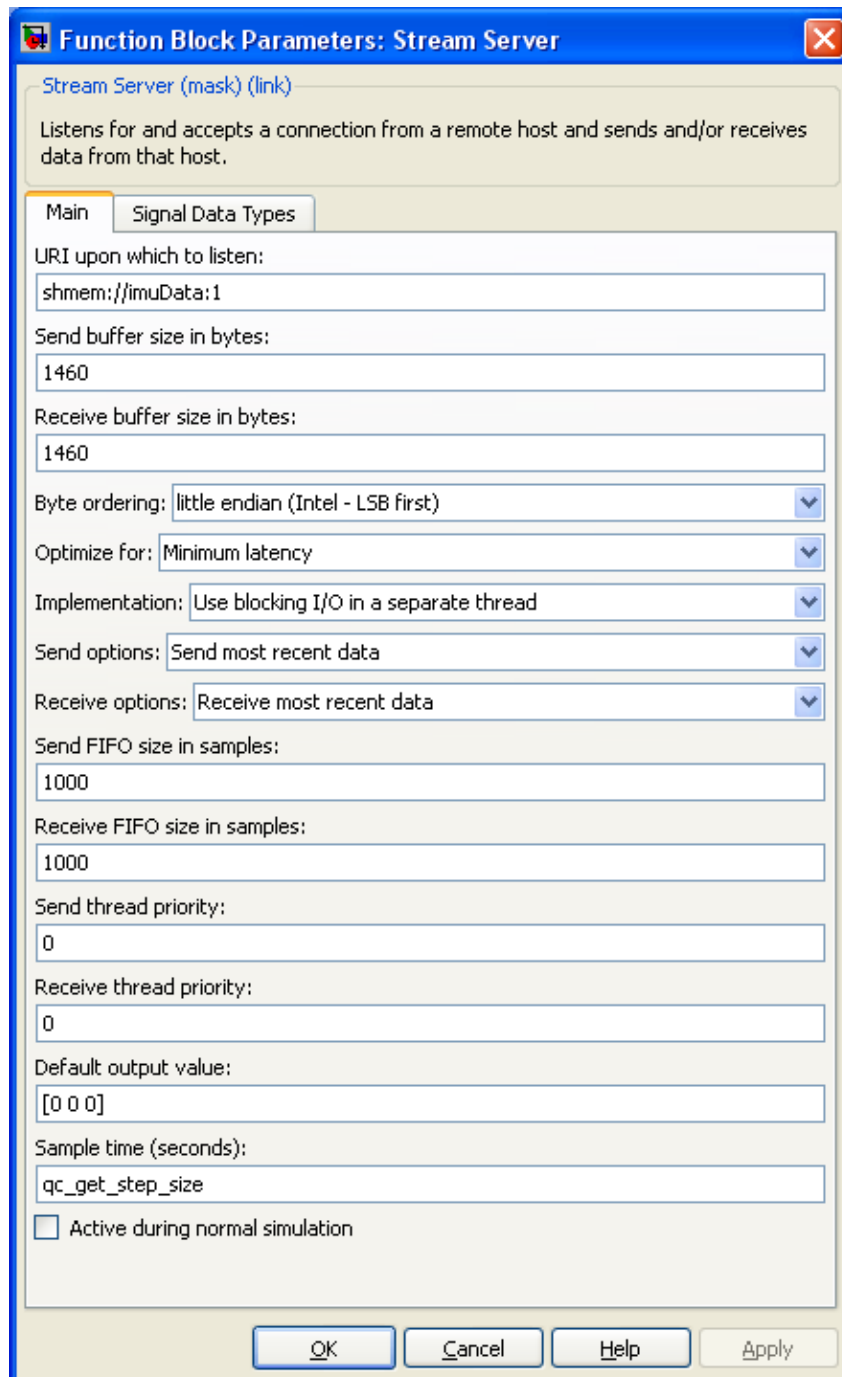


Figura 4.9: Configurações do bloco de comunicação do *Simulink/QUARC*

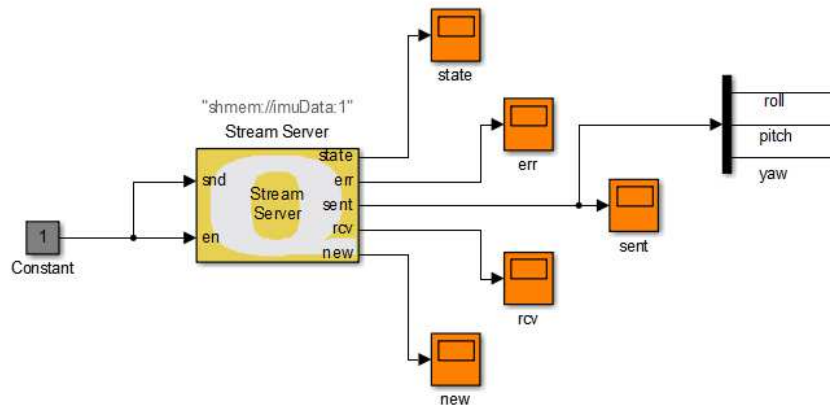


Figura 4.10: Bloco de servidor do *Simulink/QUARC* para a comunicação por memória compartilhada

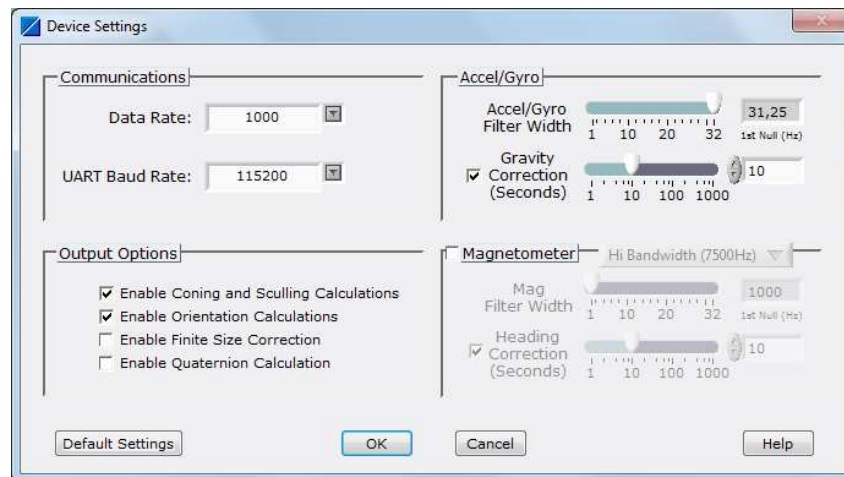


Figura 4.11: Configuração do sensor inercial por meio do software *3DM-GX3 Monitor*

local, o software *3DM-GX3 Iron Calibration* (figura 4.12) é fornecido para efetuar uma calibração do tipo *Hard* ou *Soft Iron Calibration*.

4.2 Controle do Motor DC pela placa Q8

A placa Q8, figura 4.13, é constituída de 8 canais de saída analógicos, 8 canais de entrada analógicos, 8 entradas para *encoder*, 32 pinos digitais de entrada/saída, 2 timers e 2 saídas PWM.

A alimentação de um motor DC pode ser feita por meio de sinais de saída analógicos ou sinais PWM. Como estes são sinais de baixa potência, precisam ser amplificados de

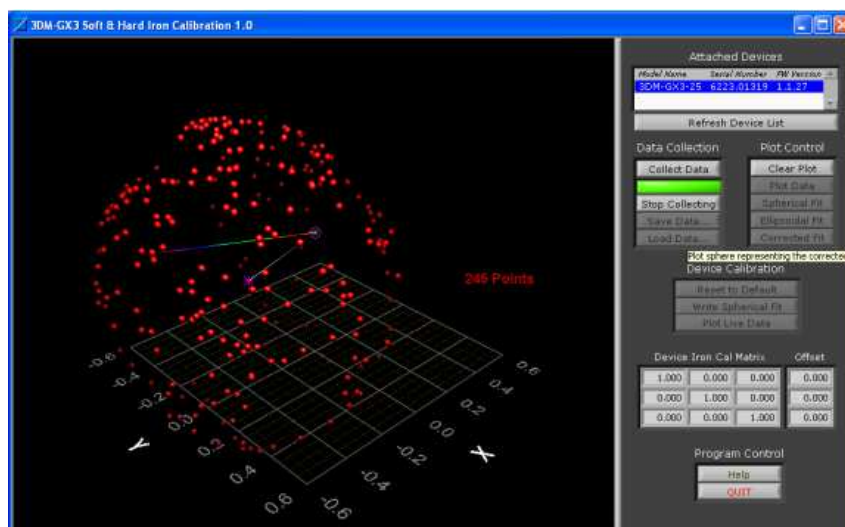


Figura 4.12: Calibração do sensor inercial por meio do software *3DM-GX3 Iron Calibration*

alguma forma. No caso do sinal analógico isto é feito por meio de um amplificador linear. Para isto foi usado um módulo com um amplificador de potência. Na falta de um segundo módulo, o outro motor foi alimentado por meio de um circuito de ponte H, acionado por um sinal PWM. Os dois casos são vistos nas próximas subseções.

4.2.1 Por meio de um canal analógico

A placa Q8 possui um conversor digital analógico (DAC) com uma resolução de 14 bits e a amplitude do sinal de saída pode variar entre $\pm 10V$, com uma taxa de variação de $2,5V/\mu s$. Antes de acionar o motor este sinal passa pelo *Universal Power Module 1503*, figura 4.14, que possui um amplificador linear de potência.

A interface entre o modelo do *Simulink* e a saída analógica da placa Q8 é feita por meio do bloco *HIL Write Analog* da biblioteca *QUARC Targets*. A configuração deste bloco, figura 4.15, necessita apenas da indicação dos canais utilizados. São 8 canais, com índices de 0 a 7. No caso apenas uma, de índice 3, é utilizada. Por segurança, uma saturação é aplicada na entrada deste bloco, de forma a garantir que o sinal de saída esteja dentro do limite aceito pela placa.

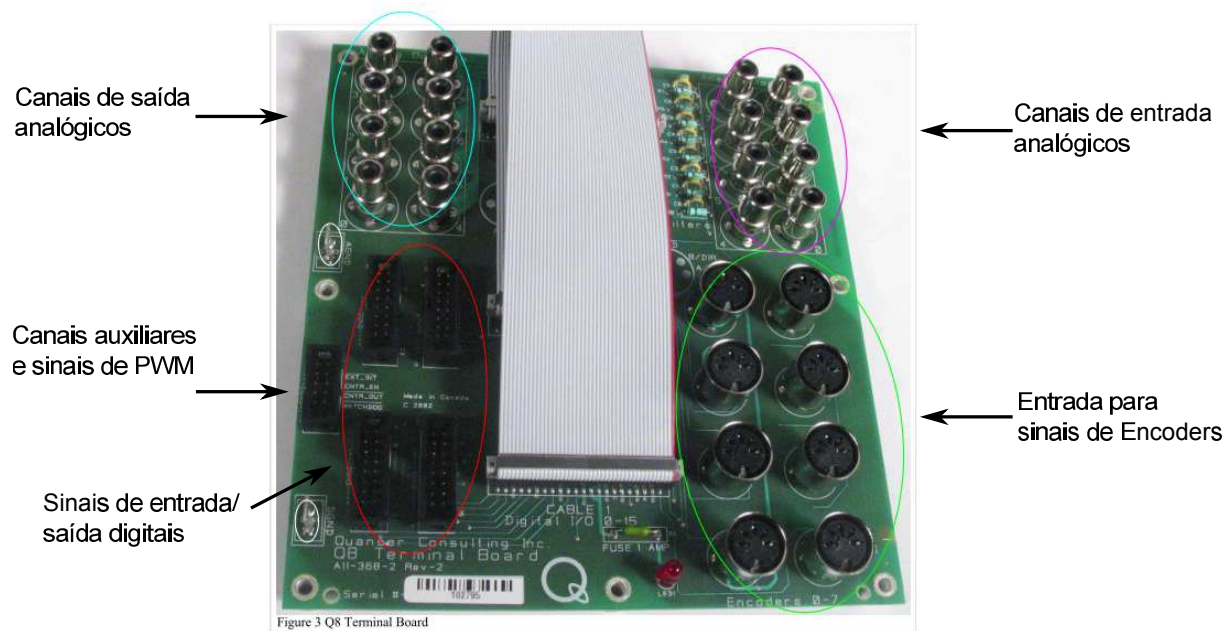


Figura 4.13: Placa de controle Q8

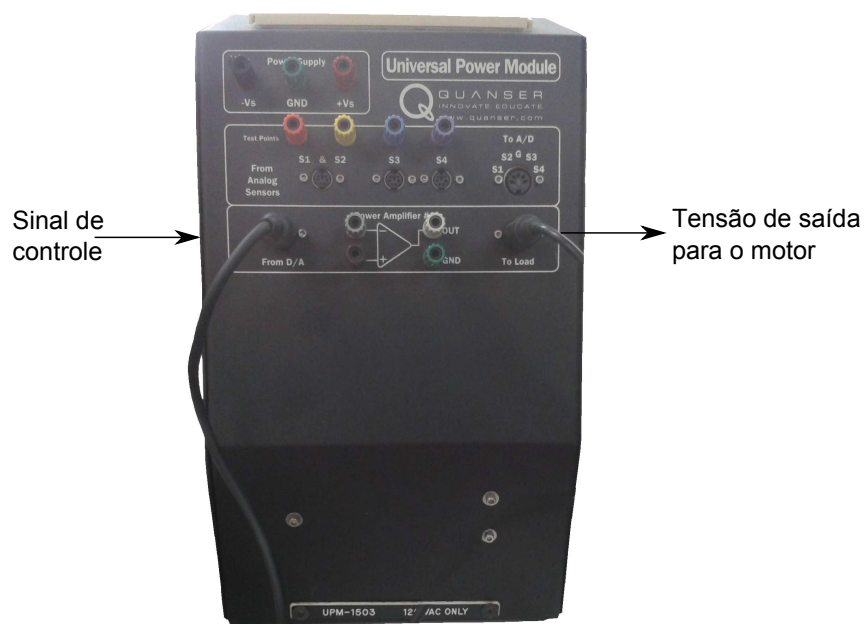


Figura 4.14: Universal Power Module 1503

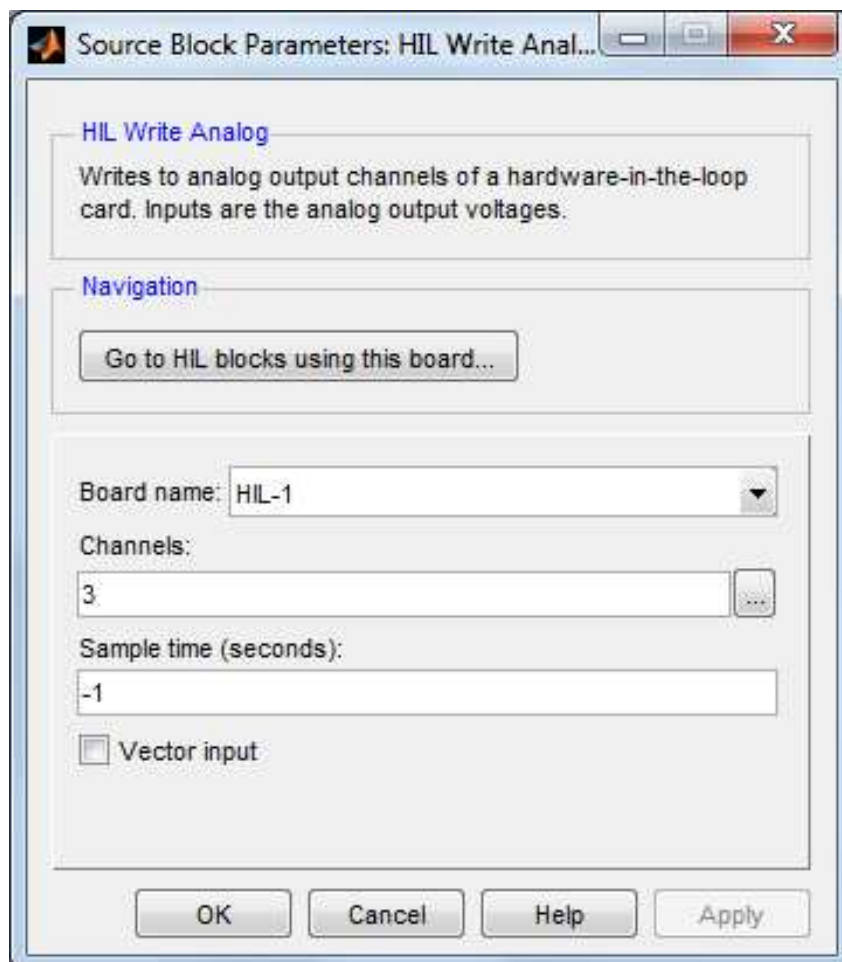


Figura 4.15: Bloco de saída analógica do *Simulink/Quarc*

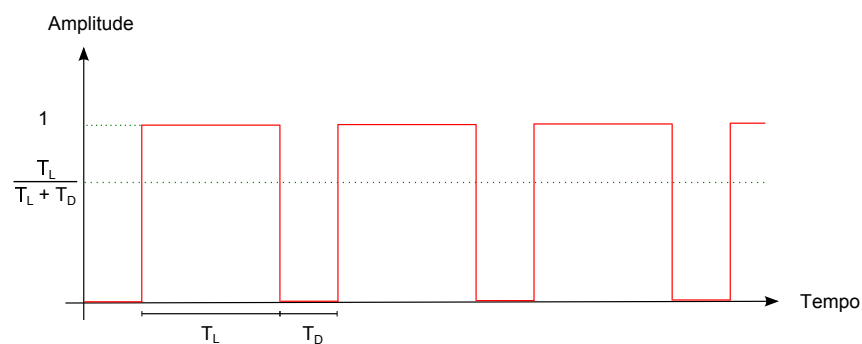


Figura 4.16: Sinal modulado por largura de pulso (PWM)

4.2.2 Por meio de um sinal PWM

O PWM é um sinal digital modulado por largura de pulso (figura 4.16). Enquanto o sinal analógico visto anteriormente é modulado diretamente por uma amplitude desejada, neste o sinal é definido por um ciclo de trabalho (*duty cycle*), que varia de 0 ($T_L = 0$) a 100% ($T_D = 0$). Este parâmetro define a porcentagem do período de tempo que o sinal está em nível lógico 1. São sinais de alta frequência, de forma que quando passam por um filtro-passa baixas, são vistos como um sinal de amplitude intermediária entre o valor lógico 1 e 0, igual a $T_L / (T_D + T_L)$.

Por se tratar de um sinal de baixa potência, o acionamento de um motor por meio de um PWM se dá com o intermédio de um circuito de potência. No caso se trata de uma ponte H, cuja estrutura está representada na figura 4.17. O sinal PWM fecha as chaves S_1 e S_4 para que a fonte V_{in} alimente o motor em um sentido, e S_2 e S_3 para o outro sentido. Uma configuração possível é aplicar o sinal PWM nas chaves S_1 e S_4 , e o inverso deste mesmo sinal nas chaves S_2 e S_3 . De forma que quando o ciclo de trabalho estiver em 50% a saída para o motor seja um sinal com média nula. Acima deste valor o motor é alimentado por uma tensão proporcional ao ciclo, no sentido definido por S_1 e S_4 . A placa utilizada, figura 4.17, é composta de um circuito integrado próprio para esta aplicação, chaves *MOSFETS* e de componentes de segurança do circuito, como diodos de barreira.

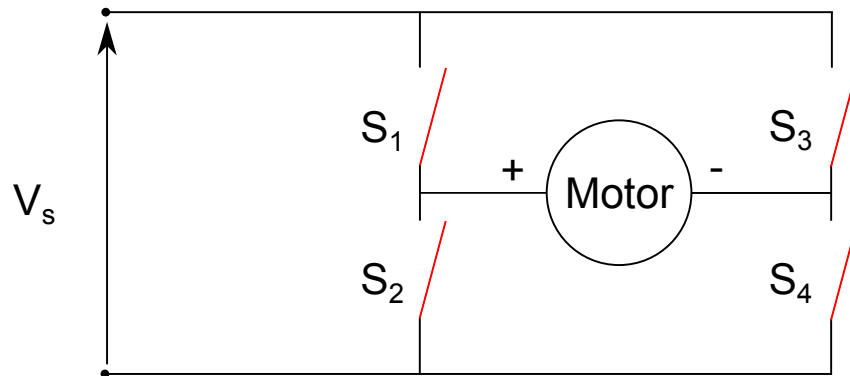


Figura 4.17: Estrutura de uma Ponte H

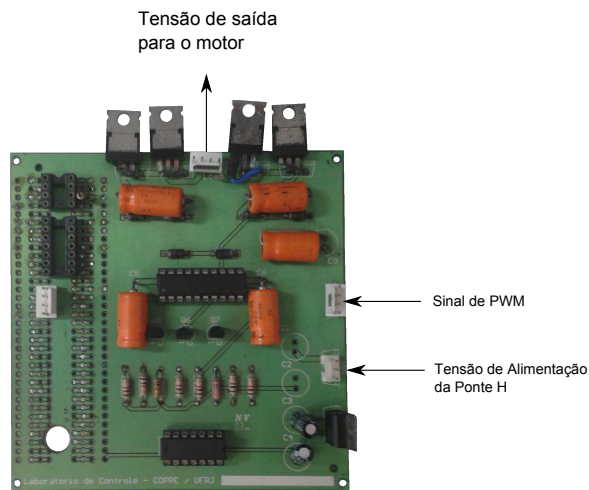


Figura 4.18: Foto do circuito de Ponte H utilizado

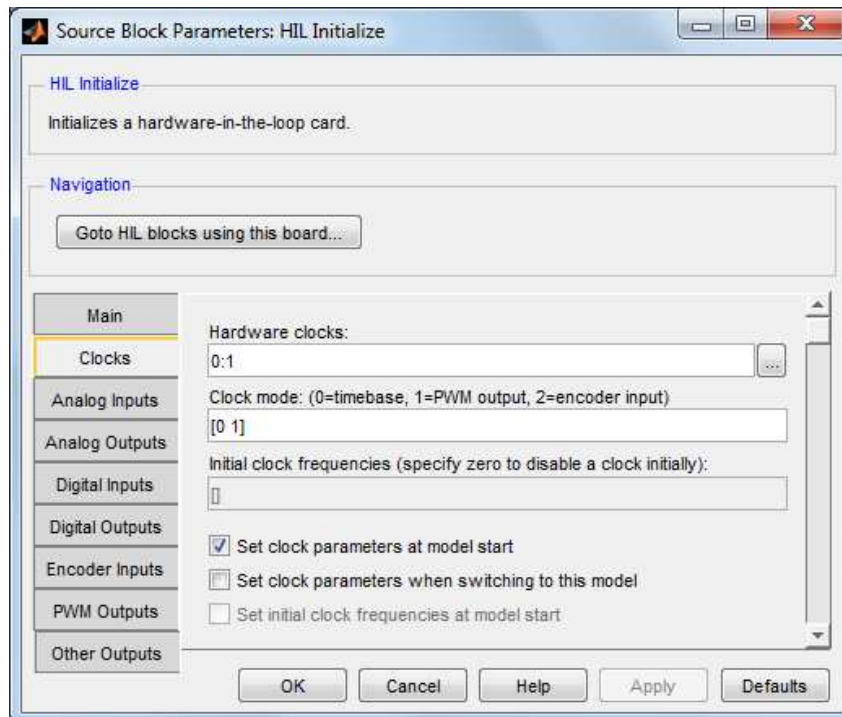


Figura 4.19: Configuração uma saída de *clock* da placa Q8 para um sinal PWM

Configuração do PWM no *QUARC*

Para utilizar o sinal PWM disponível na placa Q8, é preciso configurar uma das saídas que por padrão fornecem o *clock* do *hardware*. Isto é feito por meio do bloco *HIL Initialize* da biblioteca *QUARC Target*, conforme visto na figura 4.19.

A configuração do sinal PWM é vista no mesmo bloco, na aba *PWM Outputs*. A placa Q8 permite sinais PWM com frequências de até 16,7MHz. No entanto, quanto maior a frequência, pior é a resolução do ciclo de trabalho do sinal. No caso a frequência foi arbitrada em 65,104kHz, o que permite definir a largura do ciclo de trabalho com uma resolução de 8 bits.

Além de selecionar a frequência é possível escolher o modo de operação. O modo 0 é o que foi descrito nesta seção e que também é o mais comum. Neste caso o valor de entrada pode variar de 0.0 (0% de ciclo de trabalho) até 1.0 (100% de ciclo de trabalho).

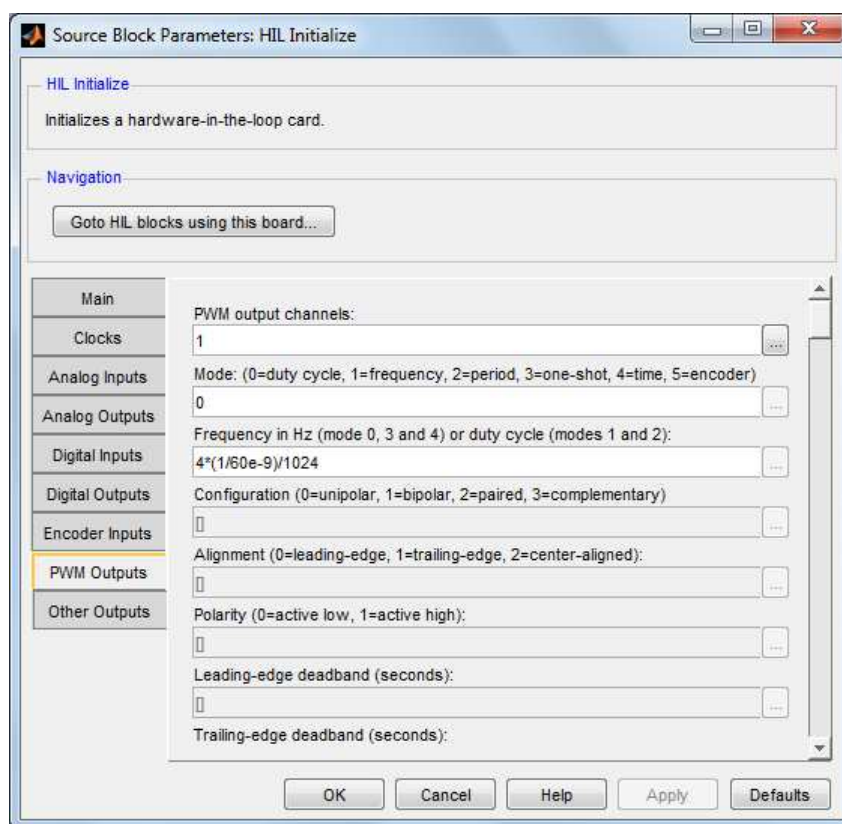


Figura 4.20: Configuração do sinal de PWM da placa Q8

4.2.3 Medida da orientação dos motores

Para o caso em que o sensor inercial é fixado no corpo do veículo, é necessário medir a posição angular do eixo dos motores.

O motor é acoplado a um *encoder* incremental óptico *Avago HEDM-5500J14*, ver apêndice A.2. O sensor consiste basicamente de um emissor de luz, no caso um LED, e um circuito de fotodiodos. Entre o LED e o circuito há uma placa dentada que é fixada no eixo interno do motor. Tal sensor, com a rotação do motor, gera duas ondas quadradas, canal A e canal B com uma defasagem de $\pm 90^\circ$, com 1024 pulsos para cada volta do rotor. Uma mudança em um estado indica uma rotação e o sentido da mesma é indicada pela defasagem entre os dois canais.

Para a leitura do *encoder* incremental é necessário um sistema que leia os canais do *encoder* e atualize uma variável contadora, que indicará a posição angular do eixo do motor em relação a sua posição inicial. Três tipos de codificação são possíveis: 1x, 2x e 4x. Na primeira codificação a medida do canal B é feita após detectar uma transição de subida no canal A. Na segunda qualquer transição de estado no canal A implica em uma medição do canal B. E na codificação 4x, qualquer transição de estado em qualquer um dos canais leva a uma medição.

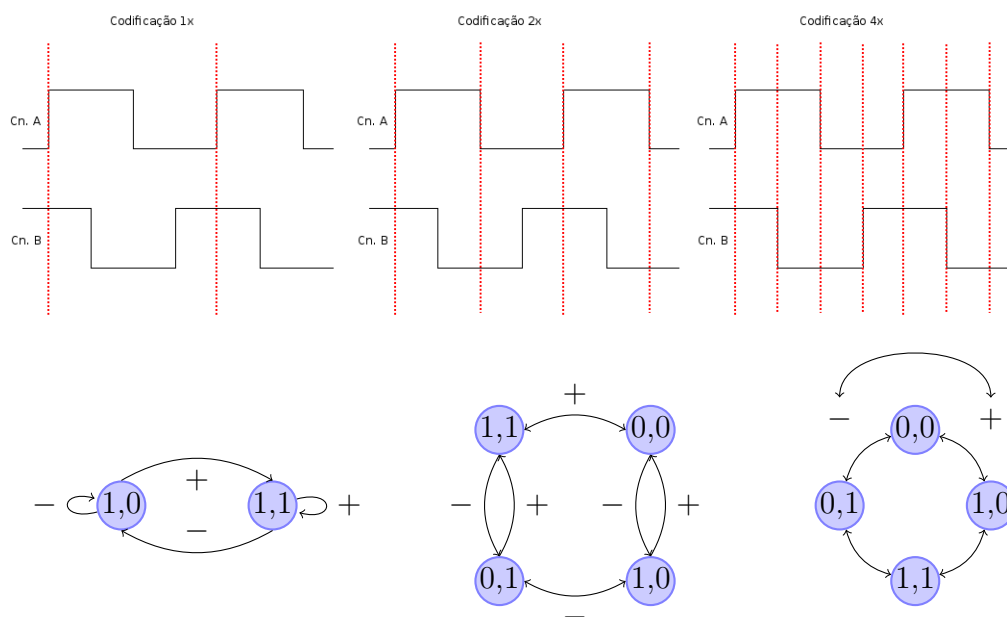


Figura 4.21: Diagrama de estados para a decodificação do encoder

O tipo de codificação implica na resolução da medição obtida com conjunto *encoder/decoder*, sendo a codificação 4x a de menor resolução possível, ou seja, a que permite medir a menor variação possível de posição. No caso, com tal codificação e com um conjunto de redução 43 : 1, temos 43 * 1024 pulsos para cada volta do eixo externo do motor, o que dá uma resolução de:

$$\frac{360^\circ}{43 * 4 * 1024} \approx 0,002^\circ \approx 7,36''$$

e a frequência máxima dos sinais gerados de $\omega_{max} * \frac{1024}{360^\circ} \approx 144\text{kHz}$. Em que ω_{max} é a velocidade angular máxima do eixo interno do motor.

Leitura dos encoders com placa Q8

A placa Q8 possui 8 entradas para leitura de *encoders*, que são computados em contadores de 24 bits. As possíveis configurações são mostradas na figura 4.22. Os três tipos de quadraturas explicados anteriormente são aceitos. Neste experimento foi optado o uso da quadratura 4x. A placa aplica um filtro ajustável aos canais A e B, sua frequência é definida por $1/(120e - 9 * N)$, com N variando de 1 até 255. Desta forma a frequência pode variar de 33kHz até 8,3MHz.

A saída do bloco de leitura do *encoder* é dada como um valor inteiro, armazenado no *counter*. É preciso multiplicá-lo por um fator, que depende da redução do motor e da quadratura usada, para convertê-lo em uma medida física.

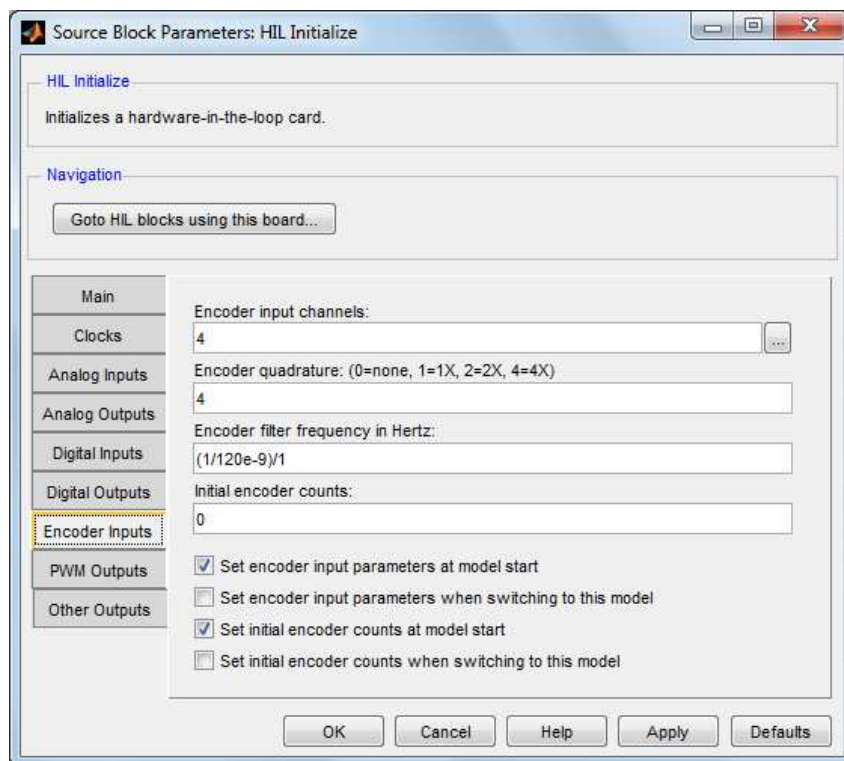


Figura 4.22: Configuração do bloco de leitura de *encoders* da *QUARC Target*

Capítulo 5

Resultados

Este capítulo apresenta os resultados dos experimentos realizados, com o controle proposto no capítulo 3 e com os componentes descritos no capítulo 4. Aqui apenas o caso em que o sensor inercial é fixado em cima da ISP é considerado. O motor da base (*Pan*) foi acionado por meio de um sinal de tensão analógico e um amplificador de potência, enquanto o motor de cima (*Tilt*) foi acionado por um sinal PWM e um *driver* de ponte H. O sistema de controle trabalhou com uma frequência de 1kHz, a mesma do sensor inercial. Os experimentos feitos consideraram a estrutura em que o sensor inercial é fixado em cima da ISP, ao lado da câmera. A plataforma de madeira, vista na imagem 4.1, foi utilizada para simular o movimento do ROV. Os resultados são analisados de duas formas distintas, por meio dos dados do sensor inercial, na seção 5.1, e por meio do vídeo captado pela câmera, na seção 5.2.

5.1 Resultados com base em dados do sensor inercial

O primeiro teste foi feito com o controle projetado de forma que a malha fechada do sistema rejeitasse distúrbios de saída com uma frequência de até 5Hz. A referência de orientação para a LOS é nula, de forma que o objetivo é mantê-la constante em seu valor inicial. Neste primeiro experimento não foi aplicado nenhum movimento na plataforma de madeira. O erro, bem como do sinal de controle, para os dois eixos são exibidos nas figuras 5.1 e 5.2.

É possível ver que o motor de base ficou suscetível a oscilações de alta frequência não

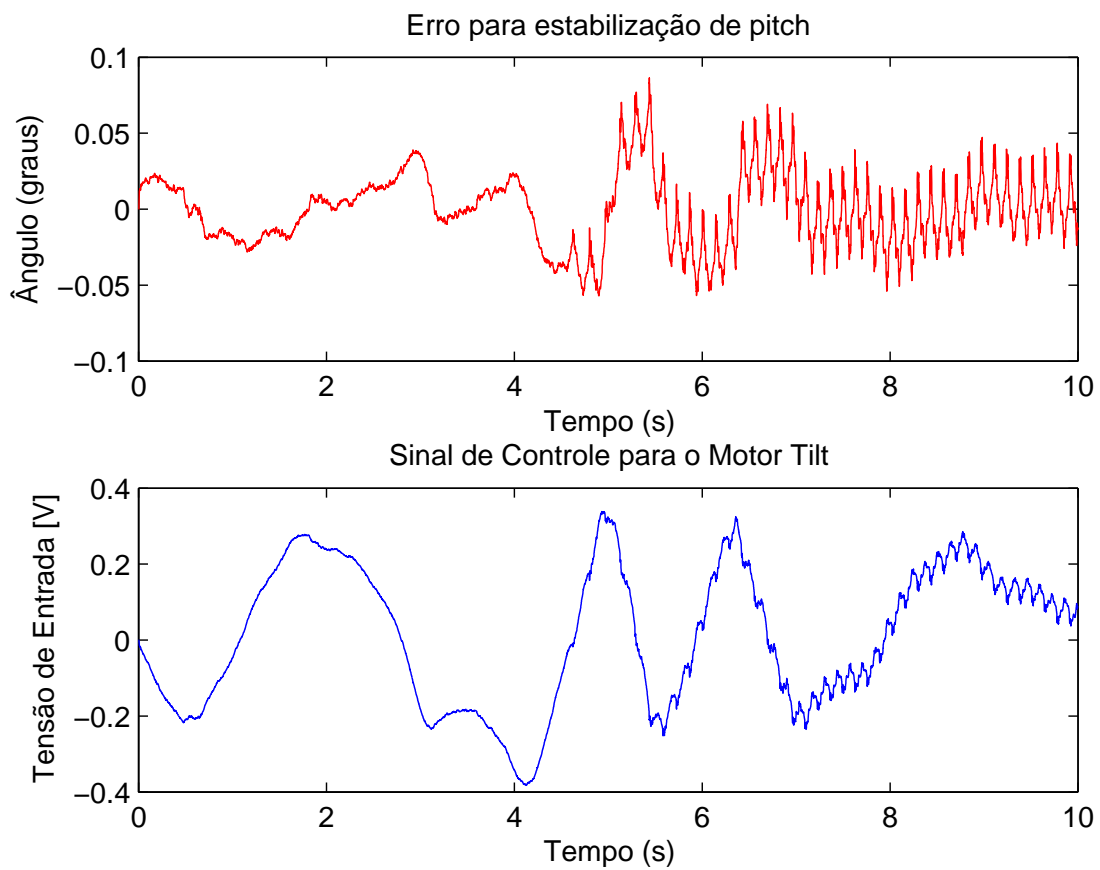


Figura 5.1: Leitura do sensor no canal de Caturro (*pitch*) e sinal de controle enviado para o motor de *Tilt* no primeiro experimento, sem movimento na plataforma

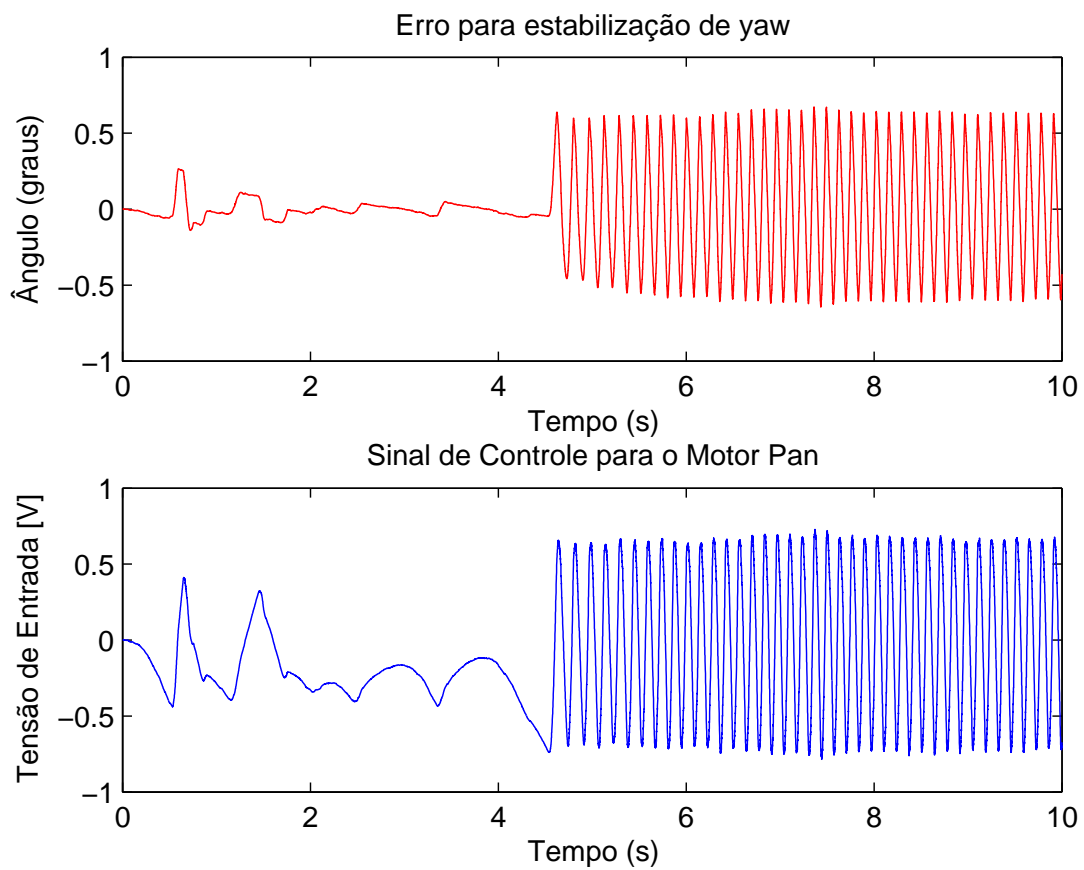


Figura 5.2: Leitura do sensor no canal de Guinada (*yaw*) e sinal de controle enviado para o motor de *Pan* no primeiro experimento, sem movimento na plataforma

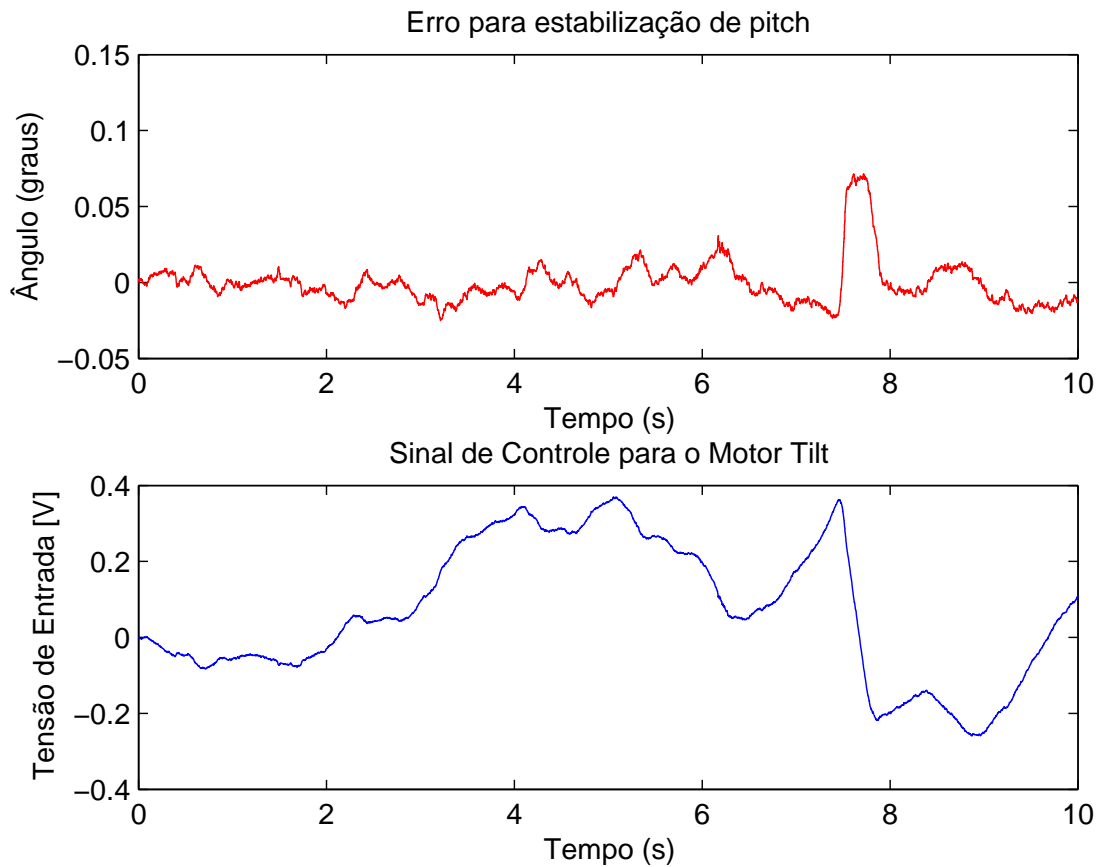


Figura 5.3: Leitura do sensor no canal de Caturro (*pitch*) e sinal de controle enviado para o motor de *Tilt* no segundo experimento, sem movimento da plataforma

previstas. Isto leva a crer que foi causado por alguma simplificação adotada. Isto é, que o sistema pode ser tratado de forma desacoplada, que os motores são ideais, a estrutura não apresenta nenhum tipo de folga ou não-linearidade e os sensores são ideais.

Um segundo teste foi feito nas mesmas condições, reduzindo o ganho de controle para a metade. De forma que agora o circuito de controle deve rejeitar distúrbios de saída compreendidos em uma banda de aproximadamente 2,5Hz. O resultado está exposto nas figuras 5.3 e 5.4. Neste caso a estabilização do ângulo de caturro obteve um erro absoluto médio de $0,015^\circ$, ou aproximadamente 0,25 miliradianos. O resultado da estabilização do ângulo de guinada foi inferior, obtendo um erro absoluto médio de $0,068^\circ$, ou aproximadamente 1,2 miliradianos. A diminuição no ganho foi, portanto, suficiente para eliminar as oscilações indesejadas.

O último teste foi feito movimentando a plataforma de base. O erro absoluto mé-

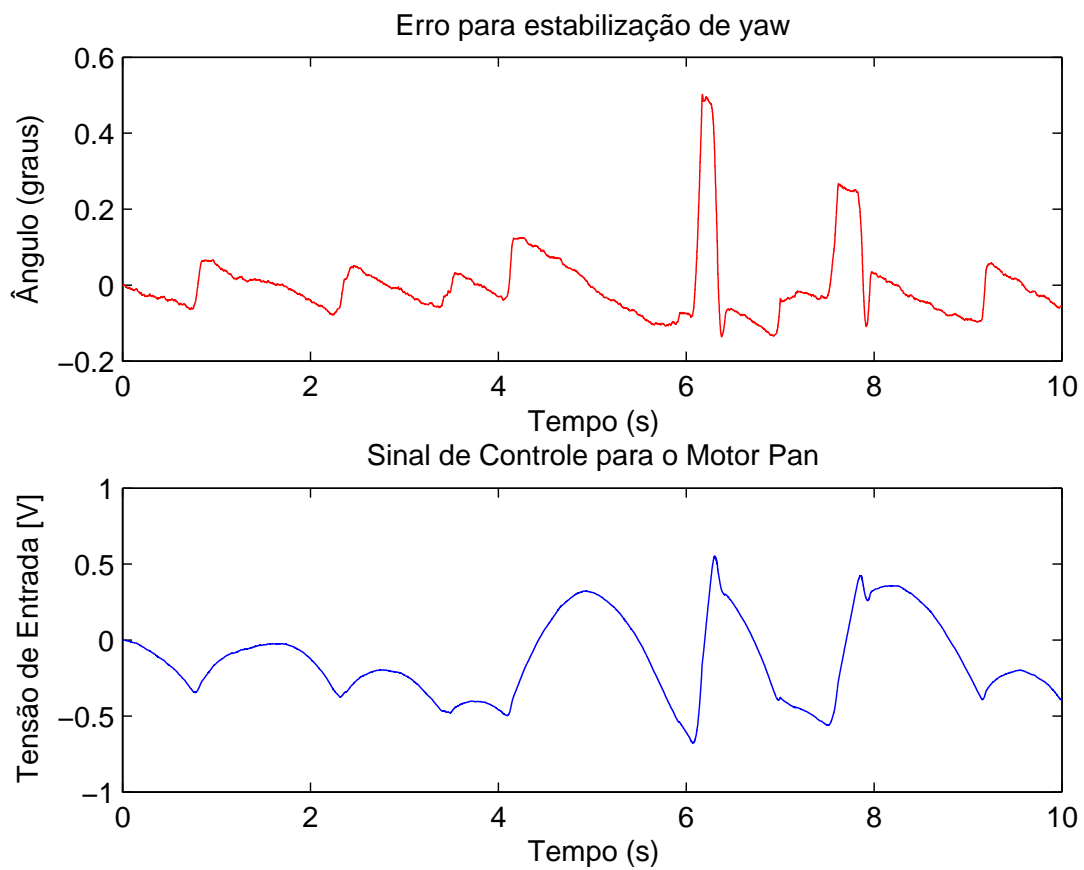


Figura 5.4: Leitura do sensor no canal de Guinada (*yaw*) e sinal de controle enviado para o motor de *Pan* no segundo experimento, sem movimento da plataforma

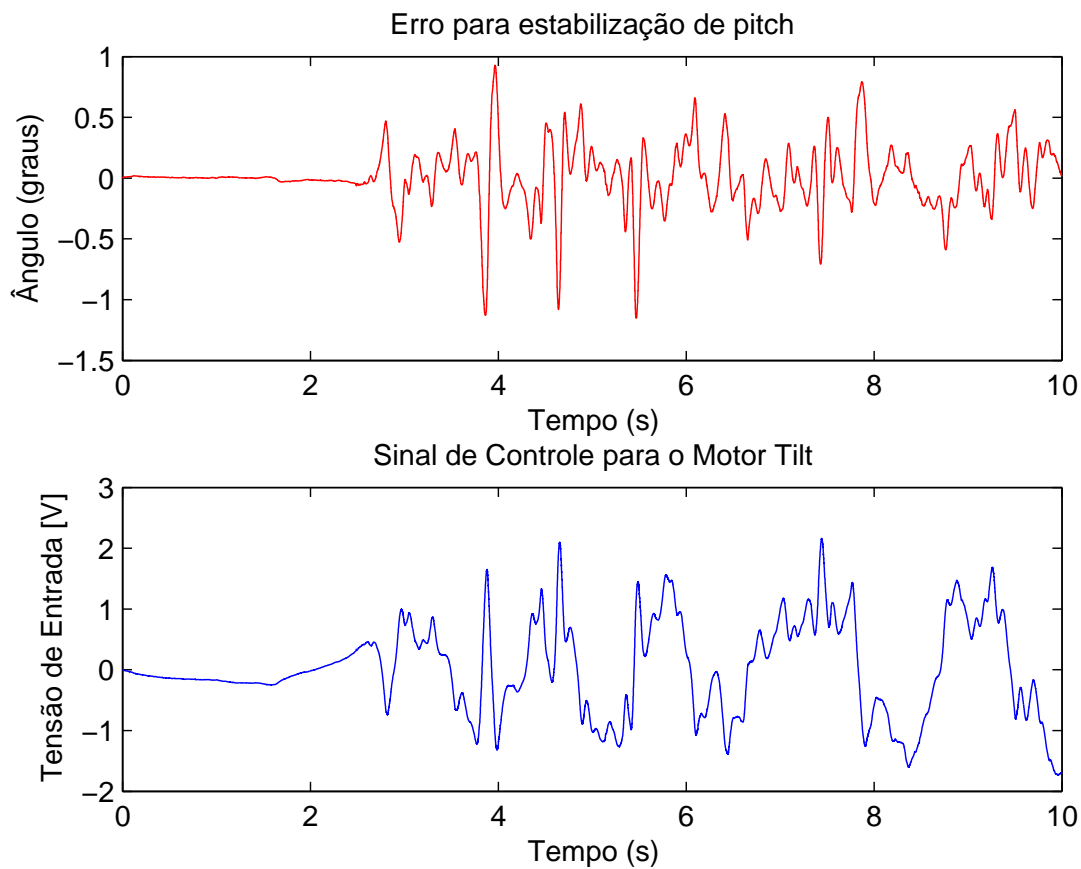


Figura 5.5: Leitura do sensor no canal de Caturro (*pitch*) e sinal de controle enviado para o motor de *Tilt* no terceiro experimento, com movimento da plataforma

Para a estabilização do ângulo de caturro foi de $0,16^\circ$, ou de aproximadamente 2,9 miliradianos. Para o ângulo de guinada, o erro foi de $0,37^\circ$, ou aproximadamente 6,4 miliradianos.

5.2 Resultados com base em imagens da câmera

A análise do sistema apenas com base nas medidas do sensor inercial não levam em conta os erros deste. Um outro teste para comparação foi feito levando em conta as imagens capturadas pela câmera e um alvo fixado em uma tela. Um algoritmo de rastreamento visual foi aplicado para revelar a trajetória desse alvo. Ainda que no caso ideal de uma estabilização perfeita o alvo se movimenta na imagem captada, uma vez que a ISP pode compensar apenas dois dos seis graus de liberdade do veículo, o movimento do alvo é bem

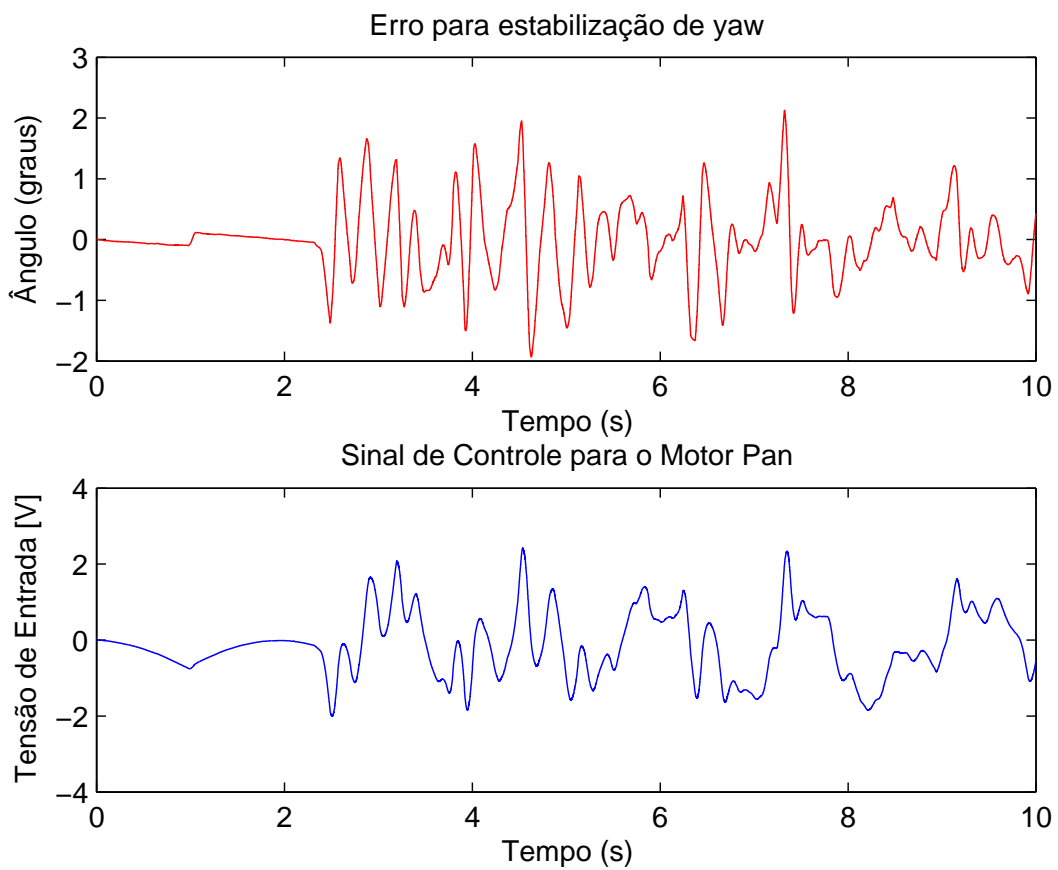


Figura 5.6: Leitura do sensor no canal de Guinada (*yaw*) e sinal de controle enviado para o motor de *Pan* no terceiro experimento, com movimento da plataforma

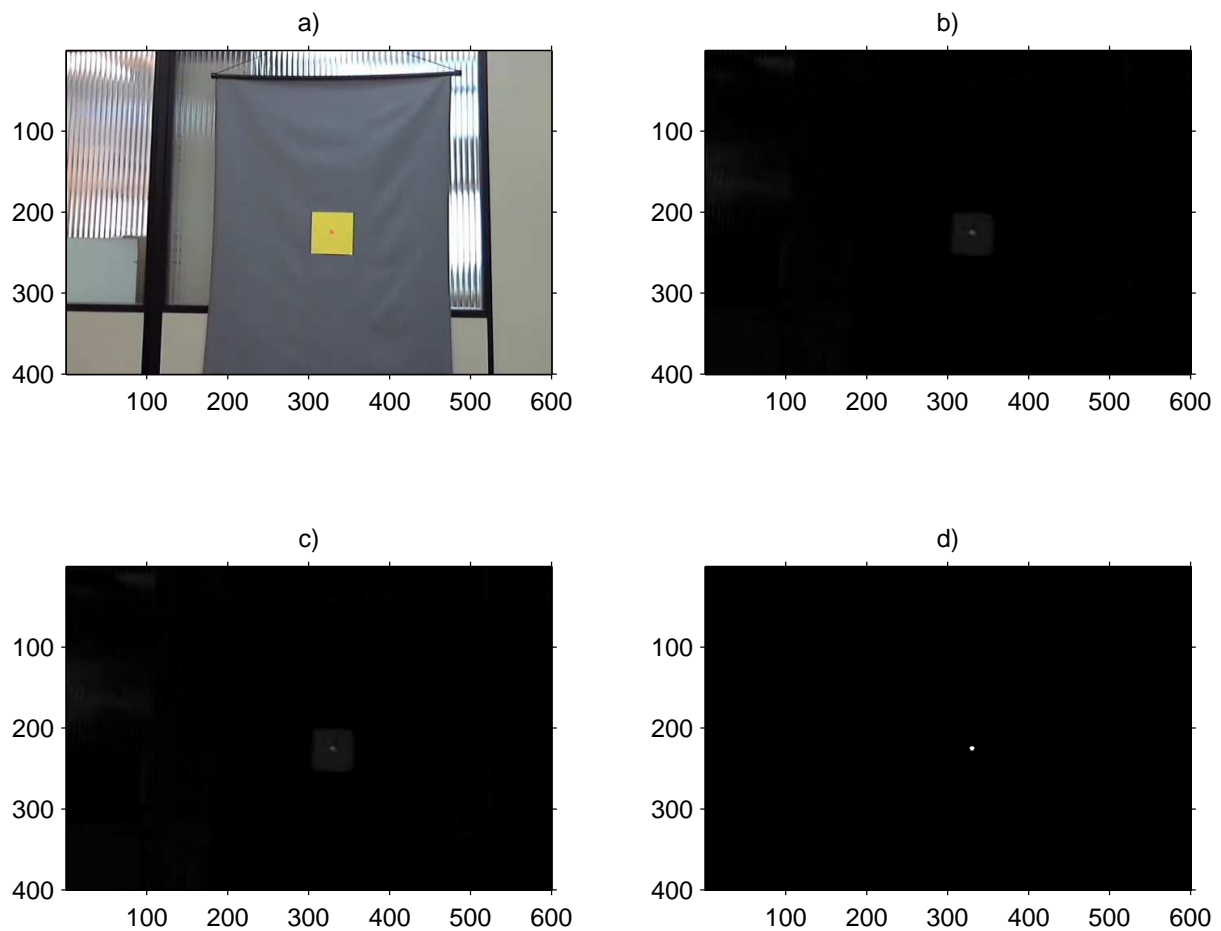


Figura 5.7: Processamento da imagem para o rastreo de um alvo: a) Imagem original b) Extração do Canal Vermelho c) Filtragem mediana d) Conversão em imagem binária

reduzido, como será visto a seguir.

Rastreo visual

O rastreo foi feito com base na *Image Processing Toolbox* do *Matlab* e considerando um alvo de cor predominante vermelha em um ambiente em que objetos desta cor não estão presentes, como foi o caso do teste realizado. O código completo do programa está no apêndice B.2, mas pode ser resumido por basicamente três passos que serão descritos a seguir.

Primeiramente uma imagem no espaço de cor RGB é representada por uma matriz com $W \times H \times 3$ elementos, em que W é a largura em *pixels* e H a altura. A dimensão restante representa os três canais de cor RGB: vermelho, verde e azul, respectivamente.

O elemento a_{ijk} é o valor da cor k do *pixel* localizado na linha i e coluna j . É um dado armazenado em 8 bits, que pode variar de 0, quando a cor k não está presente no *pixel*, a 255, quando há intensidade máxima dessa cor no *pixel*.

A imagem da figura 5.7b), é obtida da imagem original em 5.7a), por meio de uma subtração entre o canal vermelho da imagem e a imagem original convertida em uma escala de cinza. A operação de subtração com imagem não permite valores fora da faixa $[0, 255]$. Portanto, valores negativos são truncados em 0 e valores maiores que 255 são truncados em 255.

```
1 frameRed = imsubtract(frame_i(:,:,1), rgb2gray(frame_i));
```

Para atenuar ruído do tipo sal e pimenta (*Salt-and-Pepper noise*), comum em imagens e causados por *pixels* corrompidos na transmissão de dados da câmera, um filtro de suavização é aplicado. A função *medfilt2*, usando os parâmetros de padrão, faz com que o valor de cada elemento de uma matriz seja substituído pela média dos elementos de sua vizinhança. A saída de tal função é a imagem da figura 5.7c).

```
1 frameRed = medfilt2(frameRed);
```

Em seguida a imagem em escala de cinza é convertida para uma imagem binária com base em um parâmetro de limiar. Se na escala de cinza, que vai de 0 a 1, o *pixel* possui um valor menor que 0.2, ou seja mais próximo do preto, este é substituído por um *pixel* de valor 0, que representa a cor preta. Do contrário é substituído por um *pixel* de valor 1, que representa a cor branca. O resultado é visto na figura 5.7 d).

```
1 frameRed = im2bw(frameRed, 0.2);
```

Aplicando o algoritmo para cada quadro do vídeo e sobrepondo os resultados, podemos visualizar o rastro da trajetória do alvo. A figura 5.8 mostra o resultado para o vídeo captado, em a) sem o sistema de estabilização da linha de visada e em b) com a estabilização.

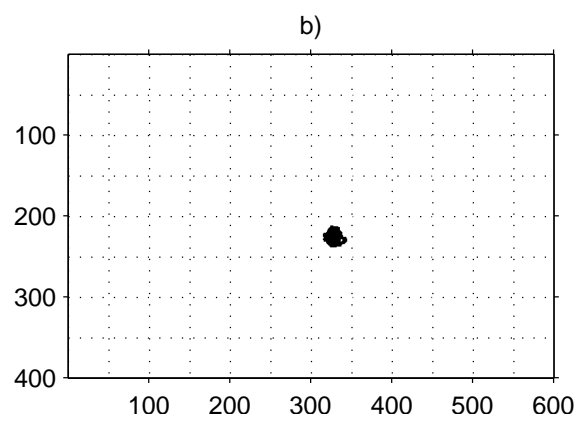
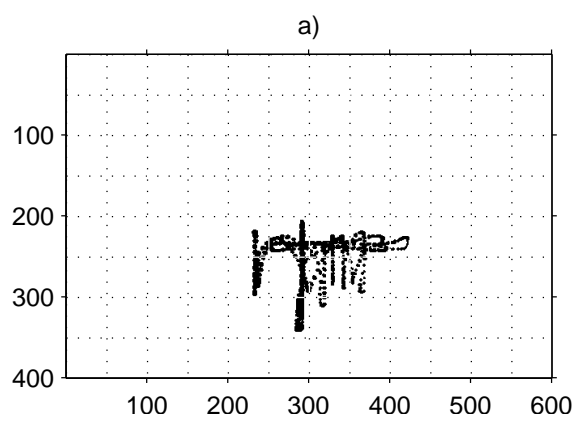


Figura 5.8: a) Rastreo visual de um alvo com estabilização da LOS b) sem estabilização

Capítulo 6

Conclusão e Trabalhos Futuros

Os resultados mostram que a ISP projetada serviu para estabilizar a direção da linha de visada com uma precisão na ordem de miliradianos. A comparação entre os vídeos capturados, com e sem a estabilização, evidenciam que o sistema conseguiu manter um alvo dentro de seu campo de visão, apesar de movimentos da base.

Ainda que os testes tenham sido realizados sem um movimento controlado da plataforma de base, que pudesse simular a dinâmica do ROV LUMA, os resultados adquiridos comprovam inicialmente o funcionamento do mecanismo. O *drift* causado pelo não uso dos magnetômetros, em torno de $0,5^\circ/min$, é lento o suficiente para ser corrigido manualmente por um operador.

Em geral ISPs utilizam sensores para medir a orientação dos eixos do gimbal, *encoders* por exemplo. Uma das estratégias consideradas, em que o sensor inercial é fixado na plataforma estabilizada, comprovou uma forma de ISP sem a necessidade de tais sensores. Isto possibilita uma redução nas dimensões do gimbal. Além disso, este trabalho abre outros caminhos que podem ser explorados em projetos futuros. Alguns são comentados a seguir:

- Uma análise simplista da dinâmica da ISP foi feita, considerando o sistema como dois sub-sistemas desacoplados, sem nenhum tipo de efeito giroscópico ou acoplamento cinemático. Isto acarretará numa piora de desempenho. Uma análise levando em conta esses fatores pode ser desenvolvida.
- O mesmo pode ser dito quanto a estrutura do controle. Uma técnica mais avançada,

como controle adaptativo, por exemplo, pode ser empregado.

- Geralmente ISPs são empregadas como componentes de um sistema maior, auxiliando no rastreamento visual de um alvo, por exemplo. Este trabalho não contemplou a tarefa de rastreamento e focou apenas na estabilização da linha de visada da câmera. Isto pode ser desenvolvida aplicando algoritmos comumente usados, como lista [9]: *contrast edge*, *contrast centroid*, *histogram projection*, *adaptive gate centroid (AGCT)*, etc.
- Como o sensor inercial foi fixado na ISP, os experimentos realizados não mediram a orientação da plataforma de teste. Um segundo sensor inercial pode ser usado. Isto deve ser feito até para obter algum parâmetro de comparação para diferentes controles empregados.
- Para cargas de maior dimensão ou peso, como a câmera de alta definição do ROV LUMA, uma estabilização indireta, por meio do uso de elementos ópticos pode ser considerada.
- O sistema foi desenvolvido sem considerar a aplicação direta ao sistema do ROV LUMA. De forma que é desejável o desenvolvimento de uma interface entre a ISP e o PC/104, que compõe a eletrônica embarcada do LUMA.
- É preciso adaptar a ISP para ambientes sub-aquáticos, seja por meio de um domo isolante, ou por meio da marinização dos motores.
- Não foi implementado nenhum tipo de filtro à parte para eliminar ruídos de medida do sensor inercial. Foi considerado apenas o filtro ajustável já embutido no sensor. Seria também possível obter os dados dos sensores (acelerômetros, giroscópios e magnetômetros) em forma bruta e aplicar algum tipo de filtragem ótima para a fusão dos dados.
- O controle foi projetado com um modelo contínuo, considerando que a frequência de amostragem fosse grande o suficiente. Uma análise no domínio discreto deve ser feita para garantir um melhor funcionamento do sistema.

Referências Bibliográficas

- [1] Configuring a communications resource. Microsoft. [Online]. Available: [http://msdn.microsoft.com/en-us/library/aa363201\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa363201(v=vs.85).aspx)
- [2] K. J. Aström and R. M. Murray, *Feedback Systems - An Introduction for Scientists and Engineers*. Princeton University Press, 2008.
- [3] A. Battistel, “Rastreamento visual e estabilização de plataformas inerciais usando apenas duas medidas de sensores giroscópicos,” Master’s thesis, COPPE/UFRJ, 2011.
- [4] C.-T. Chen, *Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods*. Oxford University Press, 2006.
- [5] W. Commons/Brosen. (2006) Ship movements on the wave. [Online]. Available: http://commons.wikimedia.org/wiki/File:Brosen_shipsmovemensonthewave.svg
- [6] W. Commons/NASA. (1997) Hubble space telescope in space. [Online]. Available: http://commons.wikimedia.org/wiki/File:Hubble_01.jpg
- [7] R. Costa, “Estabilização da linha de visada da câmera pan/tilt do robô submarino luma,” Master’s thesis, UFRJ/Escola Politécnica, 2010.
- [8] J. M. Hilkert, “Inertially stabilized platform technology,” *IEEE Control Systems Magazine*, vol. 28, no. 1, pp. 26–46, Fevereiro 2008.
- [9] M. K. Masten, “Inertially stabilized platforms for optical imaging systems,” *IEEE Control Systems Magazine*, vol. 28, no. 1, pp. 47–64, Fevereiro 2008.
- [10] R. McEwen, H. Thomas, D. Weber, and F. Psota, “Performance of an auv navigation system at arctic latitudes,” vol. 30, no. 2, pp. 443–454, Abril 2005.
- [11] *3DM-GX3 Data Communications Protocol*, MicroStrain, Inc., Abril 2010.
- [12] *3DM-GX3-25 Up and North Compensation*, MicroStrain, Inc., 2012.

- [13] T. Mitsutomi, "Characteristics and stabilization of an inertial platform." Downey, Calif.: Autonetics, Div. of North American Aviation, Inc., 1957.
- [14] W. C. Navy. (2005) Scaneagle nose camera closeup. [Online]. Available: http://commons.wikimedia.org/wiki/File:ScanEagle_nose_camera_closeup.jpg
- [15] W. C. Wang. (2008) Stabilized camera on beijing olimpics. [Online]. Available: http://commons.wikimedia.org/wiki/File:Stabilized_camera_on_Beijing_Olimpics.jpg

Apêndice A

Folhas de Dados

A.1 Motor DC: Faulhaber 2342S024CR

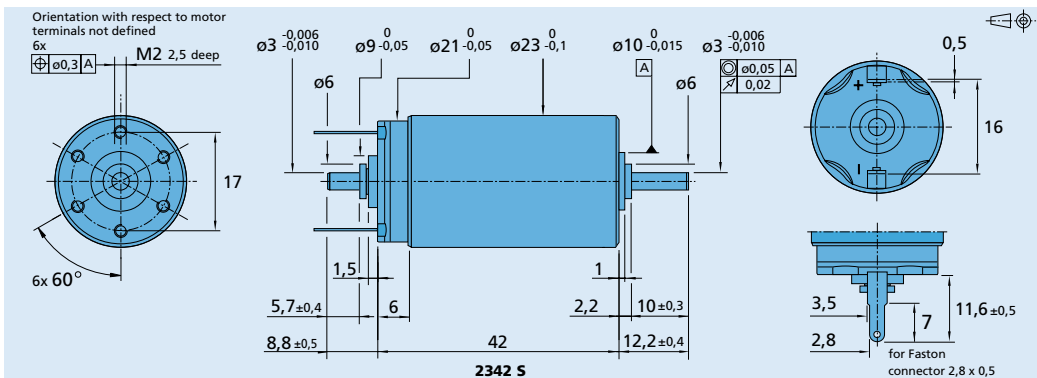
DC-Micromotors Graphite Commutation

16 mNm

For combination with
Gearheads:
22F, 23/1, 26A, 26/1, 22/7, 30/1, 38/3
Encoders:
IE2 – 16 ... 512, IE3 – 256, IE3 – 256 L, 5500, 5540

Series 2342 ... CR

| | 2342 S | 006 CR | 012 CR | 018 CR | 024 CR | 036 CR | 048 CR | |
|--|-----------------------------------|---------------------------------------|--------|--------|--------|--------|--------|------------------------|
| 1 Nominal voltage | U_N | 6 | 12 | 18 | 24 | 36 | 48 | Volt |
| 2 Terminal resistance | R | 0,40 | 1,90 | 4,10 | 7,10 | 15,9 | 31,20 | Ω |
| 3 Output power | $P_{2 \text{ max}}$ | 20,50 | 17,00 | 18,10 | 19,00 | 19,40 | 17,70 | W |
| 4 Efficiency | $\eta_{\text{ max}}$ | 81 | 80 | 81 | 81 | 81 | 81 | % |
| 5 No-load speed | n_0 | 9 000 | 8 100 | 8 000 | 8 500 | 8 100 | 8 000 | rpm |
| 6 No-load current (with shaft \varnothing 3,0 mm) | I_0 | 0,170 | 0,075 | 0,048 | 0,038 | 0,024 | 0,017 | A |
| 7 Stall torque | M_H | 87,2 | 80,0 | 86,5 | 85,4 | 91,4 | 84,4 | mNm |
| 8 Friction torque | M_f | 0,98 | 1,00 | 0,99 | 0,99 | 0,99 | 0,95 | mNm |
| 9 Speed constant | k_n | 1 650 | 713 | 462 | 366 | 231 | 170 | rpm/V |
| 10 Back-EMF constant | k_E | 0,604 | 1,400 | 2,160 | 2,730 | 4,340 | 5,870 | mV/rpm |
| 11 Torque constant | k_M | 5,77 | 13,40 | 20,70 | 26,10 | 41,40 | 56,10 | mNm/A |
| 12 Current constant | k_i | 0,173 | 0,075 | 0,048 | 0,038 | 0,024 | 0,018 | A/mNm |
| 13 Slope of n-M curve | $\Delta n/\Delta M$ | 103 | 101 | 92,5 | 99,5 | 88,6 | 94,8 | rpm/mNm |
| 14 Rotor inductance | L | 13,5 | 65 | 150 | 265 | 590 | 1 050 | μH |
| 15 Mechanical time constant | τ_m | 6 | 6 | 6 | 6 | 6 | 6 | ms |
| 16 Rotor inertia | J | 5,6 | 5,7 | 6,2 | 5,8 | 6,5 | 6,0 | gcm^2 |
| 17 Angular acceleration | $\alpha_{\text{ max}}$ | 160 | 140 | 140 | 150 | 140 | 140 | 10^3 rad/s^2 |
| 18 Thermal resistance | $R_{\text{th}1} / R_{\text{th}2}$ | 3 / 15 | | | | | | K/W |
| 19 Thermal time constant | τ_{w1} / τ_{w2} | 6,5 / 490 | | | | | | s |
| 20 Operating temperature range: | | | | | | | | |
| – motor | | – 30 ... + 100 | | | | | | $^{\circ}\text{C}$ |
| – rotor, max. permissible | | + 125 | | | | | | $^{\circ}\text{C}$ |
| 21 Shaft bearings | | ball bearings, preloaded | | | | | | |
| 22 Shaft load max.: | | | | | | | | |
| – with shaft diameter | | 3,0 | | | | | | mm |
| – radial at 3 000 rpm (3 mm from bearing) | | 20 | | | | | | N |
| – axial at 3 000 rpm | | 2 | | | | | | N |
| – axial at standstill | | 20 | | | | | | N |
| 23 Shaft play: | | | | | | | | |
| – radial | \leq | 0,015 | | | | | | mm |
| – axial | $=$ | 0 | | | | | | mm |
| 24 Housing material | | steel, black coated | | | | | | |
| 25 Weight | | 88 | | | | | | g |
| 26 Direction of rotation | | clockwise, viewed from the front face | | | | | | |
| Recommended values - mathematically independent of each other | | | | | | | | |
| 27 Speed up to | $n_{e \text{ max}}$ | 7 000 | 7 000 | 7 000 | 7 000 | 7 000 | 7 000 | rpm |
| 28 Torque up to | $M_{e \text{ max}}$ | 16 | 16 | 16 | 16 | 16 | 16 | mNm |
| 29 Current up to (thermal limits) | $I_{e \text{ max}}$ | 2,700 | 1,400 | 0,950 | 0,720 | 0,480 | 0,350 | A |



For notes on technical data and lifetime performance refer to "Technical Information".
Edition 2009 Aug. 11

Specifications subject to change without notice

www.faulhaber.com

A.2 Encoder: Avago HEDM-5500J14

HEDM-55xx/560x & HEDS-55xx/56xx

Quick Assembly Two and Three Channel Optical Encoders



Data Sheet

HEDM-55xx/560x HEDS-550x/554x, HEDS-560x/564x



Description

The HEDS-5500/5540, HEDS-5600/5640, HEDM-5500/5540 and HEDM-5600 are high performance, low cost, two and three channel optical incremental encoders. These encoders emphasize high reliability, high resolution, and easy assembly.

Each encoder contains a lensed LED source, an integrated circuit with detectors and output circuitry, and a codewheel which rotates between the emitter and detector IC. The outputs of the HEDS-5500/5600 and HEDM-5500/5600 are two square waves in quadrature. The HEDS-5540/5640 and HEDM-5540 also have a third channel index output in addition to the two channel quadrature. This index output is a 90 electrical degree, high true index pulse which is generated once for each full rotation of the codewheel.

The HEDS series utilizes metal codewheels, while the HEDM series utilizes a film codewheel allowing for resolutions to 1024 CPR.

These encoders may be quickly and easily mounted to a motor. For larger diameter motors, the HEDM-5600, and HEDS-5600/5640 feature external mounting ears.

The quadrature signals and the index pulse are accessed through five 0.025 inch square pins located on 0.1 inch centers.

Standard resolutions between 96 and 1024 counts per revolution are presently available. Consult local Avago sales representatives for other resolutions.

Features

- Two channel quadrature output with optional index pulse
- Quick and easy assembly
- No signal adjustment required
- External mounting ears available
- Low cost
- Resolutions up to 1024 counts per revolution
- Small size -40°C to 100°C operating temperature
- TTL compatible
- Single 5 V supply

Applications

The HEDS-5500, 5540, 5600, 5640, and the HEDM-5500, 5540, 5600 provide motion detection at a low cost, making them ideal for high volume applications. Typical applications include printers, plotters, tape drives, positioning tables, and automatic handlers.

Note: Avago Technologies encoders are not recommended for use in safety critical applications. Eg. ABS braking systems, power steering, life support systems and critical care medical equipment. Please contact sales representative if more clarification is needed.

ESD WARNING: NORMAL HANDLING PRECAUTIONS SHOULD BE TAKEN TO AVOID STATIC DISCHARGE.

A.3 Sensor Inercial: MicroStrain GX3-25-OEM

3DM-GX3[®] -25-OEM

Miniature Attitude Heading Reference System

The **3DM-GX3[®] -25-OEM** is a high-performance, miniature Attitude Heading Reference System (AHRS), utilizing MEMS sensor technology. It combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm to provide static and dynamic orientation, and inertial measurements. Its form factor is ideally suited for OEM applications.



Features & Benefits

Easiest to Integrate

- smallest, lightest industrial OEM AHRS available
- simple integration supported by SDK and comprehensive API

Best in Class

- precise attitude estimations
- high-speed sample rate & flexible data outputs
- high performance under vibration

Cost Effective

- reduced cost and rapid time to market for customer's applications
- aggressive volume discount schedule

Applications

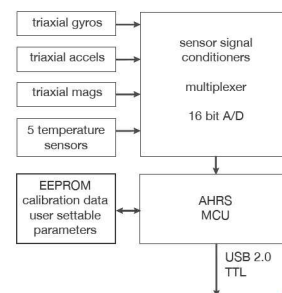
Accurate navigation and orientation under dynamic conditions such as:

- Inertial Aiding of GPS
- Unmanned Vehicle Navigation
- Platform Stabilization, Artificial Horizon
- Antenna and Camera Pointing
- Health and Usage Monitoring of Vehicles
- Reconnaissance, Surveillance, and Target Acquisition
- Robotic Control
- Personnel Tracking

System Overview

The **3DM-GX3[®] -25 OEM** offers a range of fully calibrated inertial measurements including acceleration, angular rate, magnetic field, deltaTheta and deltaVelocity vectors. It can also output computed orientation estimates including Euler angles (pitch, roll, and heading (yaw)), rotation matrix and quaternion. All quantities are fully temperature compensated and are mathematically aligned to an orthogonal coordinate system. The angular rate quantities are further corrected for g-sensitivity and scale factor non-linearity to third order. The **3DM-GX3[®] -25 OEM** architecture has been carefully designed to substantially eliminate common sources of error such as hysteresis induced by temperature changes and sensitivity to supply voltage variations. Gyro drift is eliminated in AHRS mode by referencing magnetic North and Earth's gravity and compensating for gyro bias. On-board coning and sculling compensation allows for use of lower data output rates while maintaining performance of a fast internal sampling rate.

The **3DM-GX3[®] -25 OEM** is initially sold as a starter kit consisting of an AHRS module, USB communication and power cable, software CD, user manual and quick start guide. The circuit board form-factor provides thru-holes for mounting on larger circuit assemblies and custom TTL communication and power cables can be user fabricated or purchased from the factory.



LORD MicroStrain[®]
SENSING SYSTEMS

3DM-GX3[®] -25-OEM Miniature Attitude Heading Reference System

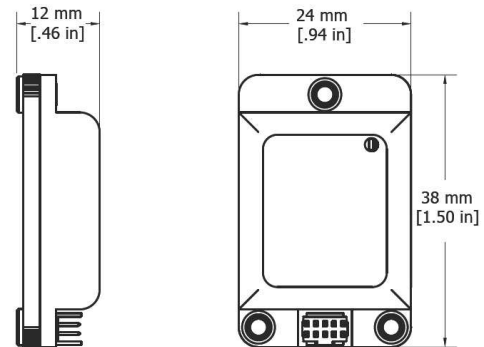
Specifications

AHRS Specifications

| Attitude and Heading | |
|--------------------------------|--|
| Attitude heading range | 360° about all 3 axes |
| Accelerometer range | ±5g standard |
| Gyroscope range | ±300°/sec standard |
| Static accuracy | ±0.5° pitch, roll, heading typical for static test conditions |
| Dynamic accuracy | ±2.0° pitch, roll, heading for dynamic (cyclic) test conditions and for arbitrary angles |
| Long term drift | eliminated by complimentary filter architecture |
| Repeatability | 0.2° |
| Resolution | <0.1° |
| Data output rate | up to 1000 Hz |
| Filtering | sensors sampled at 30 kHz, digitally filtered (user adjustable) and scaled into physical units; coning and sculling integrals computed at 1 kHz |
| Output modes | acceleration, angular rate, and magnetic field deltaTheta and deltaVelocity, Euler angles, quaternion, rotation matrix |
| General | |
| A/D resolution | 16 bits SAR oversampled to 17 bits |
| Interface options | USB 2.0 / TTL serial (3.3 volts) |
| Baud rate | 115,200 bps to 921,600 bps |
| Power supply voltage | 3.1 to 5.5 volts |
| Power consumption | 80 mA @ 5 volts with USB |
| Connector | Samtec FTSH-105-01-F-D-K |
| Operating temperature | -40 °C to +70 °C |
| Dimensions | 38 mm x 24 mm x 12 mm |
| Weight | 11.5 grams |
| ROHS | compliant |
| Shock limit | 500 g |
| Software utility | CD in starter kit (XP/Vista/Win7 compatible) |
| Software development kit (SDK) | complete data communications protocol and sample code |

IMU Specifications

| | Accels | Gyros | Mags |
|---------------------------|---------------------------------|---------------|----------------|
| Measurement range | ±5 g | ±300°/sec | ±2.5 Gauss |
| Non-linearity | ±0.1 % fs | ±0.03 % fs | ±0.4 % fs |
| In-run bias stability | ±0.04 mg | 18°/hr | — |
| Initial bias error | ±0.002 g | ±0.25°/sec | ±0.003 Gauss |
| Scale factor stability | ±0.05 % | ±0.05 % | ±0.1 % |
| Noise density | 80 µg/√Hz | 0.03°/sec/√Hz | 100 µGauss/√Hz |
| Alignment error | ±0.05° | ±0.05° | ±0.05° |
| User adjustable bandwidth | 225 Hz max | 440 Hz max | 230 Hz max |
| Sampling rate | 30 kHz | 30 kHz | 7.5 kHz max |
| Options | | | |
| Accelerometer range | ±1.7 g, ±16 g, ±50 g | | |
| Gyroscope range | ±50°/sec, ±600°/sec, ±1200°/sec | | |



© Copyright 2013 LORD MicroStrain®
 MicroStrain®, FAS-A®, 3DM®, 3DM-DH®, 3DM-GX3® and 3DM-DH3™ are trademarks of LORD
 MicroStrain®
 Specifications are subject to change without notice.
 Version 8400-0030 rev. 002

LORD Corporation
 MicroStrain® Sensing Systems
 459 Hurricane Lane,
 Suite 102
 Williston, VT 05495 USA
 www.microstrain.com
 ph: 800-449-3878
 fax: 802-863-4093
 sales@microstrain.com

Apêndice B

Códigos Fonte

B.1 Comunicação entre o Sensor 3DM-GX3-25 e o QUARC

```
1 #include "stdafx.h"
2 #include <stdio.h>
3 #include <signal.h>
4 #include <windows.h>
5 #include <tchar.h>
6 //Bibliotecas de comunicacao com o QUARC
7 #include "quanser_messages.h"
8 #include "quanser_stream.h"
9
10 static int stop = 0;
11
12 static void control_c_handler(int signum) {
13     stop = 1;
14 }
15
16 void PrintCommState(DCB dcb)
17 {
18     // Exibe parametros do DCB
19     _tprintf( TEXT("\nBaudRate = %d, ByteSize = %d, Parity = %d, StopBits = %d\n"),
20             dcb.BaudRate,
21             dcb.ByteSize,
22             dcb.Parity,
23             dcb.StopBits );
24 }
25
26 int _tmain(int argc, _TCHAR* argv[])
27 {
28
29     //////////////////////////////////////
30     //////////////////////////////////////SERIAL-SENSOR////////////////////////////////////
31     //////////////////////////////////////
32     DCB dcb;
33     HANDLE hCom;
34     BOOL fSuccess;
35     TCHAR *pcCommPort = TEXT("COM8"); //Porta Serial
```

```

37 // Handle de comunicacao
hCom = CreateFile( pcCommPort,
39     GENERIC_READ | GENERIC_WRITE,
    0,          // Acesso exclusivo
41     NULL,     // Atributo padrao de seguridade
    OPEN_EXISTING, // must use OPEN_EXISTING
43     0,       // not overlapped I/O
    NULL ); // hTemplate must be NULL for comm devices
45
46 if (hCom == INVALID_HANDLE_VALUE)
47 {
    // Verifica se houve erro
49     printf ("CreateFile falhou com o erro %d.\n", GetLastError());
    return (1);
51 }
52
53 // Inicializa estrutura DCB
SecureZeroMemory(&dcb, sizeof(DCB));
55 dcb.DCBlength = sizeof(DCB);
56
57 // Build on the current configuration by first retrieving all current
// settings.
59 fSuccess = GetCommState(hCom, &dcb);
60
61 if (!fSuccess)
62 {
    // Verifica se houve erro
63     printf ("GetCommState falhou com o erro %d.\n", GetLastError());
65     return (2);
66 }
67
68 PrintCommState(dcb);          // Print DCB parameters
69
70 //Parametros de comunicacao
71 dcb.BaudRate = 921600;        //BaudRate (ignorado para USB)
dcb.ByteSize = 8;             //Dados com 8 bits
73 dcb.Parity = NOPARITY;       //Sem bit de paridade
dcb.StopBits = ONESTOPBIT;    //1 bit de stop
75 fSuccess = SetCommState(hCom, &dcb);
76
77 if (!fSuccess)
78 {
    // Verifica se houve falha
79     printf ("SetCommState falhou com o erro %d.\n", GetLastError());
81     return (3);
82 }
83
84 // Get the comm config again.
85 fSuccess = GetCommState(hCom, &dcb);
86
87 if (!fSuccess)
88 {
    // Verifica se houve falha.
89     printf ("GetCommState falhou com o erro %d.\n", GetLastError());
91     return (2);
92 }

```

```

93 PrintCommState(dcb);          // Output to console
95
96 _tprintf (TEXT("Portal serial %s reconfigurada com sucesso.\n"), pcCommPort);
97
98 BYTE SensData [19];
99 BYTE revSensData [19];
100 BYTE getEuler = 0xCE;
101
102 DWORD dwBytesTransferred;
103
104 DWORD dwError ,
105     dwNumBytesWritten;
106
107 // Variaveis com os angulos de Roll, Pitch e Yaw
108 float yaw = 0;
109 float pitch = 0;
110 float roll = 0;
111 // Variaveis com os angulos de Roll, Pitch e Yaw da primeira leitura do sensor
112 float roll0, pitch0, yaw0;
113 t_single s_roll = 0;
114 t_single s_pitch = 0;
115 t_single s_yaw = 0;
116
117 ////////////////////////////////////////////////////
118 ////////////////////////////////////////////////////SHARED MEMORY - MATLAB //
119 ////////////////////////////////////////////////////
120
121 const char      uri[]           = "shmem://imuData:1"; //URI
122 const t_boolean nonblocking     = false;
123 const t_int      send_buffer_size = 8000; //Pilha de
124 const t_int      receive_buffer_size = 8000; //Pilha de entrada
125 const char *     locale         = NULL;
126 t_stream client;
127 t_error result;
128 char message [512];
129
130 /* Define uma handle para o comando Control+C para finalizar a execucao do programa */
131 signal(SIGINT, control_c_handler);
132
133 /* Esta funcao tenta a conexao com o server por meio da URI fornecida */
134 result = stream_connect(uri, nonblocking, send_buffer_size, receive_buffer_size, &
135     client);
136
137 if (result == 0) /* Se conectou... */
138 {
139     unsigned long count = 0;
140     t_single value = 0.0;
141
142     /* Loop principal do programa:
143        - Leitura do Sensor
144        - Envia os dados para o Quarc
145     */
146     while (!stop)

```

```

149 {
150 ///////////////////////////////////////////////////////////////////
151 ///////////////////////////////////////////////////////////////////
152 ///////////////////////////////////////////////////////////////////
153 WriteFile (hCom,          // handle
154           &getEuler,      // Array que sera enviado
155           1,              // Bytes que devem ser enviados
156           &dwNumBytesWritten, // Bytes efetivamente enviados
157           NULL);
158
159 ReadFile (hCom,          // handle
160          &SensData,      // Array que sera escrito
161          19,             // Numero de bytes lidos
162          &dwBytesTransferred, // Bytes efetivamente recebidos
163          NULL);
164
165 //Para facilitar na leitura do CheckSum, a ordem dos bytes de SensData eh revertida
166 em revSensData
167 for(int i = 0; i < 19; i++)
168     revSensData[i] = SensData[18-i];
169
170 //Define um ponteiro apontado para o CheckSum recebido
171 unsigned short* imuCheckSum = (unsigned short*) revSensData;
172 //Subtrai a soma de todos os bytes recebidos (com excecao dos de CheckSum) do dado
173 de CheckSum
174 for(int i = 2; i < 19; i++)
175     *imuCheckSum -= revSensData[i];
176
177 if ( *imuCheckSum == 0 ){//Se o CheckSum esta correto
178     float yaw = *((float*) (revSensData + 6));
179     float pitch = *((float*) (revSensData + 10));
180     float roll = *((float*) (revSensData + 14));
181
182     if(!count)//Se for o primeiro dado recebido
183     {
184         roll0 = roll;
185         pitch0 = pitch;
186         yaw0 = yaw;
187     }
188
189     count++; //Incrementa numero de dados recebidos
190
191 //array com os dados a serem enviados
192 t_single dataEuler[3] = {roll-roll0, pitch, yaw-yaw0};
193
194 //envia o array dataEuler para a comunicacao com o Quarc
195 result = stream_send_single_array(client, dataEuler, 3);
196 if (result < 0)
197     break;//Se houve erro na transmissao, sai do loop principal
198 }
199
200 result = stream_receive_single(client, &value);
201
202 if (result <= 0)
203     break;

```

```

203     result = stream_flush(client);
        if (result < 0)
205         break;
    }
207
    /* Ocorreu um erro ou a conexao foi interrompida pelo server. Fecha o cliente. */
209     stream_close(client);
    printf("Conexao fechada. Valor: %lg. Numero de itens: %lu\n", value, count);
211
    /*
213     Verifica se ocorreu algum erro durante a transmissao/recepcao de dados.
    No caso de erro, exibe a mensagem adequada.
215     */
    if (result < 0)
217     {
        msg_get_error_messageA(locale, result, message, sizeof(message));
219         printf("Erro na comunicacao com o URI '%s'. %s\n", uri, message);
    }
221 }
    /*
223     Se a funcao stream_connect encontrou um erro, o cliente nao consegue se conectar
    com o server. Neste caso, a mensagem de erro eh exibida.
225     */
    else
227     {
        msg_get_error_messageA(locale, result, message, sizeof(message));
229         printf("Nao foi possivel conectar com o URI '%s'. %s\n", uri, message);
    }
231     printf("Pressione Enter para fechar\n");
    getchar();
233 }

```

B.2 Rastreio de Alvo por Cor

```

%Programa testado no Matlab R2013a com o Image Processing Toolbox

clc;
clear;
5
% abre o video desejado
video = VideoReader('sem.mp4');
frameNumber = video.NumberOfFrames;

10 gravarvideo = 0;

if gravarvideo
    mov(1:frameNumber) = struct('cdata', [], 'colormap', []);

```

```

end
15
% variavel com a imagem do rastro do alvo
rastreo = zeros(video.Height, video.Width);
rastreo = im2bw(rastreo);

20 for i = 1:frameNumber
    frame_i = read(video, i); % matriz com o frame i do video WxHx3

    % subtrai do canal vermelho da imagem RGB a imagem convertida
    para gray_scale
    frameRed = imsubtract(frame_i(:,:,1), rgb2gray(frame_i));

25
    % filtragem mediana em uma matriz de duas dimensoes
    frameRed = medfilt2(frameRed, [3 3]);

    % produz uma imagem binaria preta e branca com base em um
    limiar
30
    frameRed = im2bw(frameRed, 0.2);

    % incrementa a imagem de rastro com o novo ponto detectado
    rastreo = rastreo + frameRed;

35
    if gravarvideo
        % converte a imagem para rgb e adiciona ao frame do video
        rastreo_c(:,:,1) = rastreo*255; % rastro vermelho
        rastreo_c(:,:,2) = 0; rastreo_c(:,:,3) = 0;
        rastreo_c = im2uint8(rastreo_c);
40
        mov(i).cdata = frame_i + rastreo_c;
    end
end

imshow(rastreo);

45
% salva o video gravado
if gravarvideo
    movie2avi(mov, 'com.avi', 'compression', 'None', 'fps', video.
    FrameRate);
end

```