



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

José Wilton Oliveira de Arruda

**Controle Automático de Rumo de uma Embarcação de Superfície
não Tripulada**

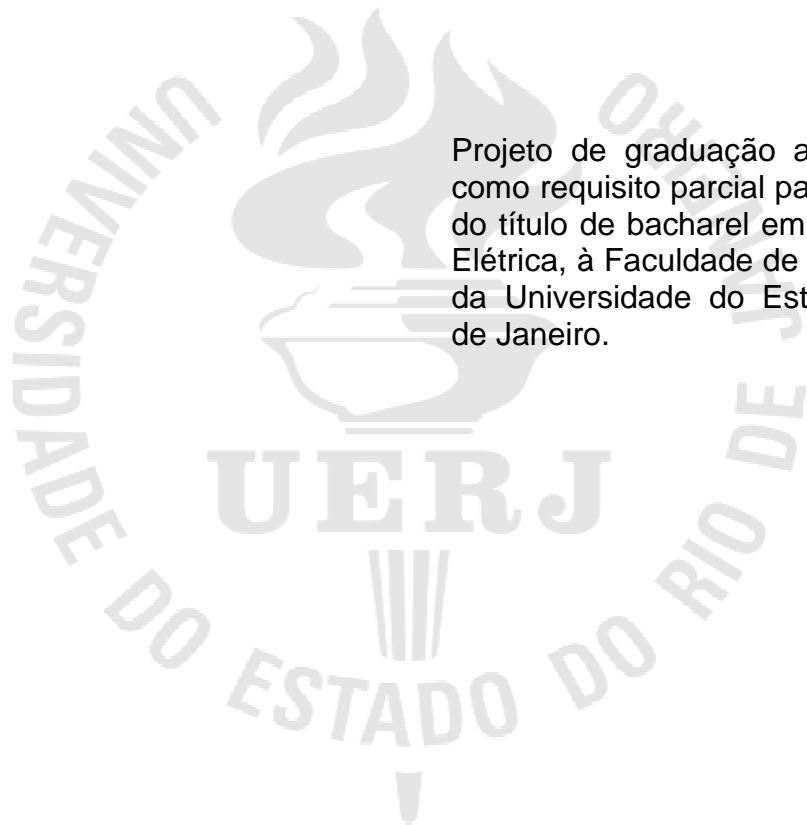
Rio de Janeiro

2012

José Wilton Oliveira de Arruda

**Controle Automático de Rumo de uma Embarcação de Superfície não
Tripulada**

Projeto de graduação apresentado,
como requisito parcial para obtenção
do título de bacharel em Engenharia
Elétrica, à Faculdade de Engenharia,
da Universidade do Estado do Rio
de Janeiro.



Orientadores: José Paulo Vilela Soares da Cunha

Tiago Roux de Oliveira

Rio de Janeiro

2012

CATALOGAÇÃO NA FONTE
UERJ/REDE SIRIUS/CTC/B

A779 Arruda, José Wilton Oliveira.
Controle automático de rumo de uma embarcação de superfície não tripulada. / José Wilton Oliveira Arruda – 2012. 96 f.: il.

Orientador: José Paulo Vilela Soares da Cunha
Coorientador: Tiago Roux de Oliveira

Projeto (graduação) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.
Bibliografia: f.78

1. Controle automático. 2. Correntes contínuas - acionamento elétrico. I. Cunha, José Paulo Vilela Soares da. II. Oliveira, Tiago Roux de. III. Universidade do Estado do Rio de Janeiro. Faculdade de Engenharia. III. Título.

CDU 621.313.13

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese.

Assinatura

Data

José Wilton Oliveira de Arruda

Controle Automático de Rumo de uma Embarcação de Superfície não Tripulada

Projeto de graduação apresentado, como requisito parcial para obtenção do título de bacharel em Engenharia Elétrica, à Faculdade de Engenharia, da Universidade do Estado do Rio de Janeiro.

Aprovado em 10 de Setembro de 2012.

Banca Examinadora:

Prof. Dr. José Paulo Vilela Soares da Cunha (Orientador)
Faculdade de Engenharia - UERJ

Prof. Dr. Tiago Roux de Oliveira (Orientador)
Faculdade de Engenharia - UERJ

Prof. Dr. Téo Cerqueira Revoredo
Faculdade de Engenharia - UERJ

Prof. Dr. Eduardo Vieira Leão Nunes
Escola Politécnica - UFRJ

Rio de Janeiro

2012

AGRADECIMENTOS

À Deus por ter me dado saúde e forças para superar todas as dificuldades encontradas ao longo desta caminhada.

Aos meus orientadores José Paulo Vilela Soares da Cunha e Tiago Roux de Oliveira, pela paciência e dedicação que tiveram comigo, e por todo o conhecimento que me foi transmitido, principalmente na área de controle, sensores de navegação e microcontrolador.

A todos os meus amigos, em especial, Hendrik Jürgen Schultze, Daniel Vieira de Moura e Vinicio Rodrigues Mendes que colaboram para o sucesso deste projeto.

À minha família que sempre esteve ao meu lado me apoiando e acreditando em mim.

DEDICATÓRIA

À minha família

RESUMO

ARRUDA, José Wilton Oliveira. Controle Automático de Rumo de uma Embarcação de Superfície não Tripulada. 2012. 96 f. Projeto Final (Graduação em Engenharia Elétrica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2012.

Este projeto visa o desenvolvimento de um sistema de controle automático que seja capaz de manter uma embarcação de superfície não tripulada na orientação desejada independente das perturbações externas. Foi desenvolvido um algoritmo de controle capaz de corrigir a orientação da embarcação com base no erro entre a posição de referência e a posição medida por uma bússola eletrônica. Este erro e a velocidade angular da embarcação, medida por um girômetro, são processados pelo algoritmo de controle que gera um sinal de comando para os propulsores a fim de estabelecer a correção do rumo. Foi utilizado o controle a estrutura variável que se baseia na escolha de uma superfície no espaço de estado denominada de superfície de deslizamento. Esta superfície é definida como uma função linear das variáveis de estado da dinâmica do movimento de rotação do barco. Uma vez que o sistema entra em modo deslizante, sua dinâmica passa a ser regida pela dinâmica da superfície de deslizamento escolhida e torna-se imune a perturbações.

Palavras-chave: Controle automático de rumo. Embarcação de superfície não tripulada. Controle a estrutura variável.

ABSTRACT

This project aims to develop an automatic control system that is capable of maintaining unmanned surface vehicles in the desired orientation independent of external disturbances. It was developed a control algorithm capable of correcting the orientation of the vessel based on the error between the reference position and the position measured by an electronic compass. This error and the angular velocity of the vessel, measured by a gyro are processed by the control algorithm generates a command signal to the drivers to establish the correctness of the orientation. It was used a variable structure control which is based on the choice of a surface in the state space called sliding surface. This surface is defined as a linear function of the state variables of the dynamic rotational motion of the vessel. Once the system is in sliding mode, its dynamics is now governed by the dynamics of the sliding surface chosen and becomes immune to disturbances.

Keywords: Automatic control guidance. Unmanned surface vehicles. Variable structure control.

LISTA DE FIGURAS

Figura 1.1 – Veículo SCOUT. Imagem extraída de [12].	15
Figura 1.2 – Imagem feita por computação gráfica do protótipo do Trimarã, desenvolvido pelos Pesquisadores da Escola de Engenharia de São Carlos. Imagem: Prof. José Roberto Monteiro.	17
Figura 1.3 – Embarcação não tripulada ROAZ 1 desenvolvida pelo Engenheiro Roberto Böell Vaz para o monitoramento ambiental. Imagem extraída de [20].	18
Figura 1.4 – Embarcação de superfície não tripulada MARINA.	20
Figura 1.5 – Fluxo de sinais do sistema de controle de rumo proposto.	21
Figura 2.1: Diagrama em blocos dos elementos principais do circuito.	24
Figura 2.2: Arduino Duemilanove. Imagem extraída de [15].	27
Figura 2.3: Bússola eletrônica HMC6352. Imagem extraída de [14].	30
Figura 2.4: Dispositivos conectados a um barramento I ² C.	31
Figura 2.5: Sinal de <i>Start</i> do protocolo de comunicação I ² C.	31
Figura 2.6: Sinal de <i>Stop</i> do protocolo de comunicação I ² C.	32
Figura 2.7: Esquemático de ligação do HMC6352 ao Arduino Duemilanove.	33
Figura 2.8: Unidade de Medição Inercial 6DOF Razor. Imagem extraída de [14].	38
Figura 2.9: Esquemático de interligação da Unidade de Medição Inercial ao Arduino Duemilanove.	40
Figura 2.10: Esquemático de interligação da bússola e girômetro ao Arduino Duemilanove.	44
Figura 3.1: Sistemas de coordenadas adotados para a embarcação.	46
Figura 3.2: Diagrama em blocos do modelo simplificado da dinâmica do sistema.	48
Figura 3.3: Disposição dos propulsores na embarcação.	54
Figura 3.4: Diagrama de blocos do sistema de controle.	55
Figura 4.1: Diagrama em blocos do VSC desenvolvido no Simulink.	57
Figura 4.2: Simulação do comportamento do sistema para $\psi_{ref} = 280,7^\circ$.	58
Figura 4.3: Sinais de comando dos propulsores da simulação do controle VSC.	59
Figura 4.4: Diagrama em blocos do VSC para sinais amostrados desenvolvido no Simulink.	60
Figura 4.5: Oscilações do sistema para frequência de amostragem de 10 Hz.	61

Figura 4.6: Sinais de comando dos propulsores para a frequência de amostragem de 10 Hz.....	61
Figura 4.7: Oscilações do sistema para a frequência de amostragem de 3,33 Hz....	62
Figura 4.8: Sinais de comando dos propulsores para a frequência de amostragem de 3,33 Hz.....	63
Figura 4.9: Oscilações do sistema para a posição de referência $\psi_{ref} = 140,3^\circ$	64
Figura 4.10: Sinais de comando dos propulsores para $\psi_{ref} = 140,3^\circ$	64
Figura 4.11: Canal da Capela Ecumênica da UERJ.....	65
Figura 4.12: Oscilações da embarcação para $\psi_{ref} = 280,7^\circ$	67
Figura 4.13: Velocidade angular da embarcação para $\psi_{ref} = 280,7^\circ$	68
Figura 4.14: Sinal de comando dos propulsores em função do tempo.....	69
Figura 4.15: Comparação entre simulação e experimento para $\psi_{ref} = 280,7^\circ$	70
Figura 4.16: Oscilações da embarcação para $\psi_{ref} = 280,7^\circ$ do segundo teste.....	71
Figura 4.17: Velocidade angular da embarcação para $\psi_{ref} = 280,7^\circ$ do segundo teste.	71
Figura 4.18: Sinal de comando dos propulsores em função do tempo do segundo teste.	72
Figura 4.19: Oscilações da embarcação para $\psi_{ref} = 140,3^\circ$	73
Figura 4.20: Velocidade angular da embarcação para $\psi_{ref} = 140,3^\circ$	73
Figura 4.21: Sinal de comando dos propulsores em função do tempo.....	74
Figura 4.22: Comparação entre simulação e experimento para $\psi_{ref} = 140,3^\circ$	75
Figura A.1 – Caiaque Hunter Fishing.	82
Figura A.2 – Suporte de metal para fixação dos propulsores.....	83
Figura A.3 – arranjo motores mais suporte metálico fixados no caiaque.	83
Figura A.4 – Caixa estanque do sistema de controle.	84
Figura A.5 – Hélice do propulsor.	86
Figura A.6 – Propulsor utilizado no caiaque.....	87
Figura A.7 – Circuito de acionamento dos relés. Imagem extraída de [22].	88
Figura B.1 – Unidade de medição composta pela bússola e unidade de medição inercial.....	89
Figura B.2 – Diagrama elétrico da interligação dos sensores ao microcontrolador...	90
Figura B.3 – (a) Suporte de alumínio em perfil de L e (b) unidade de medição fixada à caixa.....	90

LISTA DE TABELAS

Tabela 2.1: Principais características do Arduino Duemilanove.....	26
Tabela 2.2: Tabela comparativa das bússolas selecionadas.	29
Tabela 2.3: Algoritmo para a leitura do HMC6352 desenvolvido no editor do Arduino.	34
Tabela 2.4: Dados de saída da bússola apresentados no monitor serial do Arduino.	35
Tabela 2.5: Tabela comparativa dos Girômetros selecionadas.....	37
Tabela 2.6: Principais características do 6DOF Razor.....	39
Tabela 2.7: Algoritmo para a leitura do 6DOF Razor desenvolvido no editor do Arduino.....	41
Tabela 2.8: Dados de saída do 6DOF Razor apresentados no monitor serial do Arduino.....	42
Tabela A.1: Alimentação dos fios provenientes da bobina.....	85
Tabela B.1: Rotina de calibração da bússola.....	92
Tabela C.1: Código da rotina de controle desenvolvida.....	93

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	14
1.1 – Embarcação SCOUT	15
1.2 – Embarcações Brasileiras	16
1.3 – Controle Automático de Rumo	18
1.4 – Objetivo	20
1.5 – Descrição do Sistema de Controle Automático de Rumo Proposto	20
1.6 – Organização do Texto	21
CAPÍTULO 2 – MICROCONTROLADOR E SENSORES	23
2.1 – Arduino	23
2.1.1 – Seleção do Arduino	24
2.1.2 – Características Relevantes	25
2.1.3 - Comparações e Conclusões	26
2.1.4 - Características do Arduino Duemilanove	26
2.2 – Bússola Eletrônica	27
2.2.1 - Seleção da Bússola	28
2.2.2 - Comparações e Conclusões	29
2.2.3 - Características do HMC6352	29
2.2.4 - Interface de comunicação do HMC6352	30
2.2.5 – Funcionamento do HMC6352	32
2.2.6 – Contador de Voltas	35
2.3 – Girômetro	36
2.3.1 – Seleção do Girômetro	37
2.3.2 – Comparações e Conclusões	37
2.3.3 – Características do 6DOF Razor	38
2.3.4 – Funcionamento do 6DOF Razor	39
CAPÍTULO 3 – CONTROLE	45
3.1 – Sistemas de Coordenadas	45
3.2 – Modelagem Matemática da Dinâmica do Sistema	46
3.2.1 – Equação da dinâmica de rotação	47
3.2.2 – Modelagem matemática dos propulsores	49
3.3 – Controle a Estrutura Variável	50

3.3.1 – Controle a Estrutura Variável Aplicado à Embarcação	51
3.3.2 - Sinal de comando no sistema de coordenada móvel	52
3.3.3 – Sinais de comando dos propulsores	53
3.4 – Implementação do Controle	56
CAPÍTULO 4 – SIMULAÇÕES, TESTES E RESULTADOS EXPERIMENTAIS.....	57
4.1 – Simulação do Controlador a Estrutura Variável	57
4.2 – Simulação do Controle com Sinais Amostrados	59
4.3 – Testes e Resultados Experimentais.....	65
CAPÍTULO 5 – CONCLUSÕES	76
REFERÊNCIAS BIBLIOGRÁFICAS	78
APÊNDICE A – EMBARCAÇÃO E CIRCUITO DE ACIONAMENTO	82
A.1 – Embarcação.....	82
A.2 – Propulsores.....	84
A.3 – Acionamento dos propulsores	87
APÊNDICE B – UNIDADE DE MEDIÇÃO E CALIBRAÇÃO DA BÚSSOLA.....	89
B.1 – Unidade de Medição.....	89
B.2 – Calibração da Bússola.....	91
APÊNDICE C – ROTINA DE CONTROLE	93

CAPÍTULO 1 – INTRODUÇÃO

As mudanças climáticas e os impactos causados pelo homem ao meio ambiente, em especial nos mares, rios e atmosfera (tais como as emissões de gases poluentes e a descarga de efluentes em rios, lagos e oceanos) vêm motivando pesquisadores e engenheiros do mundo todo a desenvolverem projetos voltados para o monitoramento ambiental de maneira mais eficiente e sustentável.

O trabalho de coleta de variáveis físicas e químicas da água e atmosfera tem sido feita, em grande parte, por embarcações de pesquisa, bóias fixas ou bóias nômades, conduzidas por correntes marítimas e ventos [11]. Embora estes métodos de coleta de dados sejam satisfatórios, apresentam inconvenientes, como por exemplo, a limitação dos pontos de coleta por corpo aquático em função do tempo necessário, as dificuldades de se coletar dados em situações de clima e tempo adversos, o alto custo da manutenção e operação de embarcações específicas para este fim e o risco de vida dos pesquisadores que muitas vezes se expõem a áreas inapropriadas.

Os avanços das áreas de instrumentação, automação e controle e os interesses de se estudar a qualidade das águas e atmosfera têm estimulado um progresso na área de veículos de pesquisa não tripulados tais como aviões, submarinos e embarcações de superfície [11].

As embarcações de superfície não tripuladas, também conhecidas como USVs (*unmanned surface vehicles*) são embarcações que navegam de forma autônoma ao longo de trajetórias pré-definidas e possuem boa capacidade de resposta frente a perturbações externas, como por exemplo, correntes marítimas e ventos, e têm como objetivo principal o monitoramento ambiental já que podem ser utilizados para coletar dados em regiões poluídas, perigosas e em condições climáticas adversas durante longos períodos de medição e os dados podem ser acompanhados em tempo real por uma estação em terra.

1.1 – Embarcação SCOUT

Pode-se citar como exemplo de embarcação de superfície, o projeto do veículo SCOUT (Surface Craft for Oceanographic and Undersea Testing), uma plataforma de superfície autônoma de baixo custo para pesquisa em cooperativa autônoma, desenvolvida pelo Departamento de Engenharia Mecânica do Instituto de Tecnologia de Massachusetts (vide Figura 1.1). Os objetivos do projeto incluem a simplicidade, robustez, versatilidade e utilidade operacional melhorada. Foram construídos quatro veículos e uma série de experimentos de campo foi realizada utilizando estas embarcações. Estas plataformas foram construídas com o objetivo de se desenvolver algoritmos para controle e operação de múltiplas embarcações de superfície de modo a formar uma frota de veículos autônomos [12].



Figura 1.1 – Veículo SCOUT. Imagem extraída de [12].

Conforme apresentado em [12], o veículo SCOUT é um simples caiaque constituído de polietileno de alta densidade, munido de sistema de propulsão e controle autônomo realizado por um computador de bordo. O peso total do sistema com baterias instaladas é de aproximadamente 82 quilos, de modo que o veículo

possa ser facilmente transportado. Os principais componentes do veículo incluem: casco, baterias, computador a bordo, controle via rádio, sistema de direção e propulsão, comunicação Wi-Fi, cabeamento e refrigeração. Dentro do casco, dois anteparos estanques, em conjunto com duas escotilhas estanques separam o volume interno do casco em três compartimentos distintos: dianteiro, central e traseiro. O computador a bordo, é abrigado em uma caixa estanque localizada no compartimento central e interligado a todos os sistemas dentro do veículo usando plástico e conectores à prova d'água.

O SCOUT foi projetado para ser operado de três maneiras distintas: via rádio, Wi-Fi e de maneira totalmente autônoma (através do computador a bordo). As antenas para ambos os dispositivos são montadas no mastro da popa e seus fios estão diretamente conectados ao computador a bordo através das portas USB disponíveis. Os componentes principais do veículo incluem o sistema de direção e propulsão, GPS (*Global Positioning System*) e bússola. O sistema de propulsão é composto por um motor elétrico cuja velocidade é regulada através de um controlador eletrônico. O controlador do motor recebe sinais modulados em largura de pulso (*Pulse Width Modulation – PWM*) a partir do rádio ou computador a bordo de modo a fazer com que o motor gire em sentido direto ou inverso. A direção é fornecida por um servo motor, que também recebe um sinal de entrada PWM e é capaz de rotacionar o eixo do propulsor cerca de 180 graus. O GPS e bússola fornecem a velocidade, localização e as informações de direção para o algoritmo de controle do veículo utilizando a malha de controle Proporcional, Integral e Derivativo (*PID*) simples [12].

1.2 – Embarcações Brasileiras

No Brasil, existem alguns grupos de pesquisas que desenvolvem embarcações não tripuladas para o monitoramento ambiental, como é o caso dos Pesquisadores da Escola de Engenharia de São Carlos, da Universidade de São Paulo (USP) que estão desenvolvendo o protótipo de um Trimarã, uma estrutura com três cascos sendo um principal e dois laterais permitindo maior estabilidade lateral do sistema. O protótipo está em fase final de construção, e está sendo

projetado para realizar observações de peixes (por meio de câmeras submersíveis) e da qualidade da água. A embarcação possui 1,80 metros de comprimento por 1,60 metros de largura, construído em isopor de alta densidade pesando 25 quilos. O veículo é projetado para águas fluviais (rios, lagos e represas), sem quedas de água, e é dirigido automaticamente de forma remota. A navegação é feita por meio de coordenadas de um GPS, onde, os indicadores de posicionamento servirão para ajustar os ângulos da vela e do leme, conduzindo o protótipo ao local desejado. O grande diferencial do Trimarã é a autonomia, com menor exigência de reabastecimentos devido ao uso de um sistema combinado de propulsão que inclui vela, motor elétrico e energia solar [18]. A Figura 1.2 apresenta a imagem do protótipo desenvolvida por computação gráfica.



Figura 1.2 – Imagem feita por computação gráfica do protótipo do Trimarã, desenvolvido pelos Pesquisadores da Escola de Engenharia de São Carlos. Imagem: Prof. José Roberto Monteiro.

Outro grupo de pesquisas, formado pelo Engenheiro Roberto Böell Vaz e seus colaboradores, desenvolveu um barco não tripulado em forma de golfinho denominado ROAZ 1. O ROAZ 1 é uma embarcação de superfície autônoma, usada para monitoramentos, coleta de dados ambientais, ecobatimetria, levantamentos de topografia subaquática e pode ser usado em operações de busca, salvamento e vigilância em áreas de preservação. Os dados coletados são transmitidos, por um

computador a bordo, em tempo real, permitindo a avaliação mais rápida das condições ambientais [19]. O barco mede 4 metros de comprimento e é revestido por isopor. Possui sistema de propulsão a motor elétrico alimentado por bateria, e é conduzido por controle remoto ligado a um computador que mostra áudio e vídeo em tempo real em um raio de 1,5km. O projeto foi financiado por um programa de inovação mantido pelo governo de Santa Catarina por meio da Fundação de Apoio à Pesquisa Científica do Estado. A Figura 1.3 apresenta a imagem do ROAZ 1.



Figura 1.3 – Embarcação não tripulada ROAZ 1 desenvolvida pelo Engenheiro Roberto Böell Vaz para o monitoramento ambiental. Imagem extraída de [20].

1.3 – Controle Automático de Rumo

A fim de tornar factível a utilização de uma embarcação de superfície não tripulada que navegue de forma autônoma com fins de monitoração ambiental de rios, lagos, mares e defesa, torna-se necessário o desenvolvimento de técnicas de controle de navegação, posição e orientação [11]. Desta forma, este projeto visa à construção de um sistema de controle automático de rumo para a embarcação de

superfície não tripulada, desenvolvida em [22], de maneira que a mesma possa medir as grandezas físicas e químicas da água e atmosfera seguindo uma trajetória de referência pré-estabelecida em meio às perturbações externas.

O controle de rumo é responsável por manter a embarcação em orientação desejada, sendo capaz de corrigir os desvios causados por distúrbios, como por exemplo, os provocados por ventos, correntes marítimas e ondas. A correção destes desvios pode ser feita de forma manual ou automática e, normalmente, é realizada pelo controle do leme, dos propulsores ou de ambos.

O controle automático de rumo será realizado sobre o sistema de propulsão de uma embarcação de casco simples (caiaque) denominada MARINA (vide Figura 1.4), que mede 3,15 metros por 0,80 metros de largura. O sistema de propulsão é composto por dois motores elétricos alimentados por uma bateria de 12 volts (detalhes sobre a bateria, tais como: tipos, tempo de carga e descarga e vida útil estão descritos em [22]), e hélices de material plástico com três pás cada. Os motores estão fixados ao barco através de um suporte de metal, e a bateria está posicionada um pouco a frente do centro de gravidade da embarcação, de modo a distribuir o peso e manter o sistema em equilíbrio. O veículo dispõe de um computador a bordo, que é abrigado no interior de uma caixa estanque localizada na popa do caíque próxima aos propulsores. Ao computador a bordo, está conectado o microcontrolador, responsável por tratar as informações vindas dos sensores de posição e velocidade angular (responsáveis pelo controle de rumo), que também se encontram no interior da caixa estanque, e gerar sinais de comando para o acionamento dos propulsores.



Figura 1.4 – Embarcação de superfície não tripulada MARINA.

1.4 – Objetivo

Este projeto visa o desenvolvimento de um controle a estrutura variável que seja capaz de manter uma embarcação de superfície não tripulada no rumo desejado, ainda que sujeita a perturbações externas.

1.5 – Descrição do Sistema de Controle Automático de Rumo Proposto

Será desenvolvido um algoritmo de controle capaz de corrigir o rumo da embarcação por meio do erro do ângulo de rumo e da velocidade angular da mesma. Para isso, serão utilizados sensores, tais como: bússola eletrônica e girômetro, que estarão medindo continuamente a posição e a velocidade angular do barco.

As grandezas mensuradas pelos sensores serão enviadas a uma plataforma de computação física denominada Arduino, que, por sua vez, encontra-se conectado via cabo USB a um *netbook* a bordo.

O Arduino é basicamente uma placa de múltiplas entradas e saídas digitais e analógicas, microcontrolada e desenvolvida sobre uma biblioteca que simplifica a escrita da programação na linguagem Processing.

Os dados recebidos pelo Arduino serão processados pelo algoritmo de controle e, em seguida, serão gerados sinais de comando para os propulsores. Estes sinais de comando serão enviados para um circuito que irá atuar no acionamento dos propulsores. A Figura 1.5 apresenta o fluxo de sinais do sistema de controle de rumo proposto.

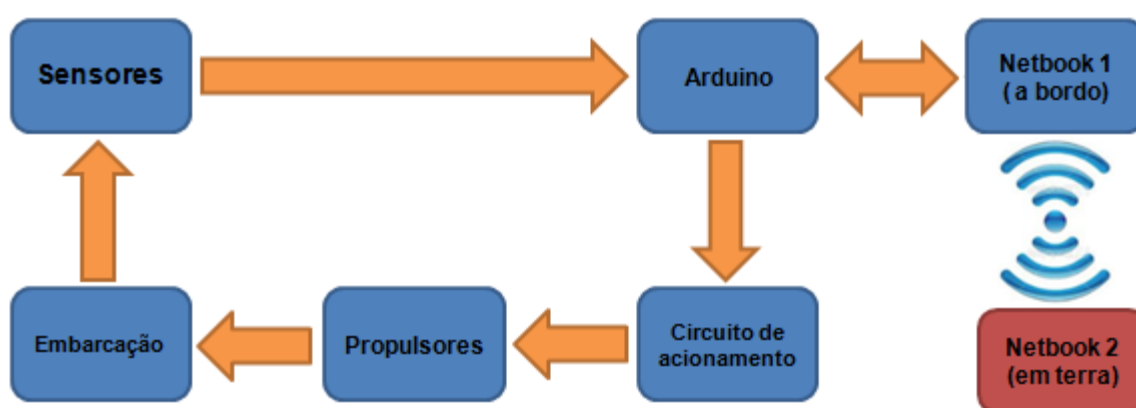


Figura 1.5 – Fluxo de sinais do sistema de controle de rumo proposto.

Haverá um *netbook* em terra que se comunicará com o *netbook* embarcado via rede *wireless* formando uma unidade de supervisão e comando. Desta forma, o comportamento do barco será monitorado a todo o momento, da mesma forma que o acesso e a alteração do código da rotina de controle poderão ser realizados remotamente.

1.6 – Organização do Texto

O texto é composto por cinco capítulos divididos da seguinte maneira:

- Capítulo 1: introduz o trabalho e faz uma breve descrição sobre o controle de orientação proposto.
- Capítulo 2: são apresentados os sensores e o microcontrolador (Arduino) utilizados no projeto.

- Capítulo 3: é desenvolvida a modelagem matemática da dinâmica do sistema e o sistema de controle.
- Capítulo 4: apresenta as simulações e os testes e resultados experimentais obtidos.
- Capítulo 5: são apresentadas as considerações finais, conclusões e propostas para trabalhos futuros.

CAPÍTULO 2 – MICROCONTROLADOR E SENSORES

Este capítulo descreve o microcontrolador responsável pelo controle automático de rumo e os sensores utilizados para medir a orientação e a velocidade angular da embarcação, as características mais relevantes ao projeto proposto e os critérios para a seleção dos mesmos.

2.1 – Arduino

O microcontrolador utilizado neste projeto é o Arduino (definido por [22]), que faz parte do conceito de *hardware* e *software* livre e está aberto para uso e contribuição de toda sociedade.

Conforme descrito em [9], o Arduino é uma plataforma de computação física (são sistemas digitais ligados a sensores e atuadores, que permitem construir sistemas que percebam a realidade e respondam com ações físicas), baseada em interfaces de Entrada/Saída microcontroladas e desenvolvida sobre uma biblioteca que simplifica a escrita da programação em Processing. O Arduino pode ser usado para desenvolver artefatos interativos *stand-alone* ou conectados a um microcomputador programável por Adobe Flash, Processing, Max/MSP, Pure Data ou SuperCollider.

O Arduino é capaz de interpretar variáveis no ambiente e transformá-las em sinal elétrico correspondente, por meio de sensores ligados aos seus terminais de entrada, e atuar no controle ou acionamento de elementos eletroeletrônicos conectados aos terminais de saída. Ou seja, é uma ferramenta de controle de entrada e saída de dados, que pode ser acionada por um sensor (por exemplo, um resistor dependente da luz - LDR) e que, logo após passar por uma etapa de processamento, o microcontrolador, poderá acionar um atuador (um motor, por exemplo) [9]. Para um melhor entendimento, na Figura 2.1 é possível identificar os elementos principais do circuito através de diagrama em blocos.



Figura 2.1: Diagrama em blocos dos elementos principais do circuito.

O Arduino é baseado em um microcontrolador Atmega, e é logicamente programável, ou seja, é possível a criação de programas, utilizando uma linguagem própria (Processing), que, quando implementados fazem com que o *hardware* execute as ações desejadas.

O grande diferencial desta ferramenta, é que ela é desenvolvida e aperfeiçoada por uma comunidade que divulga os seus projetos e seus códigos de aplicação, pois a concepção dela é *open-source*, ou seja, qualquer pessoa com conhecimento de programação pode modificá-la e ampliá-la de acordo com a necessidade, visando sempre a melhoria dos produtos que possam ser criados aplicando o Arduino [9].

Este microcontrolador foi projetado com a finalidade de ser de fácil entendimento, programação e aplicação, além de ser multiplataforma, ou seja, pode ser configurado em ambiente Windows, GNU/Linux e Mac OS. Assim sendo, pode ser perfeitamente utilizado como ferramenta educacional sem a preocupação de que o usuário tenha um conhecimento específico de eletrônica, mas pelo fato de ter o seu esquema e *software* de programação *open-source*, acabou chamando a atenção dos especialistas em eletrônica, que começaram a aperfeiçoá-lo e a criar aplicações mais complexas [9].

2.1.1 – Seleção do Arduino

O Arduino foi escolhido por ser uma plataforma de prototipagem eletrônica *open-source* baseada na flexibilidade e facilidade de utilização do *hardware* e *software*. Além disso, o software é compatível com os sistemas operacionais Windows, Mac OS X e Linux e pode ser obtido gratuitamente no seguinte site: <http://www.arduino.cc/en/Main/Software>.

Para a seleção do Arduino, foram relacionadas as características mais relevantes à execução do projeto, tais como: a compatibilidade com o sistema operacional utilizado (Linux), a capacidade de processamento, a quantidade de entradas e saídas e o custo da aquisição.

2.1.2 – Características Relevantes

- **Compatibilidade com o sistema operacional Linux**

O sistema operacional escolhido foi o Linux por ser um sistema livre. Sendo assim, definiu-se que o microcontrolador deveria ser compatível com este sistema operacional.

- **Capacidade de processamento**

Os sensores conectados ao Arduino estarão fazendo leituras continuamente, e estes valores de entrada serão processados e enviados em tempo real para a atuação dos propulsores para efeito de controle, desta forma, o Arduino deve ter uma capacidade de processamento, de forma que o sistema responda de maneira satisfatória atendendo as especificações do projeto.

- **Quantidade de entradas e saídas**

O Arduino deve ter uma quantidade de entradas e saídas digitais e analógicas suficientes para receber as informações dos sensores e acionar os atuadores utilizados no projeto.

- **Custo da aquisição**

Assim como em qualquer projeto de engenharia, os custos devem ser avaliados e reduzidos sem que sejam comprometidos os resultados do projeto. Logo, o custo de aquisição do Arduino também foi levado em consideração.

2.1.3 - Comparações e Conclusões

As pesquisas foram realizadas através da internet, em sites de vendas de componentes e circuitos eletrônicos (mais especificamente no site da *Multilógica Shop* que é uma das principais distribuidoras de componentes eletrônicos, sensores e atuadores e principalmente do Arduino). Dentre os diversos modelos disponíveis, optou-se pelo Arduino Duemilanove, por atender às especificações necessárias à implementação do projeto e pelo seu baixo custo.

2.1.4 - Características do Arduino Duemilanove

O Arduino Duemilanove ("2009") é uma placa de microcontrolador baseada no ATmega328. Ele possui 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um oscilador a cristal de 16 MHz, uma conexão USB, uma entrada para alimentação, um cabeçalho ICSP e um botão de *reset*. Para sua utilização basta conectá-lo a um computador através de um cabo USB ou ligá-lo a uma fonte de alimentação de corrente contínua ou bateria [14]. A Tabela 2.1 apresenta as principais características do Arduino Duemilanove.

Tabela 2.1: Principais características do Arduino Duemilanove.

Microcontrolador	ATmega328
Tensão de operação	5V
Tensão de alimentação (recomendada)	7-12V
Tensão de alimentação (limites)	6-20V
Pinos I/O digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente contínua por pino I/O	40 mA
Corrente contínua para o pino 3.3V	50 mA
Memória flash	32 KB (2KB usados para o bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequência de clock	16 MHz

A Figura 2.2 apresenta o Arduino Duemilanove selecionado para o projeto do controle de rumo.

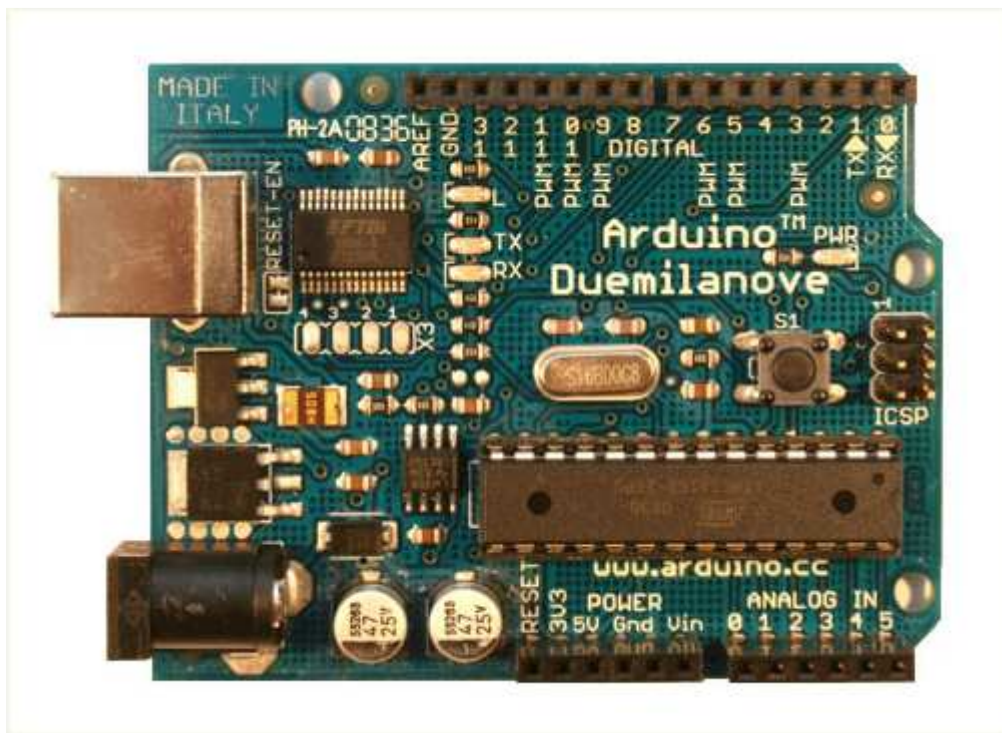


Figura 2.2: Arduino Duemilanove. Imagem extraída de [15].

2.2 – Bússola Eletrônica

A bússola é um instrumento de navegação e orientação, que se baseia nas propriedades magnéticas dos materiais ferromagnéticos e do campo magnético terrestre.

As bússolas convencionais são geralmente compostas por uma agulha magnetizada colocada em um plano horizontal e suspensa pelo seu centro de gravidade, de forma que possa girar livremente, e que se orienta sempre em direção próxima à direção norte-sul magnética, e tem a ponta destacada - geralmente em vermelho - indicando o sentido que leva ao sul magnético da Terra, ou de forma equivalente, a um ponto próximo ao polo norte geográfico da Terra [23].

As bússolas eletrônicas possuem o mesmo princípio de funcionamento que as convencionais, e classificam-se em sensores de Efeito Hall ou Magneto-Resistivo.

Conforme apresentado em [28], o sensor Magneto-Resistivo consiste em um resistor ou em uma ponte resistiva em que a resistência varia com a aplicação de um campo magnético de determinada orientação. Por outro lado, o sensor de Efeito Hall gera uma pequena tensão conforme a direção do campo magnético aplicado. O

arranjo dos elementos no sensor determina tanto a sensibilidade do dispositivo, quanto a orientação do campo que deve ser aplicado. No sensor de Efeito Hall, é possível identificar a orientação do campo, pois a polaridade da tensão gerada depende dela.

2.2.1 - Seleção da Bússola

Para a seleção da bússola levou-se em consideração a taxa de amostragem, a resolução de leitura, compatibilidade com o Arduino Duemilanove, a tensão de alimentação e o custo de aquisição.

- **Taxa de amostragem**

A taxa de amostragem consiste na quantidade de medidas por segundo que a bússola é capaz de realizar. Como a bússola será utilizada para controle em tempo real, uma baixa taxa de amostragem poderia acarretar em oscilações indesejadas no sistema de controle.

- **Resolução de leitura**

É a menor mudança incremental do parâmetro de entrada que causa uma variação detectável no valor de saída do sensor.

- **Compatibilidade com o Arduino**

Tendo em vista que o algoritmo de controle será implementado no Arduino Duemilanove, definiu-se que a bússola deveria ser compatível com alguma das suas entradas de dados.

- **Tensão de alimentação**

É a tensão que deve ser fornecida ao sensor para o seu funcionamento. O Arduino Duemilanove dispõe de duas saídas de tensões: 3,3 e 5 Vcc, que foram levadas em consideração na seleção da bússola.

- **Custo de aquisição**

Assim como para o Arduino Duemilanove, o custo de aquisição da bússola foi levado em consideração, sendo avaliado e reduzido sem que houvesse comprometimento dos resultados do projeto.

2.2.2 - Comparações e Conclusões

As pesquisas foram realizadas através da internet, em sites de eletrônica e robótica. Duas bússolas de um mesmo site foram selecionadas por atenderem os requisitos estabelecidos. A Tabela 2.2 apresenta os modelos de bússola selecionados na pesquisa.

Tabela 2.2: Tabela comparativa das bússolas selecionadas.

Bússola	Alimentação	Formato de Saída	Resolução	Taxa de Amostragem	Custo
LSM303	2,5 a 3,3V	I ² C	Ajustável ±1,3° a ±8,1°	30 Hz	R\$ 141,00
HMC6352	2,7 a 5,2V	I ² C	0,5°	Selecionável de 1 a 20 Hz	R\$ 276,00

A princípio, optou-se pela placa com bússola e acelerômetro triaxial (LSM303), pois, além da bússola, a placa contém um acelerômetro integrado e possui menor custo. Porém, como o LSM303 estava indisponível, comprou-se a bússola HMC6352.

2.2.3 - Características do HMC6352

O HMC6352 é um módulo de bússola totalmente integrado que combina dois sensores magneto-resistivos orientados ortogonalmente para detecção das componentes horizontais do campo magnético da terra. Este sensor possui circuitos de suporte analógicos e digitais tais como dois amplificadores, um circuito de *set* e *reset* e um microprocessador, e ainda permite que o usuário possa calibrá-la. A Figura 2.3 apresenta o módulo HMC6352.

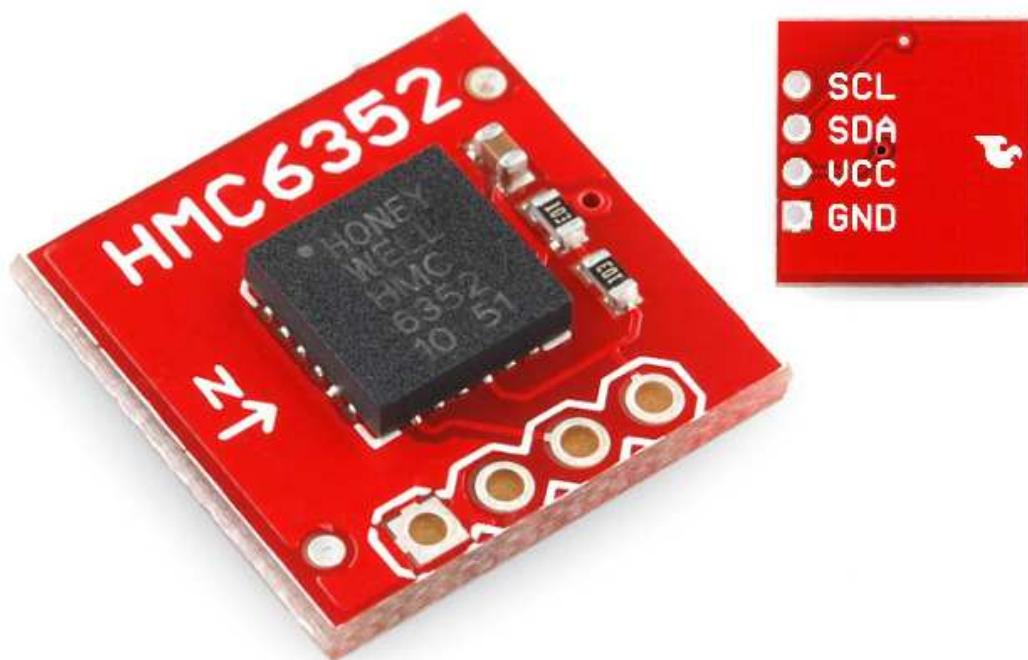


Figura 2.3: Bússola eletrônica HMC6352. Imagem extraída de [14].

O HMC6352 é alimentado com uma tensão que pode variar entre 2,7 e 5,2 Vcc, seu protocolo de comunicação é I²C (*Inter-Integrated Circuit*), sua taxa de amostragem é selecionável e varia de 1 a 20 Hertz, sua resolução de leitura é de 0,5 graus e os seus dados de saída são representados em décimos de graus entre 0 a 3599, ou seja, 0° a 359,9°.

2.2.4 - Interface de comunicação do HMC6352

- **Barramento I²C**

O módulo HMC6352 se comunica através de um barramento I²C que consiste fisicamente de dois fios e uma conexão com o terra. Os dois fios são bidirecionais, SDA (*Serial Data Line*) e SCL (*Serial Clock Line*). Nas linhas de SCL e SDA podem ser conectados diversos dispositivos, formando um barramento I²C. Cada dispositivo conectado em uma rede I²C possui um endereço, e todos podem atuar como transmissores ou receptores. No barramento I²C, um dispositivo pode atuar como mestre ou como escravo sendo o mestre, responsável pela comunicação. O barramento pode ter um único mestre e vários escravos ou pode ter uma configuração de múltiplos mestres. Todas as transferências de dados são iniciadas

pelo dispositivo mestre que é responsável por gerar o sinal de clock, e as transferências de dados são de 8 bits de comprimento. A Figura 2.4 apresenta de forma exemplificada os dispositivos em um barramento I²C:

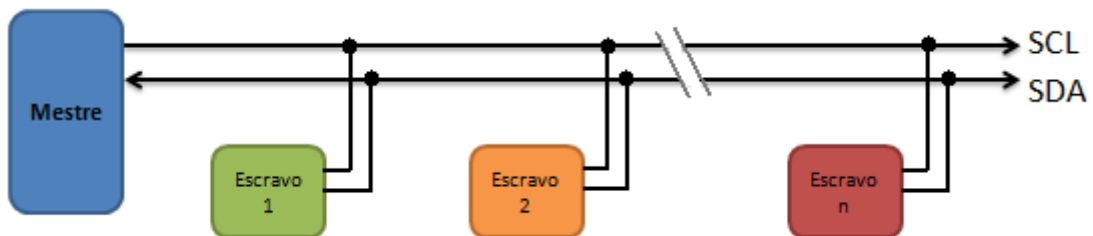


Figura 2.4: Dispositivos conectados a um barramento I²C.

- **Protocolo de comunicação I²C**

De forma geral, conforme apresentado em [16], o protocolo de comunicação I²C tem o seguinte fluxo:

1. O dispositivo mestre envia para o barramento I²C um sinal de *start*. Com isso, o dispositivo mestre tem a atenção de todos os dispositivos conectados ao barramento.
2. O dispositivo mestre envia um registro com o endereço que deseja acessar, e se deseja realizar leitura ou escrita. Todos os dispositivos irão receber este registro. Aqueles que não possuírem o endereço requisitado irão ignorar o registro e aguardar o sinal de *stop*. Aquele que tiver o endereço enviado irá responder com um sinal de confirmação.
3. Assim que o mestre receber o sinal de confirmação poderá iniciar a transmissão ou requisição dos dados. Assim que a transferência terminar, o mestre irá enviar um sinal de *stop*, que liberará o barramento para que outros dispositivos possam então atuar como mestre.

O sinal de *start* consiste em levar a linha SDA para o nível baixo, e depois a linha SCL para o nível baixo, conforme a Figura 2.5:

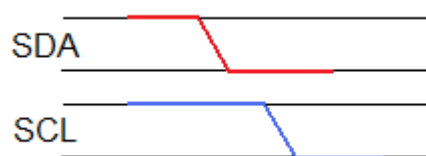


Figura 2.5: Sinal de *Start* do protocolo de comunicação I²C.

Os sinais de *clock* são gerados pelo dispositivo mestre através da linha SCL. Os dados são válidos apenas quando o sinal SCL estiver em nível alto.

O sinal de *stop* consiste em levar a linha SCL para o nível alto, e depois a linha SDA para o nível alto, conforme a Figura 2.6:

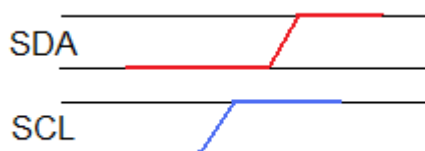


Figura 2.6: Sinal de *Stop* do protocolo de comunicação I²C.

O HMC6352 utiliza um protocolo em camadas, sendo o protocolo de interface definido pela especificação do barramento I²C e o protocolo inferior de comando definido pela Honeywell. O protocolo inferior de comando consiste em um conjunto de caracteres que são definidos pelo fabricante para que o usuário possa acessar certas funcionalidades do dispositivo, como por exemplo, entrar no modo de calibração do HMC6352.

A taxa de transmissão de dados é o modo padrão de 100kbps, tal como definido nas especificações do protocolo de comunicação I²C.

O formato dos bytes de dados deve ser caracteres em ASCII case sensitive ou dados em binário para o HMC6352 e dados de retorno.

O padrão de fábrica do HMC6352 para o endereço de 7 bits é 42 (hex) em hexadecimal para operações de escrita e 43 (hex) para operações de leitura.

2.2.5 – Funcionamento do HMC6352

O HMC6353 possui 4 terminais acessíveis ao usuário, sendo: 2 terminais para a alimentação do sensor e 2 para a comunicação I²C (SDA e SCL).

Para colocar o sensor em funcionamento, é necessário, primeiramente, conectá-lo ao Arduino Duemilanove. A tensão necessária para alimentar a bússola, é obtida através do pino de 5 Vcc disponibilizado pelo Arduino. O HMC6352 pode ser alimentado, também, através da tensão de 3,3 Vcc do Arduino, porém, como a tensão padrão para referência do conversor analógico digital (*ADC – Analog-to-digital converter*) do Arduino é de 5 Vcc, optou-se por alimentar o sensor com a

tensão padrão. Os terminais SDA e SCL são conectados respectivamente aos pinos 4 e 5 da entrada analógica pois, o Arduino Duemilanove só implementa o protocolo I²C nestes pinos. A Figura 2.7 apresenta o esquema da conexão entre o HMC6352 e o Arduino Duemilanove.

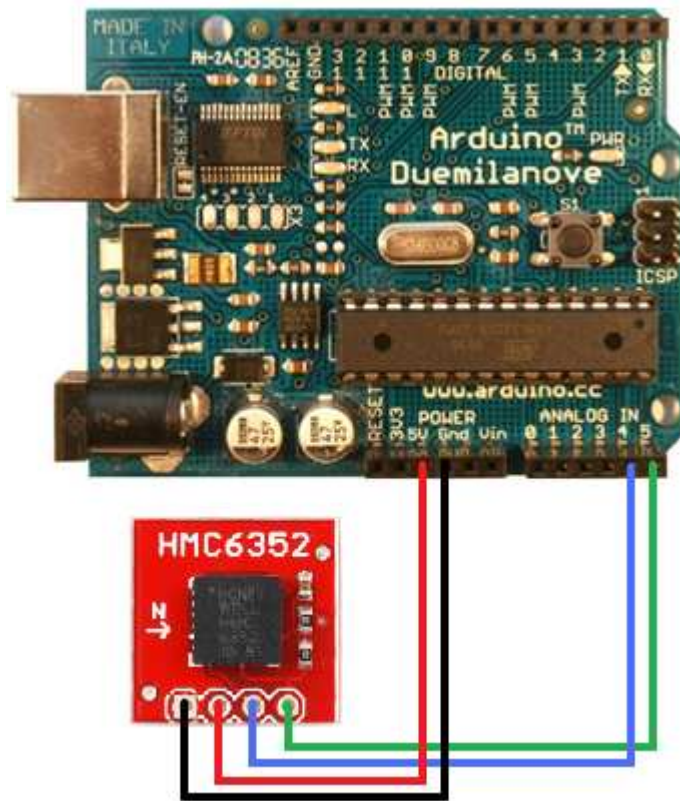


Figura 2.7: Esquemático de ligação do HMC6352 ao Arduino Duemilanove.

Para a leitura dos dados da bússola, foi desenvolvido o algoritmo apresentado na Tabela 2.3, que, fornece os dados de saída em graus (de 0° a 359,9°).

Tabela 2.3: Algoritmo para a leitura do HMC6352 desenvolvido no editor do Arduino.

```
#include <Wire.h> // biblioteca para comunicacao i2c

// o endereco i2c de acesso ao HMC6352 e 0x42
int ADDRESS=0x42 >> 1;
int heading;

void setup()
{
  Serial.begin(9600); // taxa de transmissao de dados

  // Inicializa o i2c
  Wire.begin();
}

void loop()
{
  // i2c mastre inicia a transmissao
  Wire.beginTransmission(ADDRESS);
  // envia o comando 'A', solicitando dados da bussola
  Wire.send("A");
  // encerra a transmissao
  Wire.endTransmission();
  // espera os dados serem processados
  delay(6); // aguarda 6ms

  // solicita 2 bytes a bussola
  Wire.requestFrom(ADDRESS, 2);
  // recebe os dados da bussola
  if (2 <= Wire.available()) // se os dois bytes estiverem disponiveis
  {
    // ajusta o byte mais significativo e soma
    heading = Wire.receive();
    heading = heading << 8;
    heading += Wire.receive();
  }

  // exibe o resultado final em graus
  Serial.println(heading/10.0);

  delay(100); // atraso ajustavel para a leitura dos dados
}
```

Para obter uma leitura correta dos dados, o HMC6352 deve ser mantido na posição horizontal, ou seja, paralelo ao solo, e deve estar o mais afastado possível de interferências magnéticas, tais como: motores elétricos, aparelhos de celular, materiais ferromagnéticos, e outras fontes de campos elétricos e magnéticos. Na

Tabela 2.4 têm-se os dados de saída da bússola (em graus), para um giro de 0 a 180°, apresentados no monitor serial do Arduino. Este giro foi realizado manualmente, rotacionando a bússola em torno de seu eixo vertical, apenas para a exibição da variação de seus resultados, de modo que não houve preocupação com a velocidade de rotação da mesma.

Tabela 2.4: Dados de saída da bússola apresentados no monitor serial do Arduino.

angulo em graus							
0.00	22.30	47.80	76.20	101.20	124.70	149.80	169.60
2.40	23.50	48.80	77.10	103.50	126.40	150.80	170.40
3.10	24.20	49.50	79.10	104.40	127.60	151.40	171.80
5.40	25.40	51.60	80.00	105.10	128.90	152.70	173.40
6.00	26.50	53.40	81.90	106.70	129.50	153.00	174.50
7.10	27.70	54.40	82.60	107.40	130.50	154.10	175.20
8.90	28.80	55.40	84.00	108.30	131.20	155.20	176.10
9.40	30.90	56.30	85.10	110.00	133.30	156.50	177.90
10.00	31.30	58.40	86.60	111.60	134.60	157.20	178.80
10.60	32.40	59.00	88.50	112.10	135.70	158.60	180.00
12.50	33.60	61.60	89.00	113.60	136.60	159.00	
13.20	34.80	63.40	90.00	114.20	138.00	160.60	
14.70	35.70	66.00	90.90	116.40	139.20	161.70	
15.20	37.60	67.00	92.20	117.30	139.70	162.30	
16.60	38.30	68.50	93.30	118.00	141.30	163.80	
17.60	40.20	69.30	95.40	119.30	142.70	164.00	
18.40	42.70	70.90	96.70	120.10	144.40	165.70	
19.40	43.50	71.20	98.60	121.60	145.70	166.60	
20.20	45.00	72.70	99.70	122.80	147.10	167.10	
21.00	46.30	73.10	100.70	123.50	148.70	168.40	

2.2.6 – Contador de Voltas

Conforme visto na Seção 2.2.5, o ângulo de orientação fornecido pela bússola é em graus e varia de 0° a 359,9°. Este ângulo é crescente à medida que se gira a bússola em torno de seu eixo vertical em sentido horário, e decrescente quando se gira em sentido anti-horário.

Supondo que a bússola esteja girando em sentido horário, o ângulo medido será crescente. Quando o ângulo medido chegar a 359,9°, a próxima medida será de 0° e continuará crescendo à medida que o sentido de rotação seja mantido (359,9°, 0°, 0,5°, 1°, 1,5°, 2°...). O mesmo ocorre quando a bússola gira em sentido anti-horário, o ângulo medido vai decrescendo. Quando chega a 0°, a próxima medida

será de $359,9^\circ$ e continuará decrescendo à medida que seja mantido o sentido de rotação (0° , $359,9^\circ$, $359,5^\circ$, 359° , $358,5^\circ\dots$).

Para o sistema de controle, esta variação brusca de $359,9^\circ$ para 0° , e vice-versa (0° para $359,9^\circ$), seria um grande problema, pois faria com que o mesmo se perdesse.

A fim de solucionar este problema, foi desenvolvido um algoritmo contador de voltas semelhante àquele utilizado em [2].

O contador de voltas é um algoritmo, implementado no Arduino, que percebe estas mudanças bruscas de ângulo (de 0° a $359,9^\circ$ e de $359,9^\circ$ a 0°) e compensa tais variações de forma a dar continuidade ao sentido de rotação da bússola. Ou seja, quando o ângulo chega a $359,9^\circ$, e o movimento é em sentido horário, o próximo valor medido será de $360,5^\circ$ e assim sucessivamente. Quando o ângulo está em 0° , e o movimento é anti-horário, o próximo valor medido será $-0,5^\circ$ e assim sucessivamente.

O algoritmo do contador de voltas encontra-se incorporado ao algoritmo do sistema de controle no Apêndice C.

2.3 – Girômetro

Girômetro é um sensor capaz de medir a velocidade de rotação (velocidade angular) em relação ao sistema de coordenadas inercial. Para medir a velocidade angular, geralmente os girômetros utilizam aceleração de Coriolis. A aceleração de Coriolis ocorre sobre um corpo com velocidade não nula apoiado sobre um segundo corpo que descreve um movimento de rotação, como por exemplo, um veículo se deslocando sobre a superfície da Terra [21].

Uma das tecnologias mais utilizadas baseia-se em uma massa vibrando linearmente no plano ortogonal ao eixo de medição de um corpo em repouso. Quando ocorre uma rotação, o corpo em movimento sofre a aceleração de Coriolis proporcional à sua velocidade de rotação.

Existe uma grande variedade de girômetros no mercado, podendo ser citados: os girômetros a fibra óptica e a laser. Grande parte destes sensores são triaxiais, ou seja, fornecem as velocidades angulares para os eixos x, y e z.

2.3.1 – Seleção do Girômetro

Para a seleção do girômetro foram consideradas as características mais relevantes à execução do projeto proposto. Estas características são:

- A compatibilidade com o Arduino Duemilanove.
- Medida da velocidade angular para os eixos x, y e z (triaxial).
- Tensão de alimentação.
- Taxa de amostragem.
- Dados de saída.
- Custo do sensor.

2.3.2 – Comparações e Conclusões

Foram pesquisados, em diferentes sites de componentes e equipamentos eletrônicos, alguns modelos de girômetros e Unidades de Medição Inercial (UMI), que são circuitos que contêm girômetro e acelerômetro integrados em uma só placa. Dentre os modelos pesquisados, foi escolhida uma placa com girômetro triaxial e uma unidade de medição inercial para efeito de comparação. A Tabela 2.5 apresenta os dois modelos selecionados e as suas principais características.

Tabela 2.5: Tabela comparativa dos Girômetros selecionadas.

Sensor	Graus de Liberdade	Formato de Saída	Fundo de Escala	Alimentação	Custo
Girômetro ITG-3200	3	I ² C	± 2000°/s	2,1 a 3,6 Vcc	R\$ 315,00
6 DOF Razor (Girômetro/Acelerômetro)	6	Analógico	± 300°/s (4x) ± 1200°/s (1x)	2,7 a 3,6 Vcc	R\$ 425,00

Pode-se observar na Tabela 2.5 que a placa com girômetro ITG-3200 apresenta fundo de escala maior e menor custo em relação ao 6DOF Razor, porém, visando o aperfeiçoamento da medida de velocidade angular em trabalhos futuros, optou-se pelo 6DOF Razor que, além de girômetros, possui acelerômetro.

2.3.3 – Características do 6DOF Razor

A Unidade de Medição Inercial 6DOF Razor (vide Figura 2.8) é composta pelos girômetros LPR530AL, que mede a velocidade de rotação nos eixos x e y (Arfagem e Rolagem), e LY530ALH que mede a velocidade de rotação no eixo z (Guinada) e pelo acelerômetro de 3 eixos ADX335 de modo a permitir 6 graus de medição em uma única placa.

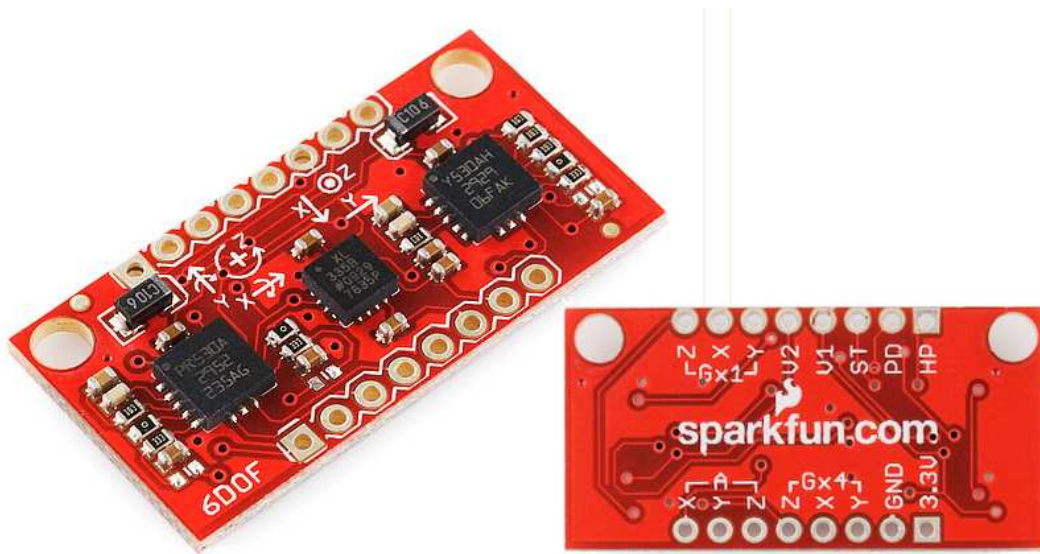


Figura 2.8: Unidade de Medição Inercial 6DOF Razor. Imagem extraída de [14].

Os girômetros fornecem a medição da velocidade angular através das suas saídas de tensão analógica, entre 0 Vcc e 3,3 Vcc, possuem amplificação de 1x e 4x, fundo de escala de $\pm 300^\circ/\text{s}$ e limitação de banda de -3dB acima de 140 Hz.

O acelerômetro é um dispositivo de escala reduzida de baixa potência. As medidas dos três eixos das acelerações são fornecidas através das suas saídas analógicas e operam com fundo de escala de $\pm 3g$, onde g é a aceleração da gravidade.

A Tabela 2.6 apresenta um resumo das principais características do 6DOF Razor.

Tabela 2.6: Principais características do 6DOF Razor.

Características do 6DOF Razor	
Alimentação	2,7 a 3,6 Vcc
Graus de liberdade do Girômetro	x, y e z
Amplificação das saídas do Girômetro	1x e 4x
Sensibilidade	0,83 mV/°/s para 1x e 3,33 mV/°/s para 4x
Faixa de medição	± 300 °/s (4x) e ±1200 °/s (1x)
Graus de liberdade do Acelerômetro	x, y e z
Sensibilidade	300 mV/g
Faixa de medição	± 3g
Acesso aos pinos	Auto-teste, desligar, filtro passa-alta, reset e referência

2.3.4 – Funcionamento do 6DOF Razor

A Unidade de Medição Inercial 6DOF Razor dispõe de 16 pinos, sendo: 2 para alimentação (positivo e negativo), 3 saídas analógicas (eixos x, y e z) do acelerômetro, 3 saídas analógicas sem amplificação (eixos x, y e z) do girômetro, 3 saídas analógicas amplificadas (4x) do girômetro, 1 pino de *auto-teste*, 1 de desliga, 1 de filtro passa-alta, 1 de *reset* e 1 de referência.

Para realizar o teste da placa, foram utilizadas as saídas amplificadas do girômetro (4x) e o acelerômetro, deixando os pinos de *auto-teste*, desligar, filtro passa-altas e *reset*, bem como as saídas não amplificadas do girômetro, em aberto.

O próximo passo foi conectar o 6DOF ao Arduino Duemilanove. As saídas analógicas (y, x e z) do girômetro foram conectadas respectivamente às entradas analógicas 0, 1 e 2 do Arduino, assim como, as saídas analógicas (z, y e x) do acelerômetro foram conectadas respectivamente nas entradas analógicas 3, 4 e 5. A tensão necessária para alimentar os sensores é obtida através do pino de 3,3 Vcc do Arduino. O pino de referencia do 6DOF, também foi conectado ao pino de tensão de 3,3Vcc. Isto foi feito, a fim de dimensionar corretamente o conversor analógico digital do 6DOF Razor uma vez que, a tensão padrão para referência analógica do Arduino é de 5 Vcc. A Figura 2.9 apresenta o esquema de ligação do 6DOF Razor ao Arduino Duemilanove.

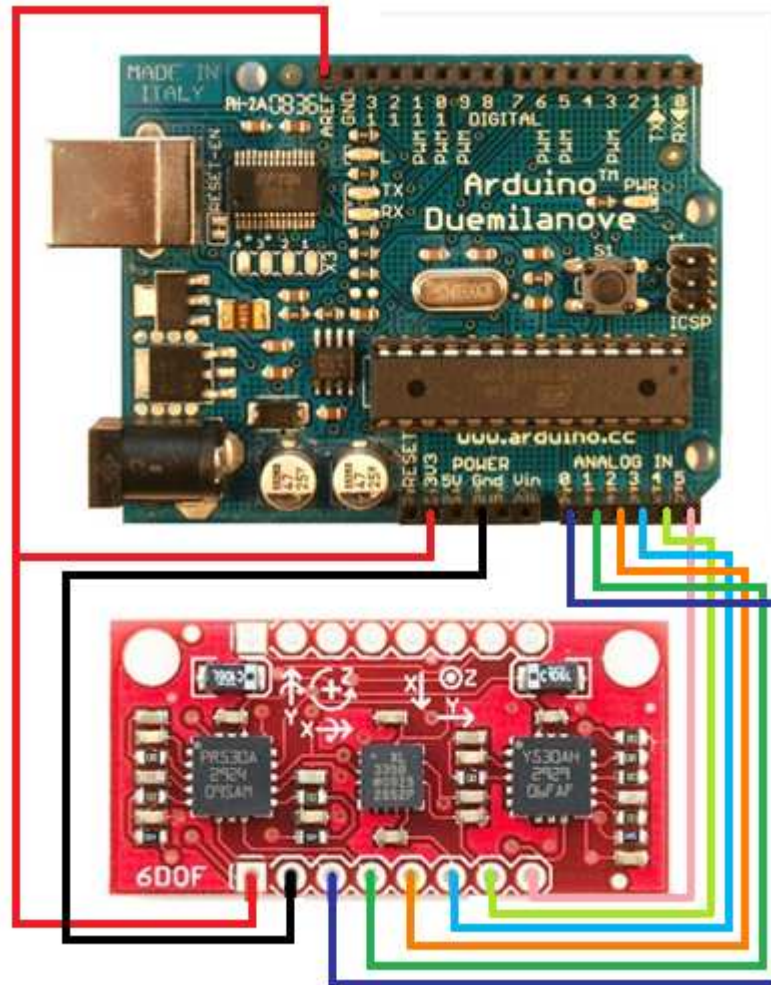


Figura 2.9: Esquemático de interligação da Unidade de Medição Inercial ao Arduino Duemilanove.

Tanto o acelerômetro, quanto as saídas x e y do girômetro, foram interligados ao Arduino apenas para efeito de testes, visto que, para a implementação do controle de rumo, apenas a medida de velocidade angular fornecida pela saída z do girômetro é necessária, uma vez que, a embarcação encontra-se no plano XY e desconsiderou-se o seu balanço.

Para a leitura dos sensores, foi desenvolvido o algoritmo apresentado na Tabela 2.7.

Tabela 2.7: Algoritmo para a leitura do 6DOF Razor desenvolvido no editor do Arduino.

```
int val = 0;

void setup()
{
  analogReference(EXTERNAL); //tensão referência externa de 3.3V
  Serial.begin(9600);        //taxa de transmissão de dados

  Serial.println("gy \t gx \t gz \t az \t ay \t ax "); //cabecalho
}

void loop()
{
  for (long i=0; i<6; i++) //lê os dados do girômetro e acelerômetro
  {
    val = analogRead(i);    //lê as entradas analógicas
    Serial.print(val);      //apresenta os dados no monitor
    Serial.print ("\t");
  }

  Serial.println("");
  delay(16);
}
```

A Tabela 2.8 apresenta os dados de saída dos sensores em repouso sob uma superfície plana onde, gy, gx e gz são os valores apresentados pelo girômetro para os eixos y, x e z, e az, ay e ax são os dados fornecidos pelo acelerômetro para os eixos z, y e x.

Tabela 2.8: Dados de saída do 6DOF Razor apresentados no monitor serial do Arduino.

gy	gx	gz	az	ay	ax
363	364	372	608	502	506
363	366	370	607	502	506
363	363	370	608	502	505
363	364	372	607	502	505
364	369	372	607	502	505
361	366	374	610	503	509
363	365	372	608	503	506
362	364	373	608	502	506
363	365	372	608	502	505
362	365	371	611	505	505
363	364	371	607	501	505
366	365	371	608	502	505
363	365	371	607	502	505
363	365	370	607	502	506
364	365	372	608	503	505

Como se pode observar, os valores estão em sua forma bruta (fornecidos pelo ADC) logo, faz-se necessário transformá-los em dados utilizáveis.

Como já mencionado, os sensores são alimentados por uma tensão de 3,3Vcc. Esta é a tensão máxima que se pode ter na saída dos sensores de maneira que, os dados fornecidos por eles, podem variar apenas na faixa de 0 a 3,3Vcc.

O girômetro mede a velocidade angular em graus por segundo (°/s) e, de acordo com o manual da Honeywell, apresenta sensibilidade de 3,33mV/°/s. Já o acelerômetro, mede a aceleração gravitacional (em g) e sua sensibilidade é de 330mV/g. O ADC fornece um valor, proporcional à tensão que recebe, de 0 a 1023 níveis discretos (10 bits de resolução). Como a tensão de referencia utilizada é de 3,3V, e esta tensão dimensiona o ADC, o nível 0 equivale à 0 V, o nível 512 equivale a 1,65V e o nível 1023 equivale a 3,3V. Como esta variação é linear, a matemática torna-se fácil de maneira que a equação básica do girômetro, para fazer a conversão, é dada por:

$$\omega = (V_g - V_{g0}) \times \frac{V_{ref}}{(2^{ADC} - 1)S_g} \quad (2.1)$$

No qual,

ω é a velocidade angular em graus por segundo (°/s)

V_g é o valor ADC do girômetro

V_{g0} é o valor ADC do girômetro em repouso

V_{ref} é a tensão de referência externa (3,3V)

r_{ADC} é o número de bits do ADC ($r_{ADC} = 10$)

s_g é a sensibilidade do girômetro (3,33mV/°s)

Como exemplo, na Tabela 2.8 pode-se observar que o valor ACD do girômetro (em repouso), para o eixo z, é de 372. Durante uma rotação do girômetro, o mesmo apresentou para o eixo z, dentre diversos valores, o valor 411. Aplicando estes valores na equação (2.1), tem-se a velocidade angular em graus por segundo:

$$\omega = (411 - 372) \times \frac{3,3V}{(2^{10} - 1)0,00333V/^\circ/s}$$

$$\omega = 37,780^\circ/s$$

Analogamente, para o acelerômetro tem-se:

$$G = (v_a - v_{a0}) \times \frac{V_{ref}}{(2^{r_{ADC}} - 1)s_a} \quad (2.2)$$

Onde,

G é a aceleração gravitacional em (g)

v_a é o valor ADC do acelerômetro

v_{a0} é o valor ADC do acelerômetro em repouso

V_{ref} é a tensão de referência externa (3,3V)

r_{ADC} é o número de bits do ADC ($r_{ADC} = 10$)

s_a é a sensibilidade do acelerômetro (330mV/g)

Para obter a aceleração gravitacional em graus, basta calcular o arco seno de G em (2.2).

A Figura 2.10 apresenta a interligação da bússola e girômetro ao Arduino Duemilanove.

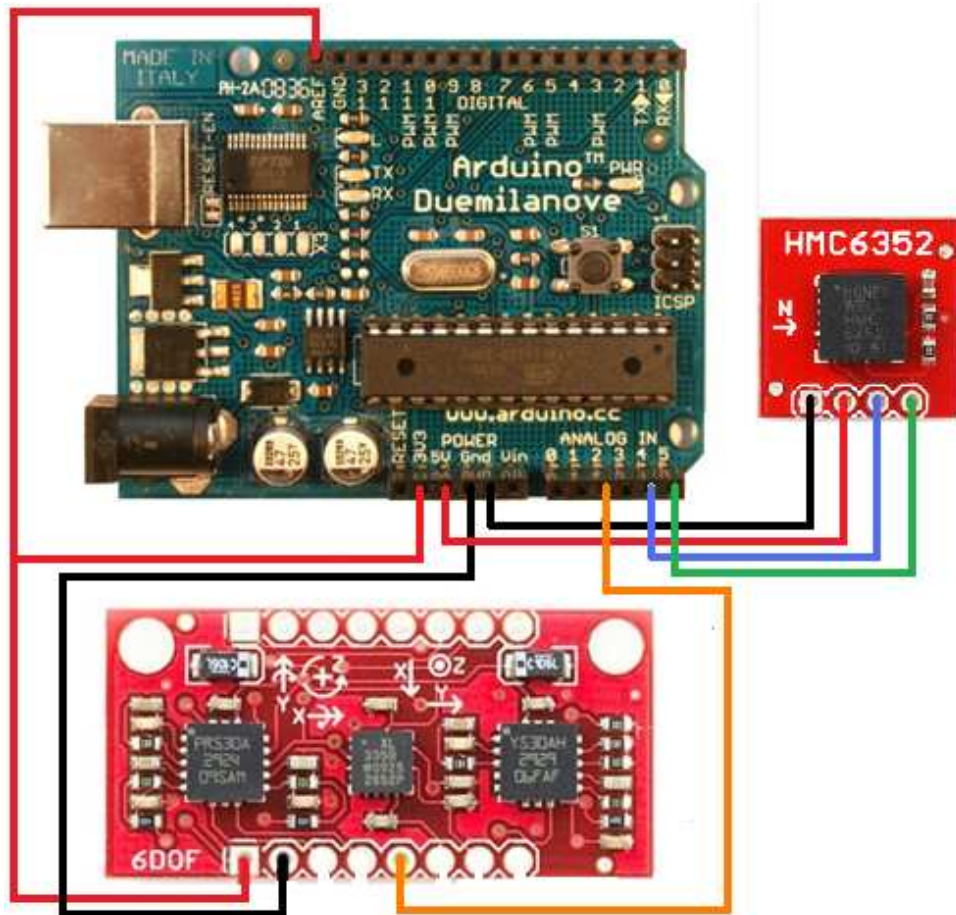


Figura 2.10: Esquemático de interligação da bússola e girômetro ao Arduino Duemilanove.

CAPÍTULO 3 – CONTROLE

O controle desenvolvido neste capítulo tem por objetivo manter a embarcação em orientação determinada em meio a perturbações externas e, para isto, será utilizado o controle a estrutura variável que será descrito na Seção 3.3.

3.1 – Sistemas de Coordenadas

Conforme apresentado em [26], os sistemas de coordenadas são utilizados para referenciar pontos no espaço.

Para se desenvolver a modelagem matemática da dinâmica da embarcação e o seu sistema de controle, foram adotados os sistemas de coordenadas inercial e móvel.

- **Sistema de coordenadas inercial**

É o sistema de coordenadas estacionário. Será representado por:

O_i – origem do sistema de coordenadas inercial;

x_i, y_i - eixos do sistema de coordenadas inercial;

z_i - eixo vertical do sistema de coordenadas inercial. Normalmente aponta para cima e é ortogonal ao plano $x_i O_i y_i$.

- **Sistema de coordenadas móvel (fixo ao veículo)**

É um sistema em que a origem é fixa em um veículo, preferencialmente no seu centro de gravidade [27]. Serão representados por:

O – origem do sistema de coordenadas móvel;

x, y, z – eixos do sistema de coordenadas móvel, onde, x aponta para frente, y aponta para a direita e z normalmente aponta para baixo.

A Figura 3.1 apresenta a embarcação e os sistemas de coordenadas adotados.

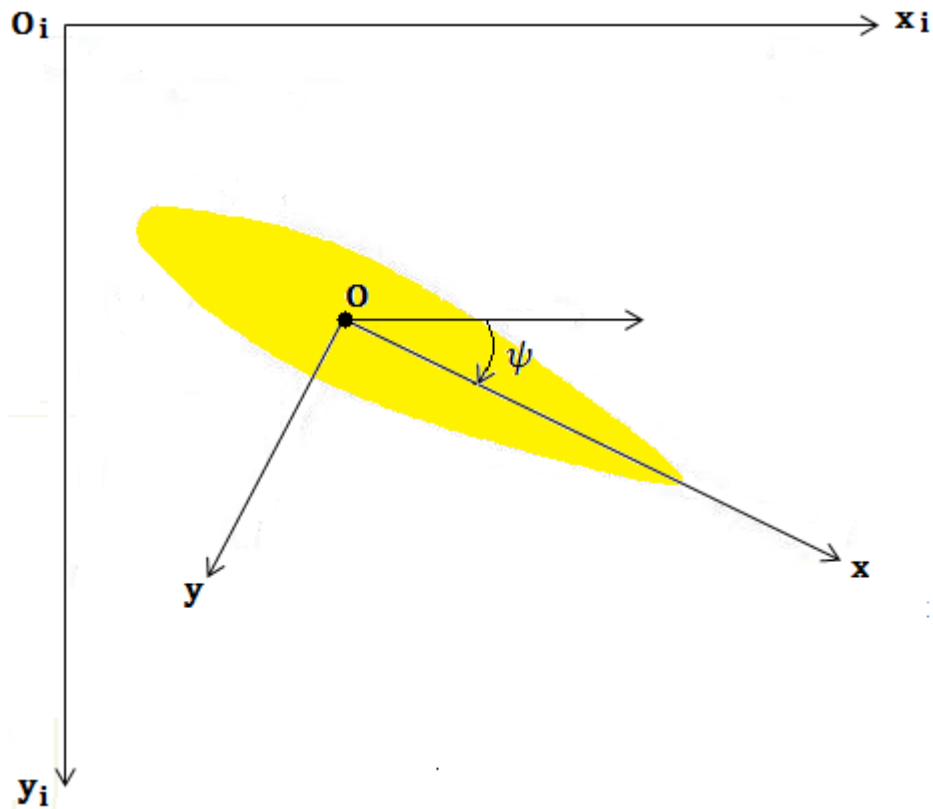


Figura 3.1: Sistemas de coordenadas adotados para a embarcação.

Neste projeto o balanço da embarcação é desprezado de maneira que os eixos z_i e z (que não estão representados na Figura 3.1) são paralelos.

3.2 – Modelagem Matemática da Dinâmica do Sistema

A modelagem matemática de um sistema dinâmico consiste na obtenção de um conjunto de equações que descrevem de forma precisa ou razoavelmente bem o comportamento do sistema. Um sistema físico pode ser representado de diversas maneiras e, portanto, pode ter vários modelos matemáticos diferentes, os quais, usualmente, são descritos em termos de equações diferenciais.

Para uma modelagem matemática mais precisa do sistema, um número maior de equações podem ser consideradas para reproduzir de forma exata o comportamento do mesmo, porém sua análise torna-se mais complexa.

Para obtenção de um modelo matemático relativamente simplificado, torna-se necessário ignorar certas propriedades físicas inerentes ao sistema.

Tendo em vista que o controle a estrutura variável será utilizado para realização do controle da dinâmica do sistema, por proporcionar alto grau de robustez as dinâmicas não modeladas e incertezas paramétricas [1], não há a necessidade de se desenvolver uma modelagem complexa.

3.2.1 – Equação da dinâmica de rotação

Para a modelagem da dinâmica do sistema é necessário definir quais variáveis serão de entrada e quais serão de saída, de maneira a se obter, através de leis físicas, uma equação diferencial que relacione tais variáveis.

Como o objetivo é o controle de rumo da embarcação, definiu-se a entrada como $M_p \in \mathbb{R}$, sendo M_p o momento produzido pelos propulsores no eixo vertical da embarcação, e as saídas são a velocidade angular ω e o ângulo de rumo da embarcação ψ .

Define-se o momento angular de um corpo em rotação como:

$$L = J \omega, \quad (3.1)$$

onde J é o momento de inércia do corpo e ω é a velocidade angular.

A variação do momento angular com o tempo é igual ao momento da resultante das forças que atuam sobre corpo sólido rígido:

$$\frac{dL}{dt} = M_p, \quad (3.2)$$

Substituindo (3.1) em (3.2), tem-se:

$$\frac{d(J \omega)}{dt} = M_p \quad \therefore \quad \dot{\omega} = \frac{M_p}{J}, \quad (3.3)$$

onde $\dot{\omega}$ é a aceleração angular do sistema.

Como o sistema está susceptível a perturbações externas, sobre as quais não se pode atuar diretamente, a equação (3.3) passa a ser:

$$\dot{\omega} = \frac{1}{J} (M_p + d), \quad (3.4)$$

onde a variável d representa os distúrbios externos, como por exemplo, o efeito do arraste, parâmetros hidrodinâmicos da embarcação e perturbações externas (provocados pela ação de ventos, correntes, ondas, etc). Desta forma, a equação (3.4) representa o modelo matemático do sistema.

Na Figura 3.2 tem-se a representação em diagrama de blocos da dinâmica simplificada do sistema.

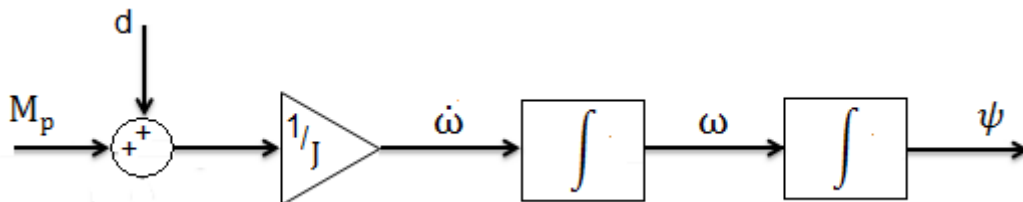


Figura 3.2: Diagrama em blocos do modelo simplificado da dinâmica do sistema.

A dinâmica do sistema pode ser representada, também, na forma de equações no Espaço de Estado:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)u(t) \\ \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + D(t)u(t) \end{cases} \quad (3.5)$$

Sendo:

$u(t)$, $\in \mathbb{R}^m$, é o sinal de controle do sistema;

$\mathbf{y}(t)$, $\in \mathbb{R}^p$, é o sinal de saída do sistema;

$\mathbf{x}(t)$, $\in \mathbb{R}^n$, é o estado e $\dot{\mathbf{x}}(t)$ sua derivada;

$\mathbf{A}(t)$, de ordem $n \times n$, é dita a matriz de estado;

$\mathbf{B}(t)$, de ordem $n \times m$, é a matriz de entrada;

$\mathbf{C}(t)$, de ordem $p \times n$, é a matriz de saída;

$D(t)$, de ordem $p \times m$, a matriz de transmissão direta.

Assim, para a dinâmica da embarcação tem-se:

$$\mathbf{x} = \begin{bmatrix} \psi \\ \omega \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \psi \\ \omega \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1/J \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = 0, \quad (3.6)$$

Substituindo (3.6) em (3.5) têm-se, respectivamente, as equações de estado e de saída da embarcação:

$$\begin{bmatrix} \dot{\omega} \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1/J \end{bmatrix} (M_p + d), \quad (3.7)$$

$$\begin{bmatrix} \psi \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \omega \end{bmatrix}, \quad (3.8)$$

3.2.2 – Modelagem matemática dos propulsores

Na modelagem matemática da dinâmica do sistema não se considerou o efeito dos propulsores. Optou-se por realizar a modelagem da dinâmica da embarcação e dos propulsores separadamente por ser mais simples de trabalhar.

A equação

$$F_{pi} = \begin{cases} F_{pmáx} & \text{se } u_{pi} > 0 \\ 0, & \text{se } u_{pi} = 0, \\ -F_{pmáx} & \text{se } u_{pi} < 0 \end{cases}, \quad (3.9)$$

representa o modelo dos propulsores, onde $i = 1, 2$ é o número do propulsor, $F_{pmáx}$ e $-F_{pmáx}$ são as forças geradas pelos propulsores em regime permanente e u_{pi} , o sinal de comando dos propulsores.

Na Equação (3.9) pode-se observar que os propulsores possuem dois níveis de velocidade (máxima e nula), e operam em modo de chaveamento (direto-parado-reverso). Este modo foi escolhido devido ao acionamento dos propulsores serem realizados por relés de acionamento simples e duplo, conforme apresentado no Apêndice A.

A ação conjunta das forças produzidas pelos propulsores resulta no momento realizado pelos propulsores no eixo vertical da embarcação, dado por:

$$M_p = \sum_{i=1}^n \mathbf{r}_{pi} \times \mathbf{F}_{pi} \quad (3.10)$$

onde:

n é o número de propulsores ($n = 2$);

\mathbf{r}_{pi} é a distância entre os propulsores e o ponto de aplicação da força;

\mathbf{F}_{pi} é a força gerada pelo propulsor.

3.3 – Controle a Estrutura Variável

O fato dos propulsores operarem em modo de chaveamento, e destes serem acionados por relés de acionamento simples e duplo, fez com que se tornasse inadequada à utilização de controladores do tipo proporcional, integral e derivativo (PID), fazendo com que o controle a estrutura variável (VSC) fosse empregado no controle de rumo da embarcação. De acordo com [1], o VSC em modo de deslizamento proporciona um alto grau de robustez quando aplicado a sistemas com incertezas paramétricas e apresenta grande capacidade de rejeição a perturbações.

O controle a estrutura variável pode ser interpretado como um tipo especial de técnica de controle não linear, onde sua principal característica, é que, uma vez que o modo deslizante tenha sido alcançado, o desempenho do sistema torna-se insensível a incertezas paramétricas da planta e a algumas classes de perturbações externas [24]. Esta característica é conhecida como propriedade de invariância [25].

O modo deslizante é o movimento feito pelo sistema devido ao chaveamento de natureza descontínua da ação de controle. De acordo com [7], o modo de

deslizamento acontece quando o sinal de controle, como função dos estados do sistema, é chaveado em alta frequência.

Uma vez que o sistema entra em modo deslizante, a dinâmica do sistema passa a ser regida pela dinâmica correspondente a superfície de deslizamento, denotada por $s(x) = 0$, que é escolhida pelo projetista.

Para o projeto de um VSC, escolhe-se uma superfície no espaço de estado para ser uma superfície de deslizamento, a qual é definida como uma função linear das variáveis de estado do sistema. Considerando que a lei de controle que gera o sinal de comando seja escolhida adequadamente, a superfície de deslizamento é alcançada em algum tempo finito t_s . A partir daí, o comportamento do sistema depende apenas da superfície escolhida, independente dos parâmetros da planta [3].

No deslizamento ideal, a trajetória do sistema fica totalmente confinada à superfície de deslizamento, e isso ocorre quando não há atrasos no chaveamento, ou seja, a mudança do sinal do controle ocorre no instante que a trajetória do sistema alcança a superfície de deslizamento, de modo a impedir que a trajetória do sistema a ultrapasse. Porém, os sistemas reais apresentam limitações que fazem com que estes atrasos existam, de modo que, a trajetória do sistema oscile em torno da superfície de deslizamento. A estas oscilações dá-se o nome de *chattering*.

3.3.1 – Controle a Estrutura Variável Aplicado à Embarcação

Para o controle da embarcação, definiu-se a seguinte superfície de deslizamento:

$$s = \lambda e + \dot{e}, \text{ onde } (\lambda > 0) \quad (3.10)$$

Da equação (3.10) tem-se:

$$e = \psi_{ref} - \psi \quad (3.11)$$

$$\dot{e} = -\dot{\psi} = -\omega, \quad (3.12)$$

onde e é o erro de rumo da embarcação e \dot{e} a sua derivada, sendo ψ_{ref} a posição de referência, ψ a posição atual e $\dot{\psi}$ (ou ω) a velocidade angular.

Após o início do modo deslizante em algum $t = t_s$, a trajetória do sistema fica restrita à superfície de deslizamento (definida como $s(\psi) = 0$). Desta forma, a equação (3.10) será representada por:

$$\lambda e + \dot{e} = 0 \quad (3.13)$$

A solução da equação diferencial (3.13) é dada por:

$$e(t) = e(t_s)e^{-\lambda(t-t_s)}, \quad t \geq t_s, \quad (3.14)$$

Onde, pode-se observar que o erro $e(t)$ decai exponencialmente e a sua taxa de decaimento depende essencialmente da constante lambda ($\lambda > 0$).

Para um grau de liberdade (ψ), definiu-se a lei de controle que gera o sinal de comando como:

$$u = \text{sgn}(s), \quad (3.15)$$

onde

$$\text{sgn}(s) = \begin{cases} 1, & \text{se } s > 0 \\ 0, & \text{se } s = 0 \\ -1, & \text{se } s < 0 \end{cases}$$

3.3.2 - Sinal de comando no sistema de coordenada móvel

Tendo em vista que o sinal de comando do controlador é representado no sistema de coordenadas móvel e que o erro de orientação no sistema de coordenadas móvel, definido por e_ψ , é igual ao representado no sistema de

coordenadas inercial, \bar{e}_ψ , não se faz necessária à transformação de coordenadas.

Desta forma temos:

$$e_\psi = \bar{e}_\psi \quad (3.16)$$

O mesmo se aplica para a derivada do erro de orientação em função do tempo

$$\dot{e}_\psi = \dot{\bar{e}}_\psi, \quad (3.17)$$

onde \dot{e}_ψ é a derivada do erro de orientação no sistema de coordenadas móvel e $\dot{\bar{e}}_\psi$ no sistema de coordenadas inercial.

Desta forma, o sinal de comando do controlador no sistema de coordenadas móvel é representado por:

$$\begin{aligned} u_\psi &= \text{sgn}[\lambda \bar{e}_\psi + \dot{\bar{e}}_\psi] \\ u_\psi &= \text{sgn}[s_\psi], \end{aligned} \quad (3.18)$$

3.3.3 – Sinais de comando dos propulsores

O sistema de controle tem por objetivo manter a embarcação em orientação definida, estando à mesma descrevendo um movimento retilíneo. Devido à disposição definida para os propulsores na embarcação, a qual pode ser observada na Figura 3.3, o ângulo de rumo (ψ) e o deslocamento retilíneo serão executados pelos mesmos propulsores (p_1 e p_2) de forma, que seja necessário à utilização de um método de desacoplamento para os sinais de comando dos propulsores.

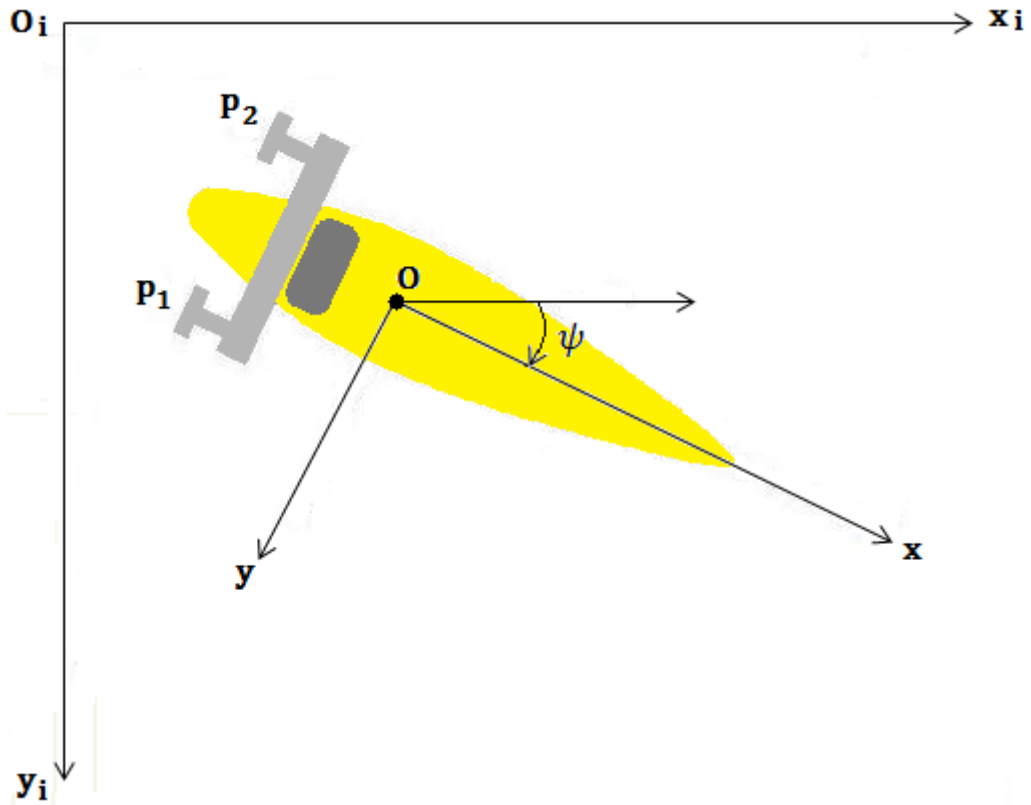


Figura 3.3: Disposição dos propulsores na embarcação.

O método de desacoplamento consiste em se obter os sinais de comando dos propulsores a partir dos sinais de comando do controlador [3].

Para que o barco se desloque em linha reta, basta que ambos os propulsores, p_1 e p_2 sejam acionados no mesmo sentido de rotação, ou seja, $u_{p1} = u_{p2}$ e mesma velocidade. Já, para que o barco possa realizar um movimento de rotação em torno do eixo z, é necessário que os propulsores sejam acionados em sentidos contrários, $u_{p1} \neq u_{p2}$, com mesma velocidade. Para que o veículo realize ambos os movimentos (retilíneo e rotação), basta que um dos propulsores esteja ligado em sentido direto e o outro desligado (e vice-versa). Desta forma, para que o sistema de controle pudesse conjugar os movimentos, retilíneo e de rotação, fez-se necessário desenvolver uma lógica de acionamento para os propulsores p_1 e p_2 como apresentada a seguir.

$$U_p = \text{sgn}\left(\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u_f \\ u_\psi \end{bmatrix}\right), \quad (3.19)$$

onde $U_p = [u_{p1} \quad u_{p2}]^T$ e u_f é um sinal de comando que está sempre em nível 1 de forma a fazer com que o barco sempre se mova para a frente.

Multiplicando as matrizes, obtêm-se os sinais de comando dos propulsores:

$$u_{p1} = \text{sgn}(u_f - u_\psi), \quad (3.20)$$

$$u_{p2} = \text{sgn}(u_f + u_\psi), \quad (3.21)$$

Desta forma, quando u_ψ for negativo ($u_\psi < 0$), o propulsor p_1 estará ligado em sentido direto com velocidade máxima, e o propulsor p_2 estará desligado, fazendo com que o barco descreva um movimento para frente e ao mesmo tempo para a esquerda;

Quando u_ψ for positivo ($u_\psi > 0$), o propulsor p_1 estará desligado, e o propulsor p_2 estará ligado em sentido direto com velocidade máxima, fazendo com que o barco descreva um movimento para frente e ao mesmo tempo para a direita;

Quando u_ψ for zero ($u_\psi = 0$), ambos propulsores estarão ligados em sentido direto com velocidade máxima ($u_{p1} = u_{p2}$), de maneira que o barco se desloque em linha reta para frente.

A Figura 3.4 apresenta o diagrama de blocos do sistema de controle.

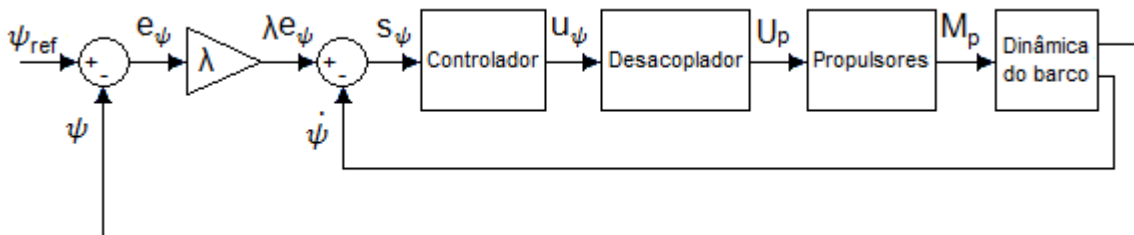


Figura 3.4: Diagrama de blocos do sistema de controle.

3.4 – Implementação do Controle

O controle será realizado pelo Arduíno Duemilanove. O código da rotina de controle encontra-se no Apêndice C.

CAPÍTULO 4 – SIMULAÇÕES, TESTES E RESULTADOS EXPERIMENTAIS

Este capítulo apresenta às simulações do comportamento do sistema para o controle a estrutura variável projetado e apresenta os resultados experimentais dos testes realizados em campo.

4.1 – Simulação do Controlador a Estrutura Variável

A fim de validar tudo o que foi apresentado até o momento, foram realizadas simulações, através do Simulink, do comportamento do sistema para diferentes condições as quais este foi submetido. Os testes consistiram em inserir diferentes condições iniciais e diferentes frequências de amostragem, de maneira a estudar o comportamento do sistema para o controle de rumo projetado. A Figura 4.1 apresenta o diagrama em blocos do sistema de controle implementado no Simulink.

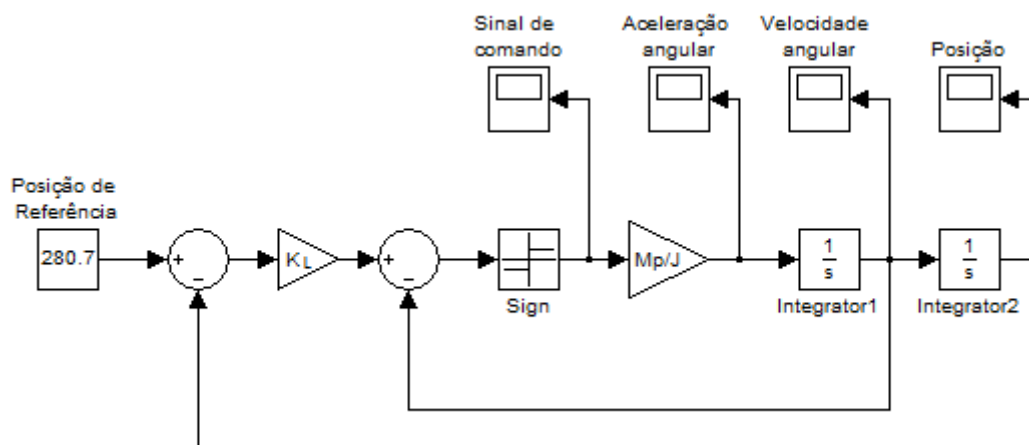


Figura 4.1: Diagrama em blocos do VSC desenvolvido no Simulink.

O bloco de ganho K_L representa a constante λ da equação (3.10). Foram realizados diversos testes do algoritmo de controle em laboratório. Alguns testes consistiram em simular os movimentos da embarcação para diferentes valores da constante λ , onde, os dados de entrada eram enviados para o microcontrolador através da unidade de medição (constituída pela bússola e girômetro), e o sistema

de propulsão era feito manualmente, rotacionando a unidade de medição, em torno do seu eixo vertical, em sentido horário e anti-horário, conforme os sinais de comando dos propulsores apresentados no monitor serial do Arduino. Através da comparação das diferentes respostas do algoritmo de controle, para os diferentes valores de λ , pôde-se observar que as variações de ângulo, em torno do ângulo de referência, eram menores para $\lambda = 2,0 \text{ rad/s}$ ou $114,6\%s$.

A razão M_p/J , na entrada da planta, representa a aceleração angular do sistema, onde M_p é o momento produzido pelos propulsores no eixo vertical da embarcação, e J o momento de inércia do sistema. A aceleração angular do sistema foi determinada através da derivada da velocidade angular, medida nos experimentos, durante um trecho em que os sinais de comando dos propulsores se mantiveram constantes. O valor da aceleração angular utilizada nas simulações foi de $0,81 \text{ rad/s}^2$ ou $46,4\%s^2$.

Considerando-se a condição inicial $\psi(0) = 237,7^\circ$ e o sinal de referência $\psi_{\text{ref}} = 280,7^\circ$ para o diagrama em blocos da Figura 4.1, obtiveram-se os resultados a seguir (Figuras 4.2 e 4.3):

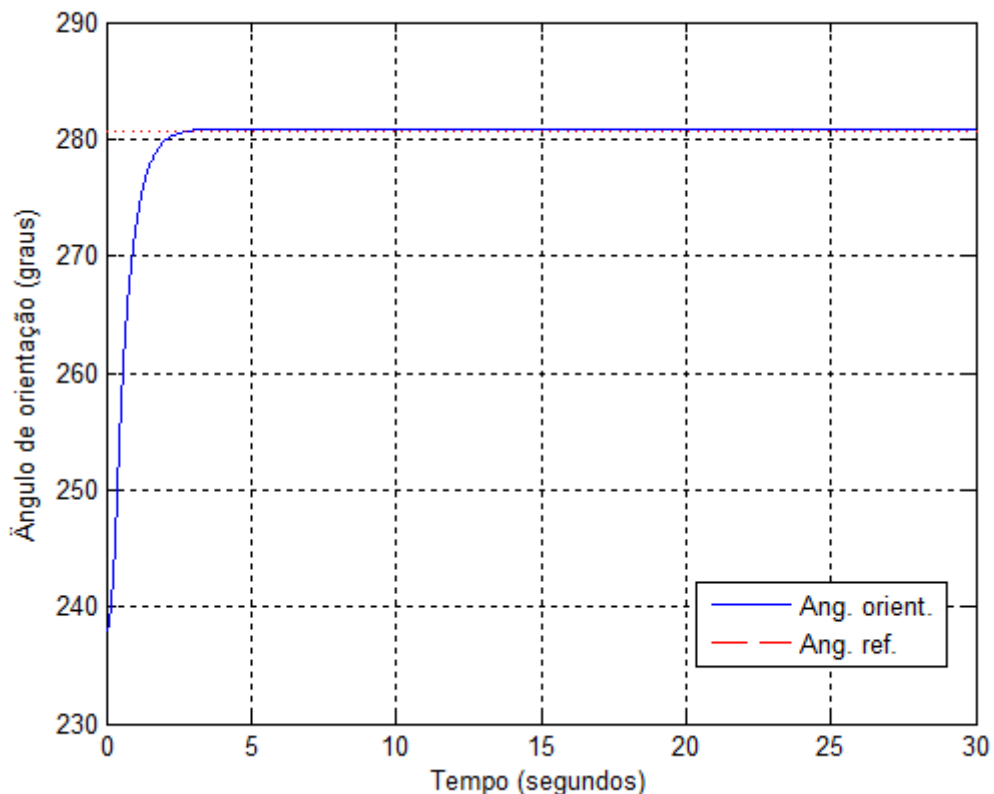


Figura 4.2: Simulação do comportamento do sistema para $\psi_{\text{ref}} = 280,7^\circ$.

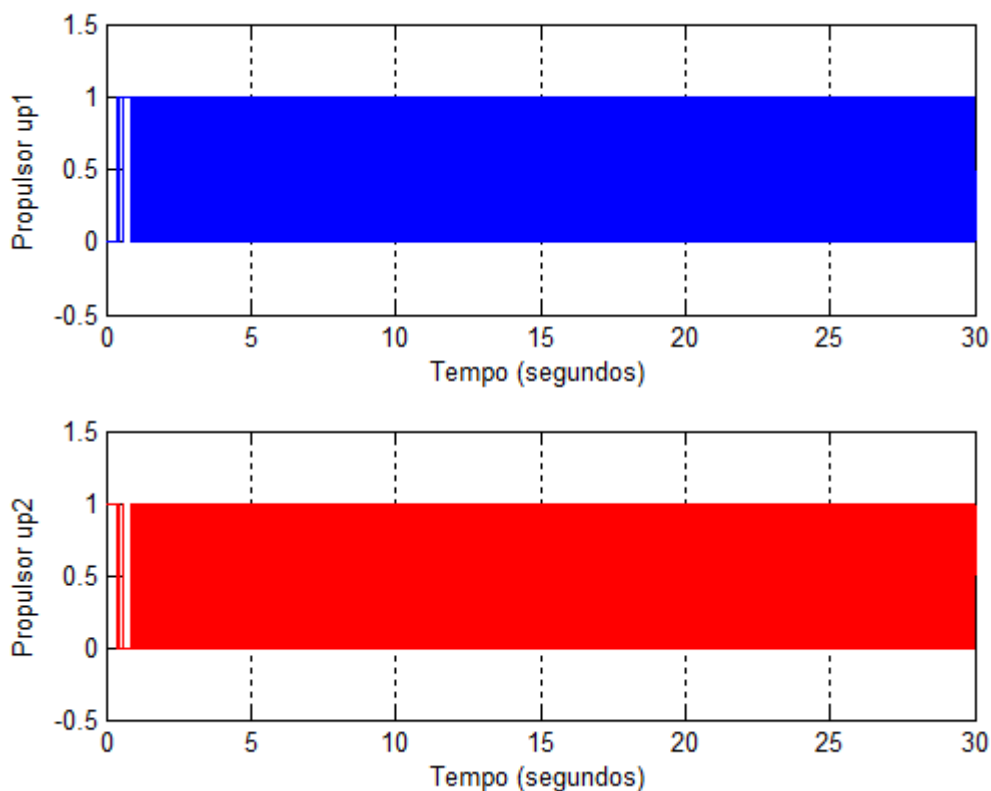


Figura 4.3: Sinais de comando dos propulsores da simulação do controle VSC.

A fim de se comparar os resultados das simulações com os experimentais, escolheu-se a condição inicial $\psi(0) = 237,7^\circ$ e a orientação de referência $\psi_{\text{ref}} = 280,7^\circ$ por terem sido as mesmas condições utilizadas nos testes em campo.

Pode-se observar na Figura 4.2, que o controle projetado está funcionando como esperado, visto que, o sistema convergiu para a orientação desejada.

A Figura 4.3 apresenta os sinais de comando dos propulsores em dois níveis, 0 e 1. Estes níveis indicam se os propulsores estão desligados (nível 0) ou ligados (nível 1).

4.2 – Simulação do Controle com Sinais Amostrados

Para a medição da posição e velocidade angular do sistema, são utilizados sensores que possuem frequências de amostragem de sinal definidas, como por exemplo, a frequência de amostragem máxima da bússola é de 20 Hz. Desta forma, as informações de posição e velocidade angular não estarão disponíveis em

qualquer instante de tempo, sendo necessária a implementação de um controle com sinais amostrados [2]. Para fazer a amostragem do controlador a estrutura variável foi adicionado um *ZOH* (*Zero Order Hold*) na entrada da planta e um em cada saída dos integradores. A Figura 4.4 apresenta o diagrama em blocos do controlador com sinais amostrados.

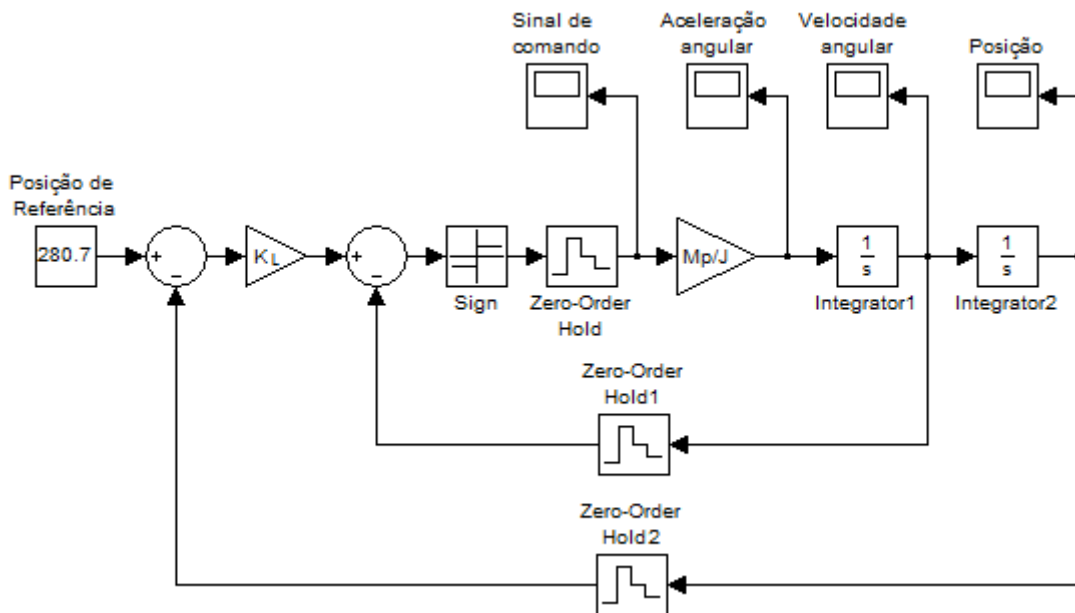


Figura 4.4: Diagrama em blocos do VSC para sinais amostrados desenvolvido no Simulink.

Logo, para a condição inicial $\psi(0) = 237,7^\circ$, sinal de referência $\psi_{ref} = 280,7^\circ$ e considerando uma frequência de amostragem de 10 Hz ($h = 0,1$ s), obtiveram-se os resultados a seguir (Figuras 4.5 e 4.6):

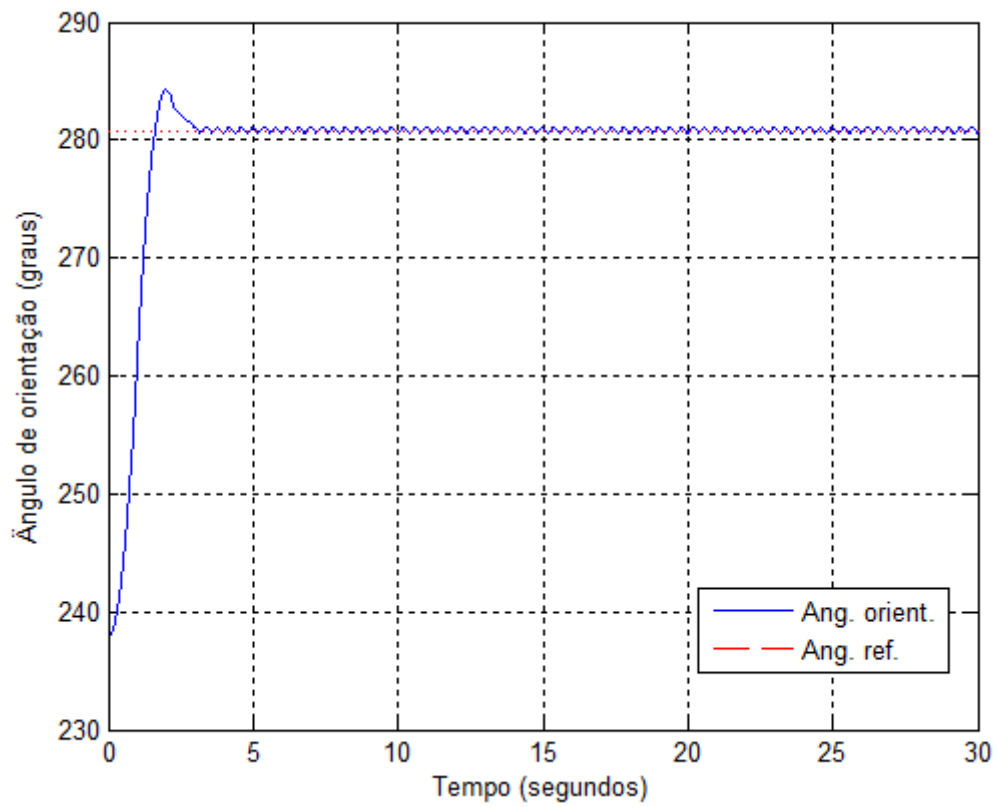


Figura 4.5: Oscilações do sistema para frequência de amostragem de 10 Hz.

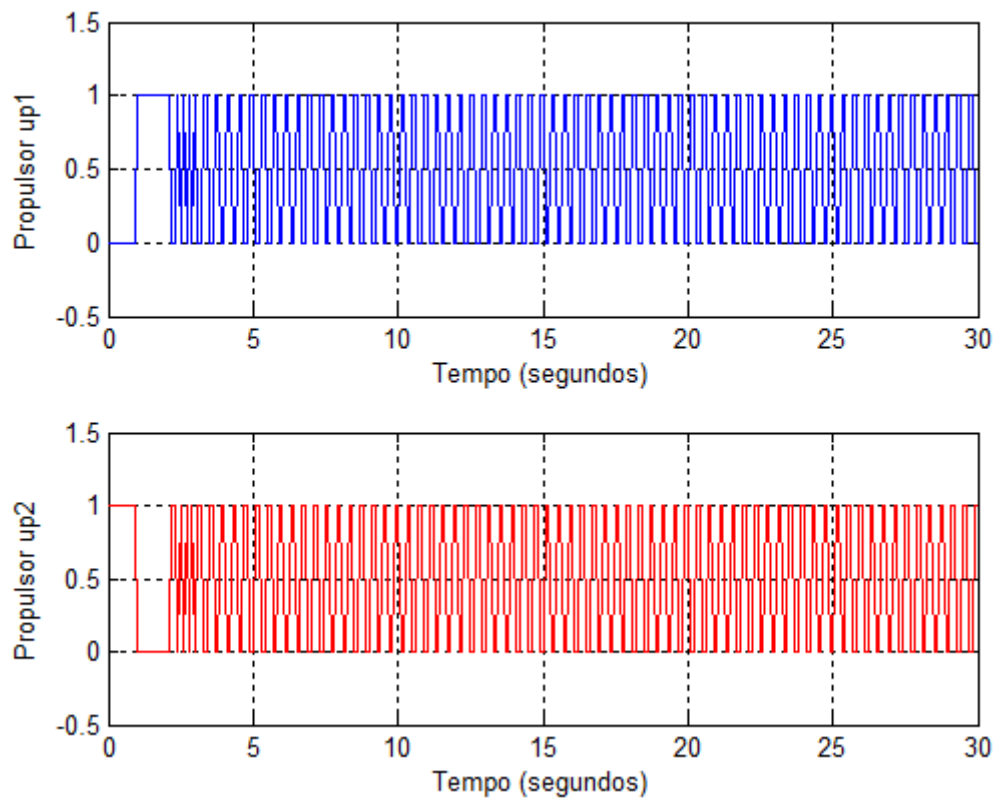


Figura 4.6: Sinais de comando dos propulsores para a frequência de amostragem de 10 Hz.

Na Figura 4.5, observa-se a presença de pequenas oscilações no controle com sinal amostrado. De acordo com [2], estas oscilações se devem ao fato de a taxa de amostragem utilizada não ser alta o suficiente para possibilitar a eliminação do erro em regime pelo controlador.

Para a frequência de amostragem de 3,33 Hz ($h = 0,3$ s), utilizada nos experimentos (mais detalhes, ver seção 4.3), e para $\psi_{\text{ref}} = 280,7^\circ$ e $\psi(0) = 237,7^\circ$, obtiveram-se os resultados a seguir (Figuras 4.7 e 4.8):

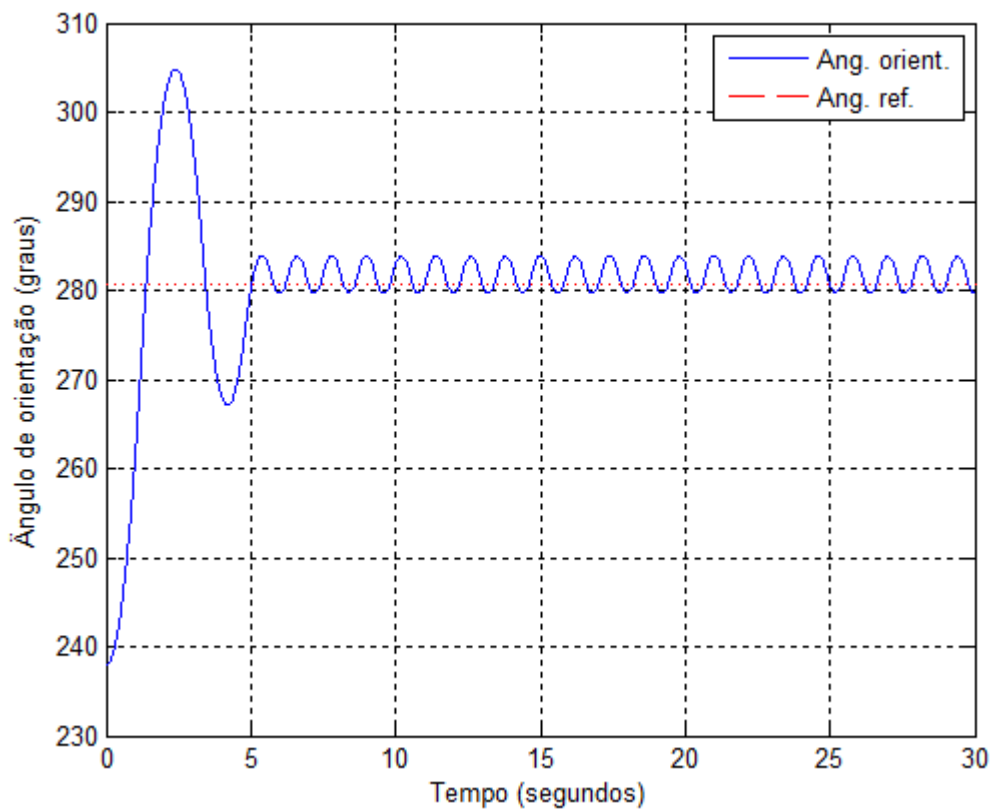


Figura 4.7: Oscilações do sistema para a frequência de amostragem de 3,33 Hz.

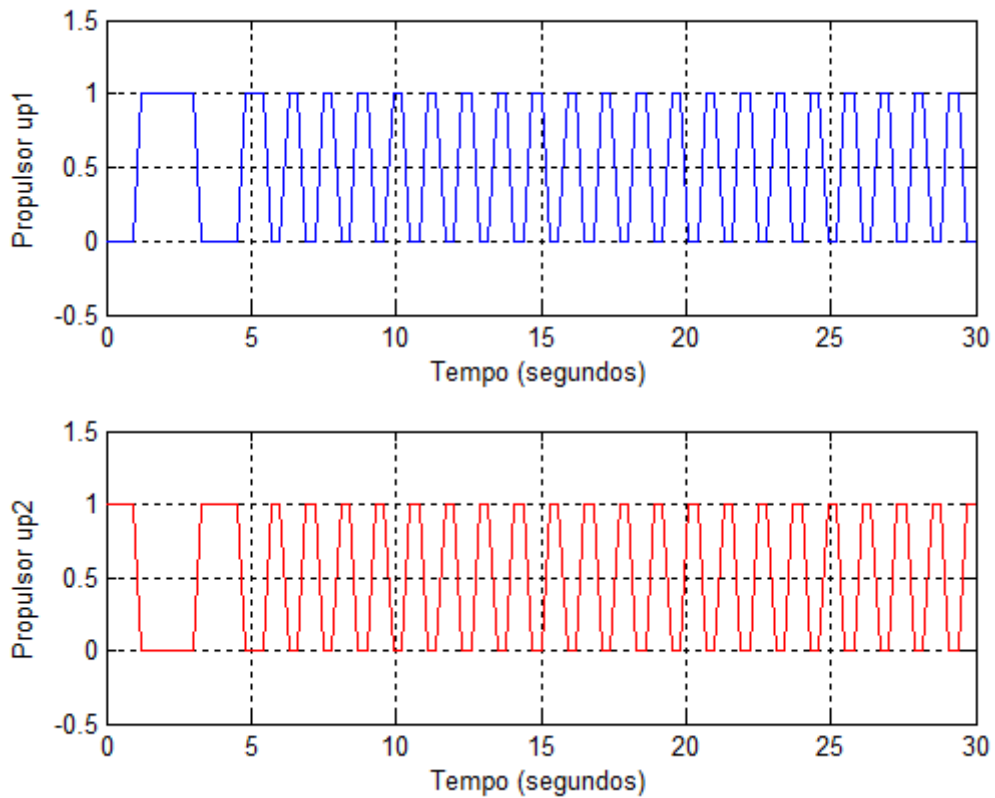


Figura 4.8: Sinais de comando dos propulsores para a frequência de amostragem de 3,33 Hz.

Na Figura 4.7, pode ser observado que à medida que o a frequência de amostragem diminui, a amplitude das oscilações aumenta e a trajetória do sistema se afasta da orientação de referência.

Para a frequência de amostragem de 3,33 Hz ($h = 0,3$ s), $\psi(0) = 135,2^\circ$ e para a posição de referência $\psi_{\text{ref}} = 140,3^\circ$, utilizada nos testes em campo, obtiveram-se os resultados a seguir (Figuras 4.9 e 4.10):

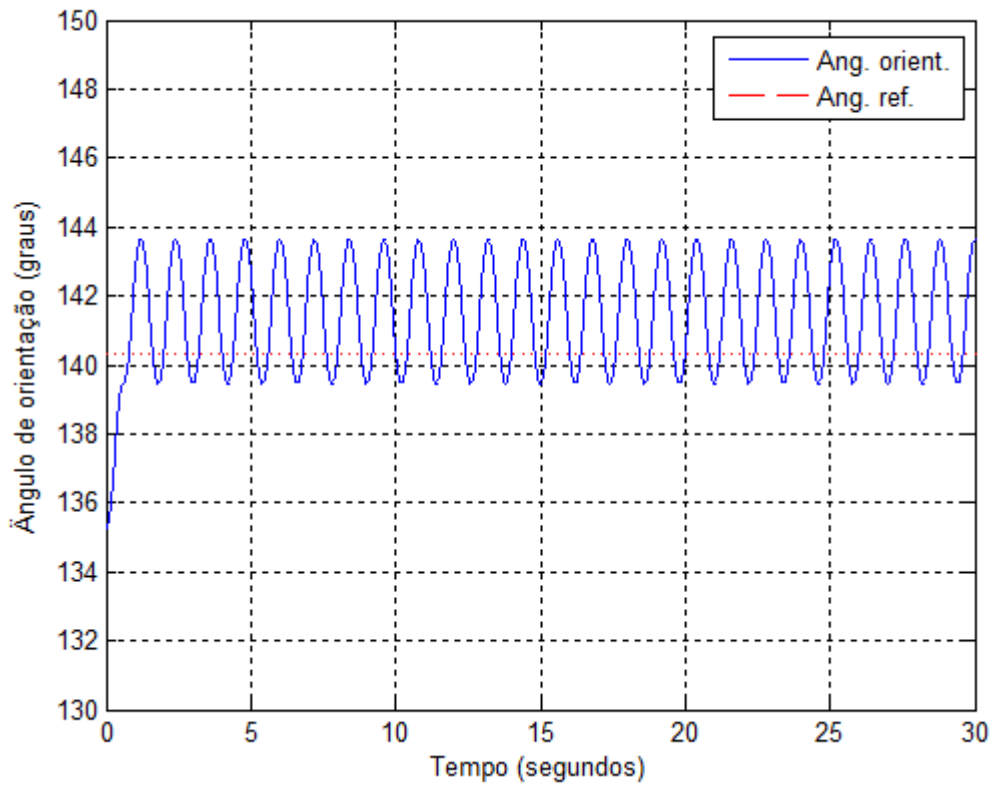


Figura 4.9: Oscilações do sistema para a posição de referência $\psi_{ref} = 140,3^\circ$.

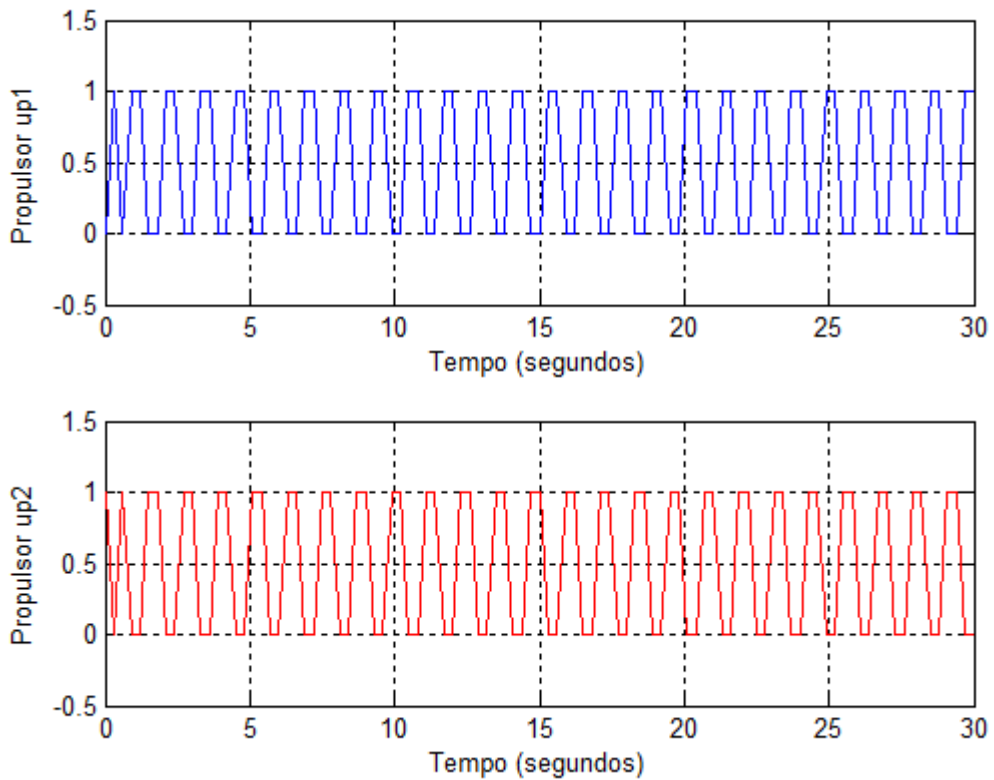


Figura 4.10: Sinais de comando dos propulsores para $\psi_{ref} = 140,3^\circ$.

Através da análise dos resultados das simulações, pôde-se concluir que o controlador está funcionando de forma satisfatória e que quanto maior a frequência de amostragem do sistema, menor é o chattering no sinal de controle. Entretanto, a resposta do sistema está limitada à frequência de amostragem dos sensores e ao tempo de resposta dos atuadores.

4.3 – Testes e Resultados Experimentais

Os testes do controle automático de rumo da embarcação foram realizados no canal da Capela Ecumênica localizado na Universidade do Estado do Rio de Janeiro (UERJ). A Figura 4.11 apresenta o canal onde os testes foram feitos.



Figura 4.11: Canal da Capela Ecumênica da UERJ.

Devido à geometria do canal, e por ser bastante estreito, os movimentos do barco ficaram limitados de modo que a embarcação executou apenas trajetórias retilíneas de ida e volta sempre com os mesmos ângulos de rumo de referência.

O barco foi centralizado paralelamente em relação às bordas de uma das extremidades do canal de maneira a ficar com a proa direcionada para a extremidade oposta. Com o barco posicionado, registrou-se o valor medido pela bússola para ser utilizado como ângulo de referência. A tensão de alimentação dos motores da embarcação também foi medida e apresentou o valor de 12,31 Volts.

Utilizou-se a constante λ da superfície de deslizamento definida na equação (3.10) igual a $\lambda = 2,0$ rad/s, conforme descrito na seção 4.1.

A frequência de amostragem utilizada durante os testes foi de 3,33 Hz. Esta frequência foi escolhida como ponto de partida para a realização dos testes, visto que seu valor é baixo, se comparada com a frequência de amostragem dos sensores (Capítulo 2), de maneira que o sistema de propulsão responderia aos sinais de comando do controlador. A ideia era realizar diversos testes para diferentes valores de frequência de amostragem, porém, não foi possível devido ao pouco tempo disponível para os testes.

As Figuras 4.12 a 4.14 apresentam os resultados experimentais do controle de rumo da embarcação para a frequência de operação de 3,33 Hz e ângulo de orientação de referência igual a $\psi_{ref} = 280,7^\circ$.

O barco foi posicionado com um ângulo de orientação qualquer e em seguida os propulsores foram acionados. Na Figura 4.12, tem-se as oscilações que descrevem o comportamento do barco para o ângulo de referência escolhido. Pode-se observar que em aproximadamente 6 segundos o barco alcançou o ângulo de referência.

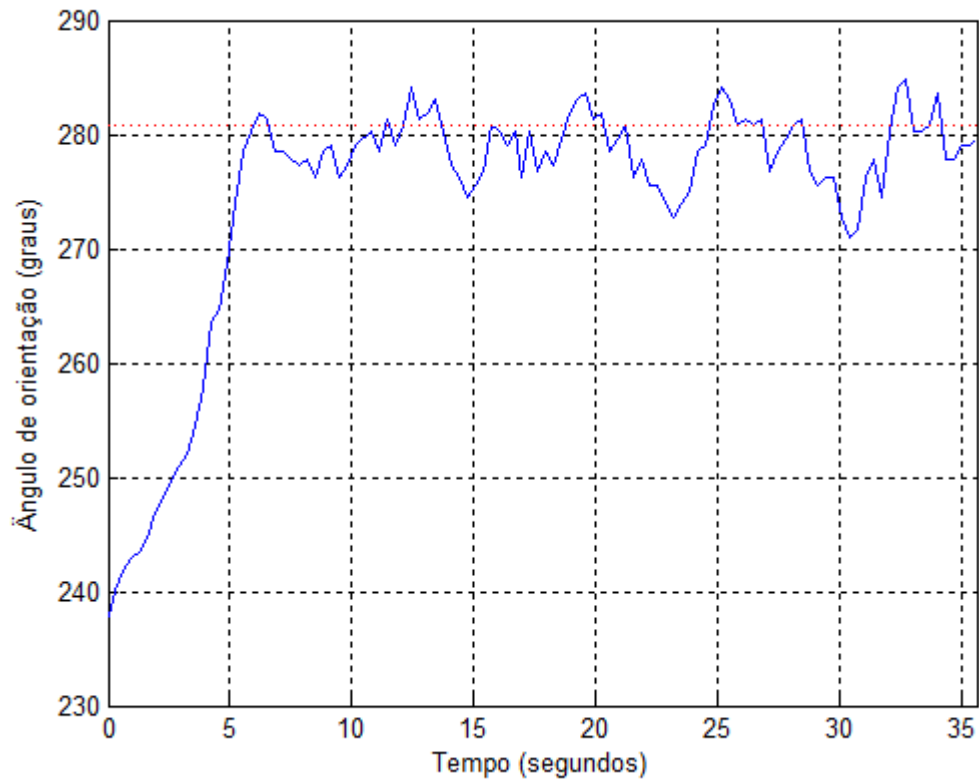


Figura 4.12: Oscilações da embarcação para $\psi_{ref} = 280,7^\circ$.

A Figura 4.13 representa a velocidade angular da embarcação durante sua trajetória.

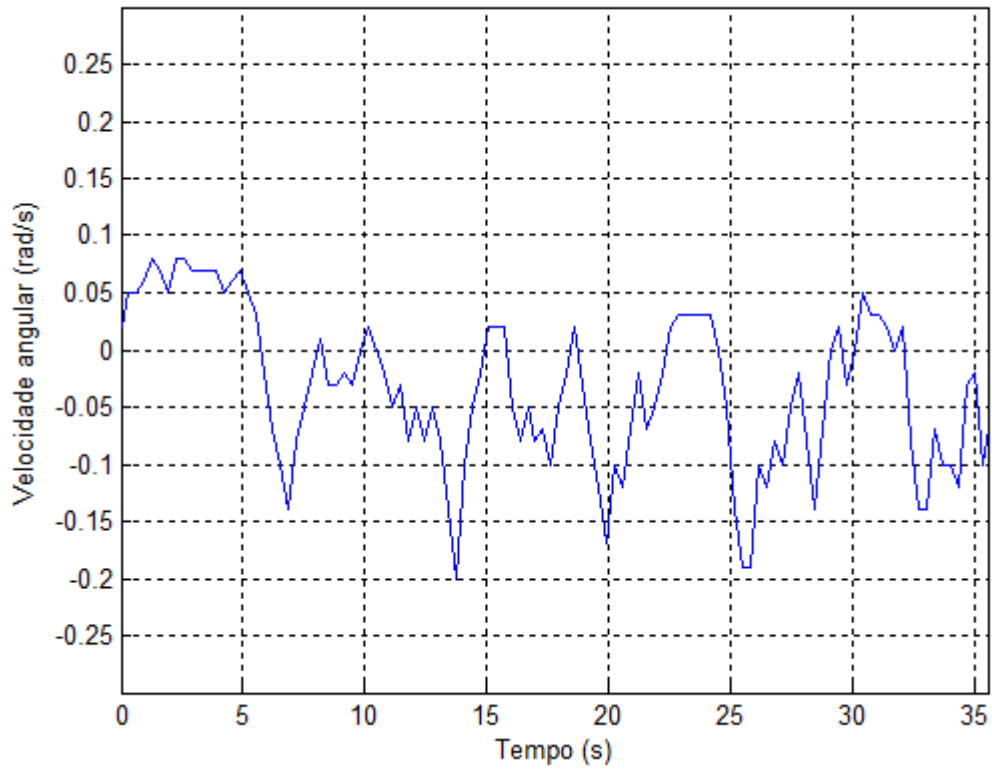


Figura 4.13: Velocidade angular da embarcação para $\psi_{ref} = 280,7^\circ$.

A Figura 4.14 apresenta o sinal de comando dos propulsores onde, o gráfico em azul representa o sinal de comando do propulsor direito (u_{p1}) enquanto o gráfico em vermelho representa o sinal de comando do propulsor esquerdo (u_{p2}).

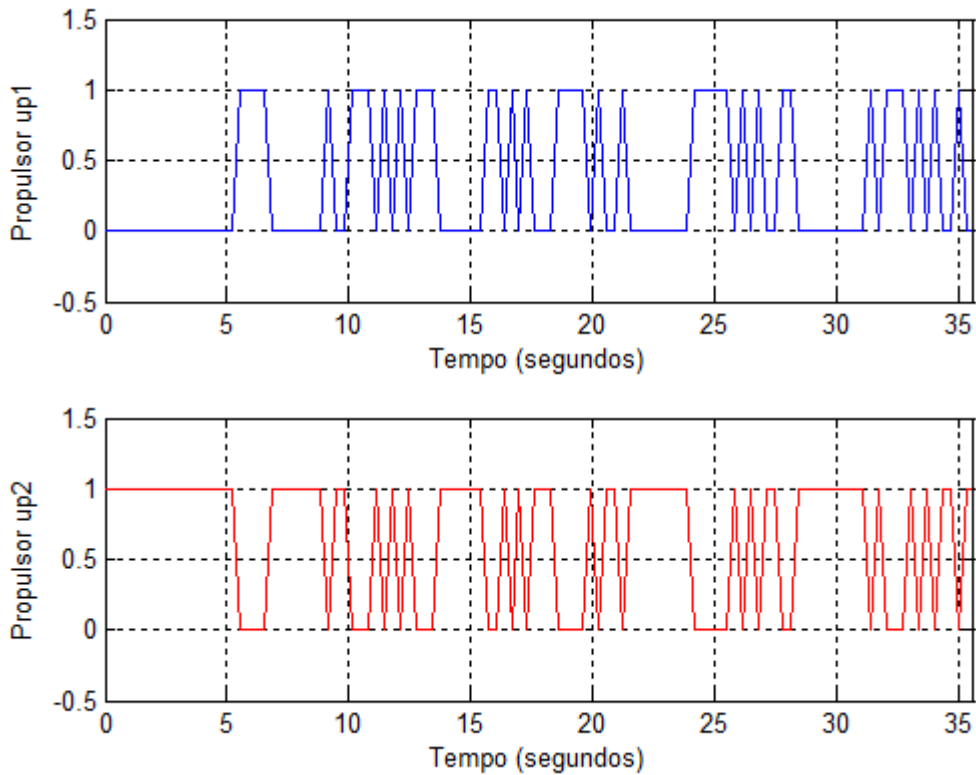


Figura 4.14: Sinal de comando dos propulsores em função do tempo.

Conforme apresentado na Figura 4.14, os sinais de comando dos propulsores variam entre os níveis 0 e 1. Estes níveis indicam se os propulsores estão desligados (nível 0) ou ligados (nível 1).

A Figura 4.15 apresenta a sobreposição dos resultados da simulação (Figura 4.7) e experimental (Figura 4.12) do comportamento do sistema.

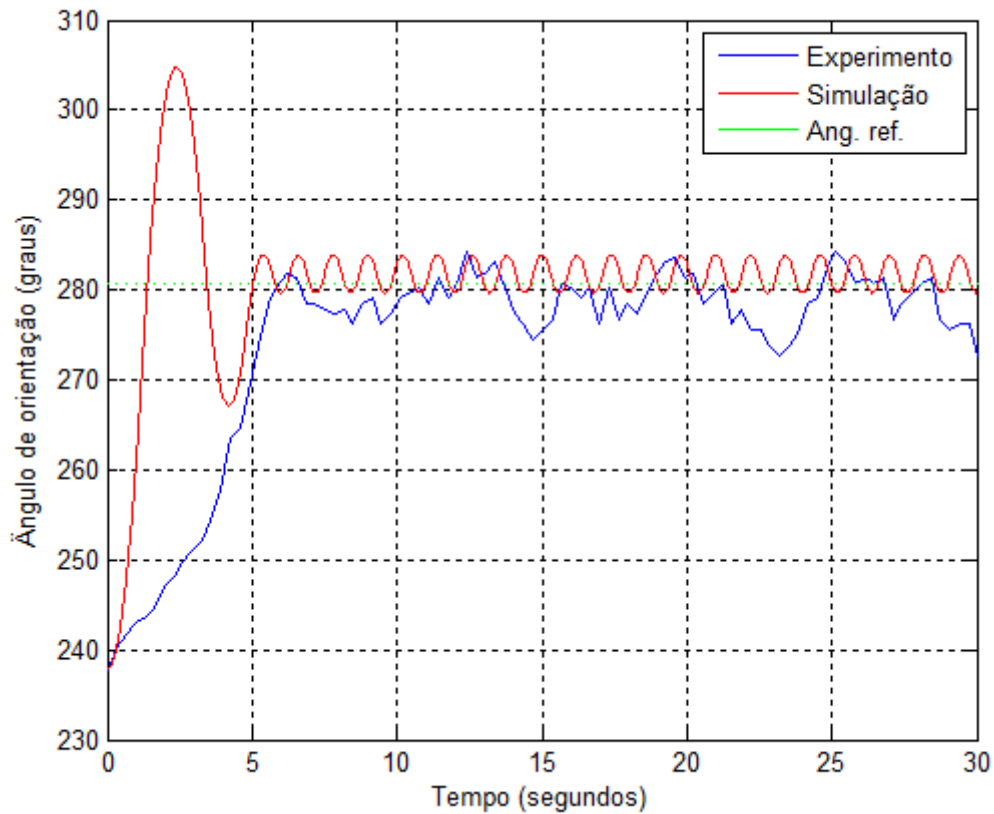


Figura 4.15: Comparação entre simulação e experimento para $\psi_{ref} = 280,7^\circ$.

Pode-se observar na Figura 4.15, que o sistema, em poucos segundos, alcançou e manteve sua trajetória entorno do ângulo de referência, porém, o fenômeno do chattering é bem visível em ambos os gráficos. Isso se deve, principalmente, ao baixo valor da frequência de operação do controlador. Observa-se também, que o regime transitório do sistema de controle foi maior no resultado experimental.

Mantidos a frequência de operação do sistema e o ângulo de orientação de referência ($\psi_{ref} = 280,7^\circ$), coletaram-se os dados com o barco em regime permanente e em seguida, desligaram-se os propulsores deixando o barco à deriva. As Figuras 4.16 a 4.18 apresentam os resultados do comportamento da embarcação.

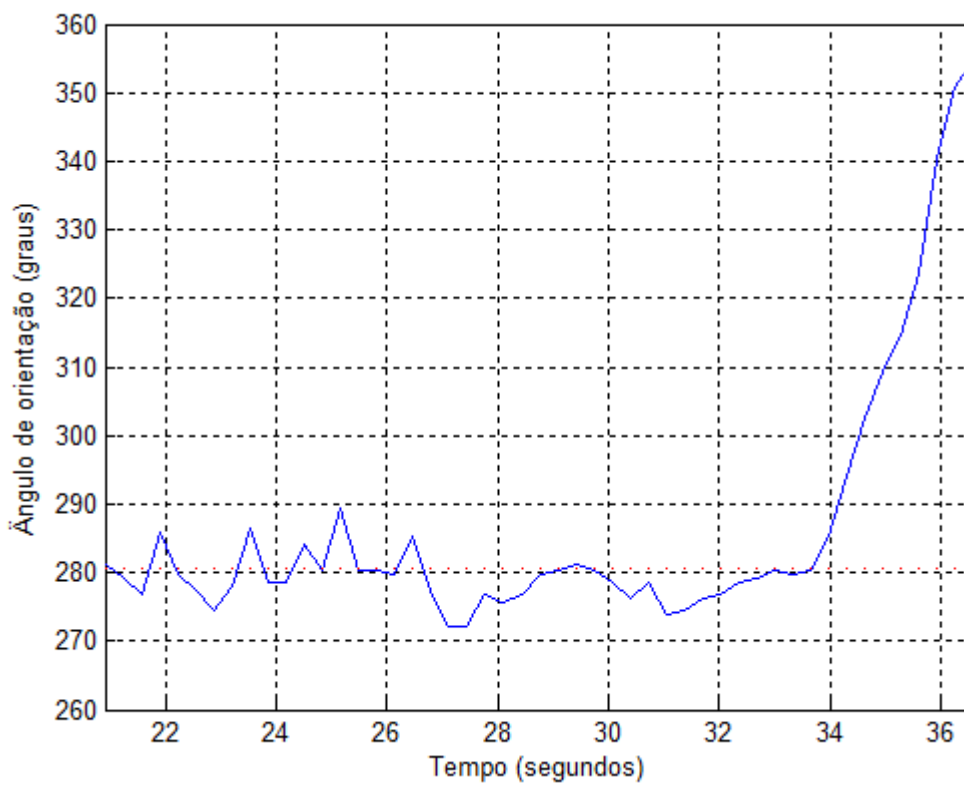


Figura 4.16: Oscilações da embarcação para $\psi_{ref} = 280,7^\circ$ do segundo teste.

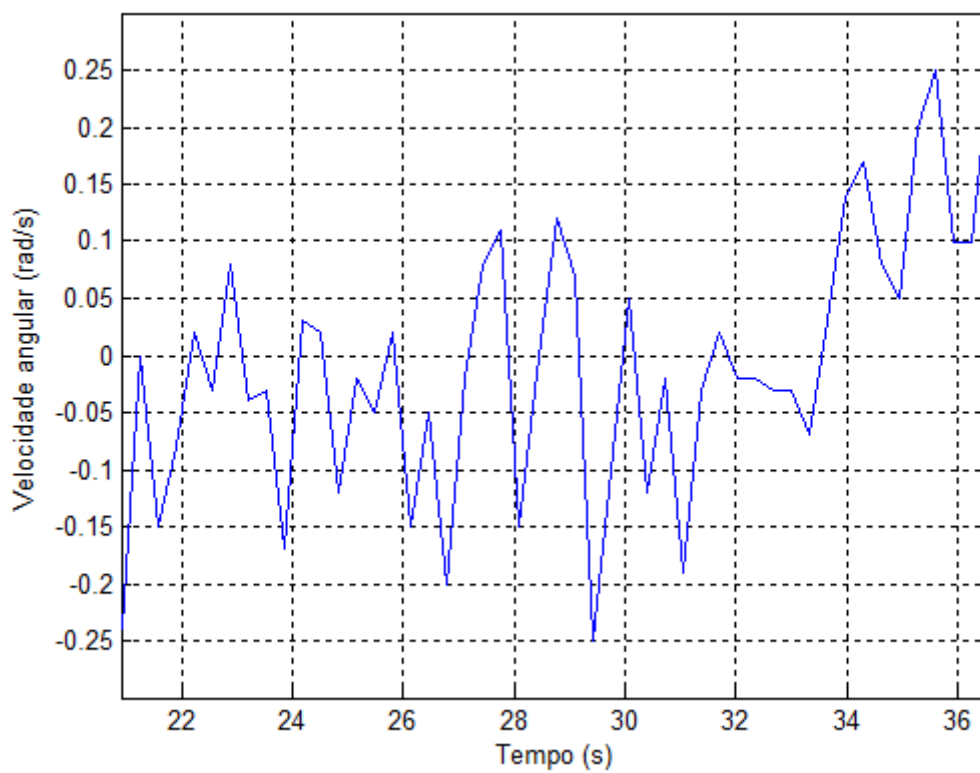


Figura 4.17: Velocidade angular da embarcação para $\psi_{ref} = 280,7^\circ$ do segundo teste.

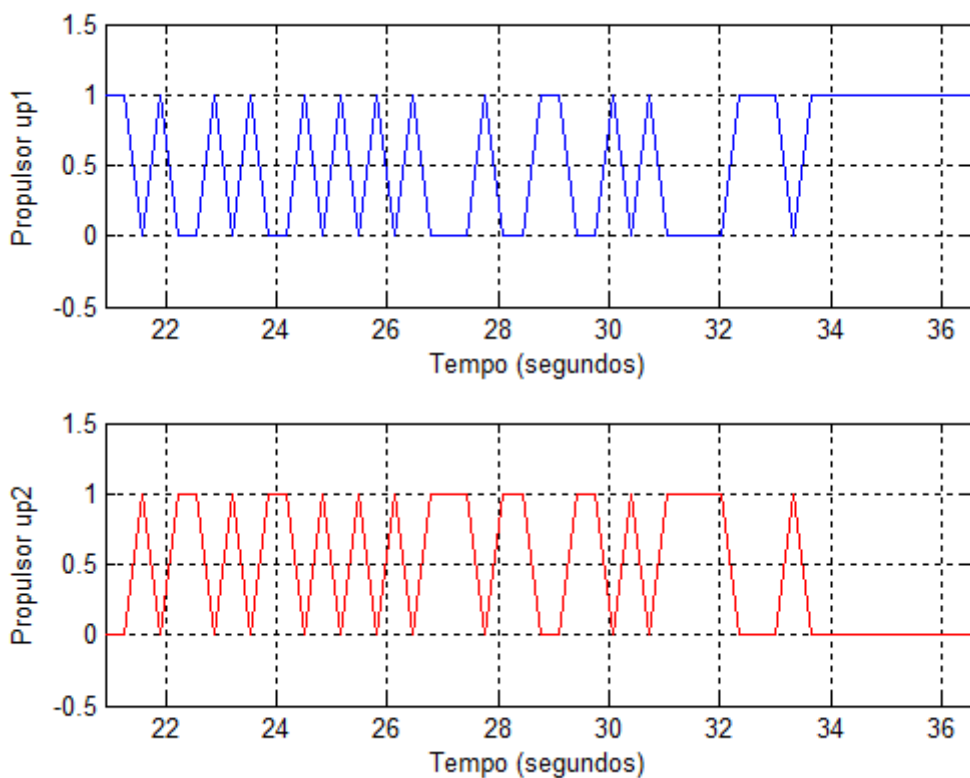


Figura 4.18: Sinal de comando dos propulsores em função do tempo do segundo teste.

Para o ângulo de orientação de referência igual a $\psi_{\text{ref}} = 140,3^\circ$ as Figuras 4.19 a 4.21 apresentam os resultados do experimento do controle de rumo da embarcação.

Na Figura 4.19 pode-se observar que os propulsores foram ligados a partir dos 4 segundos fazendo com que o barco alcançasse o ângulo de referência onde, manteve sua orientação até que os propulsores foram desligados quando se chegou há aproximadamente 25 segundos.

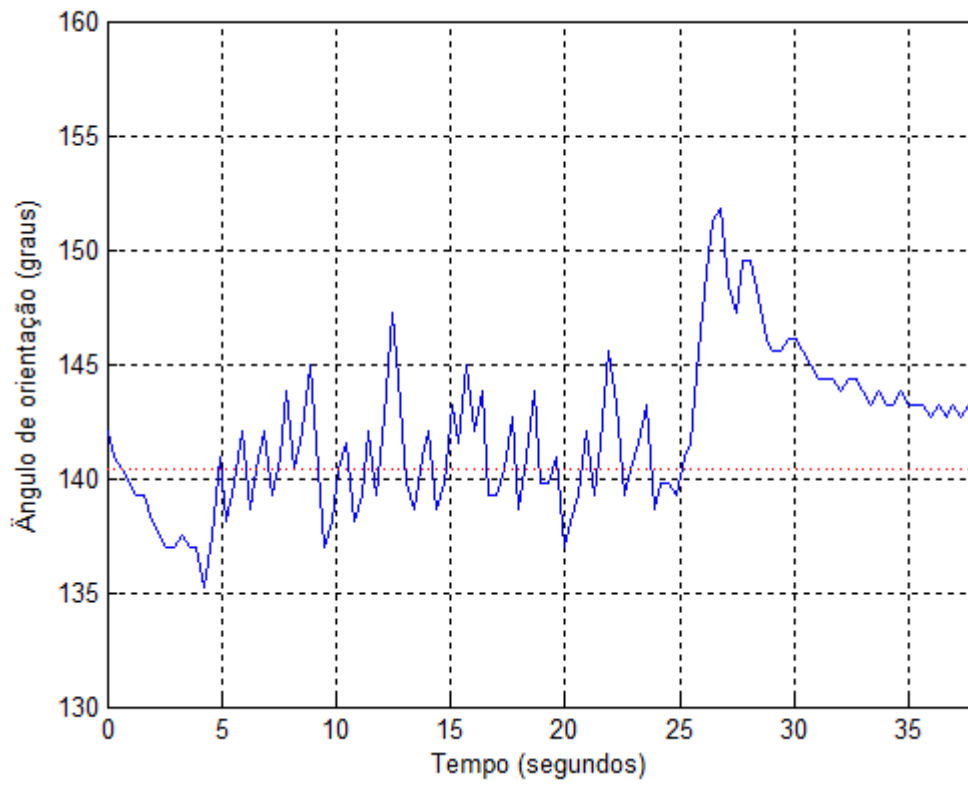


Figura 4.19: Oscilações da embarcação para $\psi_{ref} = 140,3^\circ$.

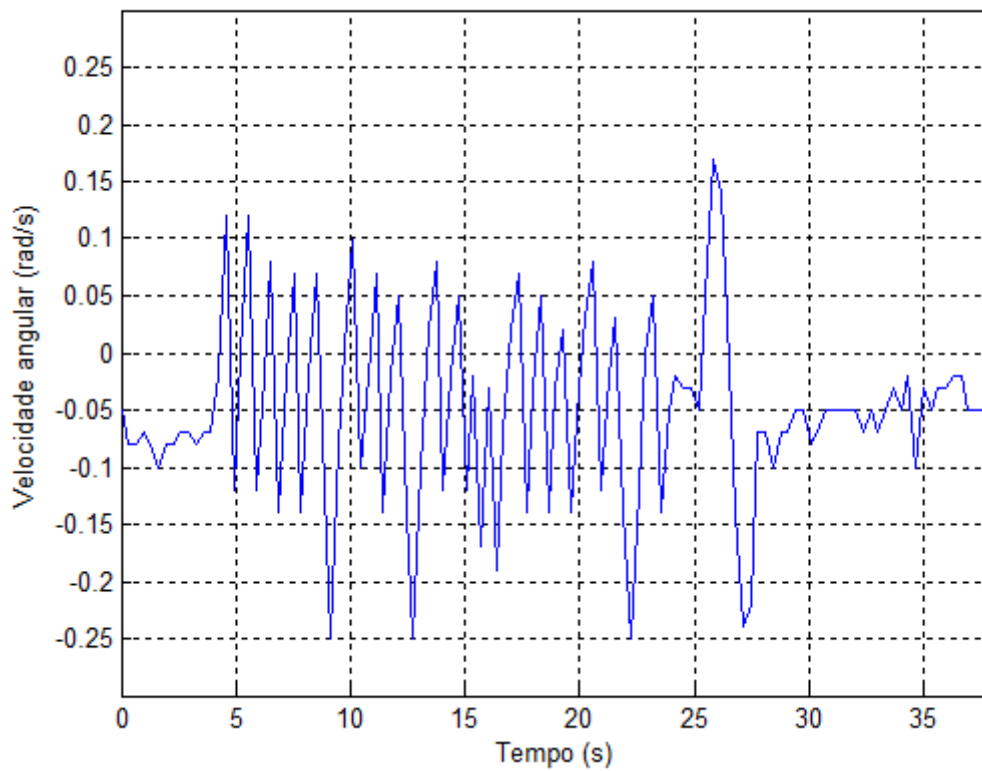


Figura 4.20: Velocidade angular da embarcação para $\psi_{ref} = 140,3^\circ$.

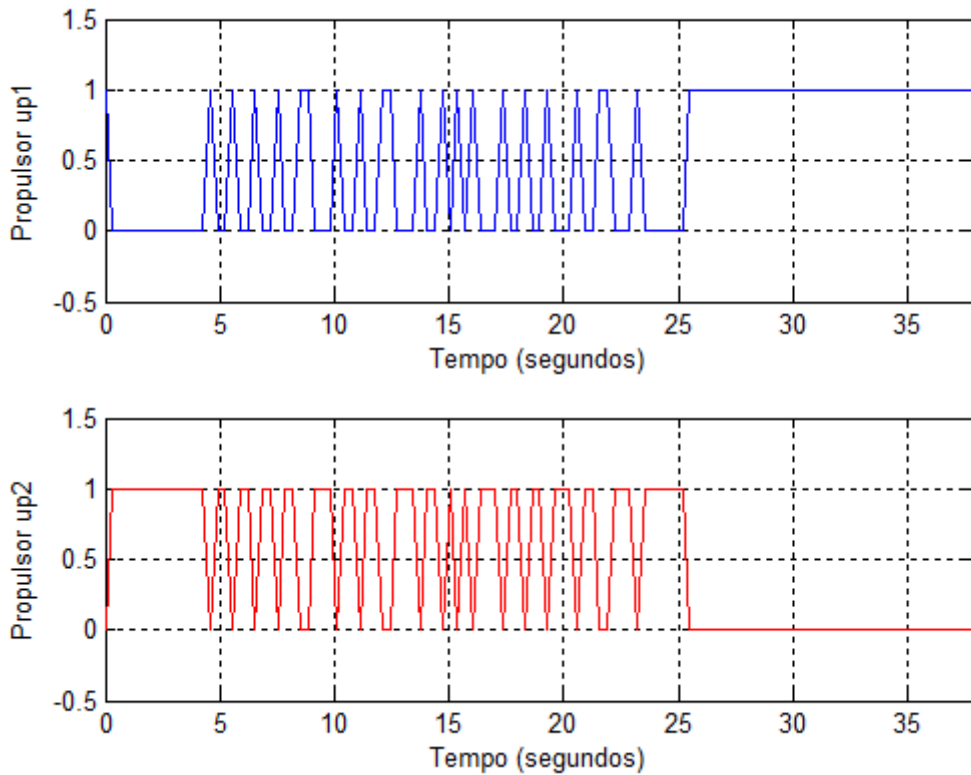


Figura 4.21: Sinal de comando dos propulsores em função do tempo.

Analisando os resultados apresentados nas Figuras 4.19 e 4.9, pode-se observar que, no resultado experimental, o sistema apresenta um transitório maior, levando mais tempo para alcançar o regime, em relação à simulação. É possível observar que o chattering do sinal de controle, para os resultados experimentais, é maior, visto que se trata de um sistema real e que está susceptível a perturbações externas e imperfeições, enquanto que para as simulações, foram desconsideradas as perturbações e as limitações do sistema. A Figura 4.22 apresenta a sobreposição dos resultados apresentados nas Figuras 4.19 e 4.9. Onde, se desconsiderou do resultado experimental (Figura 4.19), o instante de tempo inicial em que os propulsores estavam desligados, comparando-se apenas o comportamento da embarcação a partir deste instante.

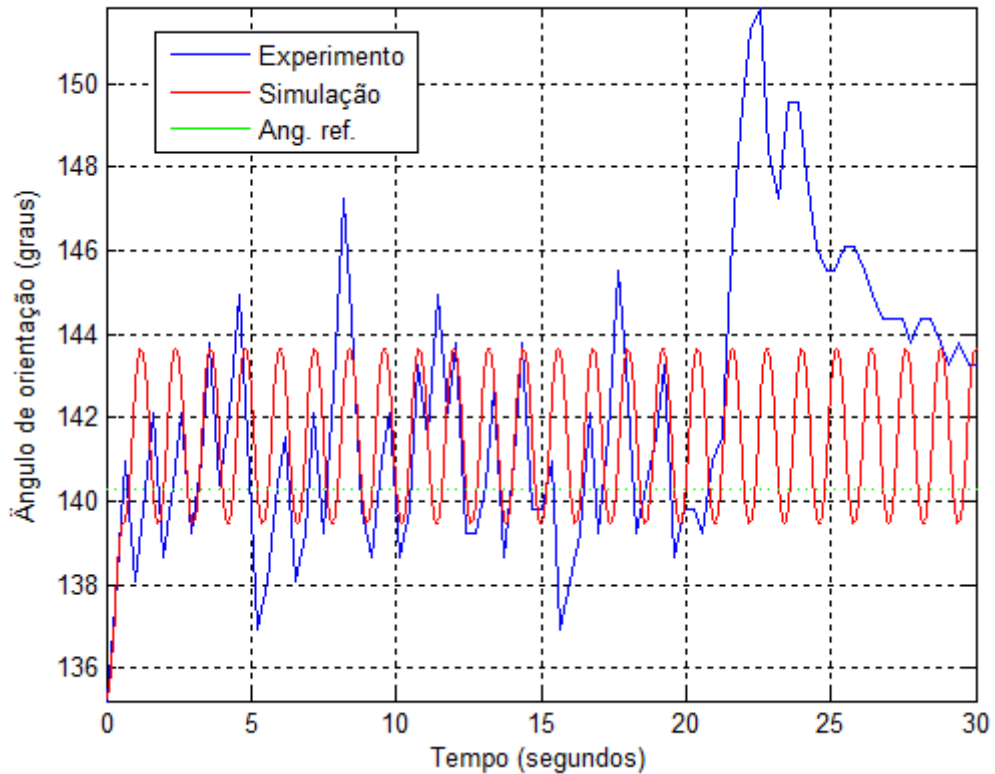


Figura 4.22: Comparação entre simulação e experimento para $\psi_{ref} = 140,3^\circ$.

Observando os gráficos nas Figuras 4.12 a 4.22 é possível perceber a ocorrência de oscilações em torno dos valores de referência estabelecidos para a superfície de deslizamento. Este efeito ocorre devido à baixa frequência de operação utilizada durante os testes, pelas imperfeições nos motores, o tempo de resposta do circuito de acionamento dos propulsores e histerese, montagem e distribuição de peso da embarcação e algumas classes de perturbações externas, as quais foram desconsideradas durante a modelagem do sistema de controle.

CAPÍTULO 5 – CONCLUSÕES

O Arduino mostrou-se eficiente no controle de rumo da embarcação visto que foi capaz de interpretar os dados de entrada provenientes dos sensores, e responder de forma satisfatória, através dos comandos para o sistema de propulsão. Além disso, a escrita do algoritmo de controle foi facilitada, pelo fato de o Arduino possuir uma biblioteca simplificada baseada em C e C++.

Tanto a bússola quanto o girômetro foram testados exaustivamente, em condições e ambientes distintos, e os resultados obtidos estavam de acordo com as especificações do fornecedor.

O controle a estrutura variável desenvolvido atuou de forma esperada corrigindo satisfatoriamente a orientação da embarcação. Porém, pôde-se observar o fenômeno do chattering. Como já mencionado durante os teste e resultados experimentais, chattering são oscilações em torno da superfície de deslizamento que se devem, em parte, à frequência de operação do sistema. Quanto maior a frequência de operação, menor serão as oscilações, ou seja, se a frequência de operação tende a infinito o chattering tende a zero. Devido à baixa frequência de operação utilizada, o atraso no circuito de acionamento e a saturação dos propulsores, pôde-se observar este fenômeno.

Para minimizar o efeito do chattering, sugerem-se como melhorias, para trabalhos futuros, o acionamento dos propulsores no mesmo sentido de rotação, porém com velocidades diferentes, a utilização de estimadores de estado em conjunto com o VSC, com o objetivo de se alcançar o deslizamento ideal apesar das eventuais dinâmicas não modeladas referentes aos sensores e atuadores do sistema e o controle linear de velocidade dos propulsores, através da Modulação por Largura de Pulso (PWM - *Pulse Width Modulation*).

A fim de minimizar o efeito de comutação dos relés, de modo a evitar que o sistema de propulsão fosse danificado, pensou em se utilizar uma região de zona morta. A região de zona morta é definida como o intervalo máximo no qual um estímulo (ângulo de orientação) pode variar em ambos os sentidos, sem produzir variação na resposta do sistema de propulsão. Durante os testes pôde-se observar que para a frequência de operação utilizada, o sistema de propulsão respondeu bem

aos estímulos dos sinais de comando de maneira que não houve a necessidade de se utilizar uma zona morta.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Cunha, J. P. V. S. (1992), Projeto e Estudos de Simulação de um Sistema de Controle a Estrutura Variável de um Veículo Submarino de Operação Remota, Tese de Mestrado em Engenharia, Programa de Pós – Graduação de Engenharia Elétrica, COPPE/UFRJ, Rio de Janeiro.
- [2] Sokal, G. J. (2010), Posicionamento Dinâmico Utilizando Controle a Estrutura Variável e Servovisão, Tese de Mestrado em Engenharia, Programa de Pós – Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro, Rio de Janeiro.
- [3] Amaral, G. S. (2008), Sistema de Posicionamento Dinâmico para um Pequeno Veículo Flutuante, Projeto de Graduação em Engenharia Elétrica da Universidade do Estado do Rio de Janeiro, Rio de Janeiro.
- [4] Nunes, E. V. L. (2004), Controle por Modos Deslizantes de Ordem Superior com Estabilidade Global, Tese de Mestrado em Engenharia, Programa de Pós – Graduação em Engenharia Elétrica, da Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- [5] Ogata, K. (2003), *Engenharia de Controle Moderno – 4ª Edição*. Pearson Prentice Hall.
- [6] Stavarengo, A. A. (2006), Telemetria de Dados e Imagens para Plataforma Autônoma para Coleta de Dados Hidrológicos, Tese de Mestrado em Engenharia, Programa de Pós – Graduação de engenharia Elétrica, São Carlos/Universidade de São Paulo, São Paulo.
- [7] Utkin, V.; Guldner, J.; Shi, J. (1999), *Sliding Mode Control in Electromechanical Systems*. Taylor & Francis.

[8] Sá, A. S. B. (2009), Modelação de Controle Dinâmico de uma Embarcação Autônoma, Dissertação/Relatório de Projecto, Mestrado Integrado em Engenharia Electrotécnica e de Computadores Major Automação, da Faculdade de Engenharia da Universidade do Porto, Porto, Portugal.

[9] Fonseca, E. G. P.; Beppu, M. M. (2010), Apostila Arduino, Programa de Educação Tutorial, Curso de Engenharia de Telecomunicações, da Universidade Federal Fluminense, Rio de Janeiro. Disponível em: http://www.telecom.uff.br/pet/petws/downloads/tutoriais/arduino/Tut_Arduino.pdf.

[10] Schultze, H. J. (2011), Projeto e Construção de uma Embarcação Teleoperada, Relatório do Projeto do circuito acionador dos motores, Departamento de Engenharia Eletrônica e Telecomunicações, Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro, Rio de Janeiro.

[11] Projeto de Pesquisa—Edital FAPERJ nº 25/2010—Apoio a Núcleos Emergentes de Pesquisa no Estado do Rio de Janeiro — PRONEM — 2010 Título do Projeto de Pesquisa: Embarcações Não Tripuladas para Monitoração Ambiental e Defesa. Solicitante: José Paulo Vilela Soares da Cunha

[12] Curcio, J.; Leonard, J.; Patrikalakis, A. (2004), SCOUT — A Low Cost Autonomous Surface Platform for Research in Cooperative Autonomy, Department of Mechanical Engineering Massachusetts Institute of Technology, Cambridge, MA. Página consultada em 26 de julho de 2012, < http://maribotics.com/publications/5_scout_oceans05.pdf >.

[13] Marchi, A. A. (2009), Implementação de uma Bússola Digital, Projeto de Graduação em Engenharia Elétrica da Escola de Engenharia de São Carlos, da Universidade de São Paulo, São Paulo.

[14] Multilógica-Shop, Open Source Hardware. Página consultada em 27 de Agosto de 2011, < <http://www.multilogica-shop.com/> >.

- [15] Arduino. Página consultada em 25 de Junho de 2011, < <http://arduino.cc/en/> >.
- [16] Prado, S. (2007), Barramento I2C, Portal Embarcados. Página consultada em 19 de Abril de 2012, < <http://www.embarcados.com.br/Artigos/Hardware-Embarcado/Barramento-I2C.html> >.
- [17] Programming Arduino To Read IMU (2011), Nuclear Projects. Página consultada em 16 de Fevereiro de 2012, < http://www.nuclearprojects.com/ins/arduino_program.shtml >.
- [18] Bernardes, J. (2012), “EESC constrói barco não tripulado que poderá monitorar águas e peixes”, Agência USP de Notícias, Universidade de São Paulo. Página consultada em 12 de julho de 2012, < <http://www5.usp.br/10679/eesc-constroi-barco-nao-tripulado-que-podera-monitorar-aguas-e-peixes/> >.
- [19] Vaz, R. B. (2010), Projeto Roaz. Página consultada em 12 de julho de 2012, < <http://www.autoboat.com.br/somos.php> >.
- [20] Escandiuzzi, F. (2011), “Barco não tripulado pode monitorar poluição e desastres em SC”, Oceanografia Brasil. Página consultada em 12 de julho de 2012, <<http://www.oceanografiabrasil.com/meio-ambiente/barco-nao-tripulado-pode-monitorar-poluicao-e-desastres-em-sc> >.
- [21] Santos, S. R. B. (2010), Fundamentos para o Projeto de um Sistema de Navegação de Baixo Custo, Estágio de Pesquisa em Engenharia Eletrônica e Computação do Instituto Tecnológico de Aeronáutica.
- [22] Schultze, H. J. (2012), Projeto e Construção de uma Embarcação Teleoperada, Projeto de Graduação em Engenharia Elétrica da Universidade do Estado do Rio de Janeiro, Rio de Janeiro.
- [23] Wikipédia, Artigo, Bússola. Página consultada em 27 de Agosto de 2011, < <http://pt.wikipedia.org/wiki/Bússola> >.

[24] Emelyanov, S. V. (1970), *Theory of variable structure systems*, Nauka.

[25] Itkis, U. (1976), *Control Systems of Variable Structure*, Wiley.

[26] Jekeli C. "*Inertial navigation system with geodetic applications*", Berlim Walter Gruyter, 2001.

[27] Britting K.R. "*Inertial Navigation System Analysis*", New York, John Wiley & Sons, Inc, 1971.

[28] Saber Eletrônica (2007), Efeito Hall x Magneto-Resistivo (MR). Revista Saber Eletrônica, abril/07, 43.

APÊNDICE A – EMBARCAÇÃO E CIRCUITO DE ACIONAMENTO

Este apêndice descreve o veículo flutuante utilizado, a disposição dos propulsores, bateria e sistema de controle na embarcação e o circuito de acionamento dos propulsores.

A.1 – Embarcação

O veículo flutuante utilizado neste projeto é uma embarcação de casco simples denominada caiaque, modelo Hunter Fishing. O caiaque Hunter Fishing é feito de um material plástico extremamente leve e mede 3,15 metros de comprimento por aproximadamente 0,80 metros de largura. A Figura A.1 apresenta o caiaque.



Figura A.1 – Caiaque Hunter Fishing.

Na popa do caiaque foram fixadas barras de metais para suporte dos dois propulsores que foram posicionados de maneira a possibilitar o controle de rumo. A Figura A.2 apresenta o suporte de metal dos propulsores e a Figura A.3 apresenta o arranjo motor mais suporte metálico fixados no caiaque.



Figura A.2 – Suporte de metal para fixação dos propulsores.



Figura A.3 – arranjo motores mais suporte metálico fixados no caiaque.

A bateria de 12 Volts modelo Moura Boat-105Ah foi posicionada à frente da parte central da embarcação (Figura A.3) de maneira a compensar o peso dos propulsores e manter o sistema em equilíbrio.

Tanto o sistema de controle, quanto o circuito de acionamento dos propulsores e netbook, foram colocados no interior de um caixa de material plástico vedada (caixa estanque) de maneira a impedir a entrada de água. A Figura A.4 apresenta a caixa.



Figura A.4 – Caixa estanque do sistema de controle.

A caixa estanque foi colocada na popa do caiaque, próximo aos propulsores. Na Figura A.3 pode-se observar a disposição da caixa no caiaque.

A.2 – Propulsores

Foram utilizados dois motores de corrente contínua (modelo Marine Sports 44lbs marinizado) no caiaque. Para a seleção dos motores, levou-se em consideração o baixo custo em relação aos demais, e o fato destes serem encontrados no mercado nacional (condições básicas exigidas na escolha do modelo).

Os motores podem operar em cinco velocidades distintas para frente (Frente 1, Frente 2, Frente 3, Frente 4 e Frente 5) e duas velocidades para ré (Ré 1 e Ré 2) que se alternam entre quatro fios (fios amarelo, vermelho, azul e preto), provenientes das bobinas do motor, e são alimentados com uma tensão de 12V fornecidos pela bateria (fios vermelho e preto). Dentre as cinco velocidades para frente, optou-se por trabalhar apenas com as velocidades Frente 1 e Frente 5, com o intuito de se reduzir o custo e o número de componentes utilizados no circuito de acionamento dos motores. Desta forma, os motores utilizados na embarcação operam apenas nas velocidades Frente 1, Frente 5, Ré 1 e Ré 2 onde Frente 1 é a velocidade mínima e Frente 5 a velocidade máxima do motor para frente. Da mesma forma que Ré 1 é a velocidade mínima, e Ré 2 a velocidade máxima do motor para ré. A velocidade Ré 2 equivale em módulo à velocidade Frente 5.

A Tabela A.1 apresenta a alimentação de cada um dos fios provenientes das bobinas em relação às quatro velocidades dos motores.

Tabela A.1: Alimentação dos fios provenientes da bobina.

Fio	Posição da chave seletora				
	Frente 1	Frente 5	Ré 1	Ré 2	Repouso
Amarelo	Positivo	Positivo	Negativo	Negativo	Negativo
Vermelho	Aberto	Positivo	Aberto	Negativo	Aberto
Azul	Aberto	Positivo	Negativo	Negativo	Aberto
Preto	Negativo	Negativo	Positivo	Positivo	Negativo

O acionamento dos motores é feito através de relés de acionamento simples e duplo, que necessitam de uma tensão sobre os seus terminais de 12V para que atuem conduzindo então uma corrente de aproximadamente 150mA.

Os hélices podem ser desenhados com diversas configurações numéricas de pás e podem ser confeccionadas com diferentes tipos de materiais. Os hélices utilizados para o sistema de propulsão da embarcação possuem três pás e são de material plástico devido ao seu alto rendimento, leveza e alta resistência à corrosão. As Figuras A.5 e A.6 apresentam respectivamente o hélice e um dos propulsores utilizados na embarcação.



Figura A.5 – Hélice do propulsor.



Figura A.6 – Propulsor utilizado no caiaque.

A.3 – Acionamento dos propulsores

O acionamento dos motores da embarcação é feito através de um circuito transistorizado, que acionará um conjunto de relés que por sua vez acionarão as bobinas do motor. Este circuito está conectado ao Arduino que, encontra-se

conectado a um netbook no qual é executado o algoritmo de controle. O Arduino tem por objetivo receber os sinais de entrada dos sensores (bússola e girômetro), interpretar estes sinais através do algoritmo de controle (Apêndice C) e apresentar como saída uma tensão que será utilizada pelo circuito transistorizado para acionar os relés. A Figura A.7 apresenta o circuito de acionamento dos relés.

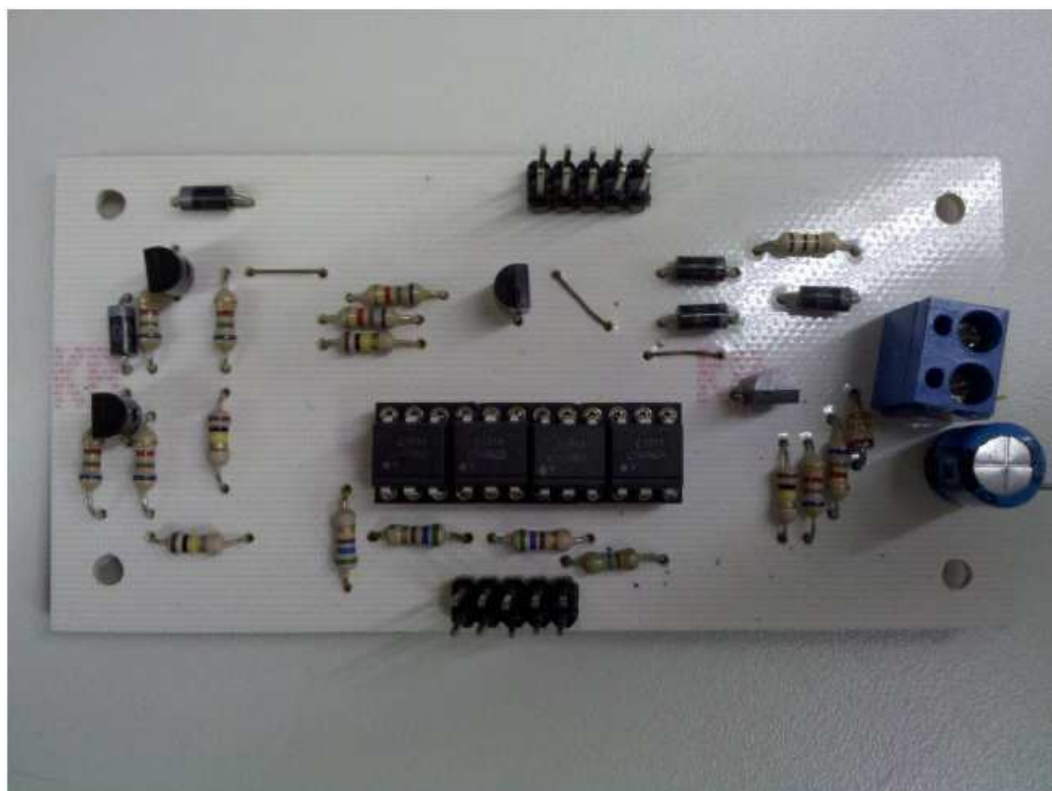


Figura A.7 – Circuito de acionamento dos relés. Imagem extraída de [22].

APÊNDICE B – UNIDADE DE MEDIÇÃO E CALIBRAÇÃO DA BÚSSOLA

B.1 – Unidade de Medição

Os sensores (bússola e unidade de medição inercial) e os fios que interligam os mesmos ao Arduino, foram fixados em uma placa de circuito impresso universal, formando uma unidade de medição. A fixação dos sensores e fios foi feita através de barras de pinos fêmea, de forma a facilitar a remoção dos mesmos, caso houvesse necessidade. Como a saída do girômetro é uma grandeza analógica, utilizou-se cabo com blindagem a fim de se evitar interferências em seu sinal.

As Figuras B.1 e B.2 apresentam, respectivamente, a unidade de medição e o diagrama elétrico da interligação dos sensores aos microcontrolador.

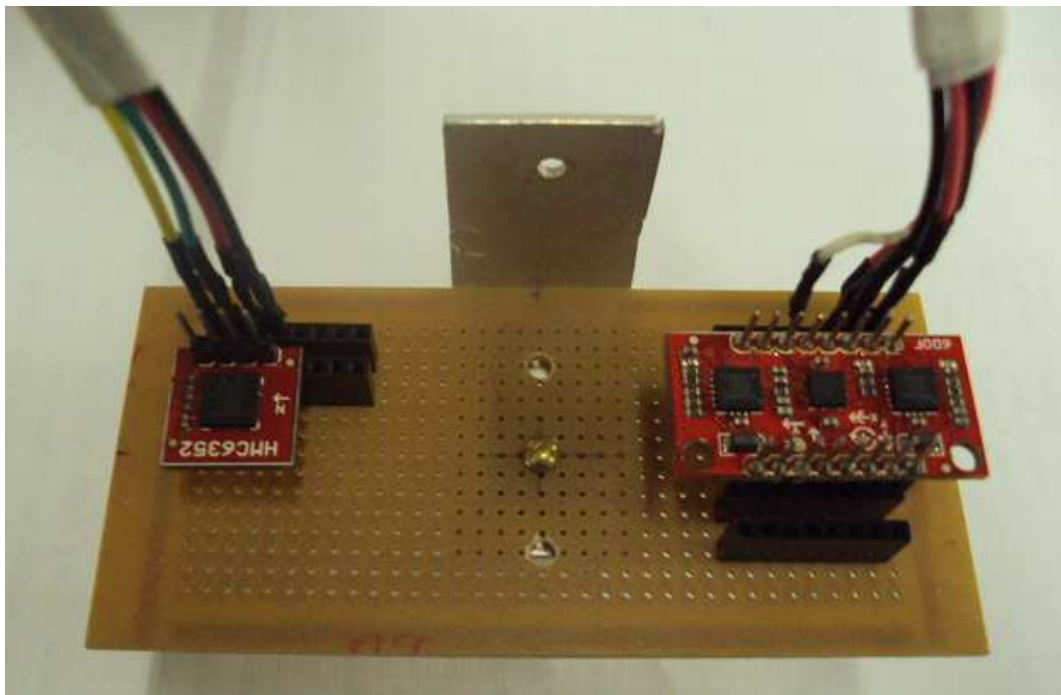


Figura B.1 – Unidade de medição composta pela bússola e unidade de medição inercial.

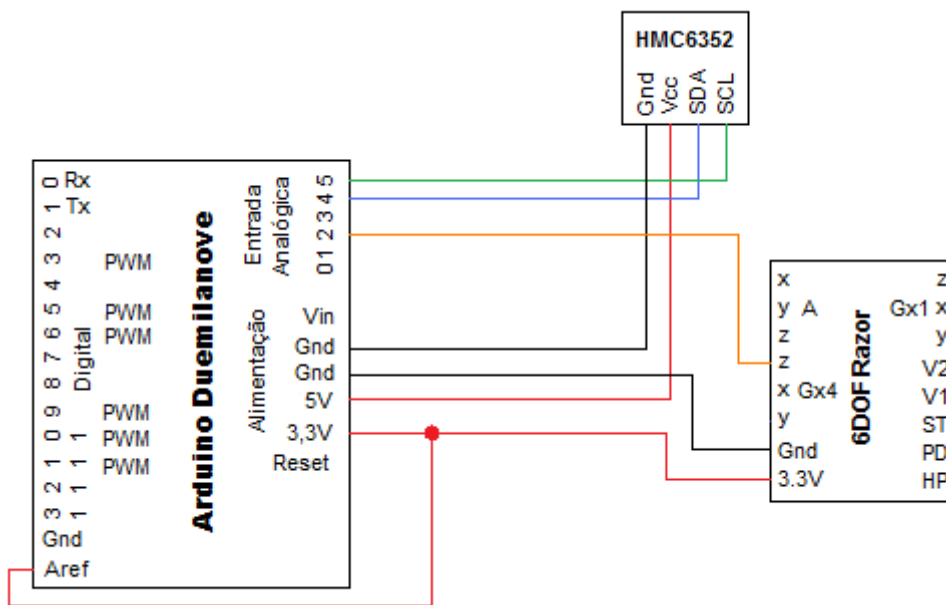


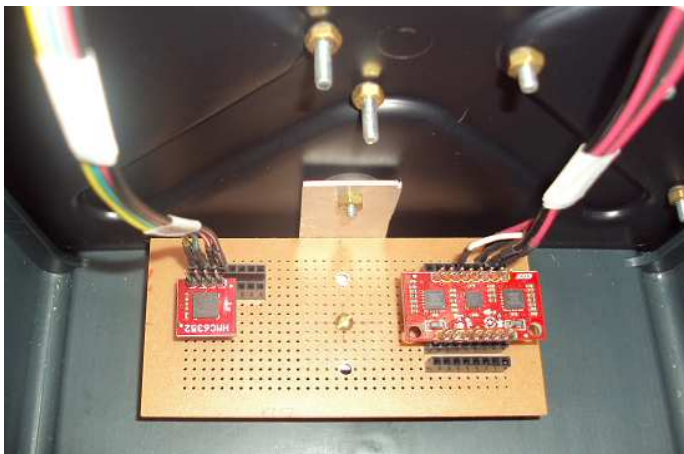
Figura B.2 – Diagrama elétrico da interligação dos sensores ao microcontrolador.

Conforme apresentado nas Figuras B.1 e B.2 e já mencionado na seção 2.3.4, somente o eixo z do girômetro foi conectado ao Arduino, uma vez que, para o controle de rumo, apenas as variações de velocidade angular em torno deste eixo serão lidas, visto que, considerou-se que o barco desloca-se apenas no plano XY, de modo que o seu balanço foi desconsiderado.

A unidade de medição foi fixada horizontalmente na caixa estanque através de um suporte em perfil de L feito de alumínio e parafusos e porcas de aço, de forma a se evitar interferências magnéticas nos dados da bússola. A Figura B.3 apresenta o suporte em perfil de L e a unidade de medição fixada na caixa estanque.



(a)



(b)

Figura B.3 – (a) Suporte de alumínio em perfil de L e (b) unidade de medição fixada à caixa.

B.2 – Calibração da Bússola

Durante a realização de alguns testes com a bússola em laboratório, observou-se que ao dar uma volta completa na bússola, a mesma apresentava valores que variavam apenas de 20° a 190° em vez de 0° a 359,9° como esperado e observado em testes anteriores.

Depois de se levantar algumas hipóteses como, por exemplo, interferências magnéticas provenientes do ambiente de teste, foram feitos novos testes em condições e ambientes diferentes e pôde-se concluir que a bússola estava descalibrada.

Para realizar a calibração da bússola consultou-se o manual do fabricante (Honeywell) que se encontra na seguinte página: <http://www.sparkfun.com/datasheets/Components/HMC6352.pdf>. Neste, observou-se que a bússola fornece uma rotina de calibração de usuário. Esta rotina de calibração é comandada pelo envio de caracteres específicos à bússola, como por exemplo, os caracteres “C” e “E”. O caracter “C” permite a entrada no modo de calibração enquanto o comando “E” encerra o modo de calibração.

Uma vez no modo de calibração, deve-se girar a bússola, ao menos uma rotação circular completa, em uma superfície plana de maneira suave a fim se obter leituras uniformemente espaçadas. O ideal é dar duas voltas completas com duração de 20 segundos.

Ao sair do modo de calibração os deslocamentos resultantes do magnetômetro e os valores de escala são atualizados. Para finalizar o processo de calibração envia-se o caracter “L” para salvar a calibração.

A tabela B.1 apresenta o código da rotina de calibração da bússola.

Tabela B.1: Rotina de calibração da bússola.

```

#include <Wire.h>

#define ADDRESS 0x42 //0 endereço i2c do HMC6352 é 0x42

void setup()
{
  // Inicializa o i2c
  Wire.begin();
}

void loop()
{
  // i2c mestre inicia a transmissão
  Wire.beginTransmission(ADDRESS);
  // envia o comando 'C', "entrar em modo de calibração"
  Wire.send('C');
  // i2c mestre encerra transmissão
  Wire.endTransmission();
  // para responder o comando "entrar em modo de calibração"
  // pode demorar até 20 s
  delay(20000);

  // i2c mestre inicia a transmissão
  Wire.beginTransmission(ADDRESS);
  // envia o comando 'E', "sair do modo de calibração"
  Wire.send('E');
  // i2c mestre encerra transmissão
  Wire.endTransmission();
  // para responder o comando "sair do modo de calibração"
  // pode demorar até 14 ms
  delay(14);

  // i2c mestre inicia a transmissão
  Wire.beginTransmission(ADDRESS);
  // envia o comando 'L', "salvar calibração na EEPROM"
  Wire.send('L');
  // i2c mestre encerra transmissão
  Wire.endTransmission();
  // para responder o comando "salvar calibrção na EEPROM"
  // pode demorar até 0,125 ms
  delay(0.125);
}

```

APÊNDICE C – ROTINA DE CONTROLE

Tabela C.1: Código da rotina de controle desenvolvida.

```
#include <Wire.h>
// orientação de referencia e constante
float orientref = 4.9, k = 2.0;
// o endereço i2c de acesso ao HMC6352 é 0x42
int ADDRESS = 0x42 >> 1;
int heading;
// declaração das portas digitais do arduino para motor
// direito (MD) e esquerdo (ME)
int MD_1 = 2;
int MD_2 = 3;
int MD_3 = 4;
int MD_4 = 5;
int ME_1 = 6;
int ME_2 = 7;
int ME_3 = 8;
int ME_4 = 9;
float Brad, Bgrau, Zrad, GyroRateZ, Psib, Psiba;
int c = 0, volta = 0, val = 0;
float Epsi, Spsi;
int upsi, up1, up2;
// Variáveis temporais
unsigned long timer = 0;
unsigned long delta_t;

void setup()
{
  int i;

  analogReference(EXTERNAL); // utilizando tensão de referência externa
  Serial.begin(9600); // taxa de transmissão de dados

  for (i = 2; i<=9; i++)
  { pinMode (i, OUTPUT); } // define as portas digitais como saída

  Serial.println("tempo posicao velocidade MD ME");

  timer = millis();

  // Inicializa o i2c
  Wire.begin();
}

void loop()
{
  int i;

  delta_t = millis() - timer; // calcula o tempo do loop
  timer = millis(); // reinicia temporizador
```

```

Serial.print(millis()); // apresenta o tempo no monitor
Serial.print(" ");

// i2c mestre inicia a transmiss o
Wire.beginTransmission(ADDRESS);
// envia o comando 'A' (get data), solicitando dados da bussola
Wire.write("A");
// encerra a transmiss o
Wire.endTransmission();
// espera que os dados seja processados
delay(6); // o datasheet sugere pelo menos 6000 microsegundos
// solicita 2 bytes a bussola
Wire.requestFrom(ADDRESS, 2);
// recebe os dados da bussola
if (2 <= Wire.available()) // se os 2 bytes estiverem disponiveis
{
    // ajusta o byte mais significante e soma
    heading = Wire.read();
    heading = heading << 8;
    heading += Wire.read();
}
// passa o valor da bussola para radianos
Psib=(3.14159265359*(heading/10.0))/180;

// Contador de voltas
if(c==0)
Psiba = Psib;
c=1;
if(Psib>5.0 && Psiba<0.6)
volta--; // decrementa volta
if(Psib<0.6 && Psiba>5.0)
volta++; // incrementa volta
Psiba = Psib; // atualiza o valor anterior da bussola
Brad=Psib + volta*6.28;
Bgrau=(Brad*180)/3.14159265359; // converte o valor da bussola para graus
Serial.print(Bgrau); // apresenta o valor da bussola em graus no monitor
Serial.print(" ");

// Giroscopio
val = analogRead(2); // le o pino 2, saida analogica
// converte o valor de tensao para graus/s
GyroRateZ=(val-372)*(3.3/1023)/0.00333;
// converte o valor do giroscopio para rad/s
Zrad=-(3.14159265359*GyroRateZ)/180;
// apresenta os dados do giroscopio em ras/s no monitor
Serial.print(Zrad);
Serial.print(" ");
Epsi=orientref-Brad; // calculo do erro de posicao
Spsi=k*Epsi-Zrad; // superficie de deslizamento

```

```

//sinal de comando no sistema movel
if($psi<0)
upsi=-1;
if($psi>0)
upsi=1;
if($psi==0)
upsi=0;

//sinais de comando dos propulsores
//propulsor 1
if((1 - upsi)<0)
up1=-1;
if((1 - upsi)>0)
up1=1;
if((1 - upsi)==0)
up1=0;

//propulsor 2
if((1 + upsi)<0)
up2=-1;
if((1 + upsi)>0)
up2=1;
if((1 + upsi)==0)
up2=0;
// Por seguranga, os propulsores serao desligados em 25 segundos
if(millis())<=25000)
{
if(up1>0) // sentido direto de up1 na velocidade maxima
{
Serial.print("1");
Serial.print(" ");
digitalWrite(MD_1, HIGH);
digitalWrite(MD_2, HIGH);
digitalWrite(MD_3, HIGH);
digitalWrite(MD_4, LOW);
}

if(up1==0) // up1 desligado
{
Serial.print("0");
Serial.print(" ");
digitalWrite(MD_1, LOW);
digitalWrite(MD_2, LOW);
digitalWrite(MD_3, LOW);
digitalWrite(MD_4, LOW);
}
}
}

```

```

if(up2>0) // sentido direto de up2 na velocidade maxima
{
Serial.print("1");
Serial.print(" ");
digitalWrite(ME_1, HIGH);
digitalWrite(ME_2, HIGH);
digitalWrite(ME_3, HIGH);
digitalWrite(ME_4, LOW);
}

if(up2==0) // up2 desligado
{
Serial.print("0");
Serial.print(" ");
digitalWrite(ME_1, LOW);
digitalWrite(ME_2, LOW);
digitalWrite(ME_3, LOW);
digitalWrite(ME_4, LOW);
}
}
else
{
Serial.print("0"); // up1 desligado
Serial.print(" ");
digitalWrite(MD_1, LOW);
digitalWrite(MD_2, LOW);
digitalWrite(MD_3, LOW);
digitalWrite(MD_4, LOW);

Serial.print("0"); // up2 desligado
Serial.print(" ");
digitalWrite(ME_1, LOW);
digitalWrite(ME_2, LOW);
digitalWrite(ME_3, LOW);
digitalWrite(ME_4, LOW);
}
Serial.println("");
delay(100); // aguarda antes de executar um novo ciclo
}

```