



UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
Faculdade de Engenharia
Departamento de Engenharia Eletrônica e Telecomunicações

PROJETO DE GRADUAÇÃO

SISTEMA DE POSICIONAMENTO DINÂMICO PARA UM PEQUENO VEÍCULO FLUTUANTE

Gustavo de Sá Amaral

Orientador: Prof. José Paulo Vilela Soares da Cunha
Coordenador da Disciplina: Prof. Paulo Sérgio Rodrigues Alonso

Março de 2008

Ficha Catalográfica

AMARAL, Gustavo de Sá

Sistema de Posicionamento Dinâmico
para um Pequeno Veículo Flutuante

x, 60 pp , 29,7 cm (UERJ, Engenharia
Eletrônica, 2008)

Projeto de Graduação – Universidade do Estado
do Rio de Janeiro.

1. Posicionamento Dinâmico
2. Estrutura Variável
3. Processamento de Imagem em Tempo Real

I. UERJ/FEN II. Título (série)

Agradecimentos

Ao meu orientador, José Paulo, que me fez aprender muito durante o projeto através dos seus conselhos e cobranças e que me deu entusiasmo para continuar aprendendo sobre a área de controle.

Aos professores Fernando Lizarralde e Lisandro Lovisollo pelas sugestões no início do trabalho.

À minha melhor amiga e namorada, Cristiane, pela paciência, compreensão e apoio em todas as horas. Sem ela, sei que teria sido muito mais difícil.

À minha família e aos meus amigos por sempre me ajudarem e acreditarem em mim.

RESUMO

Desenvolve-se um sistema de posicionamento dinâmico para um veículo flutuante de pequenas dimensões utilizando o controle a estrutura variável. O controlador é implementado em um microcomputador e seu projeto baseia-se no modelo da dinâmica do veículo, sem a necessidade de conhecer valores dos parâmetros dinâmicos nem de perturbações externas. A realimentação é feita através do processamento de imagem em tempo real. As imagens do veículo são captadas por uma câmera e processadas para que sejam obtidas as informações de posição e orientação. A partir destas informações estima-se a velocidade da embarcação, também utilizada no algoritmo de controle.

Também são descritos o circuito de acionamento dos motores e a construção da embarcação.

ABSTRACT

A dynamic positioning system for a small floating vehicle is developed using variable structure control. The controller is implemented in a microcomputer. Its design is based on the dynamic model of the vehicle without the need for the knowledge of parameters and external disturbances. The feedback is based on the real-time image processing. The images of the vehicle are captured by a camera and processed to get its position and orientation. The speed of the vehicle is estimated from this information, which is also used by the control algorithm.

The motor driver circuit and the building of the vehicle are also described.

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Objetivo | 4 |
| 1.2 | Descrição do Sistema de Posicionamento Dinâmico Proposto | 4 |
| 1.3 | Organização do Texto | 4 |
| 2 | Sistema de Visão | 6 |
| 2.1 | Seleção da Câmera | 7 |
| 2.1.1 | Levantamento das características relevantes | 7 |
| 2.1.2 | Comparações e conclusões | 10 |
| 2.2 | Processamento de Imagem | 11 |
| 2.2.1 | Transformação da imagem em imagem binária | 17 |
| 2.2.2 | Classificação dos objetos | 19 |
| 2.2.3 | Cálculo do centróide e da orientação | 20 |
| 2.3 | Escalamento | 28 |
| 2.4 | Implementação | 29 |
| 3 | Controle | 31 |
| 3.1 | Sistemas de Coordenadas | 31 |
| 3.2 | Modelagem da Dinâmica do Sistema | 32 |
| 3.2.1 | Equações de movimento | 33 |
| 3.2.2 | Modelo dos propulsores | 35 |
| 3.3 | Controle a Estrutura Variável | 36 |
| 3.3.1 | Controle da embarcação | 37 |
| 3.4 | Transformação de Coordenadas | 39 |
| 3.5 | Desacoplamento | 40 |
| 3.6 | Implementação | 43 |
| 4 | Resultados Experimentais | 44 |
| 5 | Conclusões | 50 |
| | Referências | 51 |
| | Apêndice | 52 |
| A | Embarcação e Circuito de Acionamento | 52 |
| A.1 | Embarcação | 52 |
| A.2 | Propulsores | 54 |
| A.3 | Acionamento dos Motores | 55 |
| B | Rotina de Controle | 59 |

LISTA DE FIGURAS

| | | |
|------|--|----|
| 1.1 | Exploração de petróleo em Summerland, Califórnia, 1902 | 1 |
| 1.2 | Exemplo de plataforma do tipo jaqueta | 2 |
| 1.3 | FPSO Sendje Berge | 3 |
| 1.4 | Digrama em blocos do sistema de posicionamento dinâmico proposto | 4 |
| 2.1 | Exemplo de imagem digital | 8 |
| 2.2 | Relação entre ângulo de abertura, campo de visão e distância | 9 |
| 2.3 | Disposição dos principais componentes do sistema de posicionamento dinâmico proposto | 12 |
| 2.4 | Campo de visão | 13 |
| 2.5 | Imagem do modelo da embarcação em escala de cinza | 14 |
| 2.6 | Mapa de pixels da região destacada na Figura 2.5 | 15 |
| 2.7 | Histograma da Figura 2.5 | 16 |
| 2.8 | Imagem binária da Figura 2.5 | 18 |
| 2.9 | Matriz binária após o processo de classificação | 20 |
| 2.10 | Exemplo de imagem binária com dois objetos | 21 |
| 2.11 | Centróides dos objetos na Figura 2.10 | 23 |
| 2.12 | Determinação da orientação do objeto | 25 |
| 2.13 | Raio maior e borda da embarcação | 26 |
| 2.14 | Cálculo do ângulo de orientação | 27 |
| 2.15 | Projeção da imagem no espaço de trabalho | 29 |
| 2.16 | Diagrama de fluxo do processamento de imagem e controle | 30 |
| 3.1 | Sistemas de coordenadas no espaço de trabalho | 31 |
| 3.2 | Diagrama em blocos do modelo simplificado da dinâmica do sistema | 35 |
| 3.3 | Plano de fase da dinâmica de um grau de liberdade | 38 |
| 3.4 | Disposição dos propulsores na embarcação | 41 |
| 3.5 | Diagrama em blocos simplificado do sistema de controle | 41 |
| 3.6 | Diagrama completo do sistema | 43 |
| 4.1 | Oscilação da embarcação no grau de liberdade ψ | 45 |
| 4.2 | Sinal de comando u_ψ em função do tempo | 45 |
| 4.3 | Oscilação da embarcação no grau de liberdade x_e | 46 |
| 4.4 | Sinal de comando u_x em função do tempo | 46 |
| 4.5 | Oscilação da embarcação no grau de liberdade y_e | 47 |
| 4.6 | Sinal de comando u_y em função do tempo | 47 |
| 4.7 | Trajetória da embarcação no plano horizontal | 48 |
| 4.8 | Trajetória da embarcação no plano horizontal em função do tempo | 49 |
| A.1 | Vista geral da embarcação | 52 |
| A.2 | Vista superior da embarcação sem a cobertura | 53 |
| A.3 | Aspecto dos motores e hélices | 54 |
| A.4 | Esquema básico de uma ponte “H” | 55 |
| A.5 | Esquema simplificado do circuito integrado L298HN | 56 |

| | | |
|-----|---|----|
| A.6 | Diagrama eletrônico para o controle das entradas do CI L298HN | 57 |
| A.7 | Diagrama do circuito de acionamento dos motores | 58 |
| A.8 | Placa com circuito de acionamento | 59 |

LISTA DE TABELAS

| | | |
|-----|--|----|
| 2.1 | Tabela comparativa de câmeras | 10 |
| 2.2 | Programa em MATLAB para a transformação da imagem em imagem binária | 18 |
| 2.3 | Programa em MATLAB para a obtenção da área e do centróide | 24 |
| 2.4 | Programa em MATLAB para a obtenção do raio maior e do ângulo de orientação | 28 |
| A.1 | Características do motor PM100-SG | 55 |
| A.2 | Lógica de polarização da saída do <i>driver</i> A | 56 |
| A.3 | Lógica de rotação dos propulsores | 59 |
| B.1 | Código da rotina de controle desenvolvida | 59 |

ABREVIATURAS

FPSO – Floating, Production, Storage and Offloading

FPS – Frames por Segundo

RGB – Red, Green, Blue

PD – Proporcional-Derivativo

PID – Proporcional-Integral-Derivativo

VSC – Variable Structure Control

PWM – Pulse Width Modulation

RPM – Rotações por Minuto

PC – Personal Computer

CI – Circuito Integrado

TTL – Transistor-Transistor Logic

CAPÍTULO 1 – INTRODUÇÃO

O petróleo tem se destacado ao longo dos anos por sua importância no cenário econômico mundial. A maior parte da energia consumida no mundo é produzida a partir deste recurso. Entre as reservas mundiais de petróleo, existe um número considerável de campos que se encontram afastados da costa e em grandes profundidades. No Brasil, 91% das reservas são marítimas e concentram-se predominantemente na Bacia de Campos, litoral norte do Estado do Rio de Janeiro (Tannuri, 2002).

A exploração marítima de óleo e gás teve início no final do século XIX na Califórnia e no Mar Cáspio. Naquele momento, a profundidade dos poços alcançava no máximo 6m e os equipamentos de perfuração e exploração eram apoiados por estruturas fixas construídas na costa de até 400m de comprimento como pode ser observado na Figura 1.1 (Faÿ, 1990).

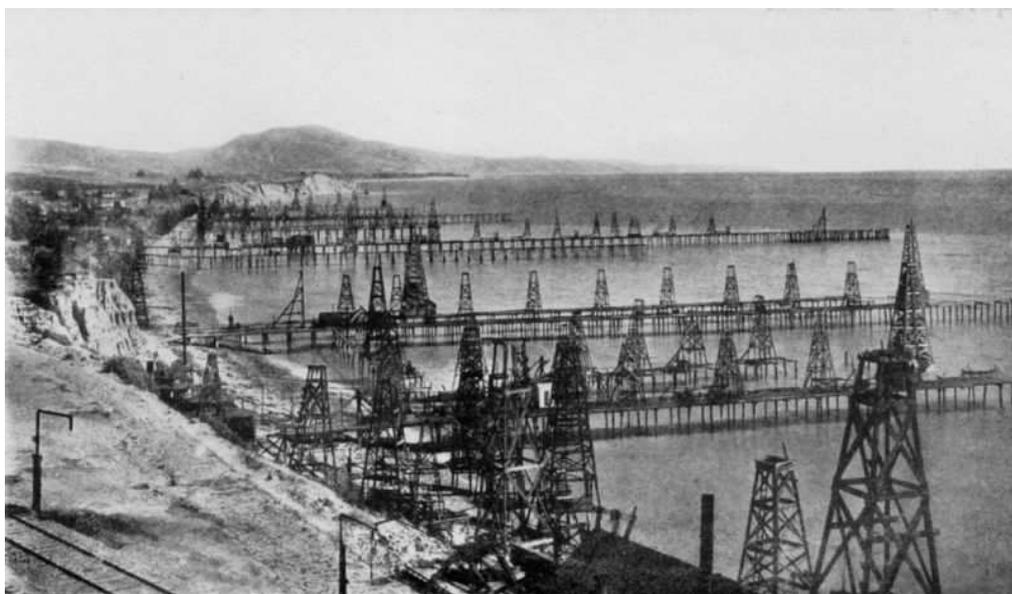


Figura 1.1: Exploração de petróleo em Summerland, Califórnia, 1902 (extraído de Faÿ, 1990).

A exploração de campos mais profundos e distantes da costa foi viabilizada por novos avanços tecnológicos no setor. Um deles foi o desenvolvimento de plataformas do tipo jaqueta. Segundo Tannuri (2002), as plataformas do tipo jaqueta consistem em estruturas de operação fixadas ao fundo do mar por treliças metálicas como a apresentada na Figura 1.2. Uma das limitações deste tipo de plataforma é

que sua viabilidade técnica e econômica é garantida apenas para profundidades inferiores a 500m.



Figura 1.2: Exemplo de plataforma do tipo jaqueta.

Após o desenvolvimento das plataformas do tipo jaqueta, surgiram os sistemas de amarração. Estes foram desenvolvidos de maneira a viabilizar as operações de prospecção a profundidades ainda maiores, em torno de 1.000m. De acordo com Faÿ (1990), sistemas de amarração são compostos por estruturas flutuantes amarradas por cabos fixados ao fundo do oceano que compensam as forças externas atuantes, como se fossem âncoras. A partir deste novo tipo de sistema, conceberam-se os FPSOs (*Floating, Production, Storage and Offloading Systems*) ou sistemas flutuantes de produção, armazenamento e descarga: navios petroleiros atuando na exploração de poços e no armazenamento e processamento do petróleo. A Figura 1.3 apresenta um exemplo de FPSO.



Figura 1.3: FPSO Sendje Berge (extraído de Equatorialoil, 2008).

Um dos principais problemas dos sistemas de amarração é que, como são constituídos por grandes massas flutuantes sustentadas por linhas flexíveis que estão sujeitas a forças ambientais, podem ocorrer movimentos oscilatórios que dificultam a execução de algumas operações (Faÿ, 1990). Desta forma, pesquisadores e engenheiros tiveram que desenvolver um método que pudesse proporcionar a continuidade das operações em mar aberto de maneira segura. Dentro deste contexto, surgiu o conceito de Posicionamento Dinâmico. Um Sistema de Posicionamento Dinâmico é um sistema que controla automaticamente a posição e a orientação de uma embarcação por meio de propulsão ativa.

O primeiro navio equipado com um sistema de posicionamento dinâmico foi desenvolvido em 1961, o qual possuía um controlador analógico que comandava os propulsores e era realimentado por um sensor *taut-wire*: sensor que utiliza um fio tensionado para a obtenção da posição da embarcação. Ao longo do tempo, outros navios foram equipados com sistemas de posicionamento dinâmico e passaram a ser utilizados em outras aplicações além da perfuração, tais como: pesquisa geofísica, suporte a veículos submarinos, lançamento de dutos e cabos e lançamento de foguetes em alto-mar.

1.1 – Objetivos

Os objetivos do Projeto de Graduação aqui proposto são: (i) construir um veículo flutuante de pequenas dimensões, (ii) desenvolver um algoritmo de processamento de imagem que possibilite a obtenção da posição do veículo através de uma câmera e (iii) desenvolver um sistema de controle para o posicionamento dinâmico do veículo.

1.2 – Descrição do Sistema de Posicionamento Dinâmico Proposto

A Figura 1.4 apresenta o diagrama em blocos do sistema de posicionamento dinâmico proposto. O controle é feito através de um microcomputador, que recebe a imagem capturada por uma câmera e processa estes dados para medir a posição e a orientação da embarcação a fim de gerar os sinais de comando para os propulsores (hélices). Os sinais de comando são enviados pelo microcomputador através de uma porta de saída paralela para um circuito com *drivers* de potência, que atuam diretamente no acionamento dos motores que movimentam os hélices.

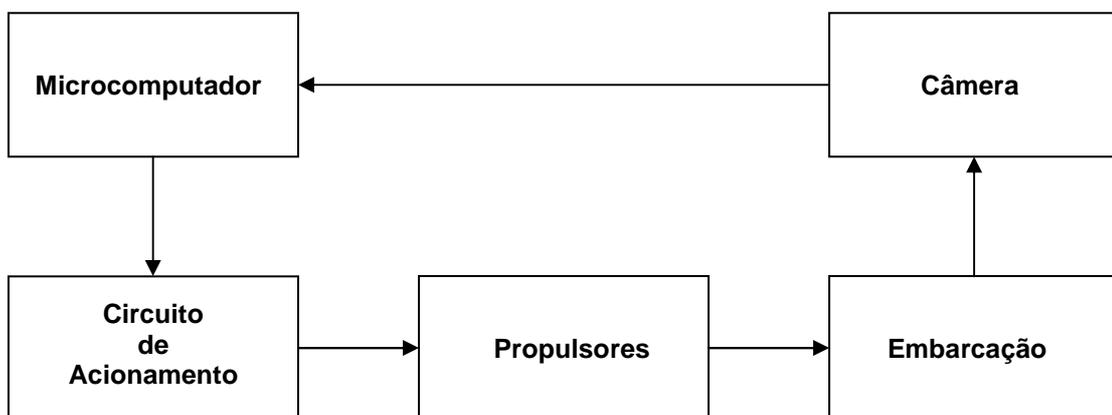


Figura 1.4: Diagrama em blocos do sistema de posicionamento dinâmico proposto.

1.3 – Organização do Texto

A organização do texto foi estabelecida da seguinte forma: primeiramente, no Capítulo 2, são descritos os métodos de aquisição e processamento de imagem utilizados para a medição da posição do veículo. No Capítulo 3 são desenvolvidos o

modelo da dinâmica da embarcação e o sistema de controle. O Capítulo 4 apresenta os resultados experimentais obtidos. Por fim, no Capítulo 5, são realizadas as considerações finais e as conclusões sobre o projeto desenvolvido.

CAPÍTULO 2 – SISTEMA DE VISÃO

Sistemas de visão são sistemas computacionais comumente utilizados na indústria. Destinam-se geralmente à inspeção, identificação e servo-movimentação visual. Um sistema de visão é basicamente constituído por uma câmara, a qual capturará uma imagem e retornará uma série de sinais elétricos que serão digitalizados e enviados a um processador principal através de uma interface de transmissão. Os dados enviados ao processador principal serão processados de acordo com o algoritmo de processamento de imagem definido, obtendo-se assim as informações desejadas da imagem (Pazos, 2002).

Segundo Pazos (2002), alguns fatores devem ser cuidadosamente ajustados durante a implementação de um sistema de visão. O posicionamento da câmara deve ser bem observado para que tanto o tamanho quanto a orientação do campo de visão fiquem adequados. Campos de visão maiores do que o necessário diminuem a resolução dos objetos, o que pode tornar mais difícil a detecção de determinadas características. Em contrapartida, campos de visão muito pequenos podem ser insuficientes para determinadas aplicações. A iluminação do ambiente é outro fator importante. Pouca luminosidade pode dificultar a discriminação entre objeto e fundo. Neste projeto, como o sistema de visão trabalhará fundamentalmente distinguindo níveis de luminância para a detecção da embarcação, método que será detalhado na Seção 2.2, uma boa luminosidade será essencial. Por fim, pode-se destacar o problema da interferência visual. Trabalhar em ambientes com muita interferência pode tornar o resultado do processamento inconsistente, ou mesmo consideravelmente mais lento. Se, por exemplo, fosse necessário detectar um objeto claro em um ambiente escuro, seria interessante que não houvesse, no ambiente de visão, muitos outros objetos de luminância equivalente ao objeto de interesse.

O método utilizado neste projeto para a aquisição da posição e da orientação da embarcação é o processamento de imagem em tempo real. Segundo Kehtarnavaz e Gamadia (2006), o conceito de “tempo real” possui três interpretações principais: tempo real no contexto da percepção, tempo real no contexto da engenharia de *software* e tempo real no contexto do processamento de sinais. Kehtarnavaz e Gamadia (2006) descrevem o conceito de tempo real no contexto da percepção como a interação entre um humano e um dispositivo

computacional quando há a percepção de que as respostas aos comandos são instantâneas. Já a interpretação sobre o tempo real no contexto da engenharia de *software* relaciona a satisfação de um tempo limite para que não haja erro em um sistema. No contexto do processamento de sinais, o conceito de tempo real baseia-se na idéia de que o processamento seja executado no tempo disponível entre duas amostras de entrada.

Para um sistema de controle, nota-se que o conceito de tempo real está diretamente relacionado ao tempo de resposta do sistema, através do qual será definida a frequência de amostragem a ser utilizada.

Neste capítulo, será feita a descrição do processo de seleção da câmera utilizada e dos algoritmos de processamento de imagem desenvolvidos.

2.1 – Seleção da Câmera

Para a especificação da câmera a ser utilizada, primeiramente foram relacionadas as características mais relevantes à execução do projeto. Em uma segunda etapa, foram pesquisadas as câmeras disponíveis no mercado doméstico.

2.1.1 – Levantamento das Características Relevantes

- **Compatibilidade com o sistema operacional Linux**

Tendo em vista a possibilidade de desenvolvimentos adicionais para o projeto e o fato do Linux ser um sistema livre, definiu-se que a câmera deveria ser compatível com este sistema operacional. Além disso, soma-se o fato de já existirem diversos programas de aquisição de imagens *open source* (código aberto) para esta plataforma.

- **Taxa de amostragem**

Um vídeo é formado por várias imagens projetadas em seqüência. Cada imagem é chamada de quadro ou *frame*. A taxa de amostragem consiste na quantidade de quadros por segundo ou FPS (*frames* por segundo) que uma câmera é capaz de captar (HDTV Source, 2008). Como a câmera será utilizada para o controle em tempo real de um sistema dinâmico, uma baixa taxa de amostragem

poderia gerar oscilações no sistema de controle. Desta forma, definiu-se que a câmera deveria suportar uma taxa de amostragem de 30 *frames* por segundo operando em sua resolução máxima.

- **Resolução**

A resolução pode ser definida como a quantidade de *pixels* (*picture elements*), que compõe uma imagem. De acordo com Pazos (2002), os *pixels* representam o menor elemento de uma imagem capaz de ser discriminado por um dispositivo. Imagens representadas em alta resolução possibilitam uma definição mais precisa das propriedades dos objetos presentes. Definiu-se que a resolução máxima da câmera deveria ser no mínimo, 640x480 *pixels*. A Figura 2.1 é um exemplo de imagem digital.

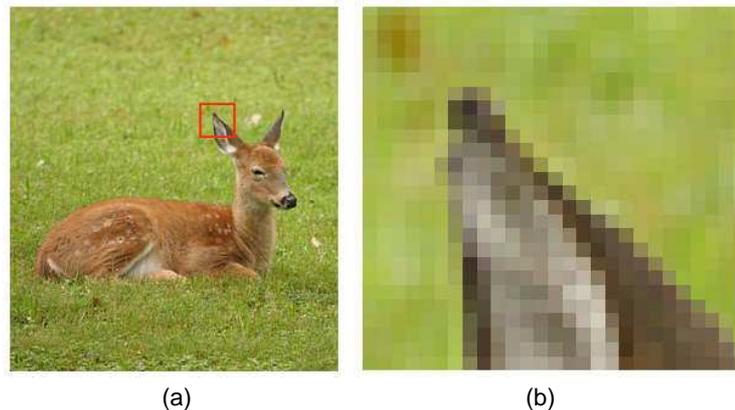


Figura 2.1: (a) imagem completa; (b) detalhe da imagem (a) ampliado, onde se vê cada *pixel* como um pequeno quadrado.

- **Ângulo de abertura**

O ângulo de abertura estabelece a relação entre a distância da câmera à superfície observada e a área do campo de visão. A Figura 2.2 apresenta a relação mencionada.

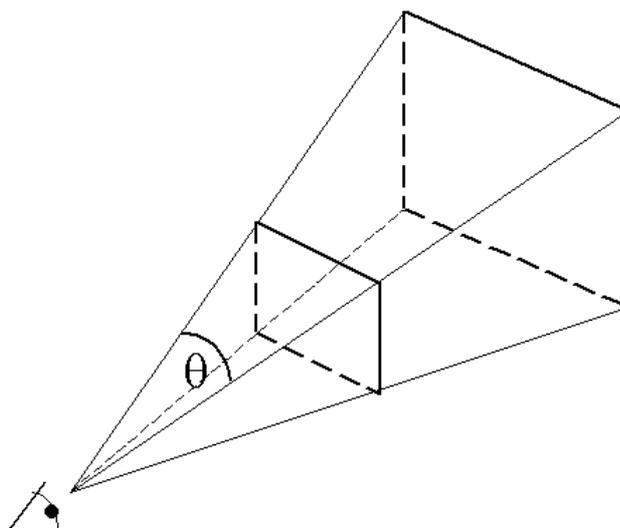


Figura 2.2: Relação entre ângulo de abertura, campo de visão e distância.

Um ângulo de abertura maior possibilita que a câmera seja posicionada próxima ao objeto de interesse. Entretanto, devido à geometria das lentes que proporcionam maiores ângulos de abertura, denominadas *grande-angulares*, é possível que sejam observadas alterações na perspectiva da imagem: os objetos podem parecer esticados ou mais distantes do que realmente são, o que poderia atrapalhar sensivelmente a definição das características destes objetos (Bock, 2008).

- **Custo de 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 da câmera também foi levado em consideração.

2.1.2 – Comparações e Conclusões

A Tabela 2.1 apresenta um comparativo dos modelos de câmeras encontrados na pesquisa.

Tabela 2.1: Tabela comparativa de câmeras. Legenda: N/D – Não disponível; fps – *Frames por segundo*.

| Marca | Chipset | Modelo | Resolução máxima | Ângulo de abertura | Taxa de amostragem (fps**) | Preço (R\$) (01/2007) |
|-----------|--------------------|------------------------------|------------------|--------------------|------------------------------|-----------------------|
| Creative | Sq930C | WebCam Live Pro | 640x480 | N/D* | 30 (640x480) | 299,00 |
| Creative | Zc0301 | WebCam NX | 640x480 | 40° | N/D | 119,00 |
| Creative | Zc0301P | WebCam NX Pro | 640x480 | 40° | N/D | 208,00 |
| Creative | Spca505b | WebCam NX Ultra | 640x480 | 75° | 30 (320x240) 15 (640x480) | 319,00 |
| D-Link | OV518+ OV519 | WebCam DSB-C310 | 640x480 | N/D | 30 (320x240) 15 (640x480) | 129,00 |
| Genius | Spca561a | Genius VideoCam Express (V2) | 352x288 | 52° | 30 (352x288) | 100,55 |
| Microsoft | Sn9c105r | LifeCam VX-1000 | 640x480 | N/D | 30 (320x240) 15 (640x480) | 80,00 |
| Microsoft | Sn9c105r | LifeCam VX-3000 | 640x480 | N/D | 30 (640x480) | 160,00 |
| Logitech | Spca561a Zc030x | WebCam QuickCam Chat | 640x480 | N/D | 30 (320x240) 15 (640x480) | 169,95 |
| Logitech | Spca525a | WebCam QuickCam Pro 5000 | 640x480 | N/D | 30 (640x480) | 525,55 |

A pesquisa foi realizada através de *sites* de busca de produtos na *internet*, onde foram selecionadas as câmeras que poderiam ser encontradas com maior facilidade no mercado doméstico. O segundo passo foi verificar, através de *sites* especializados, quais modelos possuíam *driver* para o sistema operacional Linux. Em seguida, foram obtidas todas as características relevantes ao projeto de cada modelo. Este estudo foi prejudicado devido à falta de informações disponíveis acerca de algumas câmeras, tais como ângulo de abertura e taxa de amostragem. É importante destacar que apenas as câmeras compatíveis com o sistema operacional Linux foram incluídas na tabela.

A partir das informações apresentadas na Tabela 2.1, foi possível selecionar a câmera *Microsoft LifeCam VX-3000* como a melhor opção para o projeto. A princípio, pensou-se em escolher o modelo *Creative WebCam NX Ultra*, pois este possuía um grande ângulo de abertura. Esta opção foi desconsiderada quando constatou-se que sua taxa de amostragem à resolução máxima era de apenas 15fps. Considerou-se como segunda opção a câmera *Logitech WebCam QuickCam Chat*, porém não foi possível obter informações exatas sobre a taxa de amostragem do modelo.

Uma característica interessante observada na *Genius VideoCam Express (V2)* foi o fato da câmera transmitir as imagens em *bitmap* (sem compactação), o que seria muito importante para o projeto em termos de velocidade e qualidade, já que não haveria o custo computacional da decodificação do vídeo. Porém, a resolução máxima ficou abaixo do limite estabelecido.

Os modelos *Logitech WebCam QuickCam Pro 5000* e *Creative WebCam Live Pro* possuíam as mesmas características do modelo escolhido (*Microsoft LifeCam VX-3000*), mas como pode ser observado através da tabela, estes modelos tinham custo mais elevado.

O fato de não terem sido obtidas informações sobre o ângulo de abertura da câmera selecionada não foi considerado um problema, já que será possível posicioná-la de modo a ajustar o campo de visão.

2.2 – Processamento de Imagem

De acordo com Pazos (2002), o processamento de imagem consiste basicamente em um conjunto de técnicas que permitem extrair informações de imagens. Possui aplicações em diversas áreas, tais como reconhecimento de formas, reconhecimento de padrões e robótica. Na maioria dos casos o processamento de imagem é feito digitalmente. Tal fato deve-se à facilidade de implementação em computadores, confiabilidade e precisão.

Segundo Jain (1989), uma imagem digital é uma matriz de elementos reais ou complexos representados por um número finito de *bits* denominados *pixels*.

Nesta seção, será detalhado o algoritmo de processamento de imagem desenvolvido para o sistema de posicionamento dinâmico. O algoritmo deverá extrair duas informações básicas: o centróide do objeto, neste caso a embarcação, e sua orientação em relação ao campo de visão. A Figura 2.3 apresenta o esquema básico de montagem do sistema.

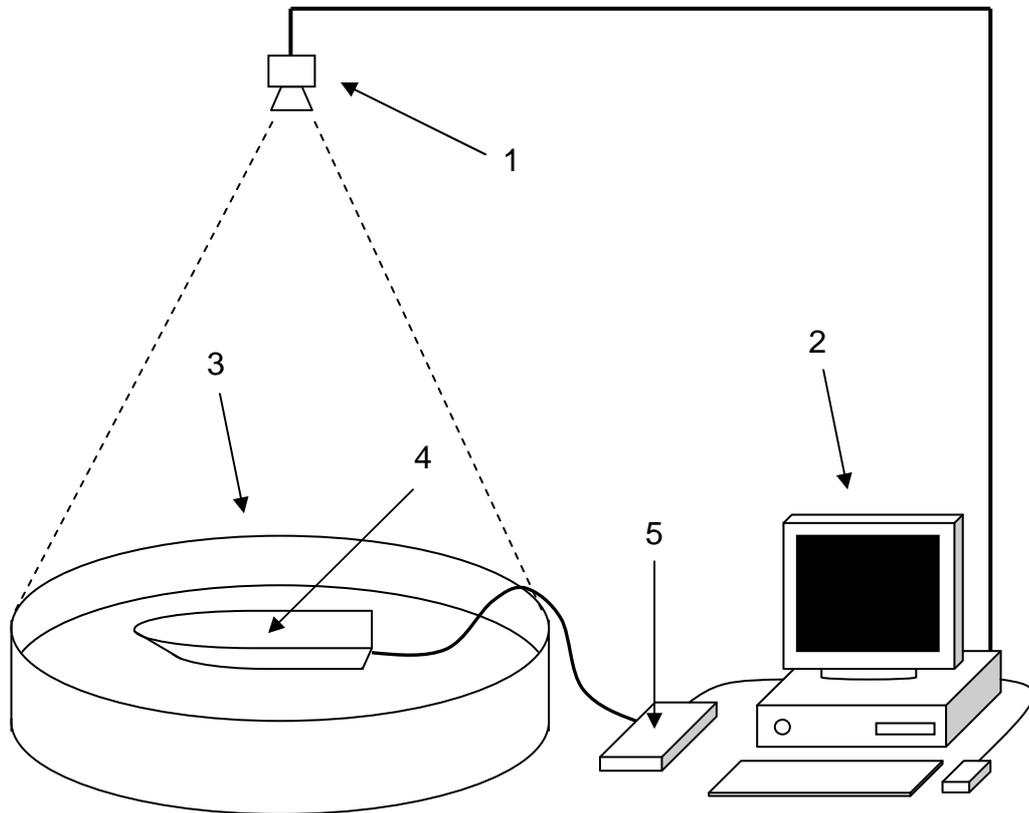


Figura 2.3: Disposição dos principais componentes do sistema de posicionamento dinâmico proposto: (1) câmera; (2) microcomputador; (3) piscina; (4) embarcação; (5) circuito de acionamento dos propulsores (*driver*).

A Figura 2.4 apresenta o campo de visão do sistema. O sistema de coordenadas utilizado é o sistema da câmera, com unidades em *pixels*.

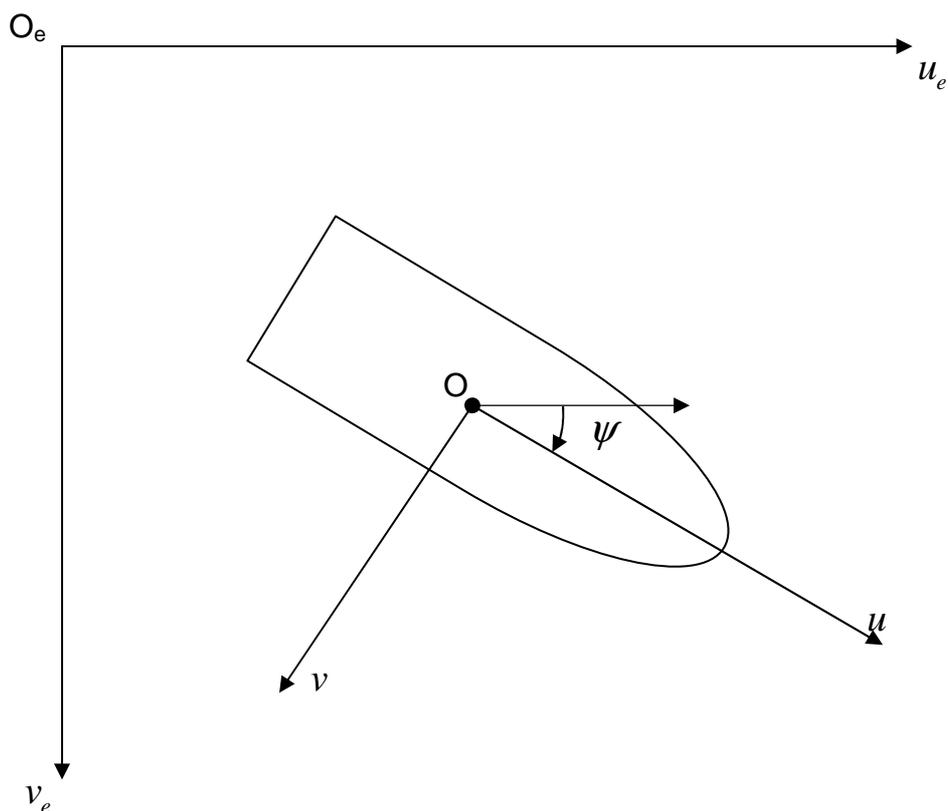


Figura 2.4: Campo de Visão

Como é possível observar na Figura 2.4, a embarcação será visualizada de cima.

Em um primeiro momento, o objetivo será apenas discriminar o objeto de interesse do fundo. Neste caso, discriminar a embarcação da água. Existem diversas maneiras de realizar esta tarefa. Neste projeto, optou-se por uma forma simples. O método escolhido consiste em distinguir níveis diferentes de luminância, ou seja, distinguir um objeto claro de um fundo escuro ou o contrário. Foi definido que a embarcação deveria ser branca para facilitar o processo. Sendo assim, será necessário identificar um objeto branco na água, a qual, obviamente, deverá ter uma tonalidade mais escura.

Para identificar níveis de luminância é necessário apenas que a imagem seja representada em escala de cinza. Caso fosse necessário trabalhar com imagens coloridas, a carga de dados a ser processada seria muito maior, pois cada *pixel* seria representado por um vetor com três elementos. No sistema de cores *RGB* (*red*, *green*, *blue*), os elementos do vetor seriam *R*, *G* e *B*. Utilizando imagens em escala

de cinza, cada *pixel* da imagem será representado apenas por uma grandeza escalar de 0 a 255 (8 *bits* de resolução) que equivale à luminância do *pixel*.

A Figura 2.5 apresenta o modelo utilizado para a descrição dos métodos de processamento de imagem. O modelo é uma representação da vista superior do veículo flutuante de pequenas dimensões e foi escolhido para facilitar a demonstração dos métodos.

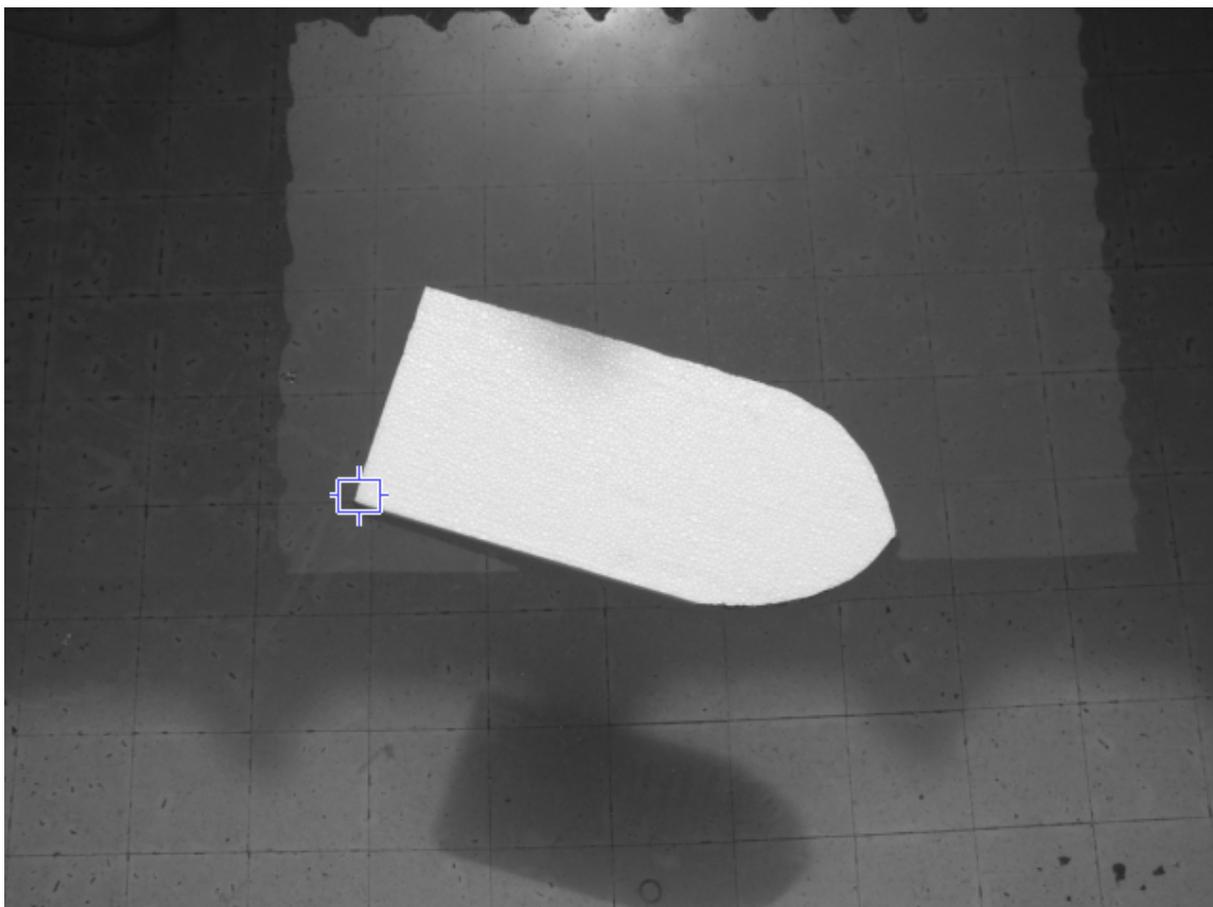


Figura 2.5: Imagem do modelo em escala de cinza. O retângulo em destaque é apresentado na Figura 2.6

| | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| 81 | 81 | 82 | 83 | 82 | 83 | 82 | 81 | 84 | 83 | 89 | 103 | 204 | 221 | 223 | 221 | 221 | 225 | 236 | 2 |
| 82 | 82 | 82 | 82 | 82 | 84 | 82 | 82 | 84 | 86 | 84 | 134 | 209 | 220 | 219 | 218 | 223 | 226 | 231 | 2 |
| 81 | 83 | 81 | 83 | 82 | 83 | 82 | 83 | 85 | 85 | 94 | 178 | 213 | 218 | 221 | 219 | 222 | 222 | 223 | 2 |
| 82 | 82 | 81 | 79 | 79 | 82 | 83 | 82 | 83 | 83 | 100 | 202 | 216 | 223 | 221 | 220 | 224 | 221 | 223 | 2 |
| 81 | 81 | 80 | 75 | 78 | 81 | 83 | 84 | 82 | 80 | 121 | 213 | 219 | 218 | 220 | 225 | 225 | 222 | 230 | 2 |
| 82 | 82 | 79 | 81 | 81 | 80 | 82 | 83 | 82 | 82 | 156 | 213 | 215 | 216 | 221 | 223 | 222 | 224 | 224 | 2 |
| 82 | 79 | 79 | 78 | 80 | 82 | 83 | 79 | 80 | 85 | 182 | 211 | 216 | 218 | 218 | 219 | 229 | 231 | 231 | 2 |
| 81 | 78 | 76 | 79 | 79 | 80 | 79 | 75 | 77 | 100 | 195 | 211 | 217 | 221 | 218 | 218 | 227 | 232 | 232 | 2 |
| 80 | 78 | 80 | 80 | 81 | 81 | 79 | 82 | 79 | 124 | 203 | 210 | 217 | 216 | 222 | 222 | 223 | 225 | 224 | 2 |
| 79 | 79 | 77 | 80 | 82 | 81 | 81 | 81 | 89 | 157 | 203 | 205 | 209 | 212 | 220 | 224 | 224 | 220 | 223 | 2 |
| 79 | 75 | 74 | 80 | 79 | 76 | 76 | 78 | 82 | 144 | 194 | 201 | 202 | 207 | 214 | 216 | 216 | 221 | 224 | 2 |
| 71 | 70 | 69 | 73 | 72 | 70 | 71 | 73 | 66 | 72 | 114 | 151 | 182 | 199 | 204 | 206 | 213 | 216 | 217 | 2 |
| 76 | 74 | 74 | 75 | 79 | 80 | 78 | 76 | 72 | 62 | 70 | 82 | 99 | 126 | 154 | 183 | 202 | 206 | 206 | 2 |
| 76 | 76 | 75 | 77 | 77 | 79 | 78 | 74 | 73 | 66 | 65 | 67 | 69 | 76 | 87 | 107 | 130 | 154 | 170 | 1 |
| 77 | 77 | 78 | 77 | 79 | 79 | 75 | 73 | 73 | 66 | 69 | 65 | 66 | 66 | 66 | 73 | 82 | 92 | 103 | 1 |
| 76 | 78 | 81 | 77 | 75 | 78 | 77 | 75 | 76 | 72 | 66 | 67 | 68 | 66 | 65 | 64 | 65 | 70 | 76 | 6 |
| 79 | 79 | 82 | 78 | 76 | 79 | 79 | 79 | 77 | 75 | 72 | 69 | 67 | 67 | 63 | 65 | 65 | 64 | 65 | 6 |

Figura 2.6: Mapa de pixels da região destacada na figura 2.5.

Pode-se perceber, através da Figura 2.6, a diferença entre a intensidade luminosa da embarcação e a intensidade luminosa da água. A Figura 2.7 apresenta um histograma da Figura 2.5. Nela estão representados os níveis de luminância da imagem e suas respectivas freqüências na matriz.

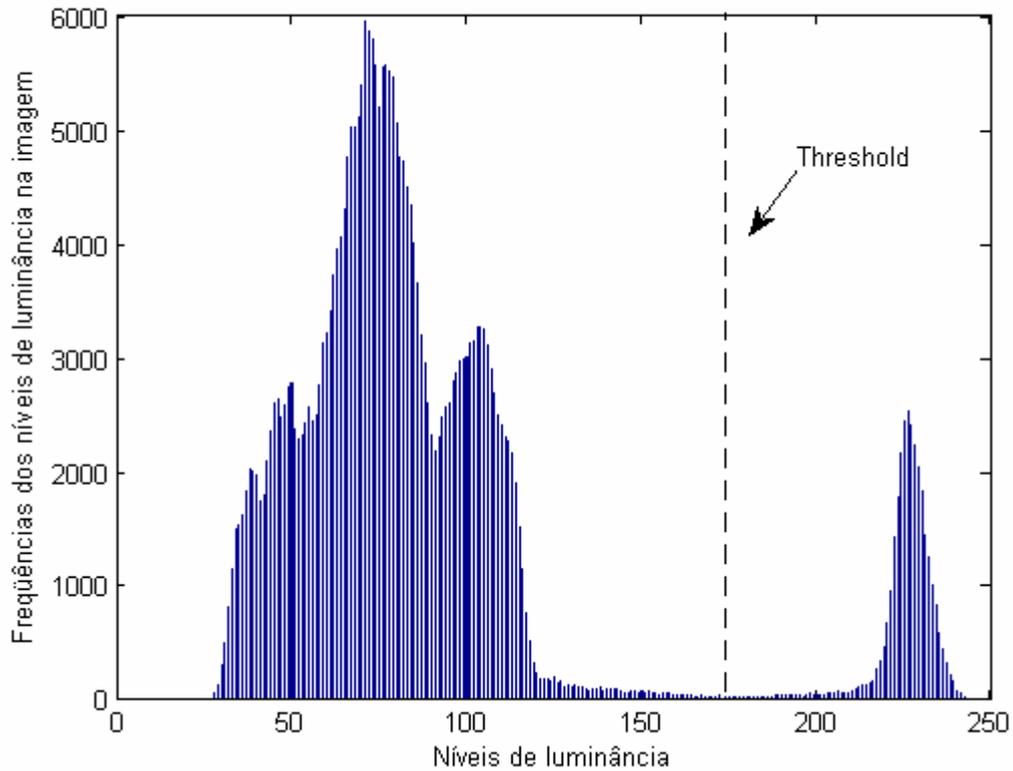


Figura 2.7: Histograma da Figura 2.5

O Primeiro grupo de barras no histograma representa os níveis de luminância da água, seus valores compreendem-se no intervalo de 23 até aproximadamente 150. O segundo grupo representa a embarcação. Seu intervalo de valores ficou entre 190 e 248. Na Figura 2.7, pode-se observar uma linha tracejada localizada entre os níveis 150 e 200 que representa o limiar escolhido para a separação do objeto e do fundo (*Threshold*). Desta forma, fica claro o método proposto para a identificação o objeto e sua separação do ambiente (água).

Para explicar de maneira mais detalhada o algoritmo de processamento de imagem, será feita uma subdivisão em três partes básicas: transformação da imagem em imagem binária, classificação dos objetos e cálculo das informações de interesse: centróide e orientação.

2.2.1 – Transformação da Imagem em Imagem Binária

A transformação da imagem captada em imagem binária faz com que a imagem em níveis de cinza seja transformada em uma imagem em preto e branco. Se o ambiente de visão fosse um ambiente ideal, o objeto de interesse seria identificado apenas utilizando-se esta técnica. A imagem binária resultante seria uma imagem com a forma da embarcação vista de cima na cor branca em um fundo preto (Pazos, 2002). Na prática, será necessário aplicar ainda outras técnicas para identificar o objeto desejado. Esta etapa consiste basicamente em identificar o nível de luminância da embarcação, varrer a matriz da imagem captada e criar uma nova matriz onde os elementos que representam os níveis de luminância do fundo sejam 0 (pretos) e os que representam os níveis de luminância da embarcação sejam 1 (brancos).

Utilizando a Figura 2.5 como exemplo, é possível perceber que a embarcação não possui um único nível de luminância, mas sim um intervalo, o qual é esperado que seja curto. Neste caso, o intervalo observado foi entre aproximadamente 190 e 248. Desta forma, será criada uma matriz binária definindo-se que, elementos com luminância maior ou igual a 160 serão 1 (brancos), e elementos com luminância menor que 160 serão 0 (pretos). A Figura 2.8 apresenta o resultado deste processo aplicado à imagem da Figura 2.5.

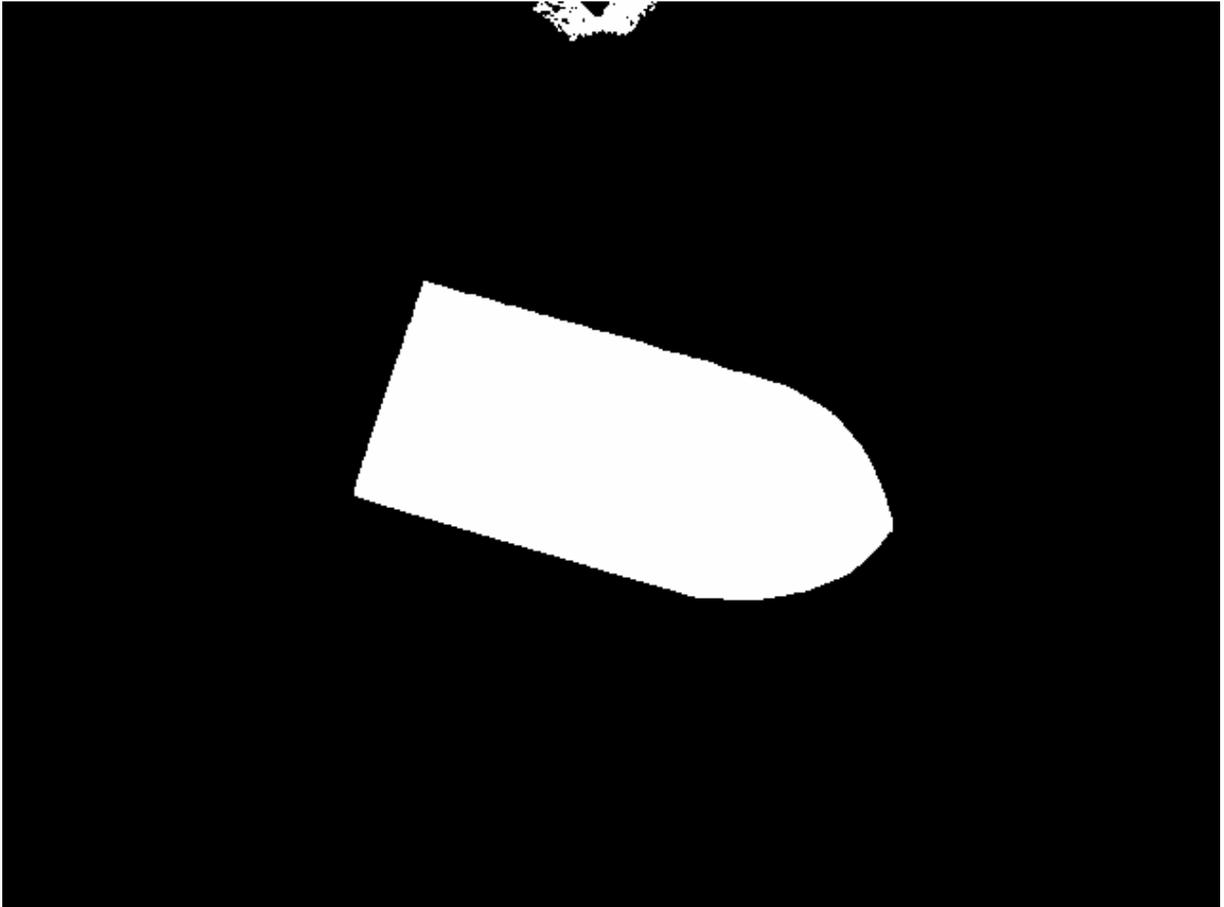


Figura 2.8: Imagem binária da Figura 2.5

É possível perceber, na parte superior da imagem na Figura 2.8, um grupo de *pixels* brancos que não faz parte da embarcação. Este grupo de *pixels* é denominado ruído. O tratamento deste efeito será detalhado na próxima etapa do processamento. A Tabela 2.2 apresenta um trecho de programa em MATLAB que converte uma imagem em escala de cinza para binária.

Tabela 2.2: Programa em MATLAB para a transformação da imagem em imagem binária.

| Programa em MATLAB |
|--|
| <pre>[m,n]=size(Imagem); %Obtem as dimensoes da imagem captada. ImagemBinaria=zeros(m,n); for i=1:m for j=1:n if Imagem(i,j)>=160 ImagemBinaria(i,j)=1; end end end end</pre> |

2.2.2 – Classificação dos objetos

A partir da matriz binária gerada na primeira etapa, será feita a classificação dos objetos presentes na imagem. Conforme mencionado na Subseção anterior, em um ambiente ideal, deveria haver na matriz binária apenas um conjunto de elementos com valor 1 que representariam a embarcação e um outro conjunto de elementos com valor 0 representando o fundo. Na prática, é provável que haja alguma interferência visual: alguns pontos no campo de visão que tenham luminância equivalente à da embarcação. Desta forma, haverá mais de um objeto branco sobre o fundo preto, e não apenas o objeto de interesse. Esta etapa do processamento consiste em segmentar a imagem em diversos objetos para que seja possível identificar o objeto desejado em meio ao ruído gerado pela interferência visual.

O método utilizado para classificar os objetos presentes na imagem consiste em detectar conjuntos de *pixels* conectados e atribuir a cada um destes conjuntos um *label*, ou etiqueta (Jain 1989). Como este método será aplicado a uma imagem binária, consideram-se objetos apenas os *pixels* com valor 1. Se, por exemplo, houvesse dois grupos de *pixels* com valor 1 (brancos) separados, seria atribuído a um dos grupos o valor 1 para cada um dos seus elementos e ao outro grupo o valor 2. Os *pixels* com valor 0, que representam o fundo, não seriam alterados. Desta forma, é possível criar uma base de dados com informações sobre as características de cada um destes grupos de *pixels* como: área, centróide e orientação. A imagem na Figura 2.8 será utilizada como exemplo para o método.

Pode-se perceber através da Figura 2.9 que agora a matriz tem dois objetos devidamente classificados. O objeto 1 que é a embarcação e o objeto 2 que é o ruído gerado pela interferência visual. A partir de agora será possível criar uma base de dados com as características de cada um destes objetos. É importante destacar que, para o processo de classificação dos objetos, foi utilizada uma função padrão do MATLAB: *bwlabel*. Esta função foi utilizada, pois o código desenvolvido para o processo de classificação dos objetos ficou muito lento e não atendeu às necessidades do projeto.

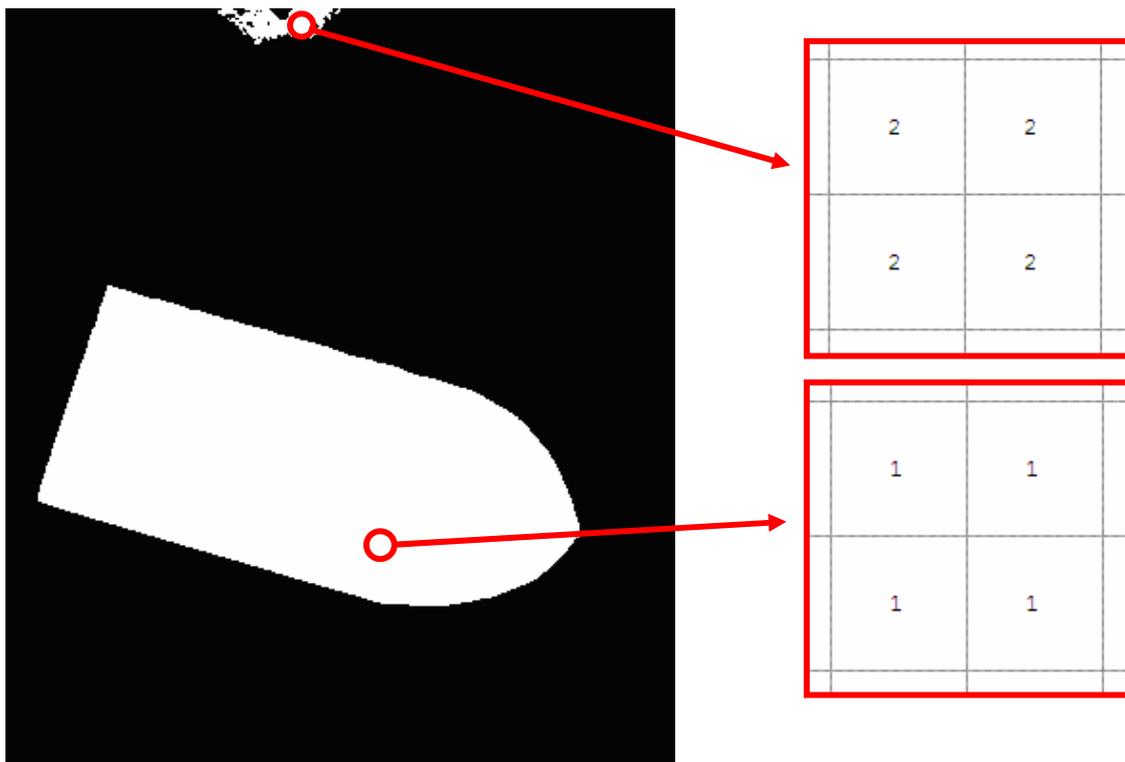


Figura 2.9: Matriz binária após o processo de classificação.

2.2.3 – Cálculo do Centróide e da Orientação

Após o processo de classificação descrito em 2.2.2, será descrita, nesta Seção, a etapa final do processamento. Esta etapa consiste na obtenção das informações referentes ao objeto de interesse: centróide e orientação.

Primeiramente, será criada a partir da matriz resultante do processo de classificação, uma base de dados relacionando o *label* de cada objeto à sua área. Considerando que o ambiente de visão tenha sido bem preparado, a interferência visual não deverá ser considerável. Desta forma, a embarcação deverá ser o objeto de maior área. A partir de então, serão calculadas apenas as propriedades do objeto de interesse: o objeto de maior área.

- **Área**

A área é a quantidade de *pixels* que representam um determinado objeto na imagem (Pazos 2002). A equação

$$Area(L) = \sum_{u=1}^{u=m} \sum_{v=1}^{v=n} \delta(p(u,v) - L), \quad (2.1)$$

onde

$$\delta(n) = \begin{cases} 1, & \text{se } n = 0, \\ 0, & \text{se } n \neq 0, \end{cases} \quad (2.2)$$

define o método utilizado para o cálculo da área.

Em (2.1), L é o *label* do objeto atribuído no processo de classificação, m e n representam o número total de linhas e colunas da matriz respectivamente, e $p(u,v)$ é o valor do *pixel* na coordenada (u,v) , que indica o *label* do objeto ao qual o *pixel* pertence.

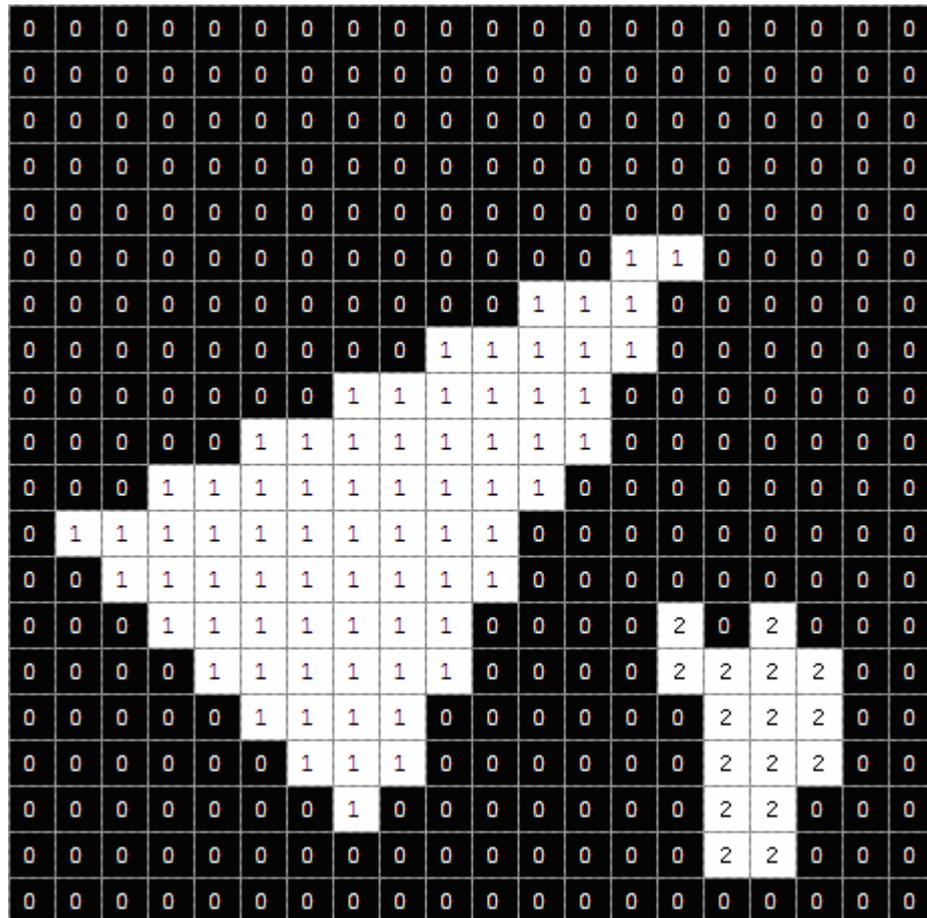


Figura 2.10: Exemplo de imagem binária com dois objetos

A Figura 2.10 apresenta um exemplo de imagem binária com dois objetos classificados. Aplicando a equação (2.1) ao exemplo da Figura 2.10, obtém-se:

Objeto 1:

$$Area(1) = \sum_{u=1}^{u=20} \sum_{v=1}^{v=20} \delta(p(u,v) - 1)$$

$$Area(1) = 73$$

Objeto 2:

$$Area(2) = \sum_{u=1}^{u=20} \sum_{v=1}^{v=20} \delta(p(u,v) - 2)$$

$$Area(2) = 16$$

Desta forma, nota-se que a área de cada objeto é igual ao número de *pixels* que o compõe.

- **Centróide**

Centróide é a posição do centro geométrico de um determinado objeto na imagem. As equações

$$CM_u(L) = \frac{\sum_{u=1}^{u=m} \sum_{v=1}^{v=n} \delta(p(u,v) - L)u}{Area(L)} \quad (2.2)$$

$$CM_v(L) = \frac{\sum_{u=1}^{u=m} \sum_{v=1}^{v=n} \delta(p(u,v) - L)v}{Area(L)} \quad (2.3)$$

descrevem o cálculo desta propriedade, onde u é a coluna do *pixel* $p(u,v)$ na matriz que representa a imagem, e v é a linha. CM_u e CM_v representam as coordenadas u e v do centróide (em *pixels*).

Aplicando as equações (2.2) e (2.3) ao exemplo na Figura 2.10, obtém-se:

Objeto 1:

$$CM_u(1) = \frac{\sum_{u=1}^{u=20} \sum_{v=1}^{v=20} \delta(p(u,v)-1)u}{73} = 8,53 \quad CM_v(1) = \frac{\sum_{u=1}^{u=20} \sum_{v=1}^{v=20} \delta(p(u,v)-1)v}{73} = 11,83$$

Objeto 2:

$$CM_u(2) = \frac{\sum_{u=1}^{u=20} \sum_{v=1}^{v=20} \delta(p(u,v)-2)u}{16} = 16,62 \quad CM_v(2) = \frac{\sum_{u=1}^{u=20} \sum_{v=1}^{v=20} \delta(p(u,v)-2)v}{16} = 16,31$$

A Figura 2.11 apresenta a localização dos centróides dos dois objetos.

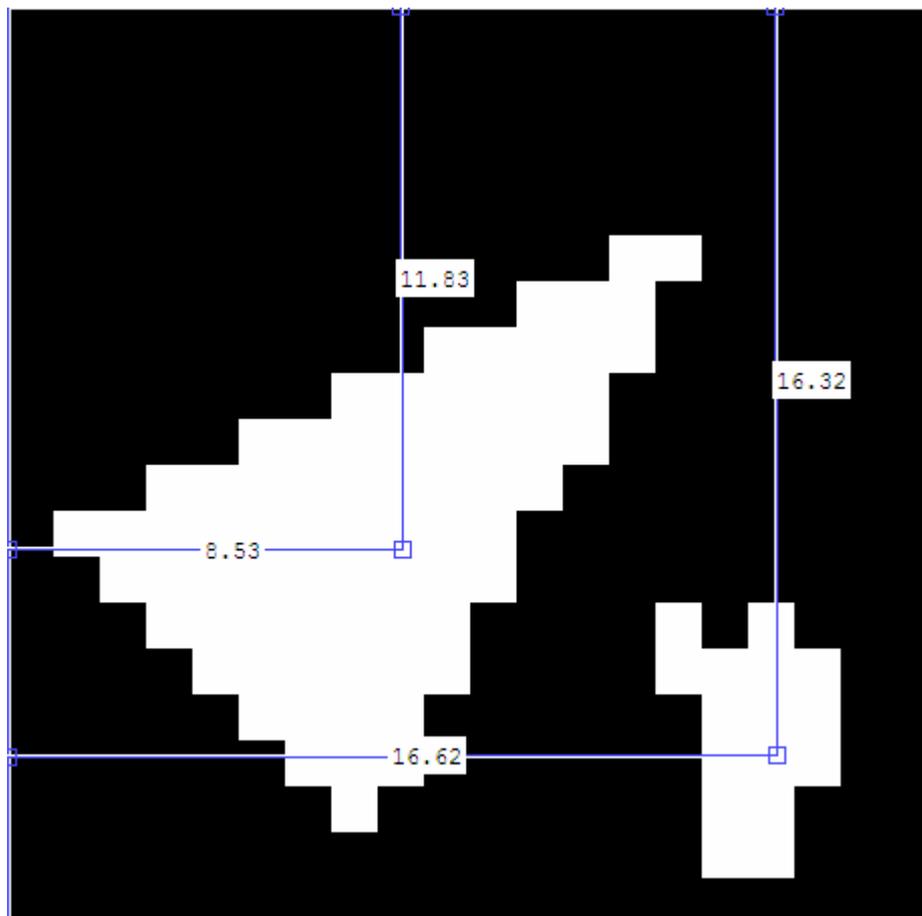


Figura 2.11: Centróides dos objetos na Figura 2.10

A Tabela 2.3 apresenta o código MATLAB utilizado para a obtenção da área e do centróide dos objetos presentes na imagem.

Tabela 2.3: Programa em MATLAB para a obtenção da área e do centróide.

| Programa em MATLAB |
|---|
| <pre> MaiorArea=0; for i=1:m for j=1:n if M(i,j)~=0 %"M" é a matriz da imagem binária. Area(M(i,j))=Area(M(i,j))+1; %Grava no vetor "Area" a area de cada um dos objetos encontrados. TotalElementos(M(i,j))=TotalElementos(M(i,j))+1; %Grava no vetor "TotalElementos" a quantidade de elementos (pixels) de cada objeto. u(M(i,j))=j+u(M(i,j)); %Somatório das colunas v(M(i,j))=i+v(M(i,j)); %Somatório das linhas if Area(M(i,j))>MaiorArea %Caso a area do objeto seja maior que a maior área encontrada até o momento. MaiorArea=Area(M(i,j)); %Grava a area Objeto=(M(i,j)); %Grava o label do objeto de maior área. end end end end cmu=u(Objeto)/TotalElementos(Objeto); %Coordenada Ue do centro de massa. cmv=v(Objeto)/TotalElementos(Objeto); %Coordenada Ve do centro de massa. </pre> |

- **Orientação**

De acordo com Pazos (2002), existem diversas definições para esta propriedade. De maneira geral, a orientação é definida como o ângulo de inclinação de um objeto dentro da imagem. Para Jain (1989), a orientação pode ser definida como o ângulo entre o eixo central de menor momento de inércia do objeto e o eixo horizontal do campo de visão. Será utilizado um método descrito em Pazos (2002) que consiste em encontrar o ângulo entre o raio maior do objeto e o eixo horizontal do campo de visão. Este método foi escolhido, pois neste caso, a frente do barco servirá como referência, sendo assim, será possível obter a orientação e o sentido do objeto. Utilizando o método de Jain (1989), seria possível apenas conhecer o eixo de orientação da embarcação.

Para que pudesse ter sido encontrado o ângulo de orientação da embarcação (ψ), primeiramente foi necessário determinar o raio maior do objeto na imagem. O raio maior é o segmento que une o centróide de um objeto ao ponto mais distante da borda deste objeto. Na Figura 2.12, pode-se observar o raio maior da embarcação.

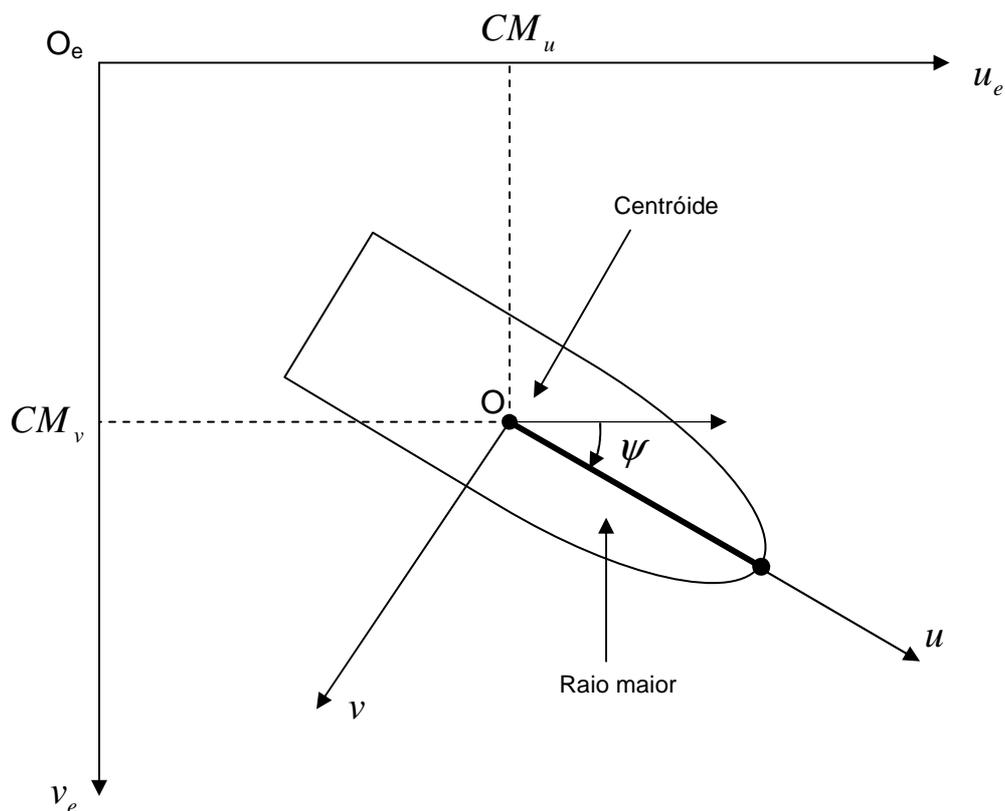


Figura 2.12: Determinação da orientação do objeto

Para o cálculo da orientação, será utilizado o exemplo da Figura 2.8, a qual representa a vista superior da embarcação em uma imagem binária com algum ruído. Considera-se que todas as etapas anteriores já tenham sido concluídas e que, portanto, o objeto de interesse já tenha sido identificado e suas coordenadas de posição já sejam conhecidas.

Primeiramente, deve-se determinar o raio maior do objeto. Conforme já definido, trata-se do vetor que liga o centróide ao ponto mais distante da borda. A borda de um objeto pode ser identificada em regiões onde há alto contraste, ou seja, regiões onde é possível observar grandes variações de intensidade luminosa entre *pixels* vizinhos, como na Figura 2.6. Em imagens binárias, pode-se dizer que um *pixel* de borda é um *pixel* branco conectado a pelo menos um *pixel* preto (Jain 1989). É através deste conceito que serão analisados cada um dos *pixels* da borda da embarcação para que seja encontrado o ponto mais distante do centróide.

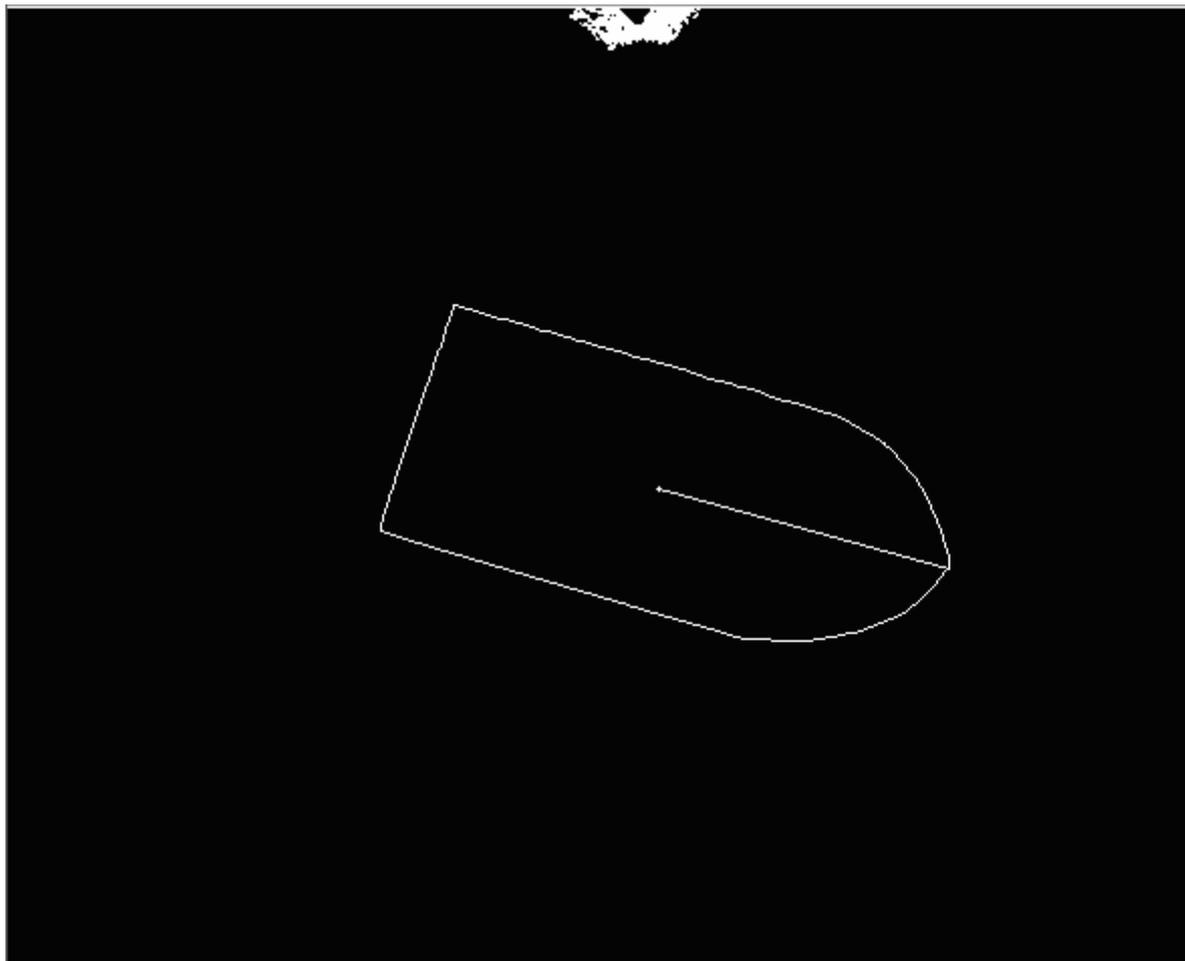


Figura 2.13: Raio maior e borda da embarcação

Na Figura 2.13 observa-se o raio maior da embarcação ligando o centróide ao ponto mais distante da borda. Desta forma, o próximo passo será obter o ângulo entre este raio e o eixo u_e do campo de visão.

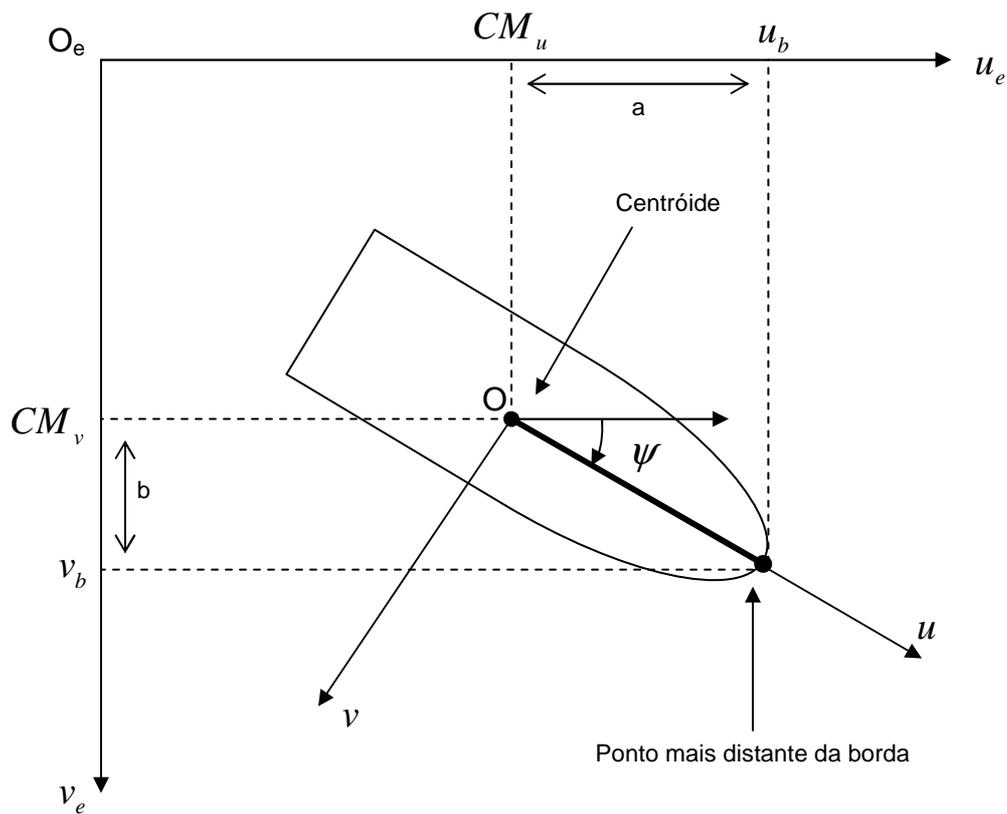


Figura 2.14: Cálculo do ângulo de orientação

A Figura 2.14 apresenta os segmentos que serão utilizados para o cálculo do ângulo de orientação. O cálculo do valor destes segmentos é definido a partir das equações

$$a = u_b - CM_u, \quad (2.4)$$

$$b = v_b - CM_v, \quad (2.5)$$

$$\psi = \cos^{-1} \left(\frac{a}{\sqrt{a^2 + b^2}} \right). \quad (2.6)$$

O ângulo de orientação obtido através da equação (2.6) é definido apenas no primeiro e no segundo quadrantes. Desta forma, verifica-se também o sinal do segmento b para que ψ possa ser definido nos quatro quadrantes: entre -180° e 180° . Para solucionar este problema, também poderia ter sido utilizada a função `atan2` do MATLAB.

A Tabela 2.4 apresenta o trecho de programa responsável pela detecção do ponto mais distante entre o centróide e a borda da embarcação e pelo cálculo do ângulo de orientação.

Tabela 2.4: Programa em MATLAB para a obtenção do raio maior e do ângulo de orientação.

| Programa em MATLAB |
|---|
| <pre> maximo=0; for i=1:m for j=1:n if (j>=2)&&(j<n)&&(i>=2)&&(i<m)&&(M(i,j)==Objeto) if ((M(i-1,j)==0) (M(i+1,j)==0) (M(i,j-1)==0) (M(i,j+1)==0)) modulo=((j-cmu)^2+(i-cmv)^2)^0.5; %Calcula o módulo do vetor if modulo>maximo %Caso o módulo do vetor seja maior que o maior modulo encontrado até o momento. maximo=modulo; %guarda o valor do módulo a=(j-cmu); %componente "Ue" do vetor de maior distancia entre a borda e o centro de massa. b=(i-cmv); %componente "Ve" do vetor de maior distancia entre a borda e o centro de massa. end end end end end Angulo=acos(a/maximo)*(180/pi); %Calcula o angulo do vetor (j-cmu,i-cmv) em graus. if b>0 %Caso o angulo esteja no terceiro ou quarto quadrantes. Orientacao=-Angulo; else Orientacao=Angulo; end </pre> |

Desta forma, foram obtidas todas as informações necessárias para o sistema de controle, ou seja, u_b , v_b e ψ .

2.3 – Escalamento

Neste capítulo, conforme já mencionado, o sistema de coordenadas utilizado é o sistema da câmera, com unidades em *pixels*. O objetivo desta seção é apresentar um método de conversão do sistema de coordenadas da câmera (no espaço de imagem) para o sistema de coordenadas no espaço de trabalho: sistema que será utilizado para a dinâmica da embarcação.

A Figura 2.15 mostra a projeção do objeto no plano de imagem para o espaço de trabalho.

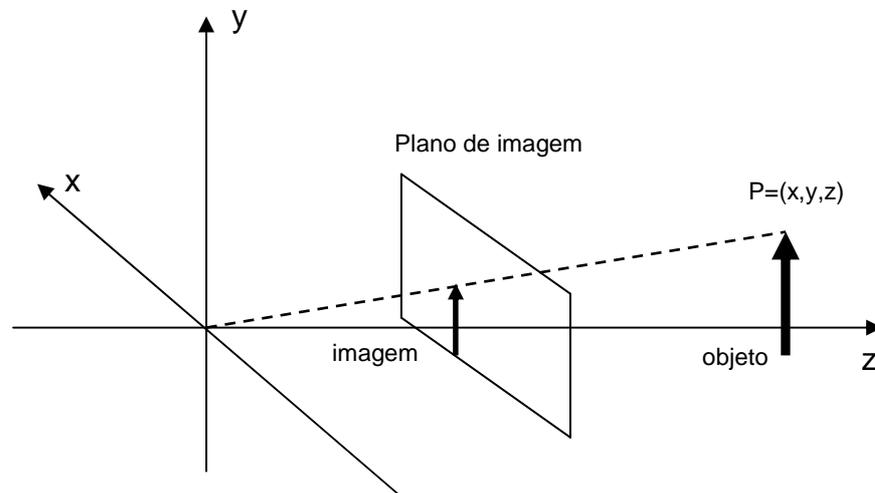


Figura 2.15: Projeção da imagem no espaço de trabalho.

A projeção perspectiva da imagem no espaço de trabalho pode ser modelada linearmente se a profundidade z for mantida constante (Hutchinson, Hager e Corke, 1996). Como neste projeto não será considerado o deslocamento da embarcação no eixo z , considera-se a profundidade constante.

O modelo que relaciona as coordenadas da imagem às coordenadas do objeto é dado por:

$$\begin{pmatrix} x \\ y \end{pmatrix} = s \begin{pmatrix} u \\ v \end{pmatrix}, \quad (2.6)$$

onde s é uma constante de dimensão $m/pixel$, e as variáveis x e y são as coordenadas no plano horizontal do espaço de trabalho.

2.4 – Implementação

Os métodos de processamento de imagem descritos neste Capítulo foram desenvolvidos no *software* MATLAB instalado em um sistema operacional Windows XP. Para a aquisição das imagens, definiu-se no *workspace* do MATLAB um objeto

de vídeo, utilizando o *driver* da *webcam* no sistema operacional, através da função padrão do MATLAB *videoinput*. Este objeto de vídeo foi configurado para que as imagens fossem captadas em *grayscale* (níveis de cinza).

A partir de então, utilizou-se o comando *getsnapshot* para que as imagens fossem captadas e armazenadas em uma variável, a qual foi utilizada pelos algoritmos de processamento de imagem como parâmetro de entrada para a obtenção das informações desejadas. A Figura 2.16 mostra o diagrama de fluxo do processamento de imagem e controle.

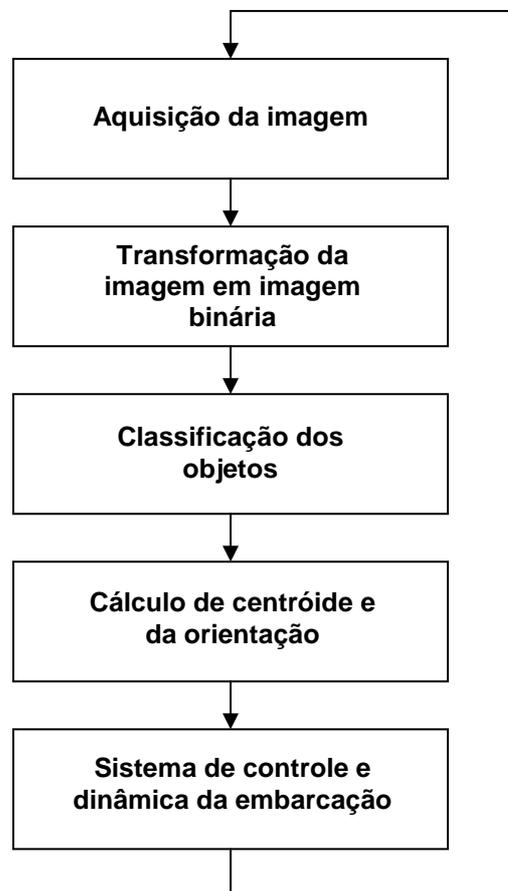


Figura 2.16: Diagrama de fluxo do processamento de imagem e controle.

CAPÍTULO 3 – CONTROLE

O objetivo do sistema de controle desenvolvido neste capítulo é manter um veículo flutuante de pequenas dimensões em posição e orientação determinadas a despeito de perturbações externas.

3.1 – Sistemas de Coordenadas

No Capítulo 2, utilizou-se o sistema de coordenadas da câmera (no espaço de imagem). Para desenvolver a modelagem e o controle da embarcação, este Capítulo adota o sistema de coordenadas no espaço de trabalho apresentado na Figura 3.1.

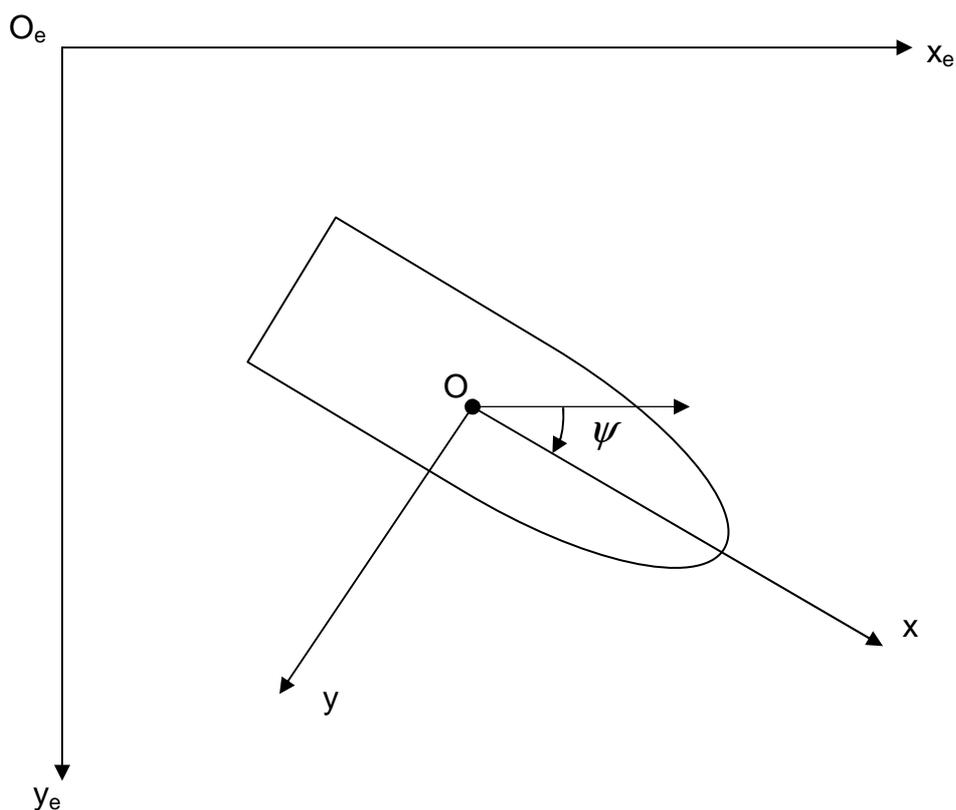


Figura 3.1: Sistemas de coordenadas no espaço de trabalho.

- **Sistema inercial (estacionário):**

O_e – origem do sistema de coordenadas estacionário;

x_e, y_e – eixos que definem o plano horizontal no sistema estacionário;

z_e – eixo vertical do sistema estacionário (profundidade). Este eixo é ortogonal ao plano $x_eO_e y_e$ e não está representado na Figura 3.1

- **Sistema móvel (fixo à embarcação):**

O – origem do sistema de coordenadas móvel;

x, y, z – eixos do sistema móvel. Neste projeto despreza-se o balanço da embarcação, desta forma os eixos z e z_e são paralelos.

A posição da embarcação pode ser definida como as coordenadas inerciais do ponto O (origem do sistema de coordenadas móvel) em relação ao ponto O_e (origem do sistema de coordenadas estacionário). O ângulo de orientação da embarcação (ψ) pode ser definido como a inclinação entre os eixos x e x_e , conforme a Figura 3.1.

3.2 – Modelagem da Dinâmica do Sistema

A modelagem de um sistema dinâmico consiste na elaboração de um conjunto de equações que definem o seu comportamento. Um sistema físico pode ser representado por diversos modelos matemáticos, os quais, usualmente, são definidos através de equações diferenciais.

A precisão dos modelos de sistemas dinâmicos pode ser aumentada à medida que é aumentada a complexidade dos mesmos. Modelos com um grande número de equações podem ser criados para reproduzir de maneira exata o comportamento de um sistema (Ogata 2003).

Nesta etapa do projeto, o objetivo foi desenvolver um modelo adequado ao desenvolvimento do controle. Na verdade, como é utilizado o controle a estrutura

variável, o qual é apresentado adiante (Seção 3.3), não foi necessário o desenvolvimento de um modelo complexo.

Tendo em vista que o objetivo do sistema de posicionamento dinâmico é manter a embarcação parada, esta apenas se deslocará em baixa velocidade de modo a corrigir o erro de posicionamento. Sendo assim, desconsidera-se o efeito do arraste no modelo da dinâmica. Distúrbios externos e parâmetros hidrodinâmicos da embarcação também foram desconsiderados na formulação do modelo.

3.2.1 – Equações de Movimento

As equações descritas nesta subseção têm o objetivo de relacionar de maneira simplificada as entradas do sistema controlado às suas saídas. As entradas foram definidas como $F_p \in \mathbb{R}^2$ e $M_p \in \mathbb{R}$, sendo $F_p = [F_{px} \ F_{py}]^T$ a força produzida pelos propulsores no plano horizontal e M_p o momento produzido pelos propulsores no eixo vertical da embarcação. As saídas são as coordenadas de posição e orientação da embarcação.

A equação inercial do sistema é

$$\begin{bmatrix} \dot{U} \\ \dot{\omega} \end{bmatrix} = I_{ma}^{-1} \begin{bmatrix} F_p \\ M_p \end{bmatrix}, \quad (3.1)$$

onde, $\dot{U} = [\dot{U}_x \ \dot{U}_y]^T$ é a aceleração linear do veículo, e $\dot{\omega}$ é a aceleração angular. A matriz I_{ma} é a matriz de inércias do sistema, dada por:

$$I_{ma} = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & J \end{bmatrix}, \quad (3.2)$$

onde

$$m_x = m + A_{11};$$

$$m_y = m + A_{22};$$

m – massa da embarcação.

Na matriz de inércias, m_x e m_y representam as massas vistas pelas entradas F_{px} e F_{py} respectivamente, e J é o momento de inércia total do sistema. As massas adicionais A_{11} e A_{22} representam forças inerciais induzidas pela pressão resultante do movimento harmônico da embarcação na água (Fossen, 2002). Define-se massa adicional de maneira simplificada como a massa de água deslocada pelo movimento da embarcação. Devido à geometria do veículo, pode-se concluir que as massas adicionais em cada grau de liberdade são distintas.

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

$$\begin{bmatrix} \dot{U}_x \\ \dot{U}_y \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 1/m_x & 0 & 0 \\ 0 & 1/m_y & 0 \\ 0 & 0 & 1/J \end{bmatrix} \begin{bmatrix} F_{px} \\ F_{py} \\ M_p \end{bmatrix} \quad (3.3)$$

Como a velocidade e a aceleração do modelo estão representadas no sistema de coordenadas móvel e as coordenadas de posição da embarcação estão definidas no sistema inercial, foi necessário utilizar a matriz de rotação

$$T(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.4)$$

para converter as coordenadas para o sistema inercial. Desta forma, tem-se

$$\begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} = T \begin{bmatrix} \dot{U} \\ \dot{\omega} \end{bmatrix}, \quad (3.5)$$

onde $\dot{V} = [\dot{V}_x \ \dot{V}_y]^T$ é a aceleração linear da embarcação no sistema de coordenadas estacionário. Substituindo (3.3) e (3.4) em (3.5) obtém-se:

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{px}/m_x \\ F_{py}/m_y \\ M_p/J \end{bmatrix}. \quad (3.6)$$

Da multiplicação das matrizes na equação (3.6) resultam:

$$\dot{V}_x = \left(F_{px} / m_x \right) \cos \psi - \left(F_{py} / m_y \right) \sin \psi, \quad (3.7)$$

$$\dot{V}_y = \left(F_{px} / m_x \right) \sin \psi + \left(F_{py} / m_y \right) \cos \psi, \quad (3.8)$$

$$\dot{\omega} = \frac{M_p}{J}. \quad (3.9)$$

Desta forma, o modelo da dinâmica do sistema é definido através das equações (3.7) a (3.9). A Figura 3.2 apresenta um diagrama em blocos simplificado da dinâmica do sistema. Nele, $P_e = [x_e \quad y_e]^T$ representa a posição da embarcação no sistema de coordenadas inercial.

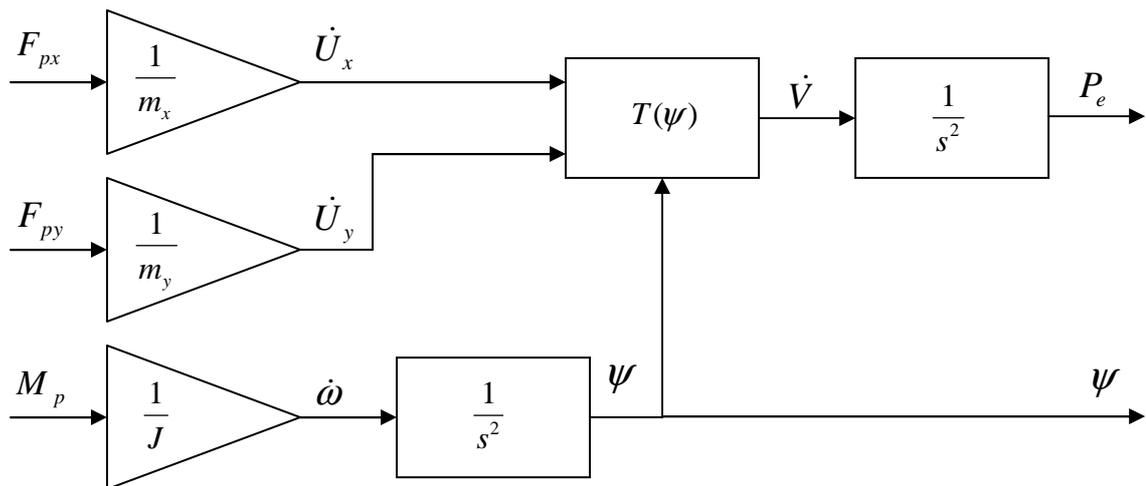


Figura 3.2: Diagrama em blocos do modelo simplificado da dinâmica do sistema.

3.2.2 – Modelo dos propulsores

Na formulação do modelo da dinâmica do sistema não foi considerado o efeito dos propulsores. A equação

$$F_{pi} = \begin{cases} F_{p\max}, & \text{se } u_{pi} > 0 \\ 0, & \text{se } u_{pi} = 0, \\ -F_{p\max}, & \text{se } u_{pi} < 0 \end{cases} \quad (3.10)$$

representa o modelo dos propulsores utilizado para o projeto, onde $i = 1, 2, 3$ é o número do propulsor, $F_{p\max}$ é a força nominal gerada pelo propulsor em regime permanente e u_{pi} é o sinal de comando.

Observa-se na equação (3.10) que os propulsores operam em modo de chaveamento (direto-parado-reverso). Este modo foi escolhido devido à maior facilidade para a implementação do *hardware* de controle, o qual é descrito no Apêndice A.

A ação conjunta dos propulsores é dada por:

$$F_p = \sum_{i=1}^n F_{pi} P_{pi} \quad (3.11.a)$$

$$M_p = \sum_{i=1}^n R_{pi} \times (F_{pi} P_{pi}) \quad (3.11.b)$$

onde:

n – número de propulsores ($n = 3$);

P_{pi} – vetor unitário (representado no sistema móvel) que define a direção da força gerada pelo i -ésimo propulsor;

R_{pi} – centro de empuxo do i -ésimo hélice (representado no sistema móvel).

Neste modelo, o momento axial produzido pelos propulsores não foi considerado.

3.3 – Controle a Estrutura Variável

Devido às dificuldades de identificação dos parâmetros do sistema e ao fato de que os propulsores deverão operar em modo de chaveamento, conforme o modelo descrito na Seção 3.2.2, a utilização de controladores lineares com ganhos fixos como os de ação proporcional e derivativa (PDs) e os de ação proporcional, integral e derivativa (PIDs) é inadequada a este projeto. Desta forma, foi escolhido o controle a estrutura variável (VSC). De acordo com Cunha (1992), o VSC em modo

de deslizamento proporciona um alto grau de robustez quando aplicado a sistemas com incertezas paramétricas e uma grande capacidade de rejeitar perturbações. O mesmo pode ser aplicado a diversas classes de sistemas dinâmicos, tais como: lineares, não-lineares, variantes no tempo, sujeitos a perturbações, monovariáveis e multivariáveis. Segundo Utkin, Guldner e Shi (1999), o modo de deslizamento acontece quando o sinal de controle, como função dos estados do sistema, é chaveado em alta frequência.

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 de referência. 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. Esta propriedade é conhecida como "invariância", e caracteriza a robustez de um VSC em relação às incertezas paramétricas (Cunha, 1992).

Segundo Cunha (1992), uma das desvantagens deste tipo de controle é que o esforço é sempre muito exigente e independe da grandeza da perturbação a ser rejeitada.

Nesta seção, apresenta-se um VSC para o controle do posicionamento da embarcação.

3.3.1 – Controle da Embarcação

Para o controle da embarcação, considerou-se o desenvolvimento de um VSC para a dinâmica de segunda ordem de cada grau de liberdade. A superfície de deslizamento escolhida é definida por

$$s = \lambda e + \dot{e}, \quad s = 0, \quad (3.12)$$

onde

$$e = x_{ref} - x \quad (3.13)$$

é o erro de posicionamento e $\dot{e} = -\dot{x}$.

Após o início do modo deslizante em algum $t = t_s$ a solução da equação diferencial (3.12) é dada por

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

na qual percebe-se que o decaimento do erro $e(t)$ ocorre exponencialmente e sua taxa de decaimento depende essencialmente da constante $\lambda > 0$.

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

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

onde

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

A Figura 3.3 apresenta o plano de fase da dinâmica de um grau de liberdade controlado pela lei (3.15).

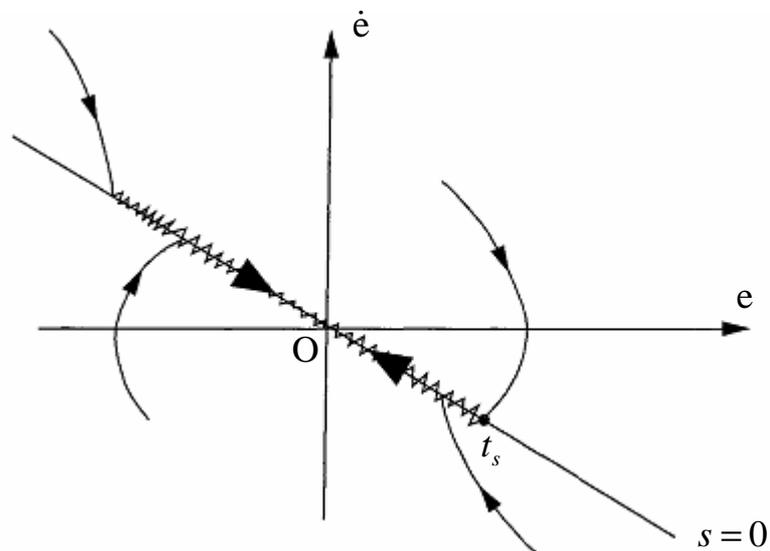


Figura 3.3: Plano de fase da dinâmica de um grau de liberdade (extraído de Utkin, Guldner e Shi, 1999).

Percebe-se no plano de fase apresentado na Figura 3.3 que a partir de t_s , quando os estados alcançam a superfície de deslizamento, os sinais de comando

são chaveados ao longo da superfície. O plano de fase apresentado na Figura 3.3 não considera a saturação dos propulsores, a qual impediria a condição de deslizamento em toda a linha de chaveamento $s = 0$.

O controle do sistema de posicionamento dinâmico é realizado em tempo discreto, com frequência de 30Hz, que é a frequência de captura da câmera. Desta forma, a variável de estado \dot{e} é estimada de acordo com a equação

$$\dot{e}(n) = \frac{e(n) - e(n-1)}{T}, \quad (3.16)$$

onde $T = 1/30 \approx 33ms$ é o período de amostragem e $n \in \mathbb{Z}$ é o tempo discreto.

3.4 – Transformação de coordenadas

Considerando que os sinais de comando dos controladores definidos para cada grau de liberdade são representados no sistema de coordenadas móvel, e que a posição da embarcação é medida no sistema de coordenadas inercial, faz-se necessária a realização de uma transformação de coordenadas para a obtenção destes sinais. Na prática, a conversão será feita sobre o erro de posicionamento (\bar{e}) e sobre a derivada do erro em função do tempo ($\dot{\bar{e}}$), estratégia descrita em Cunha (1992), os quais são considerados já convertidos para o sistema de coordenadas móvel durante a descrição do controle da embarcação na Seção 3.3.1.

A transformação é feita utilizando-se a matriz inversa da matriz T definida em (3.4), resultando em

$$T^{-1}(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.17)$$

A equação que define a conversão de coordenadas é dada por:

$$E = T^{-1}(\psi)\bar{E}, \quad (3.18)$$

onde $E = \begin{bmatrix} e_x & e_y & e_\psi \end{bmatrix}^T$ é a matriz dos erros de posicionamento de cada grau de liberdade no sistema de coordenadas móvel e $\bar{E} = \begin{bmatrix} \bar{e}_x & \bar{e}_y & \bar{e}_\psi \end{bmatrix}^T$ é a matriz de erros no sistema inercial. Desta forma tem-se:

$$\begin{bmatrix} e_x \\ e_y \\ e_\psi \end{bmatrix} = T^{-1}(\psi) \begin{bmatrix} \bar{e}_x \\ \bar{e}_y \\ \bar{e}_\psi \end{bmatrix}, \quad (3.19.a)$$

$$\begin{bmatrix} e_x \\ e_y \\ e_\psi \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{e}_x \\ \bar{e}_y \\ \bar{e}_\psi \end{bmatrix}. \quad (3.19.b)$$

Da multiplicação das matrizes em (3.19.b) resultam as equações

$$e_x = \cos(\psi)\bar{e}_x + \sin(\psi)\bar{e}_y, \quad (3.20)$$

$$e_y = -\sin(\psi)\bar{e}_x + \cos(\psi)\bar{e}_y, \quad (3.21)$$

$$e_\psi = \bar{e}_\psi. \quad (3.22)$$

Desta forma, são obtidos os erros de posicionamento no sistema de coordenadas móvel através das equações (3.20) a (3.22). A conversão de $\dot{\bar{E}} = \begin{bmatrix} \dot{\bar{e}}_x & \dot{\bar{e}}_y & \dot{\bar{e}}_\psi \end{bmatrix}^T$ é feita de maneira análoga e seus cálculos foram suprimidos.

3.5 – Desacoplamento

É possível perceber ao longo deste capítulo que a dinâmica da embarcação é multivariável e acoplada. Devido à disposição definida para os propulsores na embarcação, a qual pode ser observada na Figura 3.4, o ângulo de rumo (ψ) e o deslocamento lateral do veículo (grau de liberdade y) serão controlados pelos mesmos propulsores (p_2 e p_3). O propulsor p_1 ficará disposto de maneira a controlar apenas o deslocamento da embarcação na coordenada x .

Nesta Seção, o objetivo é descrever um método de desacoplamento para os sinais de comando dos propulsores.

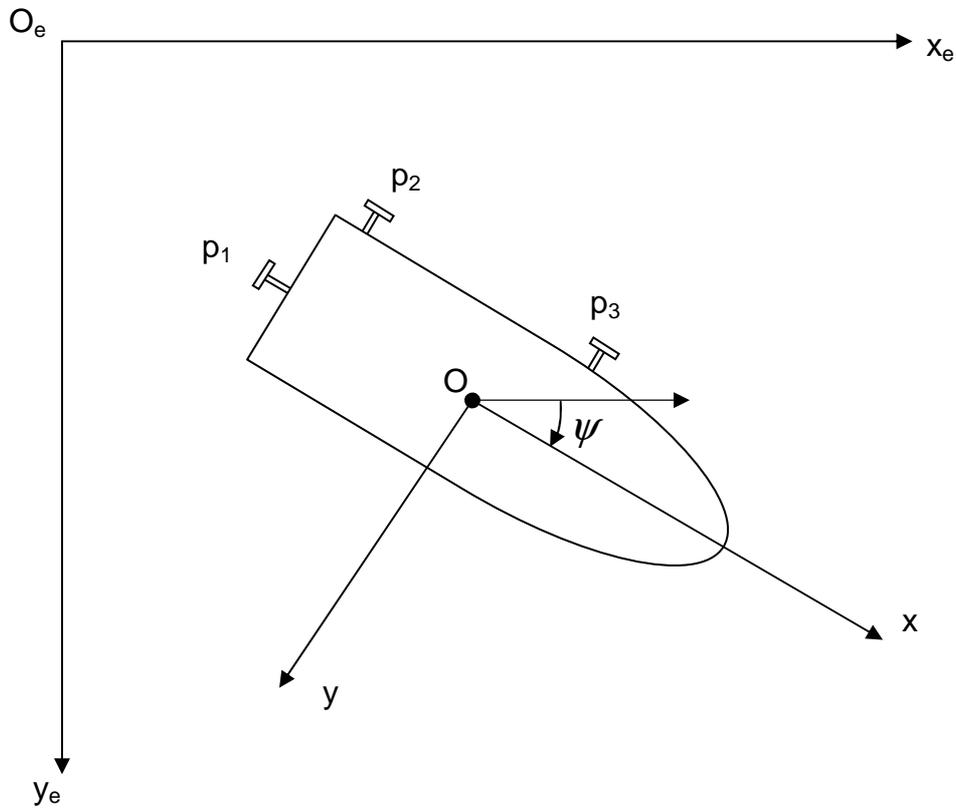


Figura 3.4: Disposição dos propulsores na embarcação.

A Figura 3.5 apresenta um diagrama em blocos simplificado do sistema de controle, destacando a conexão do desacoplador aos sinais de comando no sistema móvel e aos sinais de comando dos propulsores, onde

$$u_r = \text{sgn}(\lambda e_r + \dot{e}_r), \quad r = x, y, \psi. \quad (3.23)$$

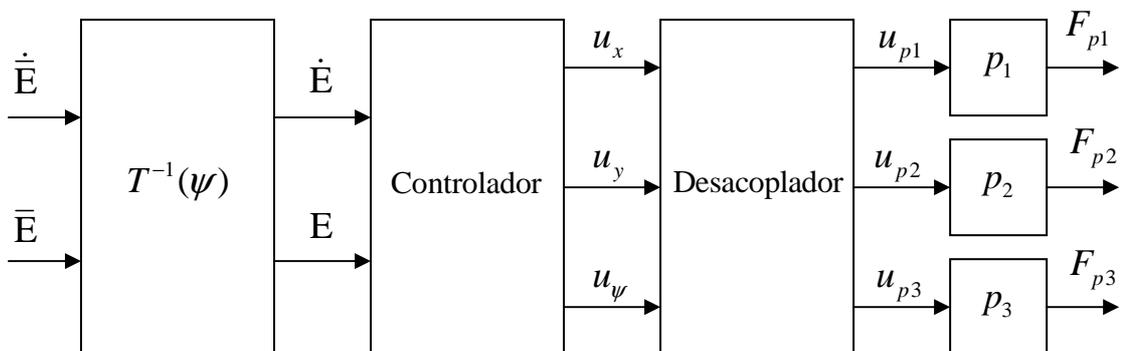


Figura 3.5: Diagrama em blocos simplificado do sistema de controle.

O método de desacoplamento a ser descrito consiste em obter os sinais de comando dos propulsores a partir dos sinais de comando dos controladores de cada grau de liberdade.

Observando a Figura 3.4 percebe-se que para que o veículo se desloque ao longo da coordenada y é necessário que os propulsores p_2 e p_3 sejam acionados no mesmo sentido de rotação, ou seja, $u_{p2} = u_{p3}$. Isso porque considera-se que os dois propulsores produzam momentos de mesmo módulo no eixo z , sendo assim, haverá apenas a força resultante que gera o movimento de translação na coordenada y . De maneira análoga, caso seja necessário realizar um movimento de rotação em torno do eixo z (grau de liberdade ψ), os propulsores p_2 e p_3 deverão ser acionados em sentidos de rotação opostos $u_{p2} = -u_{p3}$. Desta forma haverá apenas torque no eixo z , pois o somatório das forças geradas pelos propulsores será 0, sem que haja movimento de translação.

Considerando que a embarcação deverá se deslocar simultaneamente nos graus de liberdade y e ψ , foi necessário desenvolver uma lógica de acionamento para os propulsores p_2 e p_3 que possibilitasse a realização deste deslocamento simultâneo. A lógica desenvolvida consiste basicamente em verificar se os controladores dos graus de liberdade y e ψ tentam acionar um mesmo propulsor em sentidos de rotação contrários. Caso tal fato ocorra, este propulsor terá sinal de comando 0, ou seja, ficará desligado.

A equação

$$U_p = \text{sgn} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_\psi \end{bmatrix} \right) \quad (3.24)$$

define os sinais de comando dos propulsores, onde $U_p = [u_{p1} \quad u_{p2} \quad u_{p3}]^T$.

Multiplicando as matrizes em (3.24) obtém-se:

$$u_{p1} = \text{sgn}(u_x), \quad (3.25)$$

$$u_{p2} = \text{sgn}(u_y + u_\psi), \quad (3.26)$$

$$u_{p3} = \text{sgn}(u_y - u_\psi). \quad (3.27)$$

Desta forma ficam definidos os sinais de comando dos propulsores através das equações (3.25) a (3.27).

A Figura 3.6 define o diagrama completo do sistema de controle após o processo de desacoplamento.

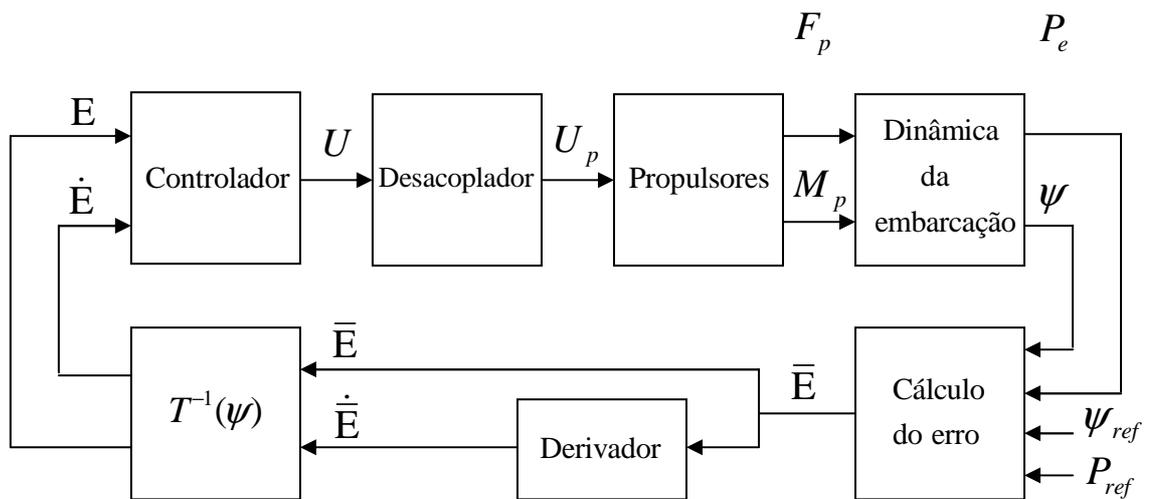


Figura 3.6: Diagrama completo do sistema.

3.6 – Implementação

Assim como o processamento de imagem, o controle também foi desenvolvido no MATLAB. O código comentado da rotina de controle utilizada encontra-se no Apêndice B.

CAPÍTULO 4 – RESULTADOS EXPERIMENTAIS

Os dados descritos neste Capítulo apresentam os resultados de testes realizados com o sistema de posicionamento dinâmico desenvolvido. A constante λ da superfície de deslizamento definida na equação (3.12) foi escolhida experimentalmente. Utilizou-se $\lambda = 2,5rad / s$. Observou-se que quando o valor de λ era muito menor que o valor escolhido, o decaimento do erro ocorria de forma muito lenta. Por outro lado, com λ muito maior que $2,5rad / s$ a embarcação oscilava com maior amplitude, de maneira semelhante à oscilação gerada por um sistema de controle apenas com ação proporcional. A tensão de alimentação dos motores da embarcação também foi escolhida experimentalmente de maneira a obter um movimento mais suave da embarcação. Utilizou-se o valor de +5,7V.

As Figuras 4.1 e 4.2 apresentam o resultado do experimento no grau de liberdade ψ . O referencial adotado foi $\psi_{ref} = 0^\circ$.

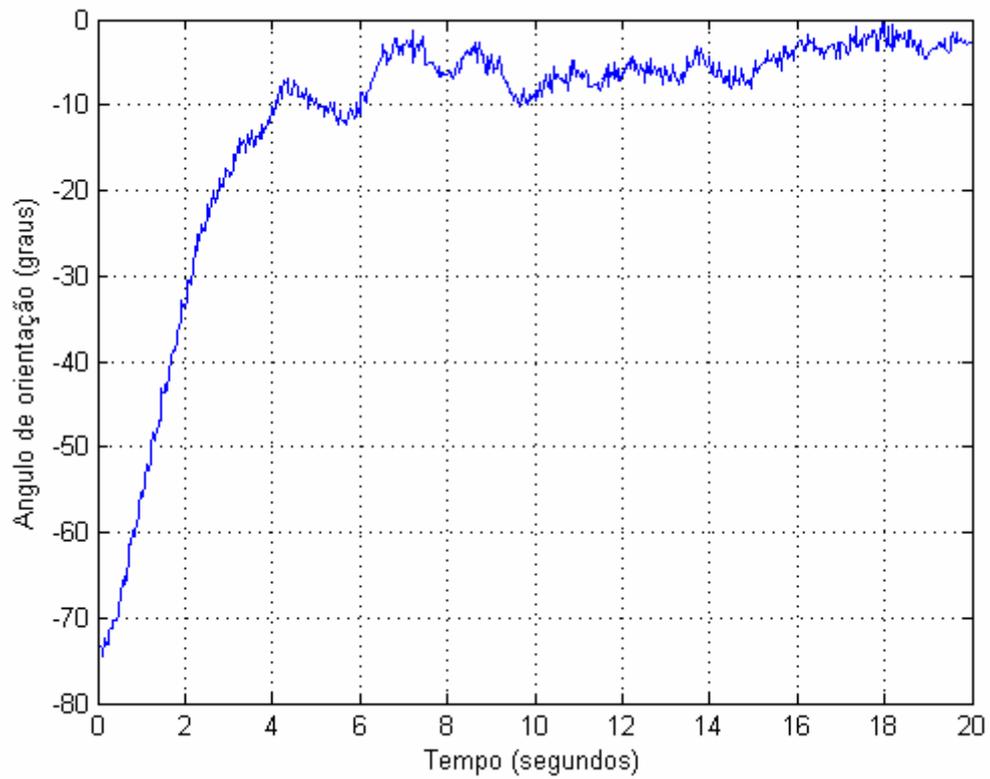


Figura 4.1: Oscilação da embarcação no grau de liberdade ψ .

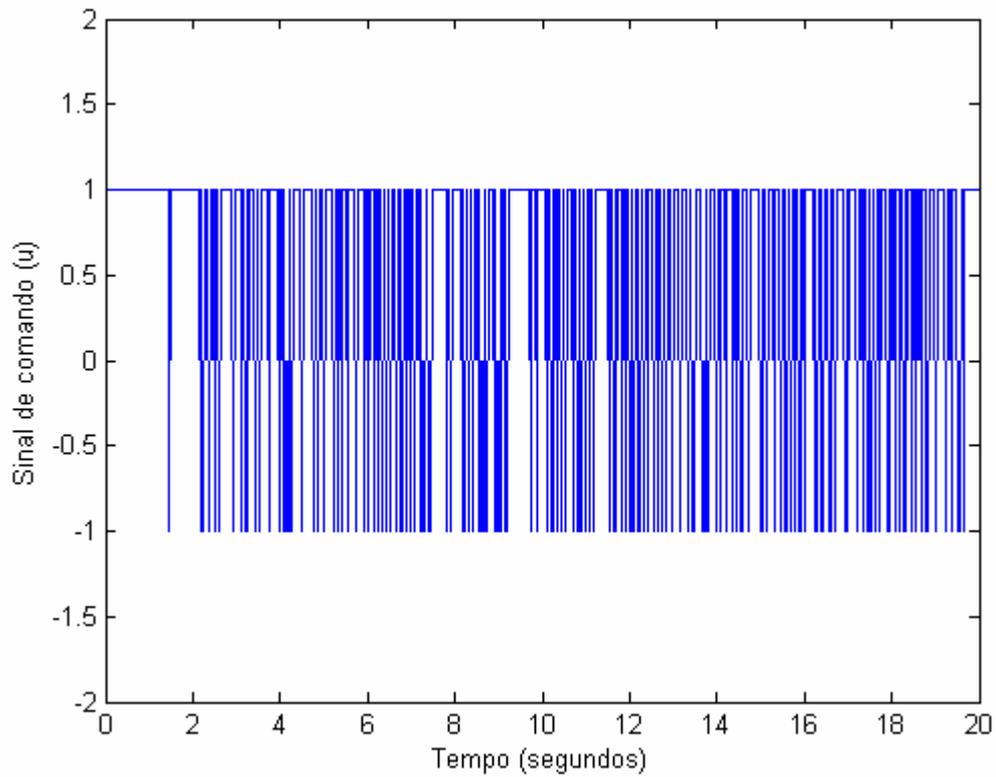


Figura 4.2: Sinal de comando u_ψ em função do tempo.

As Figuras 4.3 e 4.4 mostram o movimento da embarcação na coordenada x_e e o sinal de comando u_x em função do tempo. Utilizou-se $x_{ref} = 0,179m$.

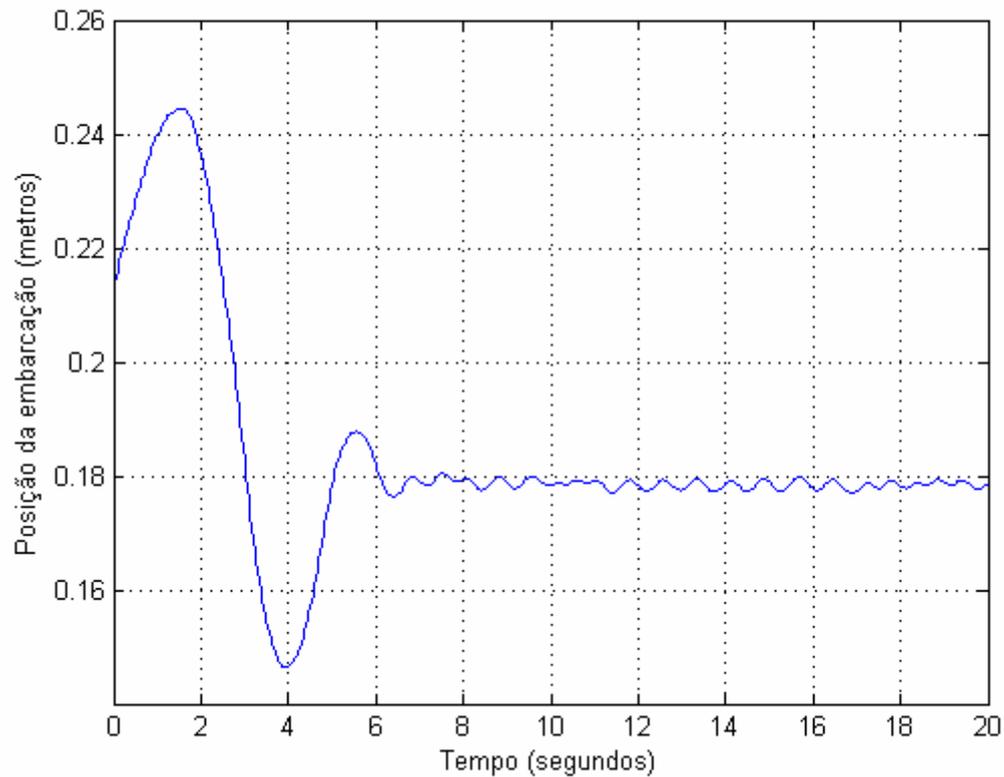


Figura 4.3: Oscilação da embarcação no grau de liberdade x_e .

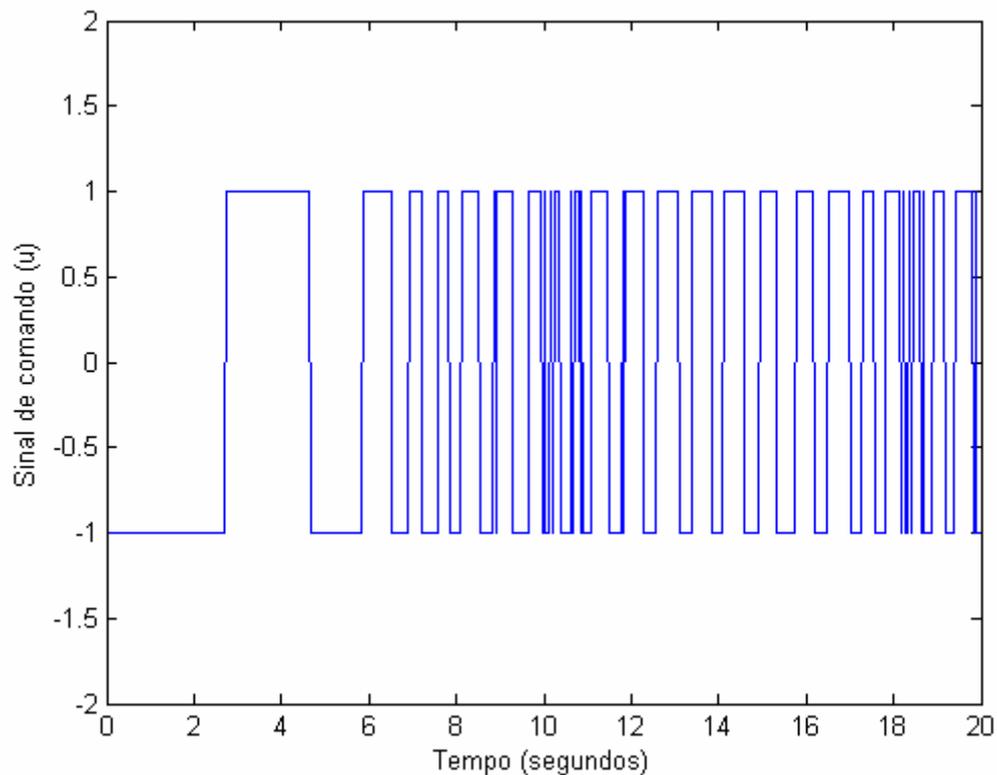


Figura 4.4: Sinal de comando u_x em função do tempo.

As Figuras 4.5 e 4.6 mostram o movimento da embarcação na coordenada y_e e o sinal de comando u_y em função do tempo com $y_{ref} = 0,128m$.

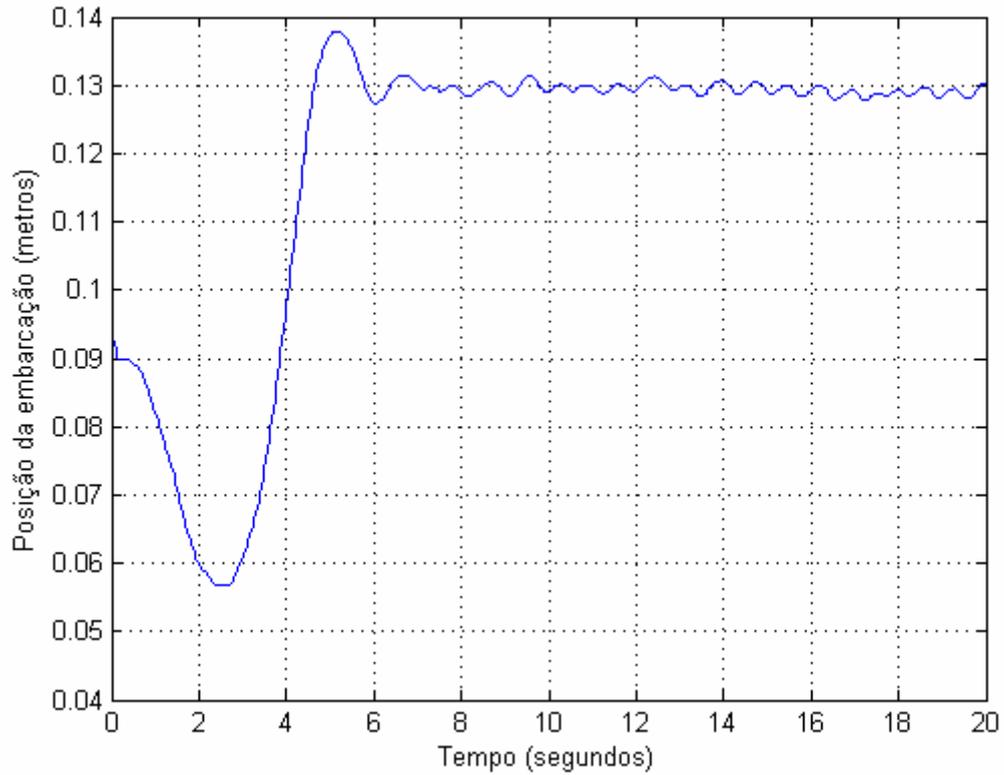


Figura 4.5: Oscilação da embarcação no grau de liberdade y_e .

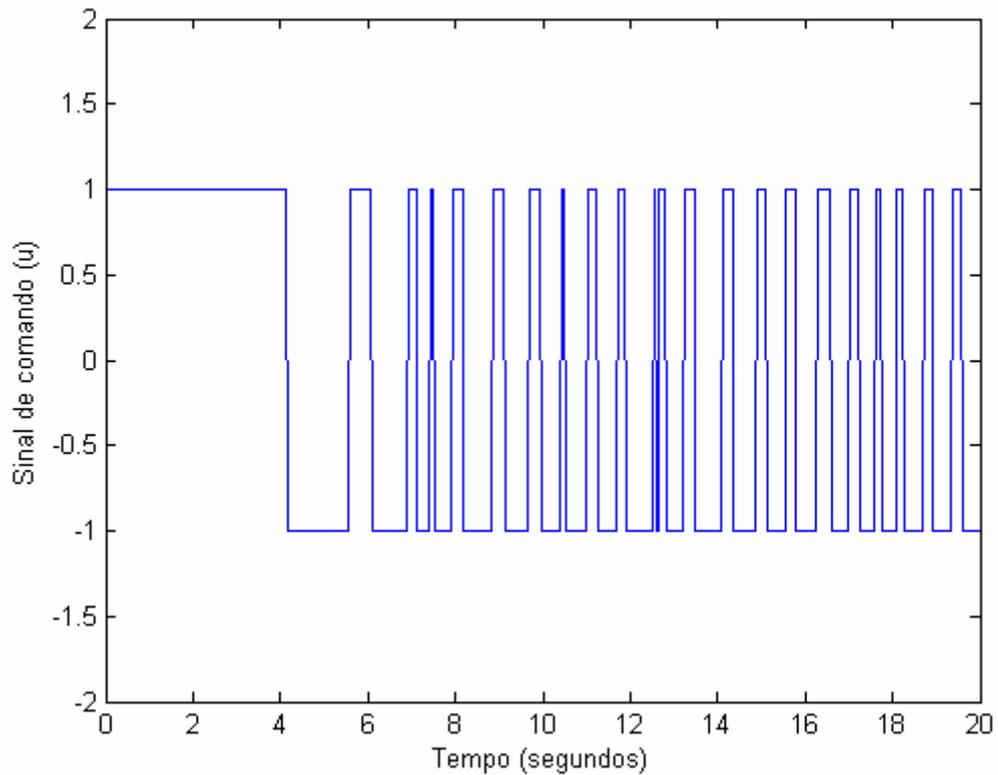


Figura 4.6: Sinal de comando u_y em função do tempo.

Observando os gráficos nas Figuras 4.1 a 4.6 é possível perceber a ocorrência de oscilações em torno dos referenciais estabelecidos para os graus de liberdade (superfície de deslizamento). Este efeito ocorre devido a imperfeições nos propulsores e no circuito de acionamento como atrasos e histerese, e denomina-se *chattering*. Neste projeto, como é utilizado o processamento de imagem em tempo real para a realimentação do sistema de controle, consideram-se também as imperfeições da imagem como possíveis causadoras do *chattering*. As imperfeições podem ser decorrentes de má iluminação, ou qualquer tipo de interferência visual que possa afetar o processamento de imagem. Um outro fator a ser considerado é a presença de perturbações externas, as quais foram desconsideradas durante a modelagem do sistema de controle. Foi possível perceber, ao longo do experimento, a formação de ondas a partir dos movimentos dos propulsores que possivelmente atrapalharam o desempenho do sistema de controle.

As Figuras 4.7 e 4.8 definem a trajetória da embarcação no plano horizontal.

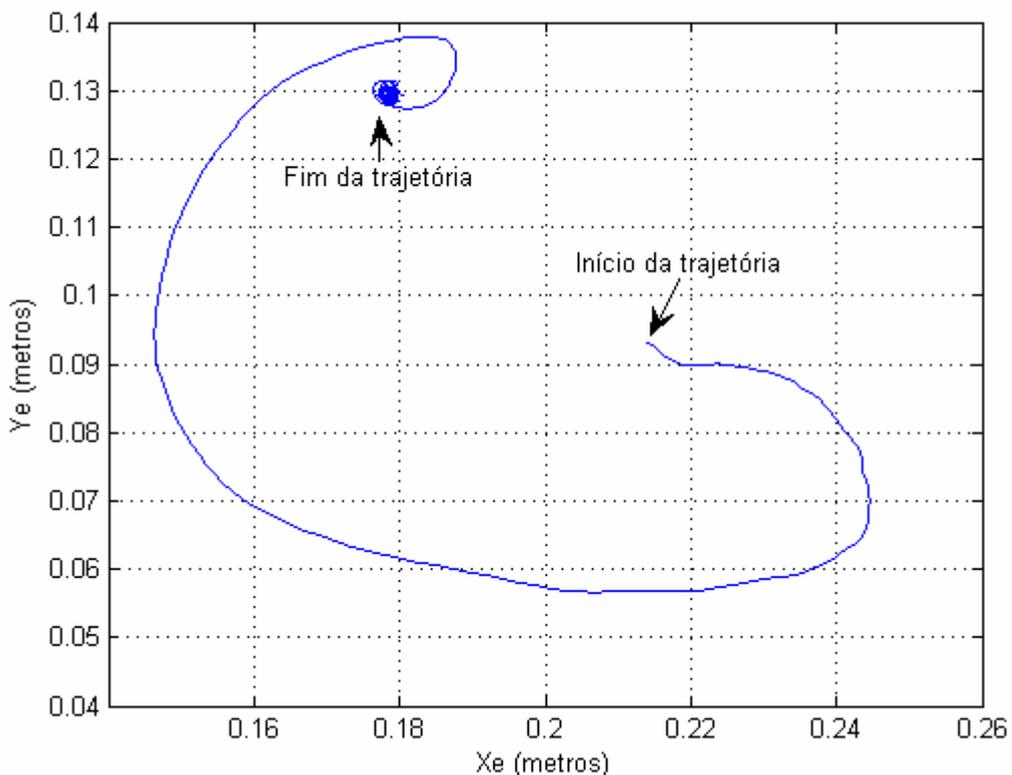


Figura 4.7: Trajetória da embarcação no plano horizontal.

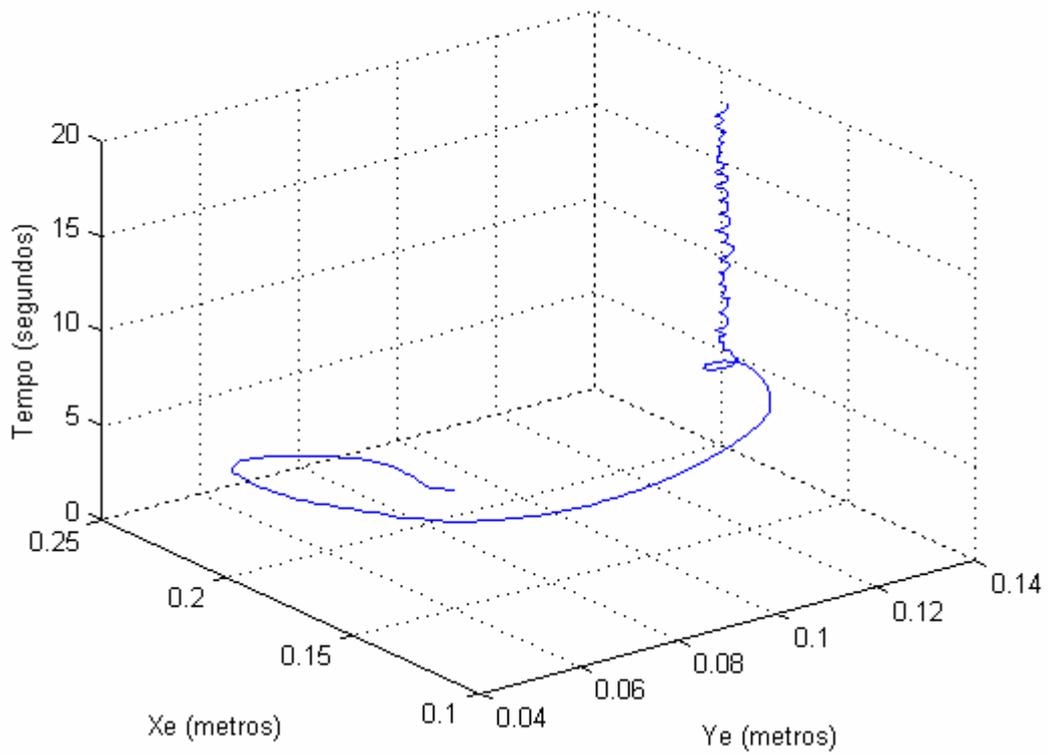


Figura 4.8: Trajetória da embarcação no plano horizontal em função do tempo

Para a conversão das unidades do sistema de coordenadas da câmera para o sistema de coordenadas no espaço de trabalho utilizou-se o fator de escala $s = 1,12 \cdot 10^{-3} \text{ m/pixel}$. Este fator foi obtido através da razão entre as dimensões reais do objeto (em metros) e as dimensões da embarcação na imagem (em *pixels*).

CAPÍTULO 5 – CONCLUSÕES

Durante o desenvolvimento do sistema de posicionamento dinâmico proposto, foi construído o veículo flutuante de pequenas dimensões (embarcação) utilizando-se uma base flutuante com três propulsores instalados conforme planejado inicialmente. A utilização dos três propulsores e a configuração escolhida para estes na embarcação se mostrou eficaz quanto à mobilidade dos graus de liberdade.

Os algoritmos de processamento de imagem desenvolvidos para a obtenção das informações de interesse se mostraram eficientes quanto à precisão das informações, o que pôde ser concluído através dos resultados experimentais. A velocidade de execução dos algoritmos também foi satisfatória, uma vez que foi possível operar à frequência máxima de captura da câmera (30Hz).

O controle a estrutura variável desenvolvido para a aplicação se mostrou eficiente e robusto. Entretanto, foi possível observar o fenômeno do *chattering*. Este fenômeno foi relatado durante a descrição dos dados experimentais e deve-se em parte à frequência de operação do sistema, que não é elevada, ao atraso no circuito de acionamento e à saturação dos propulsores. Foram considerados ainda os efeitos de interferências visuais.

Para solucionar o problema do *chattering*, sugere-se como aprimoramento futuro para o sistema de posicionamento dinâmico desenvolvido o controle linear da velocidade dos propulsores através de modulação PWM (*Pulse Width Modulation*) ou modulação por largura de pulso, o qual foi considerado no início deste projeto, porém não foi implementado. O circuito de acionamento dos motores foi projetado prevendo-se a possível utilização da modulação PWM.

REFERÊNCIAS

- Cunha, José 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.
- Fäy, Hubert (1990), *Dynamic Positioning Systems - Principles desing and applications*. Éditions Technip, Paris.
- Fossen, T. I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs, and Underwater Vehicles*, Marine Cybernetics, Trondheim, Norway.
- Hutchinson, Seth; D. Hager, Gregory; I. Corke, Peter (1996), "A tutorial on vosomal servo control". m. IEEE. Transactions on Robotics on Automation, vol.12, n^o.5, pp.651-655.
- Jain, Anil K. (1986), *Fundamentals of Digital Image Processing*. Prentice Hall, University of Califórnia, Davis.
- Ogata, Katsuhiko (2003), *Engenharia de Controle Moderno – 4ª Edição*. Pearson Prentice Hall.
- Pazos, Fernando (2002), *Automação de Sistemas & Robótica*. Axcel Books, Rio de Janeiro.
- Tannuri, Eduardo Aoun, (2002), *Desenvolvimento de Metodologia de Projeto de Sistema de Posicionamento Dinâmico Aplicado a Operações em Alto Mar*, Tese de Doutorado em Engenharia, Escola Politécnica, USP, São Paulo.
- Rashid, Muhammad H. (2001), *Power Electronics Handbook*. Academic Press, University of Florida/University of West Florida Joint Program and Computer Engineering, Pensacola.
- Kehtarnavaz, Nasser e Gamadia, Mark (2006), *Real-Time Image and Video Processing: From Research to Reality*. Morgan & Claypool Publishers, University of Texas at Dallas, USA.
- Utkin, Vadim; Guldner, Jürgen e Shi, Jingxin (1999), *Sliding Mode Control on Eletromechanical Systems*. Taylor & Francis.
- HDTV-Source (2008), "HDTV reviews & HDTV buyers guide". Página consultada em 16 de Fevereiro de 2008, <<http://www.hdtv-source.com/>>.
- Equatorialoil (s. d.), "Oil production". Página consultada em 16 de Fevereiro de 2008, <http://www.equatorialoil.com/pages/Oil_production.html>.

APÊNDICE A – EMBARCAÇÃO E CIRCUITO DE ACIONAMENTO

Neste Apêndice são descritos o veículo flutuante de pequenas dimensões (embarcação) construído e o circuito de acionamento dos motores.

A.1 – Embarcação

A embarcação construída para o sistema de posicionamento dinâmico possui três propulsores posicionados de maneira a possibilitar o controle independente do movimento em todos os graus de liberdade. A Figura A.1 apresenta a embarcação construída.



Figura A.1: Vista geral da embarcação.

A embarcação consiste basicamente em uma base de madeira onde encontram-se instalados os três propulsores: dois deles posicionados lateralmente de maneira a controlar o movimento lateral da embarcação e o ângulo de rumo, e um deles posicionado na popa (parte traseira) que será responsável pelos

movimentos para frente e para trás. Embaixo da base de madeira está fixada uma placa de isopor que funciona como flutuador.

Percebe-se na Figura A.1 que a embarcação possui uma cobertura branca. Esta cobertura é uma placa de isopor que tem o objetivo de facilitar o processamento de imagem: método que foi abordado no Capítulo 2. A princípio, a utilização desta placa cobrindo os propulsores não estava prevista, entretanto, após a realização de alguns testes, observou-se que os motores atrapalhavam consideravelmente o resultado do processamento de imagem, pois projetavam sombras na superfície do veículo, atuando assim como uma interferência visual. A Figura A.2 mostra a embarcação sem a cobertura de isopor onde pode-se observar a disposição dos motores.



Figura A.2: Vista superior da embarcação sem a cobertura.

Antes da construção do veículo apresentado na Figura A.1, foi utilizada apenas uma placa de isopor como embarcação onde foram fixados os três propulsores com a mesma configuração descrita. Porém, como a placa de isopor possuía pouca inércia, os movimentos dos propulsores geravam oscilações bruscas,

o que motivou a construção de uma nova embarcação com maior inércia devido à base de madeira.

A.2 – Propulsores

Os propulsores utilizados na embarcação são motores elétricos de corrente contínua. A eles são acoplados eixos com hélices conforme observa-se na Figura A.3. Vale mencionar que para a escolha do modelo dos motores e da disposição destes no veículo não houve qualquer preocupação com o consumo de energia.

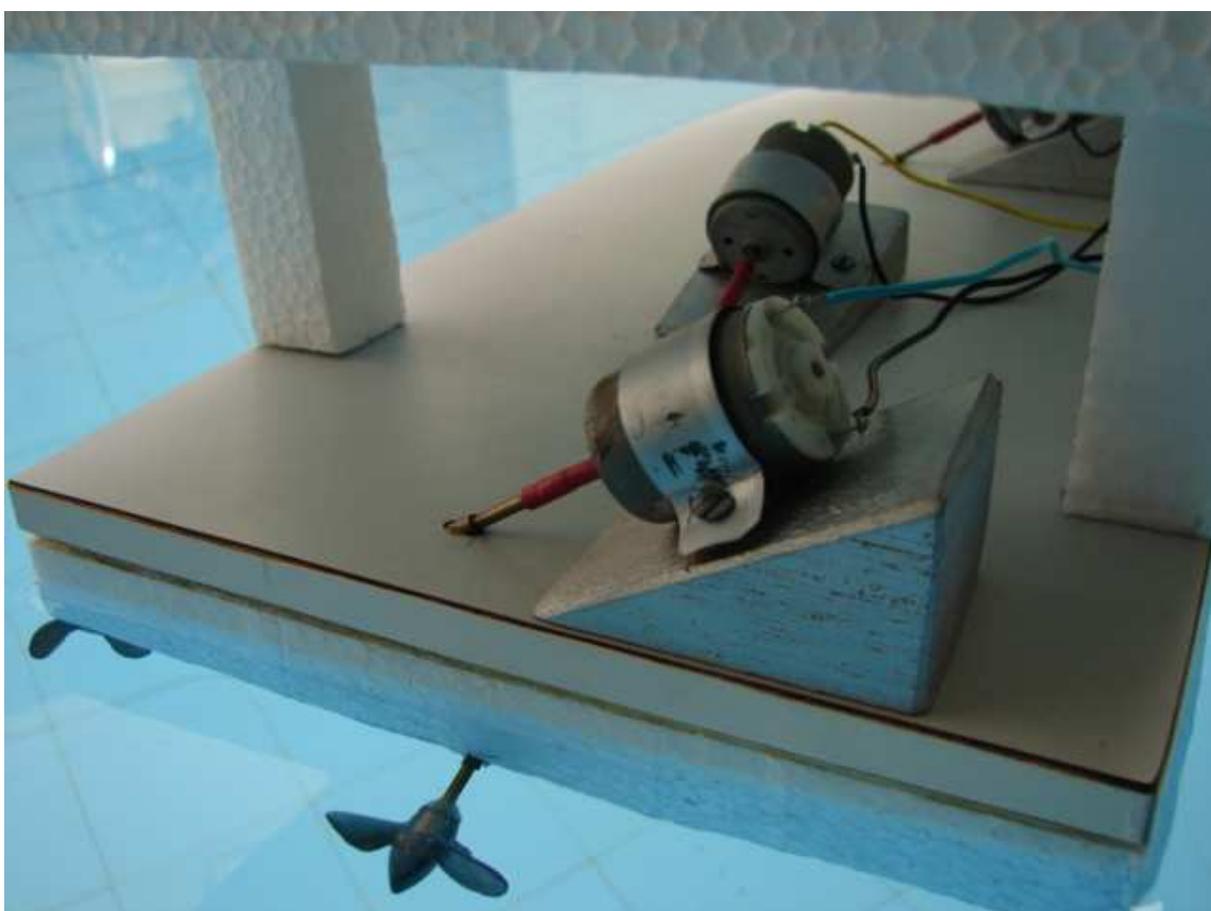


Figura A.3: Aspecto dos motores e hélices.

O modelo dos motores é o PM100-SG. A Tabela A.1 apresenta as características do motor.

Tabela A.1: Características do motor PM100-SG.

| Tensão (V) | | Em vazio | | Máximo rendimento | | | | | Bloqueio | |
|------------|---------|----------|----------|-------------------|----------|--------|----------|------------|----------|----------|
| | | Rotação | Corrente | Rotação | Corrente | Torque | Potência | Rendimento | Torque | Corrente |
| Operação | Nominal | RPM | A | RPM | A | g.cm | W | % | g.cm | A |
| 12 | 12 | 4870 | 0,03 | 3901 | 0,12 | 23,1 | 0,923 | 63,71 | 116 | 0,486 |

A.3 – Acionamento dos Motores

O acionamento dos motores da embarcação é feito através de um circuito conectado a um microcomputador do tipo PC (*Personal Computer*) no qual é executado o algoritmo de controle. O circuito recebe sinais de comando da porta paralela do microcomputador. É composto basicamente por *drivers* do tipo ponte “H” que possibilitam o controle do sentido de rotação dos motores (Rashid, 2001). Para o controle de cada propulsor foram designados dois *bits* da porta paralela. Um deles é responsável pelo sentido de rotação do propulsor e o outro pelo estado: ligado ou desligado. Desta forma, foram utilizados seis *bits* (D0 a D5) para comandar os três motores.

A Figura A.4 apresenta o esquema básico de uma ponte “H”.

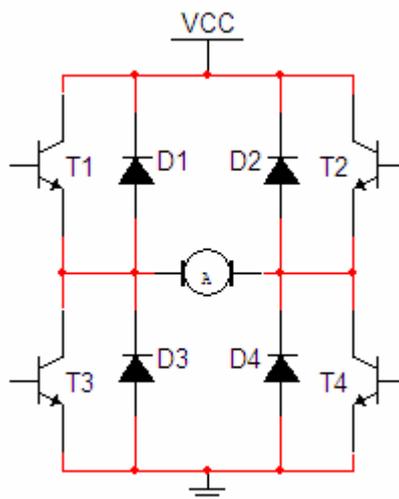


Figura A.4: Esquema básico de uma ponte “H”

Observando o circuito na Figura A.4, nota-se que o sentido de rotação dos propulsores é definido através dos estados dos transistores. Para que o motor seja polarizado diretamente, os transistores *T1* e *T4* devem estar em estado de saturação

enquanto $T2$ e $T3$ ficam em corte. Para a polarização inversa, $T2$ e $T3$ devem ficar saturados enquanto $T1$ e $T4$ ficam cortados.

Os *drivers* ponte “H” utilizados no circuito de acionamento foram desenvolvidos com o circuito integrado (CI) L298HN. Este CI possui dois *drivers* ponte “H” encapsulados e é compatível com lógica TTL (*transistor-transistor logic*). A Figura A.5 apresenta o esquema simplificado do CI L298HN.

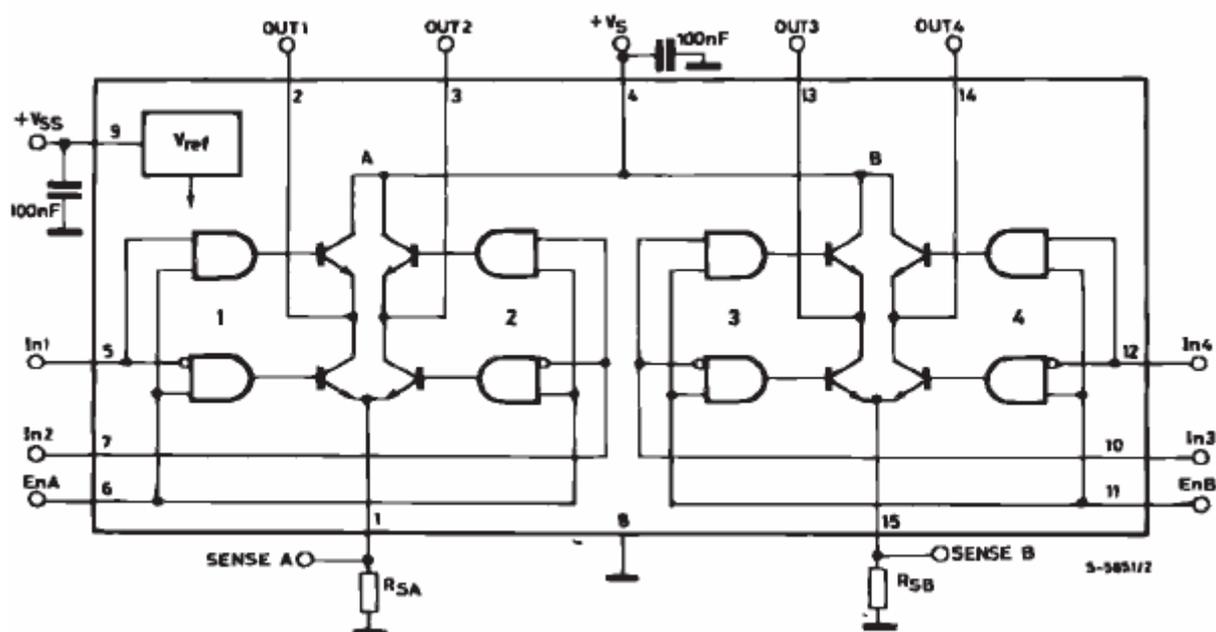


Figura A.5: Esquema simplificado do circuito integrado L298HN.

No esquema da Figura A.5 observam-se duas entradas de alimentação: V_{ss} e V_s . A entrada $V_{ss}=+5V$ é usada para a alimentação da parte lógica do CI. A entrada V_s é a tensão de alimentação dos propulsores e deve ter valor máximo de +12V (tensão nominal dos motores utilizados). O sentido de rotação dos propulsores é definido através das entradas $In1$ e $In2$ no *driver A* e $In3$ e $In4$ no *driver B*. Os estados ligado ou desligado são definidos através das entradas EnA e EnB . A Tabela A.2 define a lógica de polarização da saída entre os pinos $OUT1$ e $OUT2$ no *driver A* através das entradas $In1$, $In2$ e EnA .

Tabela A.2: Lógica de polarização da saída do *driver A*. Legenda: X – *Don't care*.

| EnA | In1 | In2 | Polarização (OUT1-OUT2) |
|-----|-----|-----|-------------------------|
| 1 | 1 | 0 | Direta |
| | 0 | 1 | Inversa |
| 0 | X | X | Transistores cortados |

Observando a Tabela A.2 pode-se concluir que, para definir o sentido de rotação dos propulsores, os níveis lógicos das entradas *In1* e *In2* serão sempre opostos. Desta forma, utilizou-se o inversor apresentado na Figura A.6 para controlar o sentido de rotação dos propulsores através de apenas um *bit* da porta paralela.

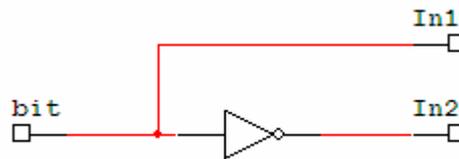


Figura A.6: Diagrama eletrônico para o controle das entradas *In1* e *In2* do CI L298HN.

O diagrama apresentado na Figura A.6 não inclui o *bit* designado para o controle do estado do propulsor no *driver A*, o qual ficará ligado diretamente à entrada *EnA*. A Figura A.7 apresenta o diagrama completo do circuito de acionamento dos motores. Nota-se que, como foram utilizados apenas três motores, um dos CIs L298HN terá ainda um *driver* ocioso.

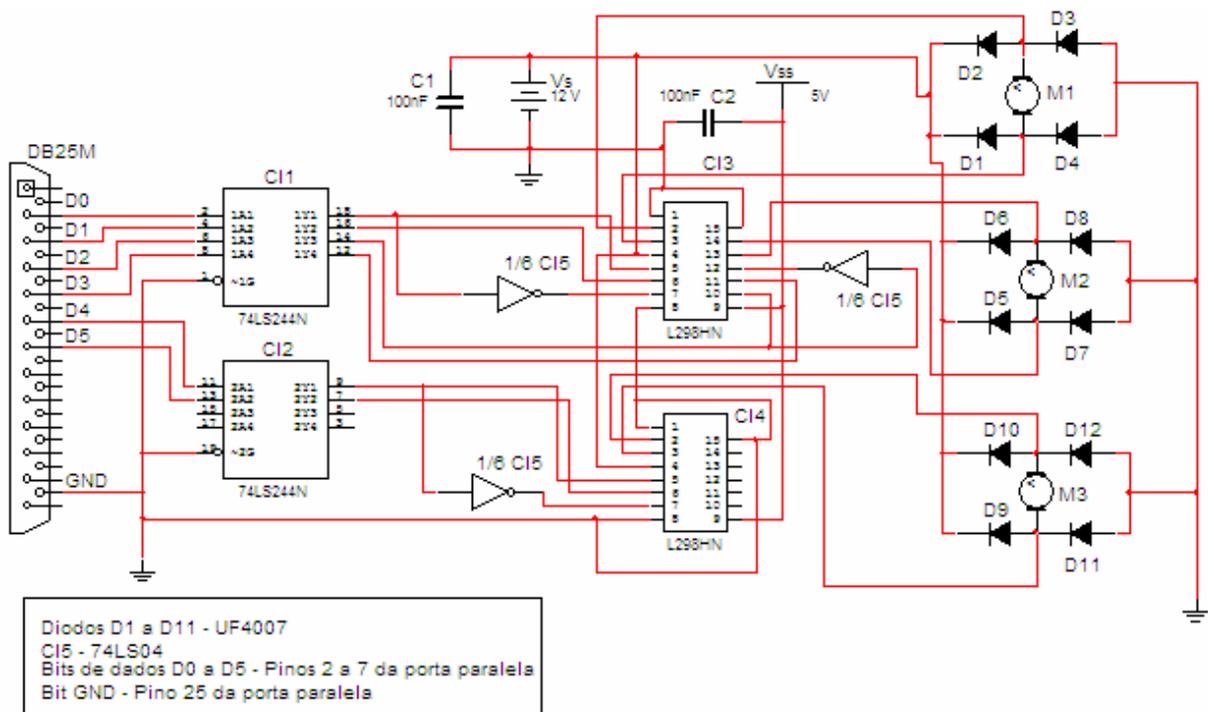


Figura A.7: Diagrama do circuito de acionamento dos motores.

Na Figura A.7 observam-se diodos colocados em antiparalelo. Estes diodos são denominados diodos de *free-wheeling* e permitem a circulação da corrente induzida pelos motores quando os transistores da ponte “H” entram em corte (Rashid, 2001).

A Figura A.8 mostra a placa onde foi montado o circuito de acionamento dos motores.

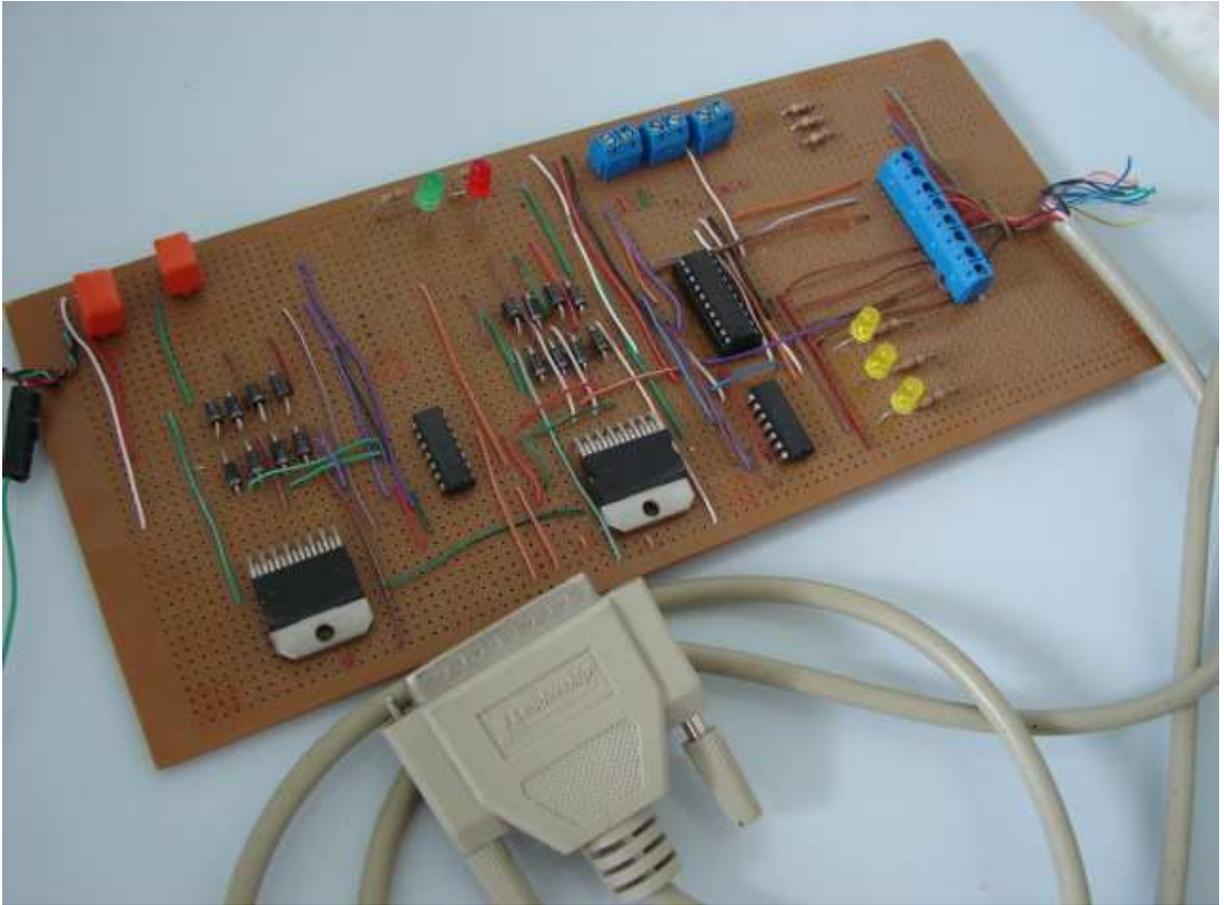


Figura A.8: Placa com circuito de acionamento.

A Tabela A.3 apresenta a relação entre os sinais dos bits da porta paralela utilizados (D0 a D5) e os estados dos propulsores.

Tabela A.3: Lógica de rotação dos propulsores. Legenda: X – *Don't care*.

| Propulsor 1 | | | Propulsor 2 | | | Propulsor 3 | | |
|-------------|----|--------------------|-------------|----|--------------------|-------------|----|--------------------|
| D0 | D1 | Sentido de rotação | D2 | D3 | Sentido de rotação | D4 | D5 | Sentido de rotação |
| 1 | 1 | Horário | 1 | 1 | Horário | 1 | 1 | Horário |
| 0 | | Anti-horário | 0 | | Anti-horário | 0 | | Anti-horário |
| X | 0 | Parado | X | 0 | Parado | X | 0 | Parado |

APÊNDICE B – ROTINA DE CONTROLE

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

Programa em MATLAB

```
uRef=160; %posição de referência no eixo Ue
vRef=115; %posição de referência no eixo Ve
sf=0.00112; %fator de escalamento
xRef=uRef*sf; %posição de referência no eixo Xe
yRef=vRef*sf; %posição de referência no eixo Ye
OrientacaoRef=0; %Orientação de referência

% Propulsor 1 (p1) move o barco em x.
% Propulsores 2 e 3 (p2 e p3) movem o barco no grau de liberdade y e em
torno do eixo z.

SentidoP1=0; %sentido de rotação inicial do propulsor p1
EnableP1=0; %condição inicial do motor 1 (OFF)
SentidoP2=0;
EnableP2=0;
SentidoP3=0;
EnableP3=0;
K=2.5; %(K equivale à constante "lambda" da superficie de deslizamento)

for n=1:inf
frame=getsnapshot(vid); %Captura a imagem.
[cmx cmy Orientacao]=ProcessarImagem(frame); %Obtem informações de
interesse da imagem.
AnguloRad=(Orientacao*pi)/180;
if n==1
    vx=0; %velocidade inicial no grau de liberdade x.
    vy=0; %velocidade inicial no grau de liberdade y.
    vor=0; %velocidade inicial no grau de liberdade "Psi".
else
    t=toc; %Pára o cronometro e armazena o tempo em "t".
    vx=(cmx-cmxAnt)/t;
    vy=(cmy-cmyAnt)/t;
    vor=(Orientacao-OrientacaoAnt)/t;
end;
ex=xRef-cmx; %Erro no grau de liberdade Xe
sx=Kx*ex-vx; %Variavel de controle em Xe

ey=yRef-cmy; %Erro no grau de liberdade Ye
sy=Ky*ey-vy; %Variavel de controle em Ye

ePsi=OrientacaoRef-Orientacao; %Erro no grau de liberdade "Psi"
sPsi=KPsi*ePsi-vor; %Variavel de controle em "Psi"

%Convertendo os sinais de comando para o sistema de coordenadas móvel:
ux=sign(sx*cos(AnguloRad)+sy*sin(AnguloRad));
uy=sign(-sx*sin(AnguloRad)+sy*cos(AnguloRad));
uPsi=sign(sPsi);
```

```

%Sinais de comando dos propulsores
up1=ux;
up2=sign(uy+uPsi);
up3=sign(uy-uPsi);

if up1<0
    SentidoP1=1; %Define sentido inverso de rotaçãõ do propulsor 1
    EnableP1=1; %Ativa o propulsor 1
elseif up1>0
    SentidoP1=0; %Define sentido direto de rotaçãõ do propulsor 1
    EnableP1=1;
elseif up1==0
    EnableP1=0; %Desativa o propulsor 1
end;

if up2<0
    SentidoP2=1; %Define sentido inverso de rotaçãõ do propulsor 2
    EnableP2=1; %Ativa o propulsor 2
elseif up2>0
    SentidoP2=0; %Define sentido direto de rotaçãõ do propulsor 2
    EnableP2=1;
elseif up2==0
    EnableP2=0; %Desativa o propulsor 2
end;

if up3<0
    SentidoP3=1; %Define sentido inverso de rotaçãõ do propulsor 3
    EnableP3=1; %Ativa o propulsor 3
elseif up3>0
    SentidoP3=0; %Define sentido direto de rotaçãõ do propulsor 3
    EnableP3=1;
elseif up3==0
    EnableP3=0; %Desativa o propulsor 3
end;

Comando=[SentidoP1 EnableP1 SentidoP2 EnableP2 SentidoP3 EnableP3 0 0];
%define vetor contendo os valores dos bits que serãõ enviados para a
porta paralela.
Putvalue(Paralela,Comando); %Envia o vetor de bits para a porta paralela
tic; %Inicia cronõmetro
cmxAnt=cmx; %armazena valor de Xe(n-1) para estimaçãõ da velocidade
cmyAnt=cmy; %armazena valor de Ye(n-1) para estimaçãõ da velocidade
OrientacaoAnt=Orientacao; %armazena valor de Orientacao(n-1) para
estimaçãõ da velocidade
flushdata(vid); %limpa buffer do objeto de video
end;

```