

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

MARCELO PARAVISI

**UM SIMULADOR ROBÓTICO PARA INUNDAÇÕES CONSIDERANDO DISTÚRBIOS
AMBIENTAIS**

Porto Alegre

2020

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**UM SIMULADOR ROBÓTICO
PARA INUNDAÇÕES
CONSIDERANDO DISTÚRBIOS
AMBIENTAIS**

MARCELO PARAVISI

Tese apresentada como requisito parcial
à obtenção do grau de Doutor em
Ciência da Computação na Pontifícia
Universidade Católica do Rio Grande do
Sul.

Orientador: Prof. Alexandre de Moraes Amory

**Porto Alegre
2020**

Ficha Catalográfica

P224s Paravisi, Marcelo

Um simulador robótico para inundações considerando distúrbios ambientais / Marcelo Paravisi . – 2020.

104.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Alexandre de Moraes Amory.

1. Robótica. 2. Simulação. 3. Unmanned Surface Vehicle. 4. Inundações. 5. Validação. I. Amory, Alexandre de Moraes. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Marcelo Paravisi

Um simulador robótico para inundações considerando distúrbios ambientais

Tese apresentada como requisito parcial para obtenção do grau de Doutor em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 15 de julho de 2020.

BANCA EXAMINADORA:

DRA. SILVIA SILVA DA COSTA BOTELHO - C3 - FURG

DR. RAFAEL HEITOR BORDINI - PPGCC – PUCRS

DR. JORGE HUGO SILVESTRINI - PGETEMA - PUCRS

DR. ALEXANDRE DE MORAIS AMORY (PPGCC/PUCRS - Orientador)

AGRADECIMENTOS

Agradeço a minha amada esposa Thaís, que me deu força e coragem, me auxiliando nos momentos difíceis. As minhas filhas Helena e Júlia, pela paciência e compreensão na minha ausência, e pela recepção calorosa em meus retornos de viagem. Aos meus pais e irmã, que sempre me apoiaram e me acolheram em Porto Alegre. Aos meus sogros e cunhado, que se fizeram presentes auxiliando minha família.

Agradeço também aos amigos que fiz no PPGCC, a todos os professores do curso que me acompanharam e me auxiliaram durante essa trajetória. Ao professor Alexandre de Moraes Amory, por seus ensinamentos, paciência e confiança ao longo das orientações.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, pelo programa de qualificação ao qual pude usufruir ao longo desse Doutorado. A CAPES, pelo projeto PRO-ALERTAS, ao qual estive vinculado durante quatro anos.

UM SIMULADOR ROBÓTICO PARA INUNDAÇÕES CONSIDERANDO DISTÚRBIOS AMBIENTAIS

RESUMO

A maioria dos desastres naturais ao redor mundo são provocados por eventos climáticos, sendo que, no Brasil, os desastres estão vinculados principalmente a chuvas e a deslizamentos de encostas. Os *Rescue Robots* tem se tornado aliados importantes das equipes de resgate para auxiliar na realização de tarefas com elevado grau de periculosidade. Contudo, o desenvolvimento de novas plataformas robóticas, capazes de lidar com ambientes tão imprevisíveis, demanda que eles sejam avaliados em condições similares ao que serão expostos no mundo real. Considerando que existem poucos locais de testes para desastres naturais, simuladores robóticos apresentam-se como uma ferramenta importante para reduzir custos de pesquisa e desenvolvimento, evitando perdas de equipamentos e identificando defeitos e limitações em protótipos. Nesse sentido, a presente tese de doutorado propõe um ambiente de simulação robótica para desastres naturais de inundações, que reproduz os distúrbios ambientais existentes em tais cenários. Para tal, são propostos protocolos para reproduzir cenários reais e seus distúrbios no ambiente virtual, bem como o protocolo e os experimentos para validar a movimentação de USVs virtuais. Por fim, são realizados estudos de casos para demonstrar o impacto dos distúrbios nas atividades realizadas por USVs.

Palavras-Chave: Simulação, Robótica, Desastres ambientais, coeficientes hidrológicos.

A ROBOTIC SIMULATOR FOR INUNDATION CONSIDERING ENVIRONMENT DISTURBANCES

ABSTRACT

Extreme weather events cause most disasters around the world, and in Brazil, the main kind of such event are landslides and rains. Rescue Robots have become important allies of rescue teams to assist in carrying out tasks with a high degree of danger. However, the development of new robotic platforms capable of handling such unpredictable environments demands that the roboticist evaluate the robot platform under similar conditions of the real world. Considering that there are few test sites for disasters, robotic simulators for natural disasters present themselves as an essential tool to reduce research and development costs, avoiding equipment losses, and identifying defects and limitations in prototypes. In this sense, the present document proposes a robotic simulation environment for natural disasters that reproduces the environmental disturbances present in flooding scenarios. Thus, this thesis presents a protocol to reproduce real scenarios and their disturbances in the virtual environment, it also describes a protocol and experiments to validate the movement of virtual USVs. Finally, case studies demonstrate the impact of disturbances on the activities performed by the USVs.

Keywords: Simulation, robotic, natural disasters, hydrological coefficients.

LISTA DE FIGURAS

3.1	Fluxograma com as etapas para modelar terrenos e simular as correntes de vento e água.	40
3.2	Geometria das edificações gerada pela ferramenta OSM2World	41
3.3	Terreno 3D com as construções transladas sobre a superfície. A geometria 3D reproduz o centro de Porto Alegre, RS, Brasil.	42
3.4	Modelos digital de terrenos e superfície da região central de Porto Alegre, RS, Brasil. A altura é representada em escala de cinza, branco para mais alto e preto para a mais baixa.	43
3.5	Grade 2D utilizada na simulação do HEC-RAS. Bordas da grade em azul. .	44
3.6	Simulação da correnteza de água ao longo tempo no arroio Dilúvio.	44
3.7	Simulação da correnteza de água, sendo que a velocidade da água pode ser observada por partículas brancas. Quanto mais longa a partícula, maior a velocidade no local.	45
3.8	Visualização das correntes de vento na ferramenta paraFoam para diferentes altitudes: 1 metro (a), 5 metros (b), 10 metros (c), 20 metros (d). Nas figuras (e) e (f), é apresentado o gradiente de velocidade para dois planos: um próximo a edificações e outro próximo a pontes.	47
3.9	Arquitetura de simulação contendo os principais módulos.	48
3.10	Sistema de coordenada do barco (v_x, v_y) e as componentes da velocidade aparente da água.	49
3.11	Implementações diferentes do efeito de boiar é apresentado em (a), (b) e (c), onde os pontos pretos representam centros de flutuabilidade, enquanto que as setas vermelhas e azuis representam as forças da gravidade e de flutuabilidade respectivamente. Em (a), o efeito de boiar do Freefloating gazebo é apresentado [Ker14]. Em (b), o efeito de boiar na embarcação é apresentando, no qual apenas a altura da onda é apresentada, dado que um barco é definido apenas com um único <i>link</i> . Em (c), o efeito similar é apresentado sendo que a embarcação passa a ser subdividido por um conjunto de <i>links</i> e <i>joints</i> como mostrado em (d). Assim, as forças da gravidade de boiar são aplicados para centro de massa e flutuabilidade de cada <i>link</i> . .	51
3.12	Da esquerda para direita: <i>rudder boat</i> , <i>airboat</i> , <i>differential boat</i> and <i>sailboat</i> disponibilizado no simulador.	53

3.13	Simulação da corrente de água com o Hec RAS para um ambiente em que ocorre a interseção de dois rios. A imagem da esquerda apresenta a trajetórias de partículas em branco movendo-se sobre a água, assim pode-se compreender a direção e módulo da correnteza de água. Na imagem da direita, o fluxo mais rápido é apresentado em vermelho, enquanto que o de menor intensidade em azul.	54
3.14	Resultado da simulação de vento com OpenFoam.	55
4.1	Sistema de coordenada do barco (v_x, v_y) e as componentes da velocidade aparente da água. Essa figura foi propositalmente repetida para facilitar a leitura.	58
4.2	Resultados para aceleração frontal comparando-se o desempenho do robô real em testes em campo com os dos barcos virtuais no ambiente simulado.	67
4.3	Resultados de três testes em campo com o <i>Lutra Airboat</i> , no qual foram realizadas curvas no sentido horário. São apresentadas as trajetórias (Figura 4.3a), as componentes da trajetória ao longo do tempo (Figura 4.3b) e a frequência e amplitude do movimento da embarcação (Figura 4.3c). . .	68
4.4	Resultado da simulação do airboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas a trajetória (Figura 4.4a) e suas componentes ao longo do tempo (Figura 4.4b), a frequência e amplitude do movimento da embarcação (Figura 4.4c).	69
4.5	Resultado da velocidade do <i>Lutra Airboat</i> ao longo dos três testes em campo. Em (a) são apresentados as componentes da velocidade no domínio do tempo, enquanto que em (b) são apresentados as componentes no domínio da frequência.	70
4.6	Resultado da simulação do airboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas as componentes da velocidade ao longo do tempo (Figura 4.6a) e as componentes da velocidade no domínio da frequência (Figura 4.6b).	70
4.7	Resultados de três testes em campo com o <i>Lutra Prop</i> , no qual foram realizadas curvas no sentido anti-horário. São apresentadas as trajetórias (Figura 4.7a), as componentes da trajetória ao longo do tempo (Figura 4.7b) e a frequência e amplitude do movimento da embarcação (Figura 4.7c). . .	71
4.8	Resultado da simulação do airboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas a trajetória (Figura 4.8a) e suas componentes ao longo do tempo (Figura 4.8b), a frequência e amplitude do movimento da embarcação (Figura 4.8c).	72

4.9	Resultado da velocidade do <i>Lutra Prop</i> ao longo dos três testes em campo. Em (a) são apresentados as componentes da velocidade no domínio do tempo, enquanto que em (b) são apresentados as componentes no domínio da frequência.	73
4.10	Resultado da simulação do diffboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas as componentes da velocidade ao longo do tempo (Figura 4.10a) e as componentes da velocidade no domínio da frequência (Figura 4.10b).	73
5.1	Trajétoria do barco diferencial no ambiente da Figura 3.13. A cor da trajetória representa a velocidade da embarcação ao longo do caminho.	78
5.2	Diferentes velocidades de navegação do barco diferencial navegando rio acima, contrário a correnteza de água.	79
5.3	Trajetoria do <i>rudderboat</i> descendo o rio sem e com correnteza de água. Trajetória em azul: a embarcação navega sem correnteza de água. Trajetória em vermelho: a embarcação navega a favor da correnteza de água com a mesma velocidade da água (módulo da velocidade da água é 1m/s; a direção/sentido é de baixo para cima).	80
5.4	<i>Rudderboat</i> descendo o rio com a mesma velocidade da correnteza de água. Imagens apresentam a colisão da embarcação com o pilar da ponte.	80
5.5	Trajétórias do <i>Rudderboat</i> ao descer o rio com corrente de água (linha azul) e sem corrente de água (linha verde). Embarcações partem lado direito da imagem.	81
5.6	Ângulo do leme ao longo tempo para cada um dos <i>Rudderboats</i> descendo o rio.	81
5.7	Barco diferencial navegando em linha reta. A linha vermelha apresenta a trajetória da embarcação quando afetada apenas corrente de água, a linha rosa quando afetado apenas pelo vento. A trajetória verde quando a embarcação é afetado simultaneamente pelo vento e água.	82
5.8	Da esquerda para a direita, diagramas polares de um veleiro real [RJS01] e do modelo de simulação do veleiro contendo uma área de vela de 2m ²	83
5.9	Cenários propostos para avaliação.	84
5.10	Trajétórias dos três cenários para cada um das quatro embarcações. As trajetórias do <i>Airboat</i> , <i>Differential boat</i> , <i>Rudderboat</i> , <i>Sailboat</i> são exibidas em linhas vermelhas, rosas, verdes e pretas, respectivamente.	85
5.11	Diagrama da arquitetura multi-camadas do USV_Sim integrado ao pacote move_base e ao pacote heatmap.	87
5.12	Experimento de <i>coverage path-planning</i> de forma a comparar os efeitos dos distúrbios e desvio de obstáculos móveis na trajetória do <i>Differential boat</i>	90

- 5.13 Inspeção de ponte com um barco diferencial. A linha em preto representa a trajetória da embarcação ao navegar com distúrbios, enquanto que em verde a trajetória sem distúrbios. Os pontos C1, C2 e C3 são os locais onde a embarcação colidiu ao ter sido arrastada pelos distúrbios naturais. . . . 91
- 5.14 Resultado da simulação do *Airboat* para mover-se numa inundação em área urbana. As trajetórias em rosa é o deslocamento realizado na presença de distúrbios, enquanto que em preto não existem distúrbios. . . . 93

LISTA DE TABELAS

2.1	Comparação entre simuladores de USVs com relação a característica desejáveis. Critério ? significa informação não disponível em publicações e na documentação. × significa características não suportada. ✓ significa parcialmente suportada ou limitação significativa. ✓✓ significa totalmente suportada e prontamente disponível para o usuário.	33
4.1	Parâmetros físicos do <i>Lutra Airboat</i> e <i>Lutra Prop</i>	59
4.2	Força de propulsão das embarcações para diferentes valores de PWM means Pulse Width Modulation.	60
4.3	Parâmetros dinâmicos do <i>Lutra Airboat</i> e <i>Lutra Prop</i> coletados na etapa 3 e 4.	62
5.1	Estatísticas da simulação realizada com cada tipo de embarcação para todos os cenários da Figura 5.10.	85
5.2	Distância e tempo para o <i>Differential boat</i> realizar a cobertura da região na presença e ausência de distúrbios de vento e correnteza de água. As consequências de haver um cruzamento de outra embarcação, provocando o desvio a direita conforme as COLREGS.	88

LISTA DE SIGLAS

ADCP – Acoustic Doppler Current Profiler
AIS – Automatic Identification System
BDI – Belief Desire Intention
CFD – Computational Fluid Dynamics
GNC – Guidance, Navigation and Control
GPS – Global Positioning System
HDF – Hierarchical Data Format
IMU – Inertial Measurement Unit
LBM – Lattice Boltzmann Method
LSA – Laboratório de Sistemas Autônomos
MDT – Modelo Digital de Terreno
MDS – Modelo Digital de Superfície
MOOS-IVP – Mission Oriented Operating Suite and Interval Programming
NIST – National Institute of Standards and Technology
OGRE – Object-Oriented Graphics Rendering Engine
ROV – Remotely Operated Vehicle
RTK – Real Time Kinematic
SAUC-E – Student AUV Challenge Europe
ELROB – European Land Robot Trial
UAV – Unmanned Aerial Vehicle
UGV – Unmanned Ground Vehicle
UMV – Unmanned Marine Vehicle
USV – Unmanned Surface Vehicle
UUV – Unmanned Underwater Vehicles
XML – Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	23
1.1	METODOLOGIA	25
1.1.1	HIPÓTESE E PERGUNTAS DE PESQUISA	26
1.2	CONTRIBUIÇÕES	29
2	REVISÃO BIBLIOGRÁFICA E DOCUMENTAL	31
3	SIMULADOR	39
3.1	MODELO DE CENÁRIOS	39
3.1.1	MODELANDO TERRENOS 3D	39
3.1.2	MODELANDO CORRENTEZA DE ÁGUA E DE VENTO	41
3.2	ARQUITETURA	46
3.2.1	CINEMÁTICA DE USVS	48
3.2.2	PLUGIN FOIL DYNAMICS	49
3.2.3	APRIMORAMENTOS NO UWSIM E NO PLUGIN FREEFLOATING GAZEBO	50
3.2.4	MODELOS DE ROBÔS PROPOSTOS	52
3.2.5	MÓDULOS DE CORRENTEZA DE ÁGUA E VENTO	53
4	VALIDAÇÃO	57
4.1	METODOLOGIA	57
4.1.1	ETAPA 1: ESPECIFICAÇÕES FÍSICAS DOS ROBÔS	58
4.1.2	ETAPA 2: EXTRAÇÃO DE FORÇAS DE PROPULSÃO	59
4.1.3	ETAPA 3: EXTRAÇÃO DA DINÂMICA DE MOVIMENTO FRONTAL	60
4.1.4	ETAPA 4: EXTRAÇÃO DA DINÂMICA DE MOVIMENTO LATERAL	63
4.1.5	TESTES COM ROBÔS SIMULADOS	65
4.2	COMPARAÇÃO ENTRE ROBÔS REAIS E SIMULADOS	66
4.2.1	RESULTADOS PARA ACELERAÇÃO FRONTAL	66
4.2.2	RESULTADOS PARA MOVIMENTO LATERAL	67
4.3	OUTRAS CONSIDERAÇÕES	73
5	ESTUDOS DE CASOS	77
5.1	EMBARCAÇÃO EM SITUAÇÕES DE DISTÚRBIOS VARIÁVEIS	77
5.2	RIO COM DIFERENTES CORRENTES DE ÁGUA	78

5.3	RUDDERBOAT DESCENDO RIO	79
5.4	EFEITOS DOS DISTÚRBIOS SOBRE O BARCO DIFERENCIAL	81
5.5	DESEMPENHO DO VELEIRO	82
5.6	COMPARAÇÃO ENTRE OS VEÍCULOS	83
5.7	PLANEJAMENTO DE ROTAS DE COBERTURA	85
5.8	INSPEÇÃO DE PONTE	89
5.9	INUNDAÇÃO	91
6	CONCLUSÕES	95
6.1	LIMITAÇÕES	96
6.2	TRABALHOS FUTUROS	97
	REFERÊNCIAS	99

1. INTRODUÇÃO

Desastres são definidos como eventos meteorológicos, geológicos ou provocados por homens, que demandam a intervenção de agências compostas por equipes especializadas para lidar com danos às edificações e às estruturas, minimizando a perdas de vidas [SK16]. Conforme World Meteorological Organization [RM15], aproximadamente 90% dos desastres naturais ao redor do mundo são provocados por tufões, tornados e ciclones tropicais. De maneira similar, no Brasil, os desastres naturais estão relacionados a eventos climáticos extremos [DDC⁺13] [HCE16] [dBWSF17], dentre os quais, chuvas e deslizamentos de encostas desalojam muitas famílias e ceifam muitas vidas.

Não é apenas após a ocorrência de um desastre natural que devem ser tomadas ações. Na área de gerenciamento de desastres, o ciclo de ações é subdividido em 4 fases, sendo que as duas primeiras, mitigação e preparação, precedem o momento do evento em si, enquanto que as duas últimas, resposta e recuperação, ocorrem durante e após o desastre. A fase de mitigação é responsável por identificar e eliminar os riscos de eventos futuros, tais como, reforçar estruturas que possam colapsar e remover famílias de áreas de riscos. A fase de preparação pressupõe que o evento não poderá ser impedido, conseqüentemente devem ser realizados treinamentos e elaborados planos para serem executados durante o desastre. Já a fase de resposta consiste nas ações de avaliações dos riscos existentes, de remoções e de resgates de pessoas. A última fase é a de recuperação, na qual ocorre a retirada de detritos e a reconstrução das áreas afetadas.

Inicialmente o uso da robótica em situações de desastres estava restrito a fase de resposta, ou seja, a realização de tarefas que visa a localização e resgate de vítimas, identificação dos riscos e da extensão da zona de desastre [SK16]. Posteriormente, ampliou-se para utilização na fase de recuperação, já que os robôs conseguem auxiliar em tarefas importantes para a revitalização da área afetada pelo desastre. Um exemplo de tarefa na fase de recuperação é a localização de automóveis e embarcações que estão submersos em rios e lagos, pois esses podem provocar vazamentos de combustível e óleo. A inspeção de estruturas, tais como pontes e edifícios, por veículos autônomos, era realizada primeiramente apenas na fase de resposta, passou a ser feita também na fase de mitigação, uma vez que deve ser avaliado o estado da estrutura para identificar as fragilidades que possam levar ao futuro colapso da edificação.

Contudo, para que um robô possa operar em locais de desastres como os citados acima, ele deve perceber e atuar num mundo imprevisível, estando ciente do ambiente em que navega e evitando causar novos danos ou ampliar os existentes [Mur14]. A dificuldade em atingir esses objetivos está no fato de que as condições dos locais de desastres podem ser extremas, o que acaba por afetar o desempenho de sensores, já que limita o

uso destes com tamanho e peso elevados. Além disso, as embarcações podem não ter potência e estabilidade suficientes para superar as forças geradas pelo vento e pela água.

Ainda nesse contexto, Murphy [Mur14] identificou que o design de algumas plataformas robóticas está limitando o uso dessas em situações de desastres, um exemplo disso são os *Unmanned Marine Vehicles* (UMVs), que tem sido projetados para ambientes aquáticos abertos, em que os distúrbios ambientais são limitados ou reduzidos. Os *Rescue Robots* deveriam ser projetados para atuar regiões litorâneas (rios, lagos, região costeira, portos), em que distúrbios ambientais (como ondas, ventos e correnteza de água, marés) estão presentes. Um exemplo em que os *Rescue Robots* não puderam ser utilizados devido as condições severas do ambiente é o caso do desabamento de uma ponte em Minnesota (EUA) em 2007. Nesse desastre, pretendeu-se utilizar um ROV (*Remotely Operated Vehicle*) para documentar o lugar e localizar vítimas presas embaixo de destroço, porém, a turbidez e correnteza da água dificultaram a operação do ROV.

Por outro lado, para que os robôs sejam melhor utilizados em situações de desastres, é necessário encontrar formas para avaliar novos protótipos, já que existem poucos locais de teste que se aproximam das condições apresentadas em situações de desastres. Dentre as poucas iniciativas existentes, destacam-se Robocup-Rescue Project [TKT+00], League [BCK+07], competição de robótica marinha SAUC-E [SAU20], experimentação de UAVs (*Unmanned Aerial Vehicles*) chamada ELROB [Sch20], e um *benchmarking* no mundo real chamado *Disaster City* do *National Institute of Standards and Technology* (NIST) [KK09].

Considerando que ambientes de desastres apresentam elevados riscos para plataformas robóticas caras, a realização de testes em simulação pode reduzir custos de pesquisa e desenvolvimento, evitando perdas de equipamentos, identificando antecipadamente os erros e as limitações no design de novos protótipos robóticos. Contudo, ainda não existem simulações robóticas realísticas de ambientes com distúrbios complexos e severos, como as correntes de água que UMVs enfrentam em regiões litorâneas e em inundações em áreas urbanas. Da mesma forma, ainda carecem simulações robóticas realistas de vento para realização de testes em zonas urbanas, em que a passagem de vento entre prédios forma vórtices e corredores de ventos, podendo arrastar a plataforma robótica.

Devido a essas limitações, a presente tese tem como objetivo propor o simulador robótico USV_SIM para situações de desastres de inundações, capaz de apresentar a dinâmica de fluídos de maneira aero- e hidrológicamente precisa, aproximando a movimentação de *Unmanned Surface Vehicles* (USVs) na simulação dos observados no mundo real. Salienta-se que o tema dessa tese possui enfoque em USVs, uma vez que o doutoramento esteve vinculado ao projeto de pesquisa Pro-alertas da CAPES, o qual possuía foco em barcos robóticos.

Para atingir esse objetivo geral, elencou-se os seguintes objetivos específicos: a simulação dos distúrbios ambientais deve considerar a topologia do terreno bem como a presença de grandes obstáculos, por exemplo, edifícios, pontes e árvores. Consequente-

mente devem ser modelados virtualmente o ambiente em que deseja-se realizar os testes, já que os simuladores utilizam a geometria 3D do terreno e dos obstáculos como entrada. Para realizar a validação dos movimentos das embarcações simuladas, devem ser realizadas coletas de dados com os barcos reais. Por fim, utilizando o simulador proposto, cabe demonstrar como os distúrbios ambientais impactam nas atividades geralmente realizadas por USVs.

Ao atingir esses objetivos específicos, pode-se viabilizar o desenvolvimento e comparação de novos algoritmos e técnicas GNC (*Guidance, Navigation and Control*), bem como de novas plataformas robóticas a serem utilizadas em condições ambientais severas. Permitindo a execução de diferentes algoritmos num mesmo cenário, é possível avaliar o desempenho desses em circunstâncias iguais. O que dificilmente será atingido, ao tentar-se realizar a comparação no mundo real, uma vez que as situações de desastres oferecem riscos e não existem meios para garantir que o estímulo em ambos os testes será o mesmo. Isso acaba por prejudicar a avaliação dos resultados obtidos.

O presente documento está organizado da seguinte forma: ainda no Capítulo 1 são descritas as perguntas de pesquisa, a hipótese e as etapas para a concretização da metodologia. No Capítulo 2, é apresentada uma revisão dos simuladores robóticos existentes, no Capítulo 3 é descrito o modelo de simulação proposto. Já no Capítulo 4, é exposto o protocolo utilizado para realizar a validação do modelo de simulação dos USVs, bem como os resultados obtidos na validação e as dificuldades encontradas. No Capítulo 5, são apresentados diversos estudos de casos em que são demonstrados os impactos do modelo de simulação no desempenho das embarcações ao realizar diferentes atividades. Por fim, tem-se as considerações finais e as referências.

1.1 Metodologia

Dentre as etapas envolvidas no processo de desenvolvimento no campo da Robótica, a realização de testes e de validações é importante para identificar possíveis erros e limitações na solução proposta. Com relação a *Disaster Robotics*, a execução de testes no mundo real torna-se inviável, pois os ambientes de desastres apresentam elevado risco e não estão disponíveis a todo momento. Alternativamente, o uso de simuladores destaca-se como um recurso valioso, já que permite a repetição de testes e não oferece riscos aos pesquisadores e aos equipamentos.

Nesse contexto, observa-se que ainda carecem simuladores robóticos capazes de reproduzir desastres naturais em que os distúrbios ambientais (vento, ondas, corrente de água) variam ao longo do tempo e do espaço, e que são aero- e hidrologicamente precisos. A distância entre a modelagem de simulação e a realidade é um dos limitadores do grau de

confiabilidade de que os resultados obtidos se aproximam dos experimentos no ambiente real [Mur14].

Por esse motivo, na presente tese, foi desenvolvido um ambiente de simulação robótico capaz de reproduzir cenários de desastres naturais contando com distúrbios ambientais como ondas, ventos e correntes de água. Visando que esses distúrbios sejam realistas, foram utilizadas ferramentas aero- e hidrologicamente precisas, ou seja, ferramentas CFD (*Computational Fluid Dynamics*) já consolidadas na literatura e gratuitas, capazes de reproduzir o movimento de fluídos considerando o formato do terreno. Como exemplo dessas ferramentas tem-se OpenFoam [Ope20a], HEC-RAS [Eng20] [USA18], Incompact3d [LL11] [Inc20].

Dentre os cenários de desastres modelados, foram simulados inundações em áreas urbanas e inspeções de pontes, bem como disponibilizados modelos virtuais para os *Unmanned Surface Vehicle*. Sendo que, para garantir que a movimentação dessas embarcações modeladas seja realista, foi realizada a calibração da dinâmica desses veículos virtuais. Assim, as características físicas (peso, tamanho, etc) e cinéticas (aceleração, velocidade máxima) foram obtidas por meio de medições dos robôs reais existentes no Laboratório de Sistemas Autônomos (LSA) e por meio de testes em campo.

A seguir serão apresentadas a hipótese, as perguntas de pesquisa e as etapas para demonstrar a validade hipótese.

1.1.1 Hipótese e perguntas de pesquisa

A simulação robótica pode reproduzir com precisão a movimentação de USVs reais e os distúrbios naturais presentes em inundações, uma vez que a experimentação em campo permite identificar os parâmetros cinemáticos das embarcações reais e a utilização de ferramentas CFDs existentes simula as características de correntes de vento e de água.

Assim, por meio dessa hipótese, pode-se subdividir em diferentes perguntas específicas de pesquisa:

1. *Como os simuladores robóticos existentes podem ser aperfeiçoados com ferramentas CFDs para que ofereçam suporte a simulação de desastres naturais?*

Com a existência de ferramentas CFD, a reprodução de distúrbios ambientais em simulações apresenta elevado grau de precisão. A incorporação dessas em simuladores robóticos podem viabilizar a emulação de ambientes tão imprevisíveis quanto inundações, rompimento de barragens e tempestades.

2. *Existe um método, um benchmark ou experimento para realizar a validação de simulações de Unmanned Surface Vehicle?*

Espera-se identificar procedimentos para validar a precisão e acurácia do modelo de simulação proposto, bem como se a validação é realizada apenas em ambientes controlados ou também na presença de distúrbios naturais.

3. *Quais as ferramentas CFDs ou os modelos de simulação do estado da arte da mecânica de fluídos podem ser utilizados para simular correntes de água e vento?*

Por meio dessa pergunta, consultou-se especialistas na área de mecânica de fluídos para identificar uma ferramenta CFD gratuita, que atenda as necessidades de simulação para ambientes urbanos e litorâneos (costa, rios e lagos), já que nesses locais os distúrbios ambientais variam mais frequentemente ao longo do tempo e do espaço, do que nos espaços abertos (como em alto mar). Para a escolha das ferramentas CFDs, considerou-se as seguintes restrições nas escalas temporal e espacial das simulações. Do ponto de vista da escala espacial, deve ser capaz de simular um bairro e região de uma cidade. Com relação a escala temporal, a simulação terá duração entre poucos horas até mesmo alguns dias. Portanto, foram desconsiderados os métodos e ferramentas CFDs capazes de simular cidades e estados ao longo de diversas semanas.

De natureza aplicada, a presente pesquisa foi conduzida com uma abordagem mista, sendo que os procedimentos tiveram uma estratégia exploratória sequencial [Cre10], ou seja, os achados de um método de pesquisa são utilizados pelo método seguinte. Portanto, as análises realizadas por esses métodos qualitativos e quantitativos foram utilizadas para demonstrar que a hipótese descrita é válida. As etapas relacionadas a esta prova consistiram em:

1. A primeira etapa da pesquisa teve uma abordagem qualitativa, na qual os métodos utilizados foram a pesquisa bibliográfica e documental. A coleta de dados na pesquisa bibliográfica foi realizada através do método de *snowballing*, tendo sido identificados os simuladores robóticos existentes e as principais características necessárias para simular USVs. Consequentemente, apesar da existência de simuladores marítimos, eles não fazem parte do escopo da revisão, uma vez que todos os simuladores que não possuem integração com ambientes de desenvolvimento de software para a robótica foram desconsiderados. Assim, a revisão foi conduzida no início do doutoramento, sendo que os resultados são apresentados no Capítulo 2. Inicialmente, foi elaborada uma lista das principais características para uma simulação realista de USVs. Em seguida, foi verificada a existência dessas em cada simulador por meio da revisão dos seus artigos. Posteriormente, realizou-se uma pesquisa documental, na qual foram

coletados dados em seus manuais, da execução dos simuladores, da leitura e da interpretação de seus códigos fontes. A execução e leitura do código fonte foi necessária porque diversos artigos não relatam explicitamente se determinada característica existia e como ela era realizada.

2. Por meio da análise dos dados coletados da revisão bibliográfica e documental, foi escolhido o FreeFloating Gazebo para ser aperfeiçoado a fim de atender as principais características necessárias para simular USVs. Conforme descrito no Capítulo 3, foram adicionados distúrbios de ondas, correnteza de vento e água, bem como implementado modelos de leme, quilha e vela. O simulador é distribuído com diferentes tipos de USVs: diferencial (2 hélices), 1 hélice e 1 leme, airboat, 1 veleiro. A existência desses modelos de veículos USVs está relacionada com a proposta inicial de aperfeiçoamento do simulador, ou seja, provisionar diferentes modelos de USVs prontos para uso, e disponibilizar meios para que novos tipos de USVs fossem prontamente modelados apenas com os módulos existentes no simulador. Assim, foram concebidos e/ou aperfeiçoados módulos que facilitam a definição dos principais tipos de propulsão e controle de direção de USVs, tais como thrusters, lemes, quilhas, velas.
3. Para reproduzir os distúrbios em diferentes tipos de ambientes (urbano, lagos, rios e regiões costeiras), inicialmente cogitou-se simular utilizando o método LBM (Lattice Boltzmann methods). No entanto, ao consultar especialistas em mecânica de fluídos, foi possível identificar outros simuladores CFD que poderiam realizar a simulação viabilizando uma maior escalabilidade do ambiente e das velocidades dos fluídos a serem simulados. Assim, as ferramentas CFD gratuitas HEC-RAS, OpenFoam e Incompact3d foram apontadas para serem utilizadas nessa tese, tendo sido realizadas integrações com o HEC-RAS e o OpenFoam conforme a Seção 3.2.5. O simulador utilizado para reproduzir a corrente de água foi o HEC-RAS. O OpenFoam foi aplicado para reproduzir as correntes de vento. Dependendo da escala do ambiente, essas simulações podem demandar alto processamento durante muitas horas para serem executadas. Conseqüentemente, as simulações dos distúrbios ambientais foram conduzidas antecipadamente e seus resultados foram exportados para serem carregados e utilizados pelo simulador robótico proposto nessa tese.
4. Após o modelo de simulação ter sido implementado, fez-se uso do método experimental de pesquisa, que demandam a comparação entre dois grupos (de controle e experimental) [GS09], para validar a movimentação das embarcações virtuais. Nesse sentido, para o grupo de controle, escolheu-se os robôs reais a serem testados num lago com pouca influência de distúrbios de vento e de água. Para o grupo experimental, tem-se os robôs virtuais, realizando os mesmos testes no simulador USV_SIM. Além disso, o método de pesquisa experimental demanda que sejam identificadas as variáveis que influenciam o objeto de estudo e, ao manipulá-las nos grupos de controle

e experimental, pode-se comparar os efeitos produzidos [GS09]. Conseqüentemente, identificou-se as variáveis que influenciam a movimentação das embarcações, tendo sido conduzidos os testes descritos na Seção 4.1. Os testes com os robôs reais (grupo de controle) permitiram estimar os parâmetros cinemáticos de cada embarcação, o que foi utilizado na calibração dos robôs virtuais. Após essa configuração, repetiu-se os testes nos USVs virtuais (grupo experimental). Os dados gerados foram utilizados para realizar a comparação e validação da movimentação das embarcações virtuais conforme descrito na Seção 4.2.

5. Com os USVs calibrados, conduziu-se estudos de casos de forma a verificar como as características modeladas no simulador impactam na movimentação das embarcações. Além disso, esses estudos de casos demonstram que os efeitos descritos nos guias de navegação estão presentes nesse simulador robótico. Também são apresentados cenários em que é realizada uma inspeção de ponte e a navegação numa enchente em área urbana.

1.2 Contribuições

Ao longo do desenvolvimento dessa tese, foram realizadas publicações dos resultados obtidos, sendo elas: a revisão bibliográfica (Capítulo 2), o protocolo de criação de cenários (Seção 3.1), a arquitetura do modelo de simulação USV_SIM (Seção 3.2), as três primeiras etapas do método de validação dos USVs das Seções 4.1.1, 4.1.2, 4.1.3 e 4.2.1. Essas publicações foram realizadas na conferência ICINCO 2018 [PJA18] e no periódico Sensors [PSJ+19].

Dentre as contribuições realizadas, destacam-se o levantamento com as características necessárias para a simulação de USVs e o nível de implementação dessas em cada simulador analisado (Capítulo 2). Tem-se também a criação do modelo de simulação e o simulador robótico USV_SIM capaz de reproduzir a flutuabilidade sobre ondas de embarcações e não apenas sobre um plano 2D (Seção 3.2.3). Foi disponibilizado quatro tipos diferentes de modelos de USVs integrados ao simulador (Seção 3.2.4), bem como módulos que agilizam a definição dos tipos de propulsão e de direção dos USVs (Seção 3.2.2).

Ainda como contribuição, foi desenvolvido um fluxo para modelar ambientes reais, os quais são utilizados nos simuladores robóticos e como entrada para as ferramentas CFD (Seção 3.1.1). Realizou-se a construção dos modelos 3D de cenários de partes da cidade de Porto Alegre integrados ao simulador Gazebo e construídos a partir de Modelo Digital de Terreno (MDT) precisos (Seção 3.1.1), assim como a integração de módulos capazes de ler as velocidades das correntes de vento e de água geradas pelos simuladores HEC-RAS e OpenFoam. Essas velocidades são transformadas em forças a serem aplicadas nas embarcações conforme apresentado na Seção 3.2.5. Há também o protocolo de validação de

modelos de USVs virtuais (Seção 4.1), os resultados dessa validação para dois tipos de modelos de USVs (Seção 4.2). E os estudos de casos para diferentes cenários, apresentando os efeitos do modelo de simulação sobre a movimentação das embarcações (Capítulo 5).

2. REVISÃO BIBLIOGRÁFICA E DOCUMENTAL

Existem diversas revisões na literatura com relação a simulação computadorizada de sistemas robóticos autônomos. Dentre essas, Craighead [CMBG07] propõe a classificação dos sistemas em 4 critérios: fidelidade física, fidelidade funcional, facilidade de desenvolvimento, custo. Os autores concluem que o desenvolvimento de novos simuladores, não é necessário. Ao invés, eles propõe que devam ser ampliadas as capacidades e funcionalidades dos simuladores robóticos existentes, assim o compartilhamento dessas expansões contribuem com a comunidade de Robótica e Automação. Por outro lado, Harris e Conrad [HC11] complementa que também devem ser considerados como fatores de avaliação: a diversidade de sensores, a acurácia gráfica e física, a capacidade multiplataforma, o código ser aberto (*open-source*).

Já Torreti [TTACP14] ressalta a importância de comparar as tarefas em ambientes virtuais e no mundo real para a avaliação dos simuladores disponíveis. Nesses trabalhos, os autores concluem que o número de simuladores que oferecem suporte a simulação a USVs é reduzido em comparação com outros tipos de plataformas robóticas, tais como UAVs e UGVs.

Na Tabela 2.1, é apresentada a comparação entre as características selecionadas neste projeto para avaliar simuladores de código aberto que suportam USVs. Assim, foram definidos os seguintes critérios para classificar as características apresentadas na Tabela 2.1:

- **Disponibilidade:** informa se o código fonte está disponível para download pela comunidade científica e se ainda está operacional.
- **Estado:** informa se o projeto continua sendo atualizado, se está ativo mas sem atualização recente, ou se foi descontinuado.
- **Ondas:** indica se simula ondas com propósito de visualização, ou se simula ondas de forma a realizar distúrbios de ondas na *pose* dos USVs.
- **Boiar:** o princípio de Arquimedes é aplicada para realizar a flutuação dos USVs sobre as ondas. Existem três possibilidades: sem boiar (o barco desloca-se sobre um plano rígido e as ondas não tem influência no barco), boiar simplificado (realiza movimento vertical rígido de toda embarcação, sem rotações), boiar realístico (realiza o movimento do barco considerando o formato das ondas, ou seja, aplicando rotações de *pitch* e *roll*).
- **Corrente de água:** descreve o método utilizado para aplicar as forças da correnteza de água sobre a embarcação: não aplica nenhuma força; uma força constante ao longo do tempo e espaço; ou uma força variável ao longo do tempo e do espaço.

- **Corrente de vento:** similar ao item anterior, mas levando em conta forças do vento sendo aplicados apenas aos objetos acima da linha da água.
- **Propulsor:** informa como a força de propulsão são simuladas: não é simulada; uma função linear calcula o percentual da força máxima a ser aplicada pelo propulsor; simula efeitos dinâmicos das hélices do propulsor.
- **Lemes:** define como a força do leme é simulada: não é simulada; emulada ao aplicar um percentual linear da força máxima; simula os efeitos dinâmicos do formato do leme.
- **Velas:** similar ao item anterior, mas com relação as forças do vento sendo aplicadas no corpo vela.
- **Vento no casco:** informa como a força do vento é aplicada no casco do USV: não aplica; aplica um valor proporcional ao existente; ou calcula.

A Tabela 2.1 apresenta quatro níveis de avaliação para cada critério significando:

? indica que a informação não está disponível em publicações ou na documentação;

× indica que funcionalidade não é suportada;

√ funcionalidade é parcialmente suportada

√√ informa que a característica é totalmente suportada e está prontamente disponível para ser utilizada pelo usuário.

Além dos simuladores robóticos apresentados na Tabela 2.1, alguns foram removidos por apresentarem características não desejáveis, se enquadrando nos critérios de exclusão dessa pesquisa. Esses critérios foram inspirados na classificação de Craighead [CMBG07] (fidelidade física, fidelidade funcional, facilidade de desenvolvimento, custo). O primeiro critério definido nesta pesquisa bibliográfica e documental tem relação com a fidelidade física, ou seja, todo aquele simulador, que não faz uso das forças para estimar a movimentação de USVs, será excluído. Relacionado ao custo, todo o software que não for open-source e gratuito não será analisado. O último critério utilizado tem relação com a facilidade de desenvolvimento: todo o simulador que não seja modularizado, dificulte a reutilização de código, que impeça ou limite a escalabilidade. Nesse último se enquadram todas aquelas *scripts* desenvolvidos para reproduzir o movimento de uma única embarcação, sendo muitas vezes implementado em Matlab ou Python.

Um dos simuladores excluídos da análise é o Player/Stage [GVH03], o qual simula robôs em ambientes 2D sem aplicar forças físicas. Esse simulador tem como principal foco a performance e não a precisão, o que permite a simulação de enxames de robôs. Outro removido foi o WaveSim [San07], o qual representa USVs e UUVs apenas com primitivas

Tabela 2.1: Comparação entre simuladores de USVs com relação a característica desejáveis. Critério ? significa informação não disponível em publicações e na documentação. × significa características não suportada. ✓ significa parcialmente suportada ou limitação significativa. ✓✓ significa totalmente suportada e prontamente disponível para o usuário.

Simulador	Disponibilidade	Estado	Ondas	Boiar	Corrente de água	Corrente de vento	Propulsor abaixo da água	Propulsor acima da água	Leme	Vela
Kelpie	×	?	✓	✓	✓	×	?	?	×	×
UWSim	✓✓	✓✓	✓	✓	×	✓	✓	×	×	×
UW-Morse	×	✓✓	✓	✓	?	?	✓	?	×	×
Freefloating gazebo	✓✓	✓✓	✓	✓	✓	×	✓✓	✓✓	×	×
UsarSim	✓	×	?	?	?	?	✓	?	✓	×
VREP	✓✓	✓✓	✓	✓	✓	×	✓✓	✓✓	×	×
MARS	✓✓	×	✓✓	✓	✓✓	×	✓	✓	×	×
RobotX	✓✓	✓✓	✓	✓✓	×	✓	✓✓	✓✓	×	×
USV_SIM	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓

geométricas básicas (como caixas e cilindros), não permitindo a representação de *kinematic chains* e embarcações mais realistas.

MORSE [ELDL11] é um simulador baseado no Blender Game Engine [Fou20a] e na física da Bullet physics Engine [Bul20]. O simulador MORSE tem sido distribuído com os seguintes tipos de robôs [HSG16]: Unmanned Ground Vehicles (UGV), Unmanned Aerial Vehicle (UAV) e um submarino genérico. No entanto, este modelo de submarino não possui nenhuma física ou massa relacionada, ou seja, o usuário deverá implementar toda a dinâmica. Devido a essa limitação em simular UUVs e USVs, ele foi não incluído na Tabela 2.1.

UsarSim tem sido utilizado para simular UUV e UGV, tendo se tornado popular para simular times de robôs para resgate, já que ele facilita a definição de novos ambientes. Em [SC10], USarSim foi estendido com um modelo de propagação para simular diferentes aspectos de um canal acústico embaixo da água. Contudo, o código fonte do UsarSim aparentou estar desatualizado no momento da revisão, já que não foi possível compilá-lo com a atual versão do Windows (Versão 10) e com as dependências atualizadas, tal como Unreal Development Kit 4 (UDK) [Gam20]. Infelizmente, UsarSim requer a versão antiga da UDK (versão 2), a qual foi descontinuada e não está mais disponível para download no site oficial. Contudo, foram conduzidos testes com a UDK versão 3 (também descontinuada) e UDK versão 4 (versão estável), em ambas tentativas de compilação falharam com erros.

Já o framework *Mission Oriented Operating Suite and Interval Programming* (MOOS-IVP) [BSNL10] combina duas arquiteturas diferentes para realizar a simulação: um *mid-*

dleware utilizado para conectar diversas aplicações; um modelo de programação para otimização de multi-objetivos. Os módulos MOOS-IVP também inclui um simulador simplificado chamado iMarineSim. Esse simulador é utilizado principalmente para atualizar os estados dos veículos, baseado na atuação dos propulsores, sendo que as forças de perturbação podem ser incorporados em modelos de controle, mas (de maneira similar ao Player/Stage), iMarineSim não foi incluído na Tabela 2.1 por não simular as forças físicas nos robôs.

Autonomous Marine Surface Vessel Simulator [SST⁺16] é um simulador desenvolvido pelo time da Universidade de Queensland University of Technology para participar na Maritime RobotX Challenge de 2016. Este simulador foi concebido para modelar de maneira eficiente e precisa o USV utilizado na competição, já que outros simuladores (Gazebo, V-REP e UWSim) não atendiam os requisitos necessários. Contudo, este simulador aparenta não estar disponível na Internet para a comunidade científica.

Com relação aos simuladores listados na Tabela 2.1, o primeiro a ser incluído é o Kelpie [MSM⁺13], o qual possui a funcionalidade de boiar, simular ondas e correntes de água. Esse simulador oferece suporte a ROS além de boas capacidades de renderização e por afirmar que é software livre. Contudo, não encontra-se mais disponível online no momento da revisão. O principal foco é oferecer suporte a robôs terrestres e subaquáticos. Apesar do Kelpie renderizar ondas oceânicas, mas não é possível qualificar e compreender como a flutuabilidade é alcançada, tendo em vista que a documentação não informa nesses detalhes e o código fonte não estar disponível. Além disso, as forças geradas pelas correntes de água são constantes para todo o cenário, ou seja, elas não mudam ao longo do tempo e do espaço. Os autores do simulador Kelpie propõe como trabalho futuro o desenvolvimento de uma extensão de forma a simular veleiros, o que demandará a definição de modelos de simulação para correntes de vento. Assim, conclui-se que o simulador carece de funcionalidades de corrente de vento e suporte a velas e a lemes.

UWSim [PPFS12] apresenta um novo nível de realismo ao fazer uso de biblioteca de renderização OpenSceneGraph 4 e osgOcean 5. UWSim foi concebido para ser modular e facilmente ampliável, assim ele funcionaria principalmente como uma visualização das saídas de módulos externos (e.g. Gazebo), os quais seriam responsáveis pelos algoritmos de controle, dinâmicas de veículos do ambiente. UWSim é um simulador open-source que é distribuído com um modelo 3D do submarino Girona 500 (o qual inclui um manipulador ARM5). A dinâmica desse UUV é implementada por um único script em python, assim a dinâmica de outros veículos tem que ser totalmente modelados "do zero", já que a falta de design modular dificulta a reutilização de dinâmica para diferentes robôs. No UWSim, os robôs são descritos por arquivos de formato URDF, enquanto que os cenários são descritos no formato XML. Além disso, ele oferece interface com o MatLab via (`ipc_bridge`) e para um programa de controle externo por meio do ROS. UWSim facilita a configuração de tópicos ROS por permitir que usuários escrevam marcações XML nos arquivos de cenário.

No exemplo do UUV Girona 500, distribuído junto ao UWSim, o efeito de boiar é aplicado até o submarino alcançar o nível de superfície definido pelo plano em $Z=0$, portanto as ondas existentes no UWSim não afetam as forças de boiar. Outra limitação é a inexistência de simulação de correntes de água, enquanto que as correntes de vento aparentam afetar apenas a forma e o tamanho das ondas oceânicas, mas não o movimento do veículo. Como o UWSim foi concebido para robótica subaquática, ele não foi projetado para viabilizar que a dinâmica de embarcações sejam modeladas de maneira a considerar a superfície da água contendo ondas. Conseqüentemente, a principal limitação para utilizar barcos inclui o modelo simplificado do efeito de boiar, falta da dinâmica para simulação de lemes, velas, corrente de água e vento.

UW-Morse [HSG16] é uma extensão que permitiu a simulação de robôs subaquáticos com o simulador Morse [ELDL11]. Aquele inclui efeitos hidrodinâmicos e sensores subaquáticos. Ambas são desenvolvidas sobre a Blender Game Engine, o que permite a renderização com alta qualidade, enquanto que a física é desempenhada pela biblioteca Bullet. O projeto Morse é open-source e encontra-se disponível online, contudo UW-Morse não havia uma versão de distribuição pública. Assim, não foi possível saber se o efeito de boiar de veículos aquáticos era afetado pelo formato e tamanho das ondas. Além disso, os autores não mencionam a existência de dinâmica para modelar as forças da correnteza da água e do vento de forma afetar o movimento dos veículos.

Gazebo [KH04] é provavelmente, dentre os simuladores, o software de simulação mais popular para sistemas autônomos. Ele é capaz de simular ambientes 3D complexos por meio da biblioteca gráfica OGRE (Object-Oriented Graphics Rendering Engine) e oferece suporte a diversos motores de simulação física, tais como, ODE, Bullet, Simbody e DART. A sua estrutura modular permite a simulação de diferentes ambientes e modelos robóticos reutilizando descrições XMLs e estendendo as funcionalidades principais do Gazebo por meio de plugins. Durante a revisão bibliográfica, maior parte dos esforços do desenvolvimento do Gazebo foram em UGVs e UAVs. Poucos trabalhos tem introduzido novas funcionalidades capazes viabilizar a simulação de USVs utilizando o gazebo, sendo uma das poucas existentes conhecida como FreeFloating Gazebo [Ker14].

O FreeFloating funciona como um plugin do Gazebo, sendo que integra o simulador Gazebo com o UWSim por tópicos do ROS. Nessa integração, o Gazebo fica responsável por calcular a simulação física, enquanto que o UWSim realiza a visualização com maior realismo e a simulação de ondas oceânicas. Assim, esse plugin aplica forças nos links em que os propulsores estão acoplados. Ainda em relação ao Gazebo, é possível simular os propulsores com maior precisão, para tal o usuário pode simular a atuação dos efeitos aerodinâmicos das hélices dos propulsores. Isso pode ser realizado por meio do uso de um plugin do Gazebo chamado LiftDrag. Contudo, tanto o Freefloating Gazebo quanto o Gazebo carecem de modelos capazes de implementar veleiros, ou seja, lemes, velas, quilhas, cordas, roldanas.

O V-REP [RSF13] é um simulador muito utilizado na área da Robótica, sendo que ele oferece suporte a addons, plugins, comunicação via sockets, integração com o ROS e criação de script LUA [PUC20]. Esse simulador é capaz de utilizar simulação de particular para emular jatos de ar e água provenientes de motores a jato e propulsores com hélices. Contudo, observa-se que o VREP ainda carece de suporte a simulação de UMVs. A maioria dos veículos disponíveis para o VREP são terrestres e aéreos, sendo que existe uma implementação de um USV diferencial [Edm20], na qual a representação visual de ondas é feita de maneira simplificada, já que ela é implementada apenas como uma textura 2D. Contudo apenas quando o veículo está em movimento que algum nível de perturbação das ondas é percebido na embarcação. Ademais, ele não sofre influência de correntes de água e de vento, e tão pouco tem-se conhecimento de plugins ou addons disponíveis que permitam a simulação de veleiros.

MARS [TM14] é um ambiente de simulação criado para múltiplos AUVs e USVs, sendo que faz uso da jMonkeyEngine 3D-Engine para gerar gráficos subaquáticos com alto grau de realismo, simula diferentes tipos de atuadores e sensores, destacando-se a existência de sensores. Além disso, ele permite a simulação de correntes de água, ondas, calculo automático do centro de flutuação e do volume da embarcação, representação de thrusters. Apesar de emular realisticamente USVs e UUVs e do código fonte estar disponível na internet, o projeto foi descontinuado, o que restringe sua utilização e expansão futura. Além disso, o simulador não oferece suporte a dinâmica de velas, lemes e quilhas.

Ao concluir a revisão dos ambientes de simulação em 2017, observou-se a ausência de um simulador capaz de reproduzir o comportamento de USVs frente a distúrbios ambientais. Baseado no levantamento realizado nessa revisão e considerando a orientação de Craighead [CMBG07] que deve-se aperfeiçoar simuladores existentes, optou-se por reutilizar as seguintes ferramentas: Gazebo, FreeFloating Plugin e UWSim. Assim, passou-se a aprimorar os modelo de ondas, ventos e correntes de água, sendo que em 2018, com a integração da ferramenta CFD HEC-RAS, foi possível realizar a primeira simulação de desastre natural, em que foi modelada uma enchente numa área urbana na cidade de Porto Alegre.

Ainda em 2018, após aos aprimoramentos acima descritos, foi anunciada que a Robotic Challenge incluiria uma etapa de simulação, em que a Open Source Robotics Foundation [Fou20c] estaria desenvolvendo novos plugins para aperfeiçoar o suporte a USVs e distúrbios ambientais. O projeto em questão chama-se VMRC, sendo que ao fazer a análise do código fonte e documentação existente, percebeu-se que o foco desse plugin é aperfeiçoar a interação do USV com ondas, permitindo que ocorram rotações como pitch e yaw. Além disso, o vento é modelado como um vetor constante, que não se altera ao longo do espaço e do tempo. A ação do vento no casco da embarcação provoca o arrasto dessa.

Dentre as limitações identificadas no projeto VMRC, percebe-se que existem dois modelos de ondas: um com caráter apenas visual (rendering), outro exclusivo para cada

embarcação. Portanto, o movimento do USV não acompanha o movimento das ondas renderizadas no simulador, bem como, ao simular dois USVs, cada um terá o seu modelo de ondas em separado, ou seja, mesmo estando muito próximos não serão afetados pela mesma onda. Outra restrição identificada é que o plugin não oferece suporte a correntes de água. Além disso o roadmap do projeto do VMRC não contempla esse distúrbio, já que a Robotic Challenge foi realizada num local em que as correntes de água quase não estão presentes.

Posterior a publicação do modelo de simulação na Revista Sensors, o código fonte do simulador UW-MORSE [HSG16] foi disponibilizado na internet. Conforme especificado no próprio repositório do simulador, não existe documentação e nem suporte para a sua execução. Além disso, os passos de instalação e compilação é escasso e superficial, o que justifica a impossibilidade de superar os erros de compilação observadas durante a análise do UW-MORSE. Essas limitações e dificuldades encontradas inviabilizaram a análise desse simulador.

Do ponto de vista de integração de UUVs e USVs, o Freefloating Gazebo oferece um rendering do ambiente subaquático com maior realismo do que observado no Gazebo e conseqüentemente no plugin VMRC. Isso se deve ao fato de que a proposta original do FreeFloating Gazebo é oferecer suporte a simulação de UUVs, ou seja, foram adicionadas efeitos gráficos como turbidez da água, reflexão e refração na superfície da água, meios de alterar a cor da água, partículas de lodo, *god rays*.

3. SIMULADOR

Nesse capítulo, são apresentados os detalhes de como o USV_SIM está estruturado e como é realizada a criação dos dados de entrada para simulação em ambientes mais realistas. Para tal, descreve-se a criação de cenários tridimensionais (3D), a simulação de modelos aero-hidrológicos baseados no estado da arte em ferramentas de CFD (sugerida por especialistas da área), bem como as adaptações e aperfeiçoamentos realizados no simulador já existente.

3.1 Modelo de cenários

No contexto dessa tese, a reprodução de ambientes reais em simulação permite a comparação do robô real com o modelo do robô simulado. Desta forma, criou-se terrenos 3D, com o intuito de reproduzir ambientes reais em simulação, contendo os principais tipos de obstáculos que as embarcações irão se deparar (edificações e pontes), bem como computar os principais distúrbios ambientais (vento e correnteza de água) que afetam os USVs.

Na Figura 3.1, é apresentado o fluxograma para modelar os cenários a serem simulados. A explicação detalhada sobre cada etapa, processo e conjunto de dados gerados, é apresentada nas seções a seguir.

3.1.1 Modelando terrenos 3D

Como primeira etapa para recriar cenários reais em modelos geométricos tridimensionais virtuais, deve-se fazer a aquisição do Modelo Digital de Terreno (MDT). O MDT contém informações geográficas de localização e altura para cada ponto da área desejada, um exemplo é apresentado na Figura 3.4a. Combinando o MDT com a textura do terreno (foto geoespacial), pode-se, por meio de softwares específicos, recriar o terreno real no mundo virtual. No caso dessa tese, utilizou-se a ferramenta *open-source* VirtualPlanetBuilder [Gra20] para exportar terrenos de larga escala de imagens geoespaciais e de mapas digitais de terreno.

Para reproduzir a geometria das edificações em 3D, realizou-se a aquisição de informações das edificações por meio do OpenStreetMaps [Fou20b]. Essa plataforma permite que altura, forma, localização de edificações, de ruas, etc, sejam exportadas por meio de arquivos de extensão OSM. O conteúdo desses arquivos OSM é descrito em XML (Extensible Markup Language), podendo ser carregados em diversos aplicativos para o seu

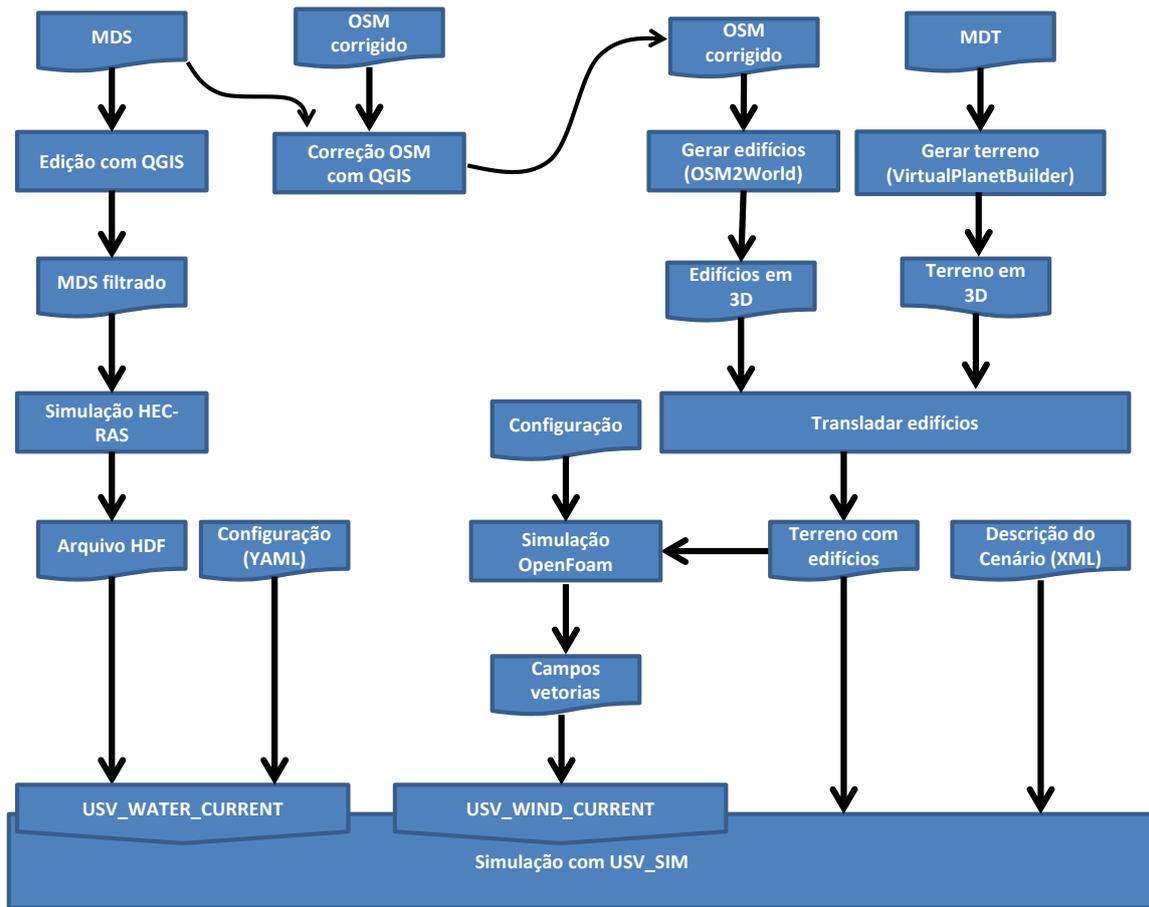


Figura 3.1: Fluxograma com as etapas para modelar terrenos e simular as correntes de vento e água.

conteúdo ser editado. Essa alteração pode ser manual ou automatizada, dependendo da aplicação utilizada. Neste projeto, utilizou-se a informação de um Modelo Digital de Superfície (MDS) para complementar e corrigir as alturas incorretas de algumas edificações. O arquivo OSM corrigido pode ser carregado na ferramenta OSM2World [OSM20] que transforma o seu conteúdo em geometria 3D. A grande limitação dessa ferramenta é que ela não conhece a topografia do terreno, ou seja, ela gera a geometria das edificações sobre um plano (veja Figura 3.2).

Assim, para combinar o terreno 3D com a geometria 3D dos edifícios, fez-se necessário a criação de um software capaz de transladar cada construção sobre a superfície do terreno. Esse software busca os vértices inferiores das edificações, e para cada um deles, o software cria uma reta na vertical a fim de localizar a intersecção com o terreno, descobrindo assim a altura correta para a edificação. Unindo o terreno 3D com as edificações das Figura 3.2, obtém-se o resultado observado na Figura 3.3.

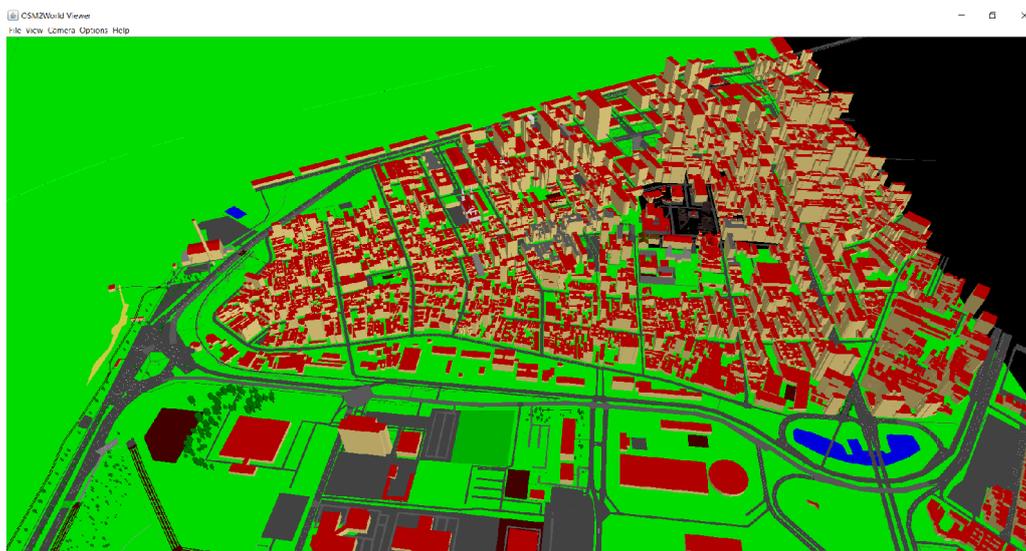


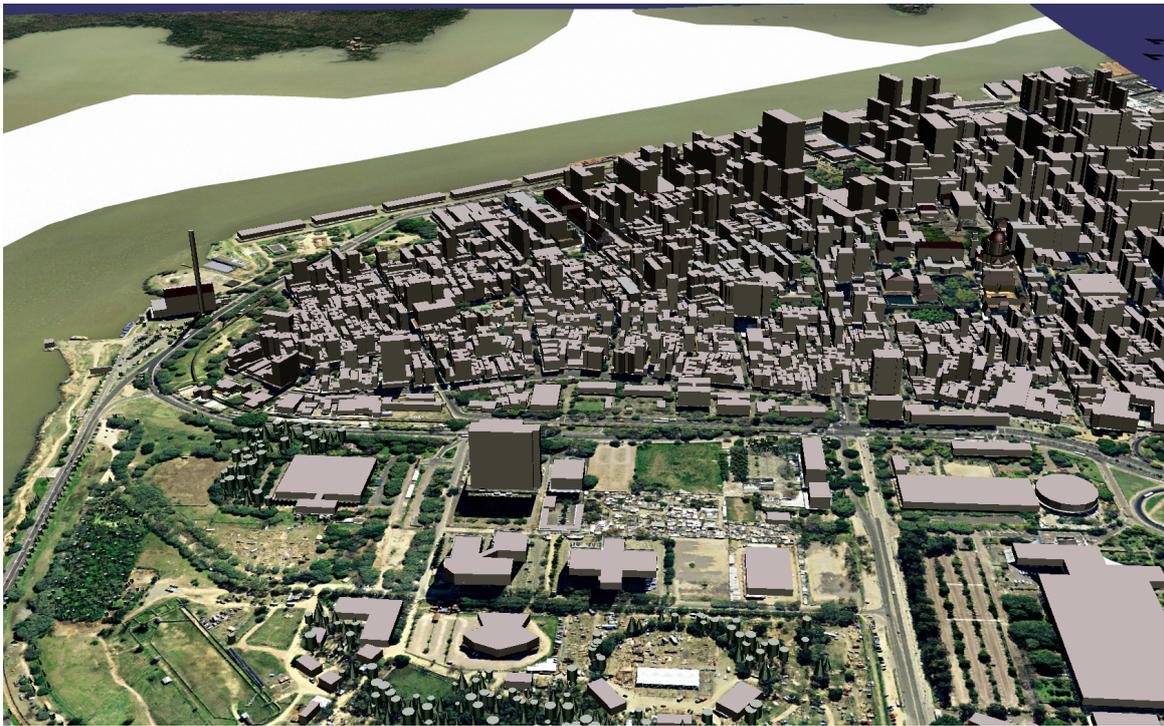
Figura 3.2: Geometria das edificações gerada pela ferramenta OSM2World

3.1.2 Modelando correnteza de água e de vento

Para simular o fluxo de rios utilizando os modelos hidrológicos, deve-se considerar o impacto direto da topologia do terreno no fluxo da água. Assim, o primeiro passo deve ser a aquisição do Modelo Digital de Superfície (MDS), ou seja, um modelo que incorpora a superfície do terreno e o formato de altura das edificações. Existem alguns repositórios na internet que permitem o download desse tipo de dados, como por exemplo EarthExplorer [USG20]. Em alguns casos, o MDS não inclui a batimetria de rios e lagos, o que impede que a profundidade e rotas seguras de USVs e barcos sejam conhecidas nessas regiões. Na Figura 3.4, é apresentado o MDT (a) e o MDS (b) da região central de Porto Alegre, RS, Brasil.

Eventualmente, algumas características dos MDS podem ser ignorados, como por exemplo, na existência de pequenas árvores em mapas com alta resolução. Contudo, isso acaba por demandar que o usuário tenha que editar e filtrar esses elementos não desejáveis. Para realizar essa tarefa, pode-se utilizar ferramentas de edição geográficas tais como o open-source QGIS [QGI20].

O modelo MDS resultante é utilizado como entrada para o simulador hidrológico HEC-RAS, o qual é responsável por simular o fluxo de água ao longo do terreno. Para tal, é necessário definir uma grade 2D ao longo do terreno (Figura 3.5) e a localização de entrada e saída do fluxo de água do rio. Posteriormente, o usuário deve definir a altura inicial da superfície da água, bem como o volume desta (em m^3) para cada instante de tempo. Com essa configuração, pode-se variar o fluxo de água ao longo do tempo. A última etapa é realizar a simulação e exportar um arquivo HDF (Hierarchical Data Format), o qual será



(a) Vista superior do modelo geométrico do centro de Porto Alegre.

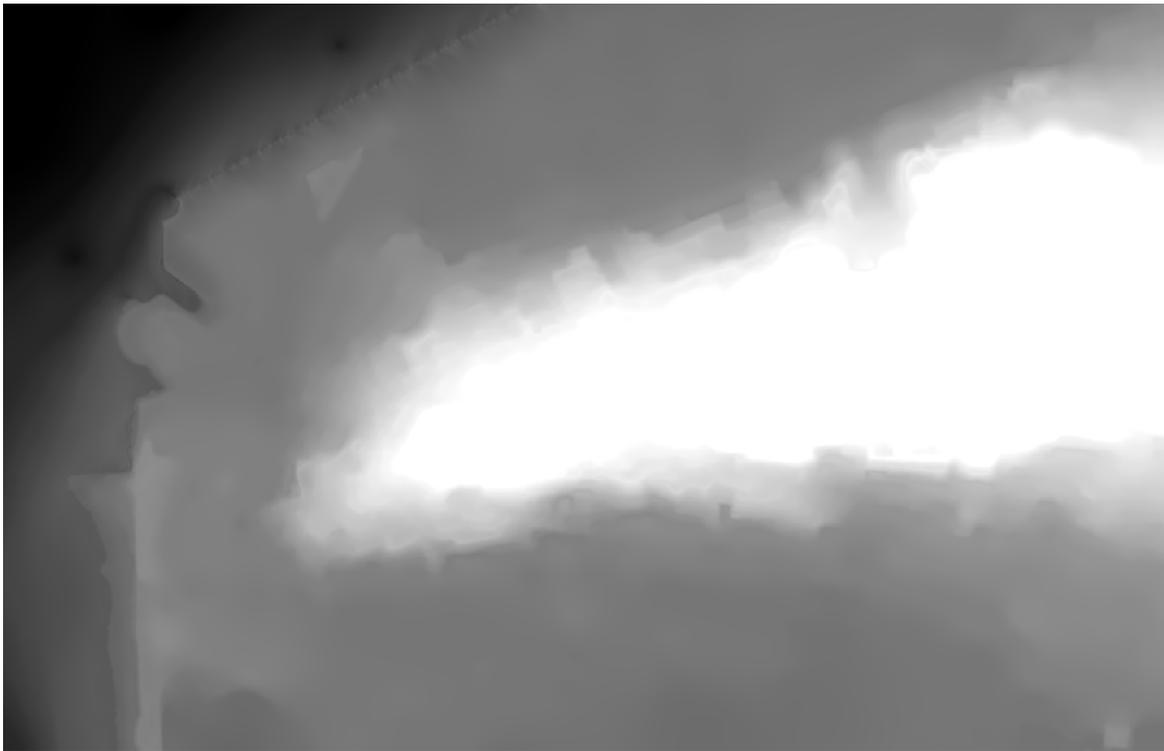


(b) Vista de topo do modelo geométrico do centro de Porto Alegre

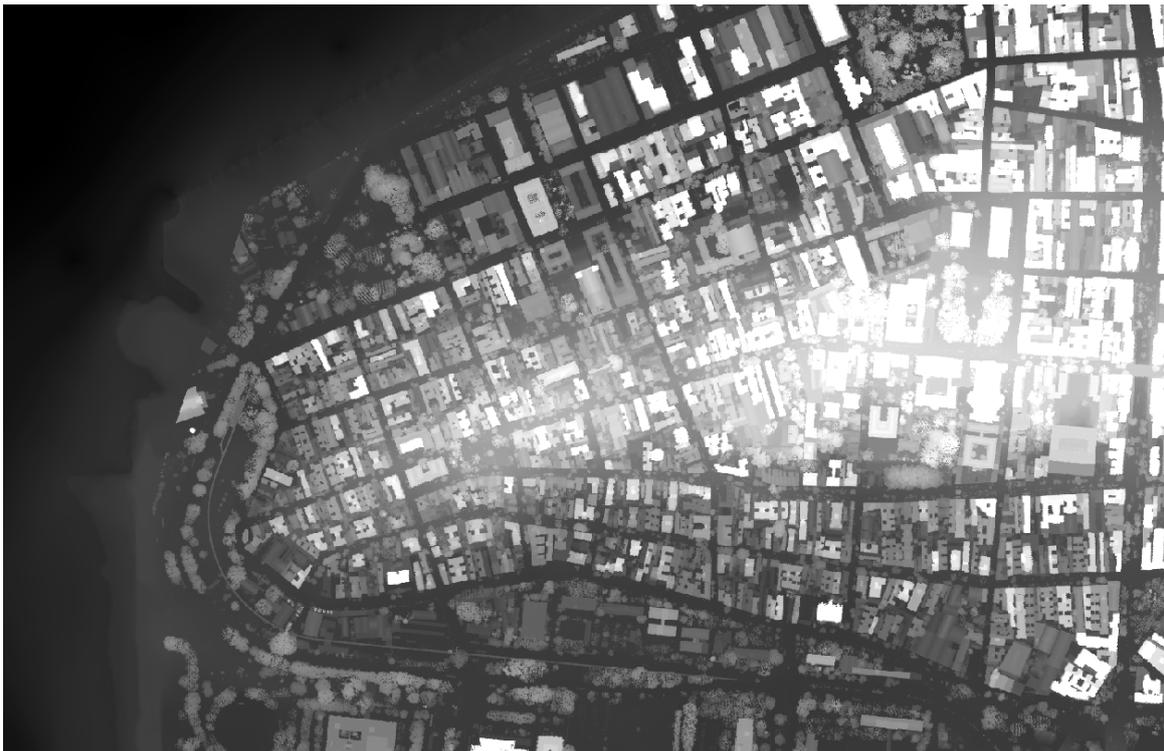
Figura 3.3: Terreno 3D com as construções transladas sobre a superfície. A geometria 3D reproduz o centro de Porto Alegre, RS, Brasil.

utilizado na configuração de entrada (arquivo YAML) do módulo de corrente de água do simulador.

A região inundada, que foi simulada, pode ser vista nas imagens da Figura 3.6, sendo que o fluxo de água é proveniente do lado direito da imagem. Essa região é loca-



(a) Modelo digital de Terreno (MDT).



(b) Modelo Digital de Superfície (MDS).

Figura 3.4: Modelos digital de terrenos e superfície da região central de Porto Alegre, RS, Brasil. A altura é representada em escala de cinza, branco para mais alto e preto para a mais baixa.

lizada no encontro do Arroio Dilúvio com o lago Guaíba, em Porto Alegre. A Figura 3.7 apresenta a direção e velocidade do fluxo de água por meio de pequenas partículas bran-

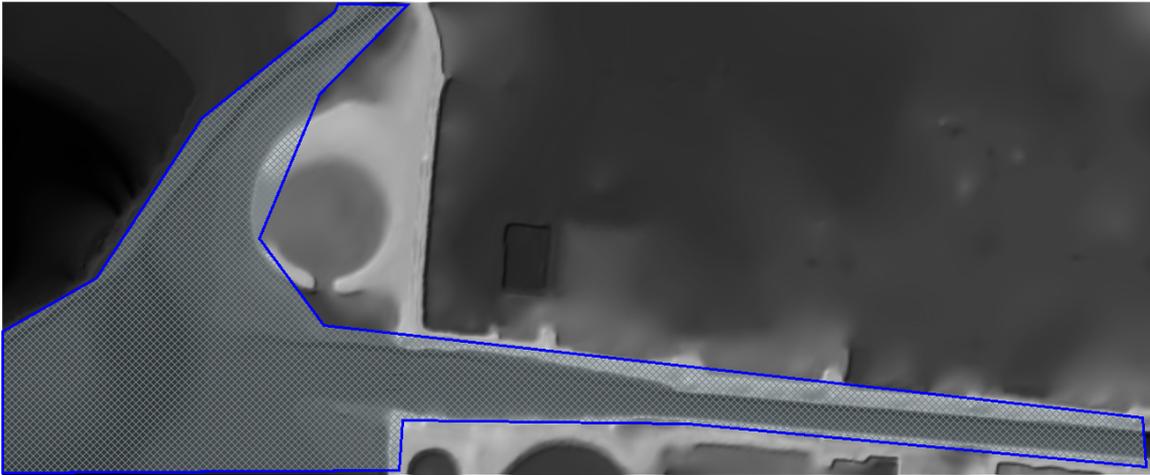


Figura 3.5: Grade 2D utilizada na simulação do HEC-RAS. Bordas da grade em azul.

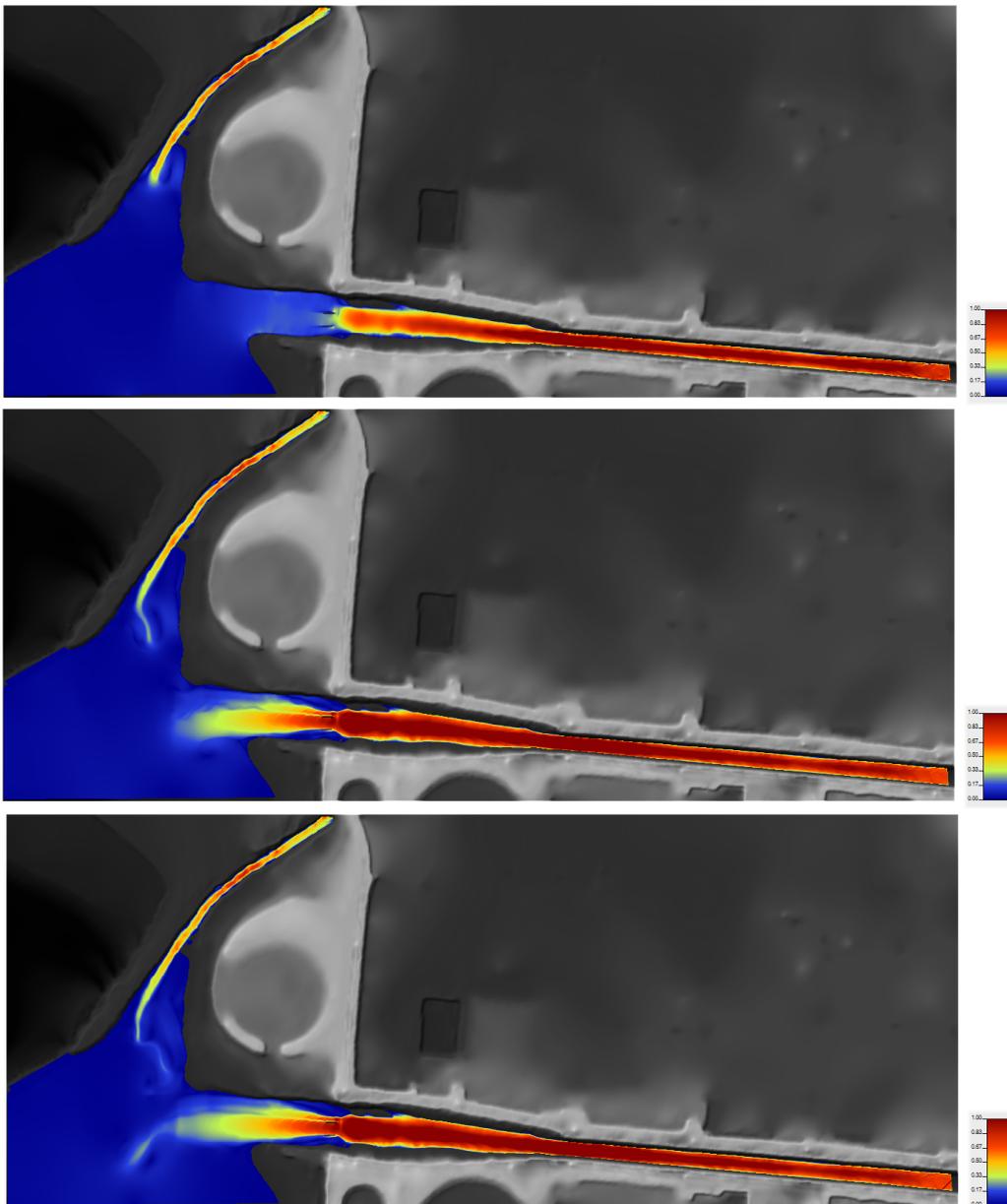


Figura 3.6: Simulação da correnteza de água ao longo tempo no arroio Dilúvio.

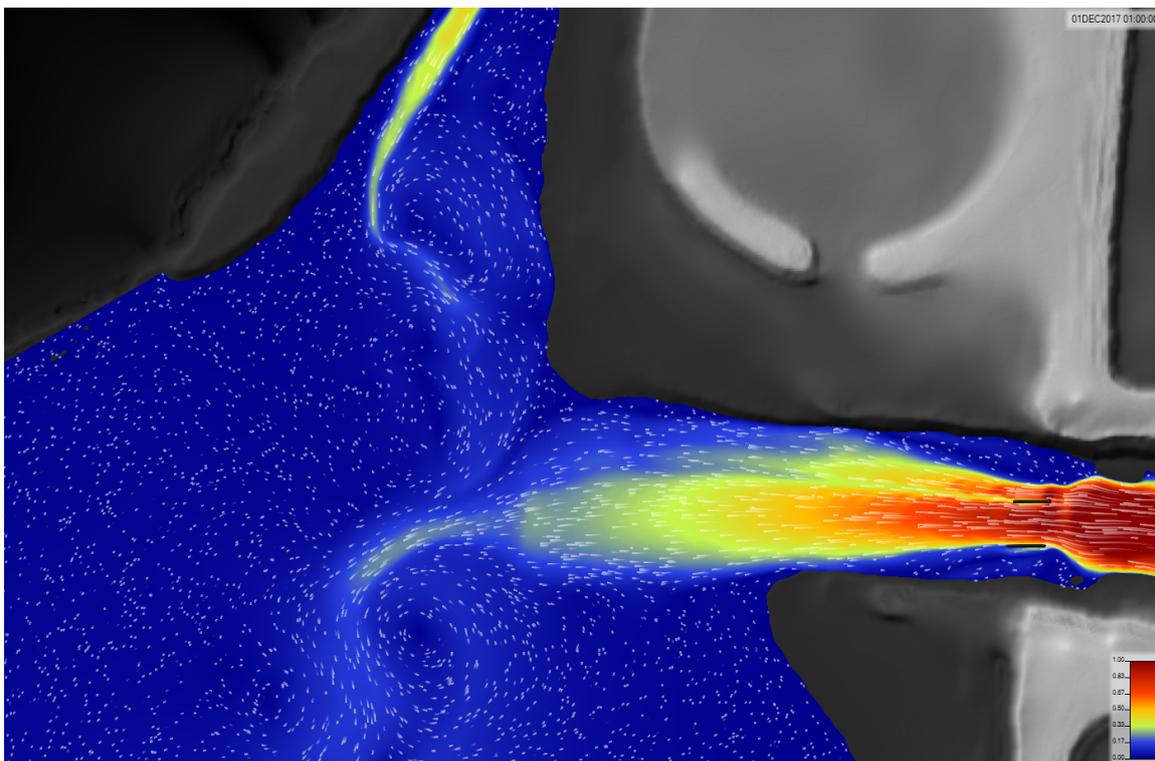


Figura 3.7: Simulação da correnteza de água, sendo que a velocidade da água pode ser observada por partículas brancas. Quanto mais longa a partícula, maior a velocidade no local.

cas, assim quanto maior a trajetória deixada pela partícula, maior é a velocidade da água naquele local. Além disso, observa-se a formação de vórtices devido ao movimento turbulento do fluxo de água.

Para a realização dessa simulação no HEC-RAS (versão 5.0.3), a região inundada foi definida por uma grade em que cada célula tem 1 metro de largura por 1 metro de altura (Figura 3.5). Como condição inicial tem-se a definição de um nível de água em 1 metro de altura. Enquanto que foram definidas três condições de contorno, sendo a primeira relacionada com a saída de água em direção ao Guaíba (região inferior esquerda na Figura 3.5), ou seja, o nível de água no estuário do Arroio Dilúvio com o Guaíba foi definido em 1 metro de altura. A segunda condição de contorno foi posicionada no topo superior da imagem de forma a simular o fluxo de água de um pequeno canal, foi configurado uma vazão de $5 \text{ m}^3/\text{s}$ de água. Já a terceira condição de contorno representa a entrada de água do próprio Arroio Dilúvio (porção inferior direita na Figura 3.5). O fluxo de água foi configurado em $20 \text{ m}^3/\text{s}$ de água. Além disso, foi utilizado o conjunto de equações de Full Momentum [Bru18], tendo sido simulada 1 hora de escoamento com intervalo de calculado de 30 segundos com número máximo de 20 iterações.

Já a criação dos mapas de vento são realizados pela ferramenta de fluidodinâmica OpenFoam, a qual consegue computar problemas de mecânica de fluídos. Portanto, a OpenFoam é uma biblioteca em C++ que realiza a simulação de escoamento por meio de

três etapas: pré-processamento, solução e pós-processamento. O pré-processamento tem como principal objetivo discretizar o modelo geométrico em estudo, numa malha de volumes otimizados para a realização do escoamento do fluido. Assim, a malha é refinada para que a forma e o tamanho do volume minimize o erro e melhore o desempenho do processamento do escoamento. A etapa de solução permite que um algoritmo realize o cálculo de escoamento, sendo que o resultado é analisado e observado por meio da ferramenta paraFoam (pós-processamento). Essa ferramenta possui diversas formas de visualização e permite que o resultado da simulação seja exportado em arquivos, os quais são utilizados como entrada na simulação robótica.

Para a simulação no OpenFoam (versão 5.4) apresentada na Figura 3.8 e utilizada nos estudos de casos do Capítulo 5, o cenário foi subdividido em cubos de 10 metros de lado, ou seja, o cenário de 1350 metros de comprimento, 560 metros de largura e 80 metros de altura, foi subdividido em 135 por 56 por 8 cubos. Essa geometria é posteriormente processada com a ferramenta blockMeshDict, a qual faz decomposição desse cubos numa malha hexahedral, ou seja, decompõe os cubos de maneira a melhorar a precisão da subdivisão considerando as informações proveniente do terreno e das edificações existentes na simulação. Além disso, foi utilizado como modelo de equações de Gauss Seidel Smoother e de Symmetric Gauss Seidel Smoother [Ope20b] para calcular a densidade e velocidade do fluido durante a simulação. Como condição de contorno, foi definido um fluxo de ar constante de 3 m/s proveniente da esquerda para direita na Figura 3.8.

A próxima seção descreve a arquitetura de simulação proposta e as principais contribuições realizadas dado o conjunto de recursos pré-existentes e descritos no Capítulo 2.

3.2 Arquitetura

A representação da arquitetura do sistema é apresentada na Figura 3.9, sendo que os módulos aprimorados são apresentados em azul. Desta forma, o Gazebo é utilizado como módulo de simulação das forças físicas, enquanto o UWSim tem caráter de ferramenta de visualização e simulação de ondas. Como pode-se observar, o Gazebo não foi modificado, mas recebeu um novo plugin chamado Foil Dynamics, além de ter sido realizado aprimoramentos no plugin FreeFloating. Já o UWSim foi modificado para que os plugins desenvolvidos pudessem fazer solicitações, a fim de conhecer a altura das ondas dada a posição da embarcação no mapa.

Os módulos de correnteza de vento e água são modelados como nodos ROS, os quais recebem solicitações do módulo FreeFloating Gazebo para aprimorar o nível de realismo do movimento das embarcações, já que passa a utilizar informações de corrente de vento e água. No caso das embarcações que possuem quilhas, lemes ou velas, o plugin Foil

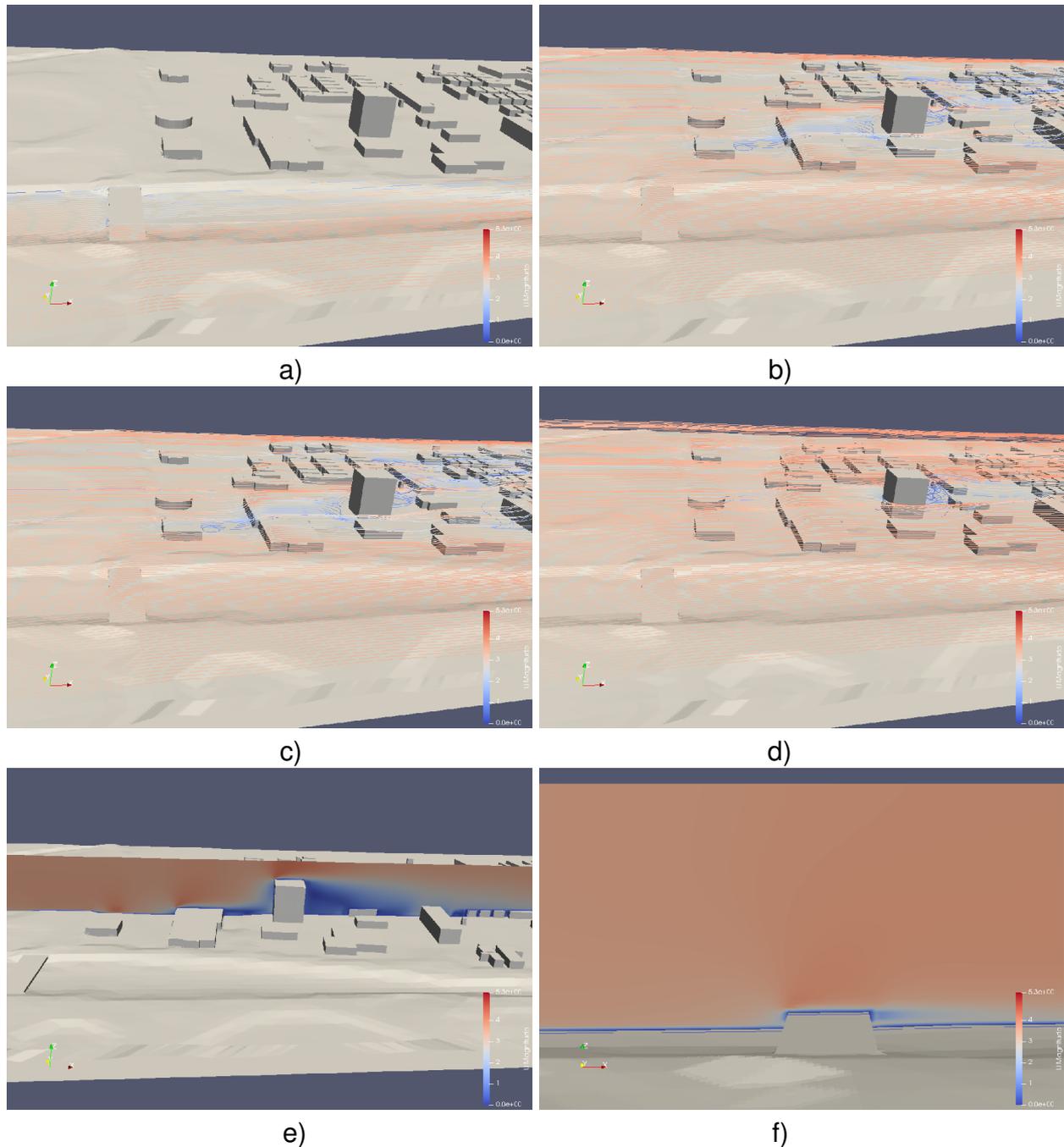


Figura 3.8: Visualização das correntes de vento na ferramenta paraFoam para diferentes altitudes: 1 metro (a), 5 metros (b), 10 metros (c), 20 metros (d). Nas figuras (e) e (f), é apresentado o gradiente de velocidade para dois planos: um próximo a edificações e outro próximo a pontes.

Dynamics faz uso de dados de velocidade do vento e de água no intuito de calcular a força a ser aplicada na embarcação. Todos os módulos fazem uso de arquivo de configuração do tipo YAML para configurar os parâmetros de simulação.

Nas próximas seções, são detalhados cada módulo constituinte da arquitetura do sistema de simulação, sendo que, na Seção 3.2.1, são apresentados aspectos do modelo matemático que regem a cinemática de movimento de USVs. Já na Seção 3.2.2 detalha

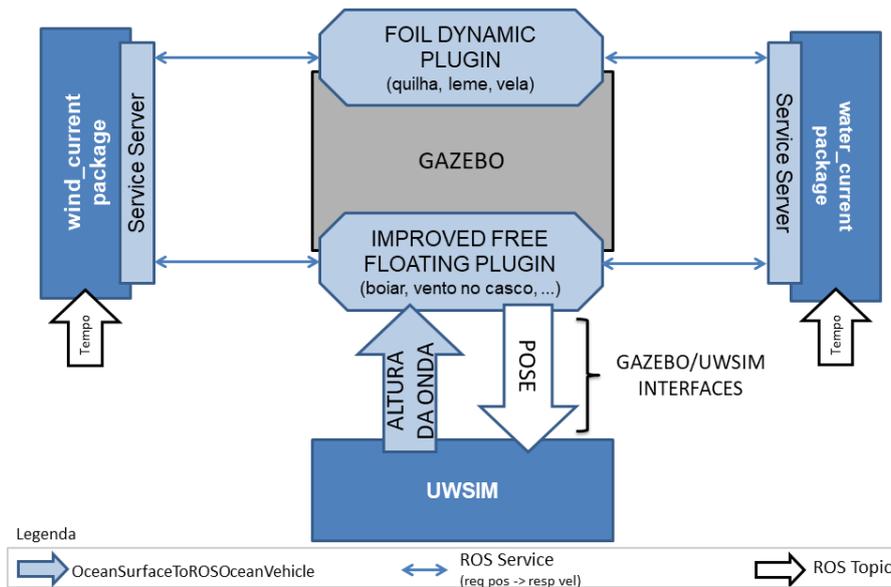


Figura 3.9: Arquitetura de simulação contendo os principais módulos.

o plugin Foil Dynamics utilizado para simular quilhas, lemes e velas de embarcações. Na seção 3.2.3, é descrito os principais aprimoramentos realizado nos simuladores existentes. Na Seção 3.2.4, são apresentados quatro modelos de USV disponibilizados nesse simulador. Por fim, na Seção 3.2.5, a integração realizada entre o simulador robótico com as ferramentas fluidodinâmica (Computational Fluid Dynamics - CFD).

3.2.1 Cinemática de USVs

Conforme Fossen [Fos11], as principais forças que governam o movimento dos USVs são:

- **Forças hidrodinâmicas** (τ_{hyd}): massa (virtual) adicionada devido ao movimento do barco, *potential damping* e *viscous damping*;
- **Forças hidrostáticas** (τ_{hs}): forças de restauração (boiar);
- **Força do vento** (τ_{wind});
- **Força das ondas** (τ_{waves});
- **Forças de controle e propulsão** (τ): dinâmica de quilhas, lemes, velas e propulsores;

O movimento do USV é obtido por meio da combinação dessas forças e momentos, como pode ser observado na Equação 3.1.

$$\tau_{RB} = \tau_{hyd} + \tau_{hs} + \tau_{waves} + \tau_{wind} + \tau \quad (3.1)$$

Nessa tese, as forças são calculadas utilizando-se plugins, sendo que o Gazebo fica responsável por unir as forças que cada plugin ficou responsável por calcular. Isso permite que o usuário possa controlar o nível de fidelidade e desempenho que deseja obter da simulação, já que pode remover um plugin, e assim, desativar determinada força. As forças Forças hidrodinâmicas (τ_{hyd}), Forças hidrostáticas (τ_{hs}), Força do vento (τ_{wind}) e Força das ondas (τ_{waves}) são implementadas por meio do plugin FreeFloating, enquanto que a força de propulsão (τ) é computado pelo plugin nomeado *Foil Dynamics*.

3.2.2 Plugin Foil Dynamics

O plugin *Foil Dynamics* é utilizado para as forças de sustentação e arrasto (τ) que agem na asa (quilha, leme ou vela) da embarcação. Para estimar essas forças, deve ser calculado a velocidade aparente entre a asa da embarcação e a velocidade do fluido (vento ou água, dependendo do tipo peça). Assim, ao utilizar coeficientes específicos de arrasto e sustentação daquela peça, as forças poderão ser calculadas por meio da seguinte equação:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho A v_a^2 C_L(\alpha) \sin\alpha - \frac{1}{2}\rho A v_a^2 C_D(\alpha) \cos\alpha \\ \frac{1}{2}\rho A v_a^2 C_L(\alpha) \cos\alpha + \frac{1}{2}\rho A v_a^2 C_D(\alpha) \sin\alpha \end{bmatrix}, \quad (3.2)$$

onde F_x é o arrasto, F_y é a sustentação, ρ é a densidade do fluido, A é a área da asa (quilha, leme, vela), v_a é a velocidade aparente do fluido, α é o ângulo de ataque da asa. $C_L(\alpha)$ e $C_D(\alpha)$ são os coeficientes de arrasto e sustentação, respectivamente, os quais estão em função do ângulo de ataque.

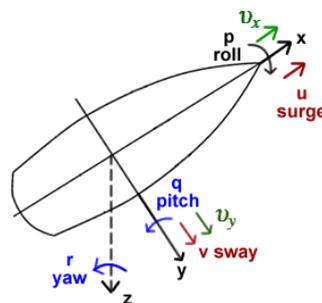


Figura 3.10: Sistema de coordenada do barco (v_x, v_y) e as componentes da velocidade aparente da água.

A velocidade aparente do fluido pode ser calculado por meio de $v_a = \sqrt{v_x^2 + v_y^2}$, onde v_x e v_y são as componentes da velocidade do fluido ao longo dos eixos longitudinal e lateral do sistema de coordenadas do barco (ver Figura 3.10), os quais são estimados por

meio da seguinte equação:

$$\begin{aligned} v_x &= v_t \cos(\alpha_t - \psi) - u + rn \\ v_y &= v_t \sin(\alpha_t - \psi) \cos\phi - v + rm + po \end{aligned} \quad (3.3)$$

onde v_t é velocidade real do fluído, α_t é o ângulo real do fluído, ψ é ângulo *yaw* do barco, e os componentes da velocidade do barco são $[u, v, p, r]^T$, (m, n, o) é o sistema de referência do barco. Portanto, para calcular o ângulo de ataque α_r do leme de um barco pode-se calcular por meio de $\alpha_r = \alpha_{aw} - \delta_r$, onde δ_r é o ângulo do leme do barco e $\alpha_{aw} = \text{atan2}(v_y, -v_x)$ é ângulo da velocidade aparente da água.

Enquanto que o plugin Liftdrag computa as forças considerando apenas a velocidade da asa, nessa tese o plugin Foil Dynamics considera a velocidade do fluído para calcular as forças de arrasto e sustentação. Além disso, esse plugin utiliza os valores de coeficientes de arrasto e sustentação descrito em Xiao [XJ14], o qual aproxima os coeficientes utilizando a equação $C_D(\alpha_a) = \sin(2\alpha_a)$ and $C_L(\alpha) = 1.5(1 - \cos(2\alpha_a))$, onde α_a é o ângulo de ataque aparente. Assim, o usuário precisa apenas informar as características físicas da asa (leme, quilha ou vela), ou seja, área e posição a ser aplicada as forças.

3.2.3 Aprimoramentos no UWSim e no Plugin FreeFloating Gazebo

Anteriormente a esse modelo, o Gazebo não simulava as correntes de água e ondas. No UWSim, as ondas são renderizadas pela biblioteca OsgOcean, a qual implementa as ondas utilizando Transformada Rápida de Fourier. Portanto, o UWSim conhece a altura da ondas, mas essa informação não é compartilhada com o Freefloating Gazebo [Ker14]. Neste caso, todas as embarcações flutuam sobre uma superfície plana ($Z=0$), ignorando o formato das ondas e limitando o realismo da simulação, como pode ser visto na Figura 3.11a.

Por isso, o UWSim foi modificado para que ocorra a sincronização entre a simulação de ondas (OsgOcean) e o Gazebo, assim cada passo de simulação do OsgOcean será o mesmo para todos os plugins do Gazebo. Por meio de uma nova interface (chamada OceanSurfaceToROSOceanVehicle), a altura da onda relativa ao centro de flutuação (buoyancy) do barco é enviado do UWSim para o FreeFloating Gazebo aprimorado, sendo assim utilizado para calcular o efeito de boiar. Contudo, isto permite que os barcos flutuem para cima e para baixo se aproximando do movimento das ondas. Em alguns momentos, ele provoca que o convés (frente do barco) e a popa (parte de trás do barco) estejam totalmente submersos ou até mesmo flutuando fora da água, como pode ser observado na Figura 3.11b.

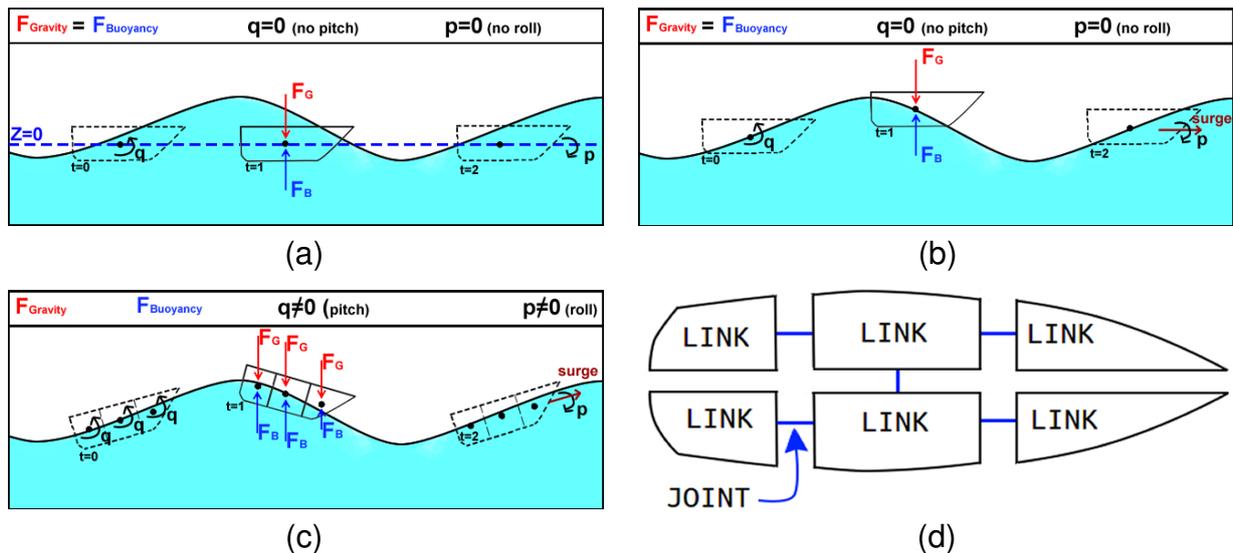


Figura 3.11: Implementações diferentes do efeito de boiar é apresentado em (a), (b) e (c), onde os pontos pretos representam centros de flutuabilidade, enquanto que as setas vermelhas e azuis representam as forças da gravidade e de flutuabilidade respectivamente. Em (a), o efeito de boiar do Freefloating gazebo é apresentado [Ker14]. Em (b), o efeito de boiar na embarcação é apresentando, no qual apenas a altura da onda é apresentada, dado que um barco é definido apenas com um único *link*. Em (c), o efeito similar é apresentado sendo que a embarcação passa a ser subdividido por um conjunto de *links* e *joints* como mostrado em (d). Assim, as forças da gravidade de boiar são aplicados para centro de massa e flutuabilidade de cada *link*.

Posteriormente, modificou-se o plugin FreeFloating Gazebo para a viabilizar o surgimento de rotações no eixo X (roll) e Y (pitch) do barco. Essas rotações são causadas pelo movimento das ondas (τ_{waves}), permitindo que os barcos reproduzam as inclinações geradas pelas ondas, como ilustrado na Figura 3.11c. Com essa estratégia, cada casco do barco deve ser subdividido em diversas partes, ou seja, modelado como um conjunto de *links* unidos por meio de juntas fixas (ver Figura 3.11d).

A subdivisão também permite aprimorar o realismo do momento inercial do casco do barco, pois cada parte (*link*) está em diferentes alturas, com diferentes volumes de casco submerso, e forças de flutuabilidade. Essas forças individuais em cada parte do casco são combinadas pela biblioteca de física Gazebo, permitindo que o torque seja automaticamente estimado. O efeito resultante das forças da gravidade (seta vermelha) e de boiar (seta azul) é apresentado na Figura 3.11c, permitindo que o barco apresente rotações (roll e pitch) devido a influência das ondas, provocando um efeito mais realista no movimento da embarcação.

A subdivisão do casco pode gerar geometrias complexas, as quais, ao estarem submersas, demanda o cálculo de seus volumes por meio de métodos computacionais caros. Para tal, a complexidade desses volumes é reduzida a medida que cada link passa a ser representado por geometrias primitivas, tais como cubos, retângulos e cilindros. Essa

substituição permite que o volume submerso seja estimado sem prejudicar a velocidade da simulação.

Ademais, o FreeFloating Gazebo foi modificado para permitir que as forças do vento afetem o casco do barco, alterando o movimento do mesmo (τ_{wind}). Então, as embarcações chamadas *Airboat*, *Rudder Boat*, *Differential Boat* (apresentados na Figura 3.12) são afetadas pela força do vento aplicada ao casco, enquanto que o veleiro (*Sailboat*) é afetado pelo vento ao aplicar força ao casco e às velas. As forças e os momentos gerados pelo vento são aplicadas a cada *link* do casco, sendo que os coeficientes aerodinâmicos das resistências laterais e longitudinais, as forças cruzadas e momentos de rolamento são definidos pelas equações em Blendermann [Ble94].

3.2.4 Modelos de robôs propostos

O USV_SIM disponibiliza quatro modelos de USV, prontos para uso, sendo que a dinâmica de cada um deles é definida de maneira padronizada. Essas embarcações são ilustradas na Figura 3.12, sendo nomeadas, da esquerda para a direita, como: *Airboat*, *Rudder Boat*, *Differential Boat* e *Sailboat*. O propulsor do *Airboat* fica em cima do casco e pode girar em torno de um eixo na vertical, permitindo assim, que o USV mude a direção de deslocamento. O *Airboat* é afetado mais significativamente pelos efeitos de *drift*, já que ele não possui nenhuma quilha submersa. Já o *Differential Boat* utiliza dois propulsores submersos, sendo que ele inverte a velocidade dos propulsores para realizar curvas. O *Rudder Boat* possui um propulsor submerso e um leme, o qual é responsável por alterar a direção de movimento. Enquanto que o veleiro (*Sailboat*) utiliza uma vela rígida para propulsão, um leme para definição de direção de movimento e uma quilha para garantir a estabilidade nas rotações (*sway* e *roll*).

Essas embarcações podem fazer uso de diversos sensores, pois o Gazebo oferece diversos plugins e uma documentação detalhada em como modelar sensores de corrente de água, temperatura, ph, etc. Os modelos disponibilizados nessa tese já contém sensores de posição, Inertial Measurement Unit (IMU) e *laser ranger finder*.

Com relação a aplicação em situações de desastres, o *Airboat* é melhor utilizado em águas rasas, já que detritos submersos podem danificar propulsores e lemes, bem como em áreas contendo vegetação subaquática. Contudo, o *Airboat* não é aconselhável para locais com grandes correntezas de vento e água. Já o barco diferencial é um veículos com grande manobrabilidade e possui potência para enfrentar correntezas de água. Porém, por consumir mais energia, uma vez que possui dois propulsores, ele tende a ficar limitado a missões de menor tempo de duração. O *Rudder Boat* possui boa manobrabilidade e capacidade de carga, possuindo um maior equilíbrio entre tempo de missão e consumo energético. Já o *Sailboat* é melhor utilizado em corpos de água de maiores dimensões



Figura 3.12: Da esquerda para direita: *rudder boat*, *airboat*, *differential boat* and *sailboat* disponibilizado no simulador.

(como os oceanos), pois possui um baixo consumo energético e menor manobrabilidade. Além disso, os *sailboats* não podem ser utilizados em águas rasas em função do tamanho da quilha.

Na Seção 4.1.1, é possível conhecer as dimensões das embarcações *Airboat* e *Differential boats*, que possuem as mesmas medidas de dois veículos reais existente no Laboratório de Sistemas Autônomos (LSA). Os cascos do *Sailboat* e *Rudder boat* possuem dimensões iguais do *Airboat* e do *Differential Boat*, porém com algumas características específicas: o *Sailboat* possui uma vela rígida com 1 m^2 de área e uma quilha de 30 Kg; enquanto que o *Rudder boat* possui um leme de 4 centímetros de largura por 6 centímetros de altura.

3.2.5 Módulos de correnteza de água e vento

Para reproduzir correntes de água de maneira mais realista, tendo diferentes valores ao longo do tempo e do espaço, desenvolveu-se um novo pacote ROS chamado *water_current*. Ele é responsável por carregar os dados exportados do simulador hidrológico HEC-RAS [Eng20] [Bru95]. Esse simulador consegue modelar fluxo constante em uma dimensão (1D), bem como fluxo não constante em 1D e 2D (duas dimensões) para canais e rios. Além disso, os usuários podem simular o fluxo de rios utilizando-se de MDS, e então exportando arquivos HDF. Esses arquivos armazenam a velocidade da água para cada instante de tempo simulado. A Figura 3.13 apresenta o mapa de velocidade da água, obtido da simulação no HEC-RAS, sendo que, em regiões de estreitamento de canal, observa-se o aumento da velocidade água. Assim, os arquivos HDF são carregados pelo plugin *water_current*, para permitir que sejam disponibilizados as velocidades da água para o modelo de simulação robótico. Isso é realizado por meio de um serviço ROS, no qual o FreeFlo-

ating Gazebo foi modificado para enviar uma solicitação contendo a posição de cada link da embarcação, e o plugin `water_current` responde com a velocidade da água para cada ponto.

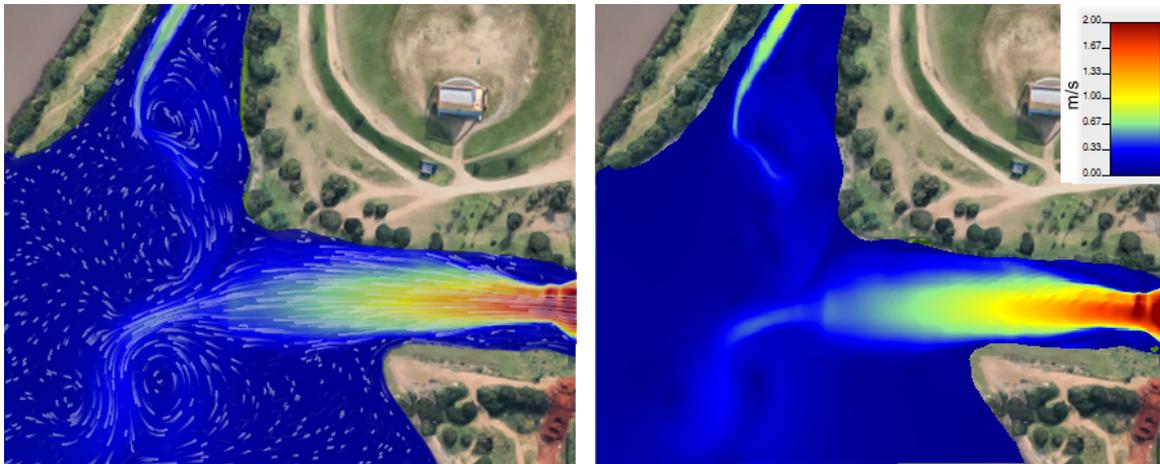
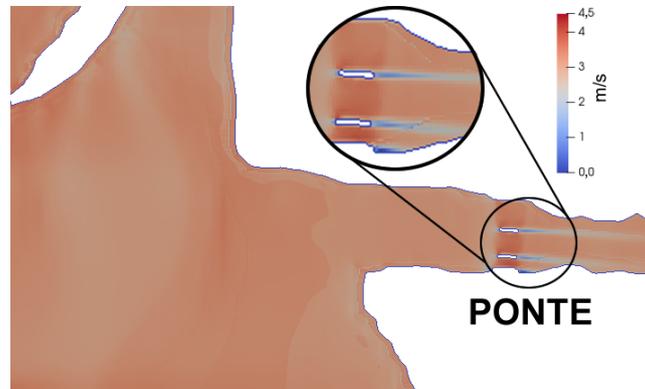


Figura 3.13: Simulação da corrente de água com o Hec RAS para um ambiente em que ocorre a interseção de dois rios. A imagem da esquerda apresenta a trajetórias de partículas em branco movendo-se sobre a água, assim pode-se compreender a direção e módulo da correnteza de água. Na imagem da direita, o fluxo mais rápido é apresentado em vermelho, enquanto que o de menor intensidade em azul.

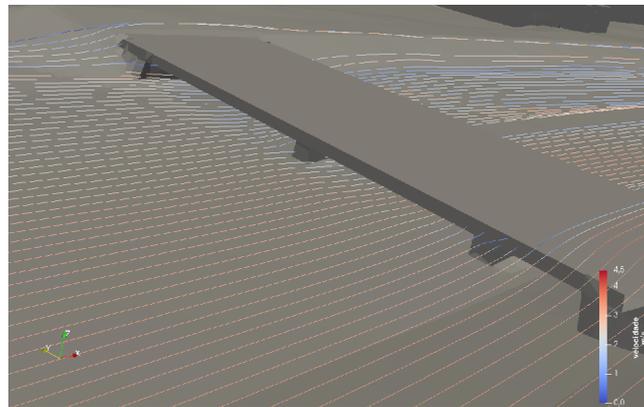
De maneira similar, o pacote ROS `wind_current` carrega os dados da velocidade de vento provenientes do CFD OpenFoam (Open source Field Operation And Manipulation) [Ope20a]. Essa ferramenta consegue realizar a simulação de fluídos considerando a existência de um terreno em 3D, composto por obstáculos como prédios e pontes. Para tal, deve-se especificar parâmetros de simulação, tais como velocidade de vento, duração da simulação e time step.

A simulação resultante, pode ser analisada e exportada para o módulo `wind_current` pela ferramenta `paraFoam` do OpenFoam. Os dados exportados contém a velocidade do vento a cada instante de tempo, podendo ser carregada e acessada na simulação robótica por meio de um serviço ROS. Por isso, que o `FreeFloating Gazebo` foi alterado para que enviasse a posição de cada link da embarcação, de forma que pacote `wind_current` respondesse com a velocidade do vento naquele ponto. A Figura 3.14c apresenta o modelo de vento ao redor de pontes e construções.

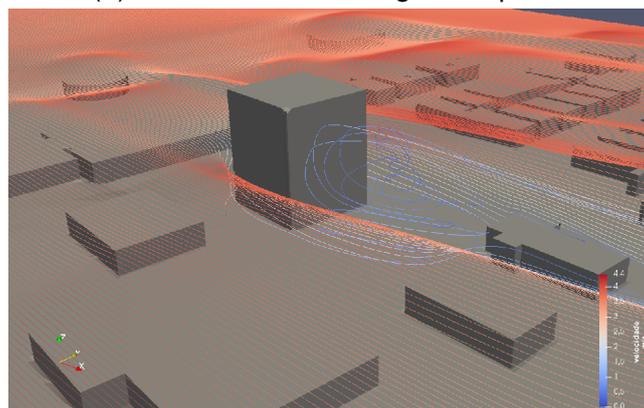
Cabe destacar que o cenário real utilizado nas figuras dessa seção referem-se a cidade de Porto Alegre, RS, Brasil, coordenadas de GPS $30^{\circ}02'50.5''S$ $51^{\circ}13'57.7''W$. Essa região fica próximo a foz do Arroio Dilúvio, sendo que o cenário foi recriado utilizando-se os MDT e MDS disponibilizados pela Prefeitura de Porto Alegre.



(a) Vista da mesma região apresentada na Figura 3.13, incluindo detalhes do vento próximo a ponte.



(b) Vista do vento na região da ponte



(c) Vista do vento ao redor de prédios.

Figura 3.14: Resultado da simulação de vento com OpenFoam.

4. VALIDAÇÃO

Nesse capítulo, é apresentado o método utilizado para realizar a validação do modelo de simulação proposto, sendo que são descritos os testes realizados com robôs reais, bem como a comparação com o movimento dos robôs virtuais. Por fim, são apresentados estudos de casos com os barcos simulados em situações em que distúrbios ambientais estão presentes.

4.1 Metodologia

De forma a verificar a fidelidade das respostas do simulador, o *Airboat* e o *Differential Boat* foram modelados em simulação de acordo com as embarcações reais existentes no Laboratório de Sistemas Autônomos (LSA). Para tal, os modelos virtuais receberam os parâmetros físicos que regem a dinâmica de movimento dessas embarcações reais. A coleta desses parâmetros foi baseada no trabalho de Wirtensohn [WRB⁺13], no qual é realizada em quatro etapas: extração de características físicas (peso, comprimento, etc), medição de força de propulsão, extração de coeficientes de arrasto linear frontal, extração de coeficientes de arrasto lateral e arrasto angular.

No trabalho de Wirtensohn (2013) [WRB⁺13], a primeira etapa contou com o levantamento das seguintes características físicas de um barco diferencial feitas em laboratório: comprimento, largura, massa, carga extra, potência do motor e nível de voltagem das baterias. Na segunda etapa, Wirtensohn estimou a força máxima dos propulsores, isso foi realizado ao amarrar a embarcação num poste por meio de um cabo conectado a um dinamômetro. Ao aplicar diferentes velocidades ao propulsor é possível relacionar velocidades com força de propulsão.

Já na terceira etapa, Wirtensohn [WRB⁺13] acelerou o robô até atingir a velocidade máxima, em seguida desligou os propulsores. Considerando que a desaceleração da embarcação é causada pelas forças de arrasto, os coeficientes de arrasto frontal (eixo x) - Figura 4.1) puderam ser estimados. Ainda nesse teste, foi possível identificar a velocidade máxima da embarcação em situações de ausência de distúrbios externos (vento, corrente de água, ondas).

Para a quarta etapa, Wirtensohn realizou dois tipos de movimento com a embarcação em testes em campo: um movimento em espiral com diferentes velocidades e direções de movimento; outro movimento em zigue-zague. Com os dados do barco real, Wirtensohn executou o algoritmo evolutivo de Enxame de Partículas [KE95] e assim estimou os coeficientes de arrasto linear no eixo y (Figura 4.1), bem como o coeficiente de arrasto angular no eixo z (Figura 4.1).

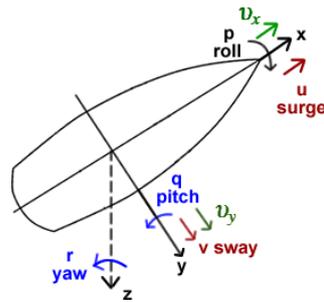


Figura 4.1: Sistema de coordenada do barco (v_x, v_y) e as componentes da velocidade aparente da água. Essa figura foi propositalmente repetida para facilitar a leitura.

Com relação a essa tese, as quatro etapas com suas adaptações e adequações são apresentadas nas próximas seções, ou seja, a Seção 4.1.1 apresenta o levantamento de características físicas das embarcações reais, a Seção 4.1.2 descreve a coleta das forças de propulsão. A Seção 4.1.3 apresenta a coleta de dados e estimativa do coeficiente de arrasto frontal, enquanto que a Seção 4.1.4 relata o teste em campo para coleta de dados e estimativa dos coeficientes de arrasto lateral e angular. Posteriormente, é apresentado como os testes com robôs virtuais foram realizados. A comparação entre o movimento das embarcações reais nos testes em campo com o movimento das simuladas no ambiente virtual é apresentada na Seção 4.2.

4.1.1 Etapa 1: Especificações físicas dos robôs

O *Airboat* e o *Differential Boat* foram modelados baseando-se nas embarcações *Lutra Airboat* e *Lutra Prop*, adquiridos da empresa Lutra Platypus [Pla20]. Ambas embarcações possuem o mesmo formato e tamanho de cascos, aproximadamente 106 centímetros de comprimento, 48 centímetros de largura e 15 centímetros de altura. Enquanto o *Lutra Airboat* pesa 9 Kg, o *Lutra Prop* pesa 9.7 Kg. O *Lutra Prop* possui abaixo do seu casco sistema de arrefecimento de motores (pequenos dutos metálicos) e suportes para fixação de sensores, tais como batimetria ou de qualidade de água. Nos testes realizados, ambas as embarcações transportaram aproximadamente 3 Kg de carga extra, já que foi necessário adicionar equipamentos para os testes, por exemplo, o sistema de *Real Time Kinematic* (RTK). Portanto, o kit do RTK incluía: antena, módulo RTK, suporte de antena, bateria, celular com acesso a internet. *Lutra Airboat* possui um propulsor acoplados a popa do casco, provendo forças para deslocar a embarcação. Já o *Lutra Prop* possui dois propulsores embaixo da água, ambos estão posicionados distantes 25 centímetros um do outro, com ângulo de 15° entre a linha horizontal e o eixo do propulsor.

A especificação física das embarcações é apresentada na Tabela 4.1. Cabe ressaltar que, durante a quarta etapa, o *Lutra Prop* teve uma redução na carga extra de 3 Kg para 0,9 Kg, já que foi possível remover parte dos equipamentos que acompanhavam o

GPS RTK. Destaca-se que isso apenas foi realizado, pois foi necessária a reconfiguração dos módulos ESC (*Eletronic Speed Control*), bem como um dos eixos dos propulsores ter sido parcialmente danificado. Contudo, essas modificações serão detalhadas na Seção 4.3.

Tabela 4.1: Parâmetros físicos do *Lutra Airboat* e *Lutra Prop*.

Parameter	Lutra Airboat	Lutra Prop	Lutra Prop v2
Comprimento	106 cm	106 cm	106 cm
Largura	48 cm	48 cm	48 cm
Altura	45 cm	15 cm	15 cm
Volume do casco	~0,02 m ³	~0,02 m ³	~0,02 m ³
Peso	9 Kg	9,7 Kg	9,7 Kg
Carga Extra	3 Kg	3 Kg	0,9 Kg

Na próxima seção, são descritos os testes em campo realizados com o *Lutra Airboat* e o *Lutra Prop* para medir as forças propulsão.

4.1.2 Etapa 2: Extração de forças de propulsão

Nessa tese, utilizou-se um procedimento similar ao utilizado em Wirtensohn [WRB⁺13] para identificar as forças de propulsão e a dinâmica do movimento das embarcações *Lutra Airboat* e *Lutra Prop*. Contudo, a entrada de dados dos propulsores são expressos em valores de PWM (Pulse Width Modulation) ao invés de valores de velocidades, isso ocorre porque os módulos ESC (*Eletronic Speed Control*) tem como valores de entrada o PWM. Assim, diversos valores de PWM foram aplicados nos atuadores, obtendo-se a força de propulsão por meio de leitura do dinamômetro. Esses dados foram coletados após os valores no dinamômetro terem se estabilizado.

No *Lutra Airboat*, foram medidas as forças dos propulsores aplicando-se os seguintes valores de PWM: 2000, 1850, 1750, 1650 e 1500. Os resultados apresentados na Tabela 4.2 demonstram um crescimento linear no módulo de força, sendo que o valor 1500 foi insuficiente para ativar a movimentação da hélice do propulsor. Já no *Lutra Prop*, aplicou-se os seguintes valores de PWM: 1800, 1700, 1600, 1553 e 1500. Similarmente ao Airboat, os propulsores não foram ativados com o PWM em 1500. Os maiores valores de PWM que foram utilizados para medir forças no *Lutra Prop* foi de 1700, já que, ao aplicar 1800, o sistema elétrico foi desativado, pois o fusível se rompeu assim que os motores drenaram correntes elétricas acima dos limites seguros do sistema da embarcação.

Devido aos problemas de sobrecarga no sistema elétrico no *Lutra Prop* descritos anteriormente, fez-se necessário a reconfiguração dos módulos ESC (*Eletronic Speed Control*), o que levou a diminuição da força de propulsão dessa embarcação. Uma nova rodada experimentos dessa etapa mostrou-se necessário para aferir a força máxima obtida pelos propulsores. Contudo, nessa etapa um dos eixos dos propulsores havia sido danificado, o

Tabela 4.2: Força de propulsão das embarcações para diferentes valores de PWM means Pulse Width Modulation.

Lutra Airboat		Lutra Prop		Lutra Prop v2	
PWM	Força	PWM	Força	PWM	Força
2000	3,13 N	1800	-	-	-
1850	2,64 N	1700	22,54 N	-	-
1750	2,35 N	1600	17,93 N	-	-
1650	1,86 N	1553	16,46 N	1350	10,14 N
1500	0	1500	0	1200	0

que diminuiu o desempenho máximo. Assim, a força máxima medida na repetição do experimento foi de 10,14 N, sendo que foram realizadas apenas medições com o maior valor de PWM, uma vez que observou-se, nos testes anteriores, a relação linear proporcional entre o valor de PWM e a força. Similarmente, a coleta de dados apresentada na próxima seção também foi repetida para o *Lutra Prop*.

Na próxima seção, é apresentado como foi realizado o experimento de aceleração frontal para estimar os parâmetros dinâmicos ao longo do eixo x de cada embarcação.

4.1.3 Etapa 3: Extração da dinâmica de movimento frontal

Para estimar a dinâmica de movimento frontal, realizou-se testes de aceleração em que cada embarcação iniciava parada, sendo, em seguida, aplicados os maiores valores de PWM possíveis. Assim, ao deslocar-se em linha reta, obteve-se o perfil de velocidade do robô durante a aceleração máxima dos propulsores. Durante os testes, não existiam correntes de vento e água significativas para afetar o movimento dos USVs. Consequentemente, as principais forças atuantes nas embarcações são as geradas pelos propulsores e pelo arrasto do movimento de cada embarcação devido ao atrito laminar entre o casco e a superfície da água.

De forma a melhorar a precisão da localização da embarcação durante o deslocamento, elas foram equipadas com Emlid's Reach RTK Global Positioning System (GPS) [Eml20], o qual foi configurado para receber correções da estação base pública provida pelo IBGE [IBG20], localizada em 30°04'26.7"S 51°07'11.2"W. Os testes foram realizados no lago localizado em 30°04'57.9"S 51°03'23.9"W, o qual possui uma área de 85 metros por 55 metros e fica no fundo de um vale, rodeado por árvores. Portanto, a posição topológica do lago limita a existência de correntes de vento, enquanto que a proximidade com a estação base do IBGE (distante apenas de 6.1 Km) garante ótimas correções de GPS. Assim, durante os testes, o GPS obteve 99% de fix (melhor sinal de gps), gerando excelente precisão, ou seja, a precisão fica no nível de poucos centímetros. Por exemplo, no teste de

aceleração do Airboat a média do desvio padrão para os eixos x e y foi de apenas 0,99 cm e 0,97 cm, respectivamente.

Assim, durante esse tipo de teste em campo, buscou-se identificar a aceleração linear, a velocidade máxima e o coeficiente de arrasto linear frontal (C_x) de cada embarcação. A aceleração e a velocidade máxima são obtidas diretamente pelos dados gerados pelo GPS RTK, enquanto que o coeficiente de arrasto C_x deve ser estimado utilizando as equações que regem o movimento da embarcação e as forças de arrasto hidrodinâmico. Zhang [ZZ18] define o movimento no eixo x da embarcação por meio da seguinte equação:

$$X = m(\dot{u} + qw - rv) \quad (4.1)$$

onde m é a massa da embarcação; v , w são as velocidades lineares do casco da embarcação nos eixos y e z (unidade em m/s); q , r são as velocidades angulares em torno dos eixos y e z (unidade em rad/s); X é a soma das forças externas ao casco em Newtons; \dot{u} é a derivada da velocidade linear.

Com base, nas condições em que os testes em campo foram executados, não havendo correntes de vento e água, pode-se utilizar a seguinte equação para listar as forças externas:

$$X_p + X_d = X \quad (4.2)$$

onde X_p é a força de propulsão (obtida no teste da etapa 2); X_d é a força de arrasto linear frontal. Essa última força é calculada no Freefloating gazebo [Ker14] pela equação a seguir.

$$X_d = C_x u^2 \quad (4.3)$$

onde C_x é o coeficiente de arrasto linear; u é a velocidade linear aparente da embarcação frente ao deslocamento do fluido. Vale lembrar que, nesse contexto, a velocidade linear aparente trata-se da velocidade do casco descontada a velocidade do fluido.

Para calcular o coeficiente de arrasto C_x , faz-se uso das Equações 4.1, 4.2 e 4.3. Isso é possível já que os seguintes termos são conhecidos: a massa m foi estimada em laboratório (etapa 1); a força de propulsão X_p foi obtida na etapa 2; enquanto que a velocidade máxima u é extraída do GPS RTK durante o teste de aceleração frontal (etapa 3). Já os valores de aceleração linear \dot{u} , velocidades lineares v e w , a velocidade angular q são consideradas como zero, já que a embarcação encontra-se deslocando em velocidade máxima (constante) e em linha reta.

A Tabela 4.3 apresenta os resultados obtidos durante essa etapa para *Lutra Airboat* e *Lutra Prop*. O *Lutra Airboat* teve como velocidade linear máxima 0,67 m/s e seu coeficiente C_x foi estimado em 6,9 Kg/m. Já o *Lutra Prop* teve como velocidade máxima 1,41 m/s, enquanto que seu coeficiente C_x foi de 11,33 Kg/m. A repetição do teste com o *Lutra Prop*

teve como valor de velocidade máxima aproximadamente 1,4 m/s. Apesar da diminuição da força de propulsão entre o primeiro e o segundo teste de aceleração frontal, a velocidade se manteve muito próxima. Isso se deve ao fato que a diminuição da força de propulsão fez com que a embarcação inclinasse menos, diminuindo o arrasto frontal C_x e mantendo as hélices submersas completamente na água. A menor velocidade das hélices também melhora o rendimento do seu movimento, pois diminui a formação de bolhas devido ao movimento rápido da hélice (cavitação) e a atração do ar da superfície devido a pequena profundidade.

Tabela 4.3: Parâmetros dinâmicos do *Lutra Airboat* e *Lutra Prop* coletados na etapa 3 e 4.

Parâmetro	Lutra Airboat	Lutra Prop	Lutra Prop v2
C_x	6,9	11,33	5,14
C_y	9,96	-	247
C_r	0,6	-	1,66
Velocidade linear X máxima	0,67 m/s	1,41 m/s	1,4 m/s
Velocidade angular Z máxima	0,78 m/s	-	0,52 rad/s

No entanto, deve-se ressaltar que o modelo de arrasto hidrodinâmico implementado no Freefloating gazebo é uma simplificação dos modelos descritos na literatura [Fal90] [Fos11]. A simplificação oferece ganhos de desempenho na simulação, mas limita o realismo do movimento da embarcação. Por exemplo, Faltinsen [Fal90] descreve as forças de arrasto X_d conforme as equações abaixo:

$$X_d = \left(\frac{0,075}{(\log_{10} R_n - 2)^2} \right) \frac{1}{2} \rho S v_a^2 \cos \beta |\cos \beta| \quad (4.4)$$

$$R_n = \frac{v_a L |\cos \beta|}{\nu}$$

onde ρ é a densidade do fluido; S é a área da superfície molhada do casco; v_a é a velocidade aparente; L é comprimento da embarcação; $\nu = 1 \times 10^{-6} m/s^2$ é viscosidade cinemática a $20^\circ C$.

Assim, ao comparar as Equações 4.3 e 4.4, pode-se concluir que o coeficiente C_x (Equação 4.3) deve contemplar todos os termos da Equação 4.4 que mantem-se constantes ou que variam pouco durante o movimento da embarcação. Como exemplo desses termos tem-se a densidade do fluido, a área da superfície do casco que está molhada, o comprimento do casco e a viscosidade cinemática. Portanto, a perda de realismo se dá por considerar que as condições sempre permanecerão as mesmas da estimativa do coeficiente de arrasto C_x , por exemplo, a superfície do casco molhada nunca variará, mesmo que a embarcação seja afetada por ondas, a densidade da água sempre será a mesma independente da temperatura da água.

Já na próxima seção, são descritos os testes em campo para coletar dados para estimar o comportamento das embarcações *Lutra Airboat* e *Lutra Prop* em movimentos circulares e curvas.

4.1.4 Etapa 4: Extração da dinâmica de movimento lateral

Para determinar os coeficientes de arrasto linear lateral (eixo y) e angular (eixo z) do *Lutra Airboat* e do *Lutra Prop* foram conduzidos testes de movimento circular no mesmo lago da etapa anterior. Nesses testes, não havia correntes de vento significativas para afetar o movimento das embarcações. Assim, com *Lutra Airboat* inicialmente parado, o propulsor é totalmente rotacionado em 0,5 radianos e a máxima propulsão é acionada, fazendo com que a embarcação atinja em alguns instantes a velocidade máxima de rotação angular. Já para o *Lutra Prop*, faz-se necessário acionar apenas um propulsor com máximo valor de PWM, enquanto que o outro propulsor é mantido parado.

Para ambas as embarcações, na ausência de forças externas (vento e correnteza de água), assim que a velocidade máxima é atingida, pode-se estimar os coeficientes de arrasto angular e linear lateral, nos eixos z e y respectivamente. Isso se dá ao fato das forças de arrasto linear e angular equilibrarem as forças geradas pelos propulsores.

Assim, durante esse teste, buscou-se identificar a aceleração angular e velocidade máxima no eixo z , o coeficiente de arrasto linear lateral (C_y) no eixo y , o coeficiente de arrasto angular C_r no eixo z . A aceleração angular e velocidade máxima angular são obtidos dos dados gerados pela bússola do sistema de localização do barco, enquanto que os coeficientes C_y e C_r , são estimados utilizando as equações que regem o movimento da embarcação e que calculam as forças de arrasto hidrodinâmico. Zhang[ZZ18] define o movimento da embarcação ao longo do eixo y e ao redor do eixo z por meio das seguintes equações:

$$Y = m(\dot{v} + ru - pw) \quad (4.5)$$

$$N = I_z \dot{w} + (I_y - I_x)pq \quad (4.6)$$

onde m é a massa da embarcação; u, w são as velocidades do casco da embarcação nos eixos x e z (unidade em m/s); p, q, r são as velocidades angulares em torno dos eixos x, y e z (unidade em rad/s); Y é a soma das forças externas ao casco no eixo y em Newtons, \dot{v}, \dot{w} são as derivadas da velocidade linear nos eixos y e z ; I_x, I_y, I_z são os momentos de inercia em torno dos eixos x, y e z ; N é o torque externo ao casco em torno do eixo z .

Já as forças externas em Y e os torques externos podem ser calculados por meio das seguintes equações:

$$Y_p + Y_d = Y \quad (4.7)$$

$$N_p + N_d = N \quad (4.8)$$

onde Y_p é a parcela da força de propulsão no eixo y ; Y_d é a força de arrasto linear lateral no eixo y ; N_p é o torque gerado pelo propulsor; N_d é o torque gerado pelo arrasto angular entre o casco e a água. Já as forças Y_d e o torque N_d são calculados no Freefloating gazebo [Ker14] pelas equações a seguir.

$$Y_d = C_y v^2 \quad (4.9)$$

$$N_d = C_r r^2 \quad (4.10)$$

onde C_y é o coeficiente de arrasto linear no eixo y ; v é a velocidade linear aparente da embarcação frente ao deslocamento do fluido ao longo do eixo y ; C_r é o coeficiente de arrasto angular no entorno do eixo z ; r é a velocidade angular no eixo z .

Para calcular o coeficiente de arrasto linear C_y , faz-se uso das Equações 4.5, 4.7 e 4.9. Isso é possível já que os seguintes termos são conhecidos: a massa m foi estimada em laboratório (etapa 1); a força de propulsão Y_p pode ser calculada utilizando-se a força estimada na etapa 2; enquanto que a velocidade máxima v é extraída do GPS do barco durante o teste circular (etapa 4). Já os valores de aceleração linear \dot{v} , a velocidade linear w , a velocidade angular q são consideradas como nulas, já que a embarcação desloca-se em velocidade angular máxima (constante), sendo que o sistema de coordenadas utilizado nessas equações é relativo ao casco da embarcação.

Para o *Lutra Airboat*, a força Y_p é calculada utilizando-se a força de propulsão máxima F_{max} (obtida na etapa 2) e o ângulo de rotação do propulsor utilizado nesse teste (0,5 rad), ou seja, $Y_p = F_{max} * \sin(0,5)$. Já para o *Lutra prop*, o valor de Y_p é zero, pois o propulsor aplica forças apenas ao longo do eixo x .

Para calcular o coeficiente de arrasto angular C_r , as Equações 4.6, 4.8 e 4.10 foram utilizadas, sendo que os seguintes termos são conhecidos: os momentos de inércia I_x, I_y, I_z são estimados com base nas dimensões e da massa do casco da embarcação (obtidos na etapa 1); p, q são as velocidades angulares nos eixos x e y ; \dot{w} é a aceleração angular em torno do eixo z ; a velocidade angular r é obtida da bússola da embarcação. Para o *Lutra Airboat*, o torque N_p é calculado multiplicando-se Y_p pela distância no eixo x do propulsor ao centro de massa da embarcação (obtida na etapa 2), ou seja, 30 centímetros. Enquanto que, para o *Lutra Prop*, o torque N_p é calculado multiplicando-se a metade da força de propulsão (obtida na etapa 2) pela distância no eixo y até o centro de massa (obtida na etapa 2), ou seja, 12.5 centímetros.

A Tabela 4.3 apresenta os resultados obtidos nessa etapa tanto para o *Lutra Airboat* quanto para o *Lutra Prop*. O *Lutra Airboat* teve como velocidade angular máxima 0,78

rad/s, enquanto que o *Lutra Prop* teve 0,52 rad/s. Já os coeficientes C_y e C_r foram estimados em 9,96 Kg/m e 0,6 Kg m^2 , respectivamente, para o *Lutra Airboat*, enquanto que para o *Lutra Prop* foram estimados em 247 Kg/m e 1,66 Kg m^2 . A grande diferença dos coeficientes pode ser observada empiricamente ao utilizar as embarcações, por exemplo, quando existe vento, este afeta mais o *Lutra Airboat* que o *Lutra Prop*. O *Lutra Airboat* tem maior instabilidade nos deslocamentos frontais em que pequenas variações de carga como, por exemplo, a posição da bateria dentro do compartimento de carga, ou mesmo, a orientação da antena, faz com que o *Lutra Airboat* realize curvas no sentido horário ou no sentido anti-horário ao enviar comandos para deslocar-se em linha reta.

De maneira similar a etapa 3, o modelo de força de arrasto para os eixos y e z , implementado no Freefloating gazebo [Ker14], é uma simplificação da literatura. Por exemplo, Faltinsen [Fal90] descreve as forças de arrasto Y_d e de torque N_d conforme as equações abaixo:

$$\begin{aligned} Y_d &= \frac{1}{2}\rho \left[\int_L dx C_d(x) D(x) \right] S v_a^2 \sin\beta |\sin\beta| \\ N_d &= \frac{1}{2}\rho \left[\int_L dx C_d(x) D(x) \right] S v_a^2 + \frac{1}{2} v_a^2 (A_{22} - A_{11}) \sin(2\beta) \end{aligned} \quad (4.11)$$

onde ρ é a densidade do fluido; L é o comprimento da embarcação; $C_d(x)$ é o coeficiente de arrasto para um fluxo transversal a um cilindro infinito que possui a mesma área do corte transversal da embarcação; $D(x)$ é profundidade do calado da embarcação; v_a é a velocidade aparente; β é o ângulo entre a velocidade aparente e o eixo x da embarcação; S é a área da superfície molhada do casco; A_{22} e A_{11} são as massas adicionadas pelo movimento da embarcação ao longo do eixo x e y ; respectivamente. Essas massas são a quantidade de água que é arrastada junto com o movimento da embarcação.

Portanto, essa simplificação oferece ganhos de desempenho em simulação, mas provoca diminuição do realismo da movimentação da embarcação virtual. Isso se deve ao fato que a área molhada do casco da embarcação não se mantém constante para todos os tipos de movimento, tão pouco os valores das massas adicionadas pelo deslocamento do USV.

Na próxima seção, são descritos os experimentos realizados em simulação do *Airboat* e do *Differential boat* para os mesmos testes realizados nas etapas 3 e 4.

4.1.5 Testes com robôs simulados

Após a realização da coleta de dados descritas nas seções anteriores, tornou-se possível recriar o *Lutra Airboat* e *Lutra Prop* com as mesmas características físicas identificadas na coleta de dados. Portanto, a etapa 1 permitiu configurar os robôs simulados com as dimensões, a massa, os momentos de inércia, a posição dos propulsores. Com a

etapa 2, os propulsores de cada embarcação virtual foram configurados com o valor máximo de força. A partir da etapa 3 e 4, as embarcações virtuais foram configuradas com os coeficientes de arrasto linear C_x e C_y , bem como com o coeficiente de arrasto angular C_r .

Com os USVs virtuais configurados, os mesmos testes em campo descritos nas etapas 3 e 4 foram reproduzidos em simulação. Assim, os ambientes virtuais foram configurados para não haver distúrbios ambientais como ondas, correntes de vento e de água. O teste de aceleração frontal (etapa 3) foi realizado no mundo virtual com as embarcações inicialmente paradas, em seguida as forças máximas dos propulsores foram ativadas. Para reproduzir o teste de movimento circular (etapa 4), o Airboat teve seu propulsor rotacionado em 0,5 rad, em seguida a máxima força foi ativada no propulsor. Para o barco diferencial, o movimento circular foi realizado com apenas um propulsor utilizando a maior força possível.

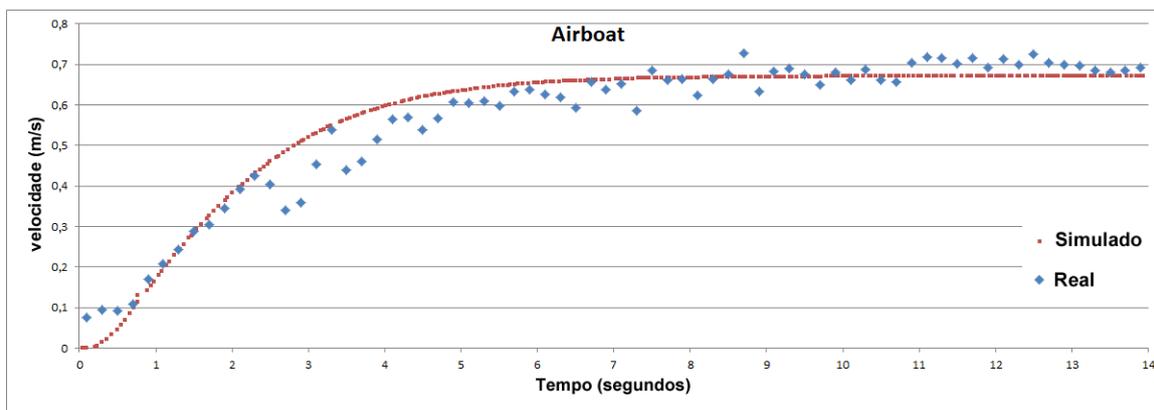
O comportamento detalhado das embarcações reais e das virtuais descritas nas Seções 4.1.3, 4.1.4 e 4.1.5 são apresentados na próxima seção, sendo realizada a comparação entre o movimento dos USVs observado no mundo real com o no mundo virtual.

4.2 Comparação entre robôs reais e simulados

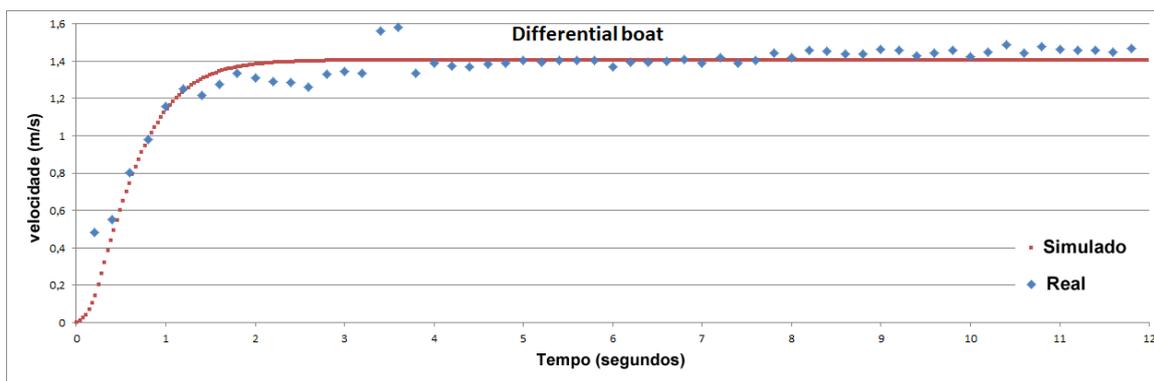
Nessa seção, são apresentados os resultados obtidos para o movimento dos robôs reais (*Lutra Airboat* e *Lutra Prop*) e seus respectivos robôs virtuais (*Airboat* e *Differential Boat*) nos testes de aceleração frontal e de movimento circular.

4.2.1 Resultados para aceleração frontal

A Figura 4.2 apresenta a comparação do perfil de velocidade obtida no teste em campo com o robô real e no ambiente simulado com o robô virtual. O gráfico mostra a velocidade em m/s ao longo do tempo em segundos, assumindo que o instante zero foi o momento em que as embarcações começaram a se mover. Durante os testes em campo, os barcos reais aparentavam estar totalmente parados, o GPS RTK capturou pequenas velocidades, aproximadamente de 0,1 m/s . Por isso, a velocidade dos barcos reais (linhas pontilhadas em azul) não estão em zero. Os gráficos mostram que a variação da velocidade durante a aceleração e durante a velocidade máxima se aproximam, demonstrando precisão do movimento do robô simulado em relação ao robô real. O erro absoluto médio e desvio padrão para o Airboat foi de 0,0312 m/s e 0,0481 m/s (respectivamente), ou seja, 94,7% dos erros estão na faixa de um desvio padrão. Enquanto que o erro absoluto médio e desvio padrão para o barco diferencial são 0,0507 m/s and 0,0742 m/s (respectivamente), ou seja 96,6% dos erros estão na faixa de um desvio padrão.



(a) Gráfico comparativo entre *Lutra Airboat* versus o airboat simulado. Utilizou-se o valor 2000 de PWM.



(b) Gráfico comparativo entre *Lutra prop* versus o barco diferencial simulado. Utilizou-se o valor 1700 de PWM.

Figura 4.2: Resultados para aceleração frontal comparando-se o desempenho do robô real em testes em campo com os dos barcos virtuais no ambiente simulado.

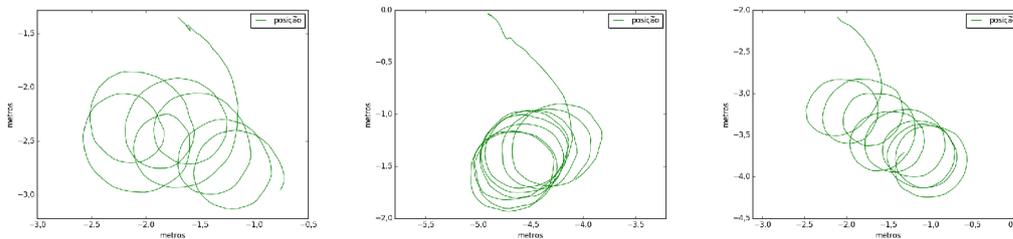
4.2.2 Resultados para movimento lateral

Durante essa etapa, foram realizados três testes de movimento circular com o *Lutra Airboat* em campo, sendo que as trajetórias coletadas são apresentadas nas imagens da Figura 4.3a. Nessas imagens pode-se observar que o movimento da embarcação possui um comportamento circular. As componentes x e y de cada trajetória no domínio do tempo são apresentadas na Figura 4.3b, sendo que, nesses gráficos, o comportamento circular pode ser visto por meio das características senoidais dos valores. Para facilitar o entendimento do comportamento da trajetória da embarcação e dos ruídos que afetam o movimento, optou-se por trocar a representação dos dados do domínio do tempo para o domínio da frequência, o que foi feito aplicando-se a Transformada de Fourier.

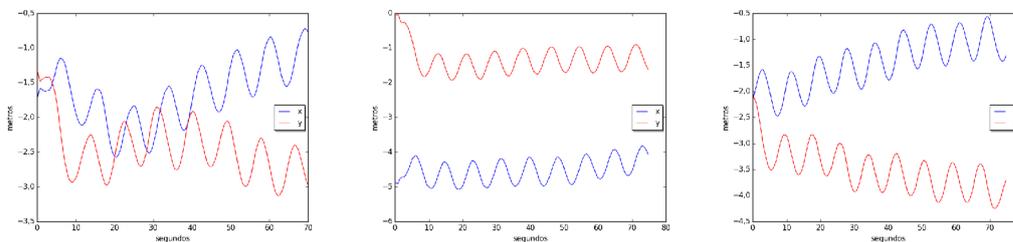
Os componentes x e y no domínio da frequência são apresentadas na Figura 4.3c, sendo que, nos três testes, as frequências para x e y são 0,12 Hz, os módulos da coordenada x são 0,37 m, 0,41 m e 0,38 m, os módulos da coordenada y são 0,32 m, 0,35 m e 0,42 m. Conclui-se, a partir desses valores, que existe uma frequência padrão próxima

a 0,12 Hz na trajetória circular do *Lutra Airboat*, sendo que o raio do círculo, gerado pelo movimento do barco, fica entre 0,32 m e 0,47 m no primeiro teste, entre 0,35 m e 0,41 m no segundo teste, entre 0,38 m e 0,48 m no terceiro teste.

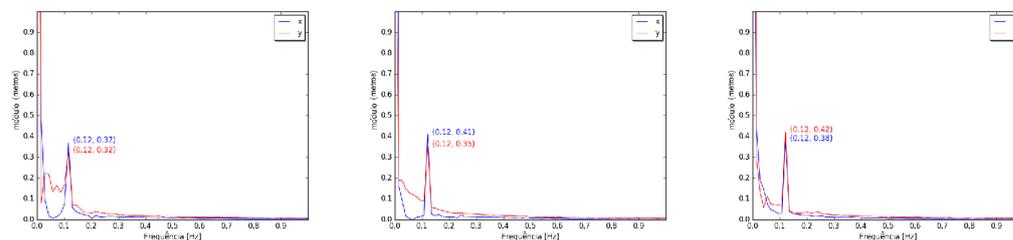
Contudo, percebe-se ainda a presença de ruídos, os quais não poderiam ser mitigados pelo pesquisador, por exemplo, ao realizar movimentos circulares, a própria embarcação produz ondulações que podem afetar o deslocamento nos próximos instantes. Conseqüentemente, os ruídos presentes nas trajetórias da embarcação podem ser observados nos pequenos valores no entorno da frequência 0,12 Hz. O valor elevado do módulo próximo a 0 Hz é consequência da embarcação não estar se deslocando próximo ao valor zero da coordenada em questão. Por exemplo, no primeiro teste, o componente x permanece sempre abaixo de -1 metros, assim esse valor aparece no gráfico da Transformada de Fourier como uma onda senoidal de módulo 1 metro com frequência que tende a zero.



(a) Trajetória do *Lutra Airboat* nos três testes em campo.



(b) Componentes x e y da posição do *Lutra Airboat* no domínio do tempo (segundos).

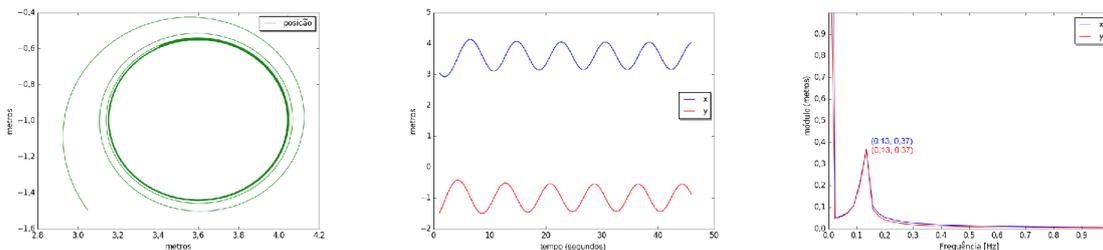


(c) Componentes x e y da posição do *Lutra Airboat* no domínio da frequência (Hz).

Figura 4.3: Resultados de três testes em campo com o *Lutra Airboat*, no qual foram realizadas curvas no sentido horário. São apresentadas as trajetórias (Figura 4.3a), as componentes da trajetória ao longo do tempo (Figura 4.3b) e a frequência e amplitude do movimento da embarcação (Figura 4.3c).

Ao replicar o teste real no ambiente virtual, obteve-se a trajetória da Figura 4.4a. Nela, percebe-se a ausência dos ruídos que interferem nos dados coletados do barco real,

sendo que a trajetória é perfeitamente circular. Para viabilizar a comparação com o robô real, realizou-se a conversão da representação do domínio do tempo (Figura 4.4b) para o domínio da frequência (Figura 4.4c). Com a Transformada de Fourier, pode-se perceber que as frequências dos componentes x e y da trajetória do barco simulado ficam em 0,13 Hz, ou seja, apenas 0,01 Hz de diferença para os testes com robôs reais. Já o módulo ficou em 0,37 metros para as duas componentes, ou seja, a circunferência da trajetória do barco simulado tem 0,37 metros de raio. Portanto, as circunferências do barco simulado se aproximam das obtidas nos testes com o *Lutra Airboat* tanto em frequência quanto em módulo.



(a) Trajetória do airboat simulado com os parâmetros extraídos dos testes em campo.

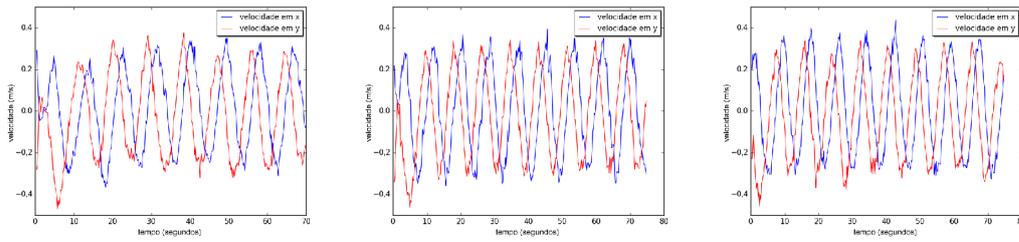
(b) Componentes x e y da posição do airboat simulado no domínio da frequência (Hz).

(c) Componentes x e y da posição do airboat simulado no domínio da frequência (Hz).

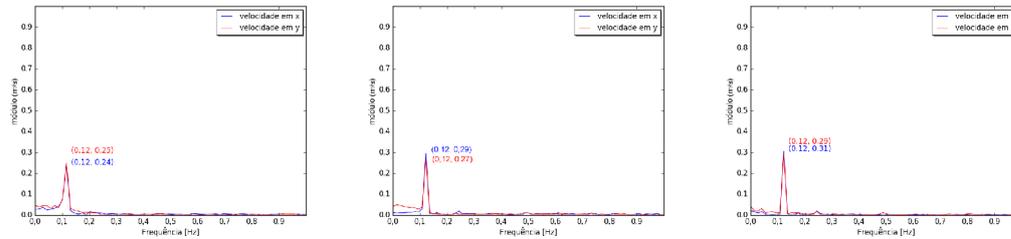
Figura 4.4: Resultado da simulação do airboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas a trajetória (Figura 4.4a) e suas componentes ao longo do tempo (Figura 4.4b), a frequência e amplitude do movimento da embarcação (Figura 4.4c).

Já do ponto de vista da velocidade da embarcação real, os gráficos que reproduzem as suas componentes x e y no domínio do tempo são apresentados na Figura 4.5a. De maneira similar aos gráficos da posição, os da velocidade foram convertidos do domínio do tempo para o domínio da frequência, sendo apresentados na Figura 4.5b. A frequência das componentes da velocidade são de 0.12 Hz, enquanto que o módulo assumiu os seguintes valores entre 0.24 m/s e 0.31 m/s. Enquanto isso, as componentes da velocidade do airboat simulado são apresentados na Figura 4.6a e os gráficos correspondentes no domínio da frequência estão na Figura 4.6b. Observa-se que a frequência obtida em simulação foi de 0.13 Hz e o módulo teve valor de 0.28 m/s. Portanto, a frequência está similar a da embarcação real, enquanto que o módulo se aproxima do segundo e terceiro teste.

Os resultados de três testes em campo para o movimento circular do *Lutra Prop* são apresentados nas Figuras 4.7 e 4.9, sendo que na primeira estão os dados relativos a posição, enquanto na última relativos a velocidade. Na Figura 4.7a são apresentadas as trajetórias da embarcação real em cada teste, enquanto que, na Figura 4.7b são apresentadas as componentes x e y de cada trajetória no domínio do tempo. Pode-se observar o comportamento circular das trajetórias por meio das características senoidais dos valores de x e y . Similarmente ao realizado para o *Lutra Airboat*, optou-se por trocar a representação

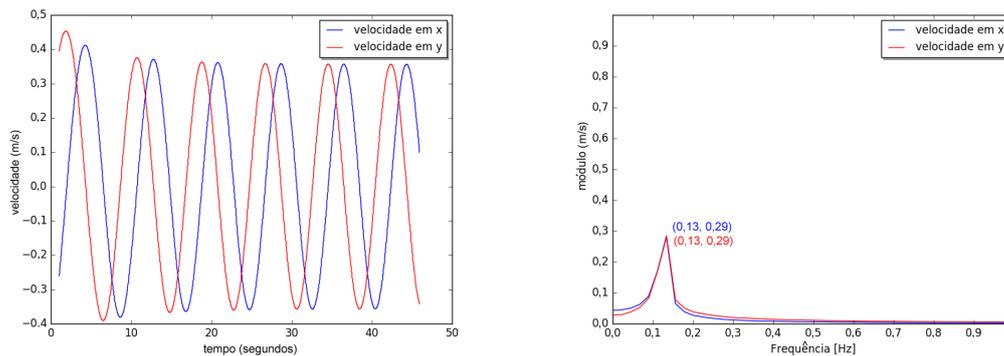


(a) Componentes x e y da velocidade do *Lutra Airboat* no domínio do tempo (segundos).



(b) Componentes x e y da velocidade do *Lutra Airboat* no domínio da frequência (Hz).

Figura 4.5: Resultado da velocidade do *Lutra Airboat* ao longo dos três testes em campo. Em (a) são apresentados as componentes da velocidade no domínio do tempo, enquanto que em (b) são apresentados as componentes no domínio da frequência.



(a) Componentes X e Y da velocidade do airboat simulado no domínio do tempo (segundos).

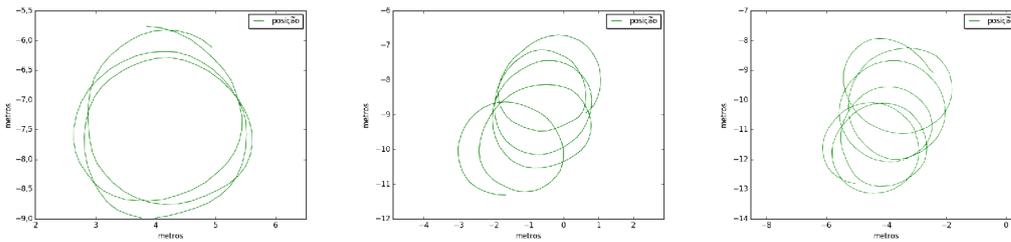
(b) Componentes X e Y da velocidade do airboat simulado no domínio da frequência (Hz).

Figura 4.6: Resultado da simulação do airboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas as componentes da velocidade ao longo do tempo (Figura 4.6a) e as componentes da velocidade no domínio da frequência (Figura 4.6b).

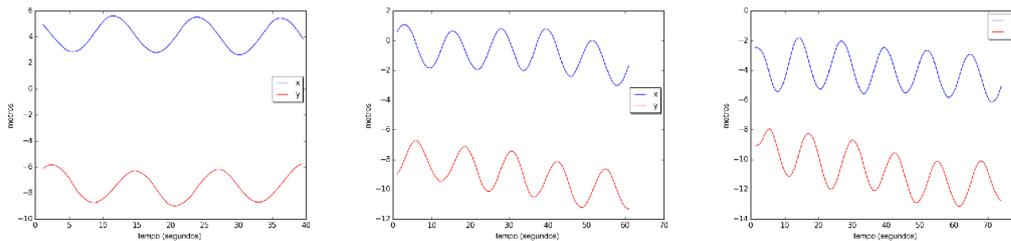
dos dados do domínio do tempo para o domínio da frequência por meio da transformada de Fourier. Essa representação encontra-se na Figura 4.7c.

Os componentes x e y no domínio da frequência são apresentadas na Figura 4.7c, sendo que, nos três testes, as frequências para x e y são 0,08 Hz, os módulos da coordenada x são 1,31 m, 1,48 m, 1,53 m, os módulos da coordenada y são 1,33 m, 1,38 m, 1,48 m. Conclui-se, a partir desses valores, que existe uma frequência padrão próxima a 0,08 Hz na trajetória circular do *Lutra Airboat*, sendo que o raio do círculo, gerado pelo movimento

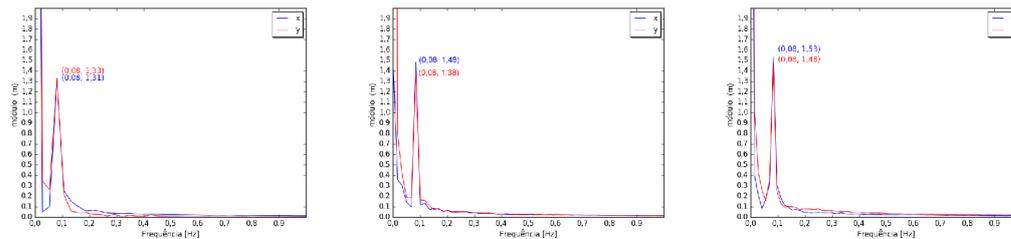
do barco, fica entre 1,31 m e 1,33 m no primeiro teste, entre 1,38 m e 1,48 m no segundo teste, entre 1,48 m e 1,53 m no terceiro teste. Os ruídos presentes nas trajetórias da embarcação podem ser observados nos pequenos valores no entorno da frequência 0,08 Hz, o valor elevado do módulo próximo a 0 Hz é consequência da embarcação não estar se deslocando próximo ao valor zero da coordenada em questão. Por exemplo, no primeiro teste, o componente x (Figura 4.7a) permanece sempre acima de 2 metros, assim esse valor aparece no gráfico da Transformada de Fourier como uma onda senoidal de módulo 2 metro com frequência que tende a zero.



(a) Trajetória do *Lutra Prop* nos três testes em campo.



(b) Componentes x e y da posição do *Lutra Prop* no domínio do tempo (segundos).

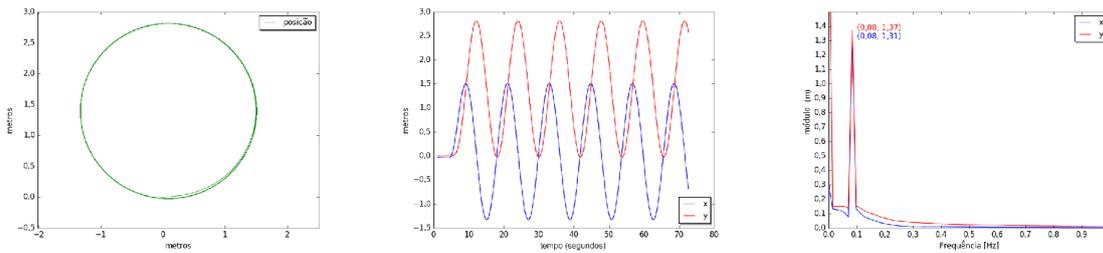


(c) Componentes x e y da posição do *Lutra Prop* no domínio da frequência (Hz).

Figura 4.7: Resultados de três testes em campo com o *Lutra Prop*, no qual foram realizadas curvas no sentido anti-horário. São apresentadas as trajetórias (Figura 4.7a), as componentes da trajetória ao longo do tempo (Figura 4.7b) e a frequência e amplitude do movimento da embarcação (Figura 4.7c).

Com base nos dados coletados nesses três testes em campo, foi conduzido um teste no mundo virtual. A trajetória obtida pelo *Differential Boat* é apresentada na Figura 4.8a, podendo-se observar que ausência de ruídos da coleta de dados do barco real. Para viabilizar a comparação com o robô real, realizou-se a conversão da representação do domínio do tempo (Figura 4.8b) para o domínio da frequência (Figura 4.8c). Com a Transformada de Fourier, pode-se perceber que as frequências dos componentes x e y da

trajetória do barco simulado ficaram em 0,08 Hz, ou seja, a mesma obtida nos testes com robôs reais. Já o módulo ficou em 1,31 m e 1,37 m para as componentes x e y , ou seja, a circunferência da trajetória do barco simulado tem entre 1,31 m e 1,37 m de raio. Portanto, as circunferências do barco simulado se aproximam das obtidas nos testes com o *Lutra Prop*.



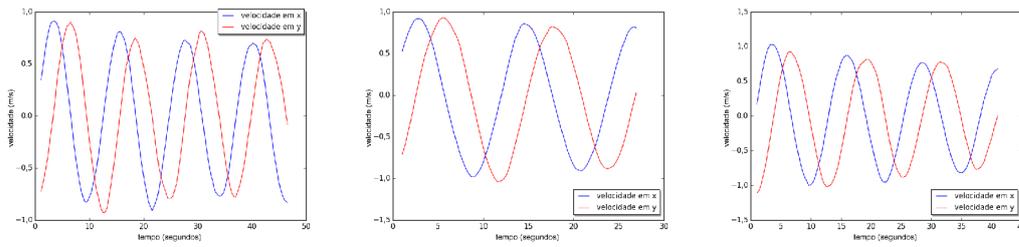
(a) Trajetória do diffboat simulado com os parâmetros extraídos dos testes em campo.

(b) Componentes x e y da posição do diffboat simulado no domínio da frequência (Hz).

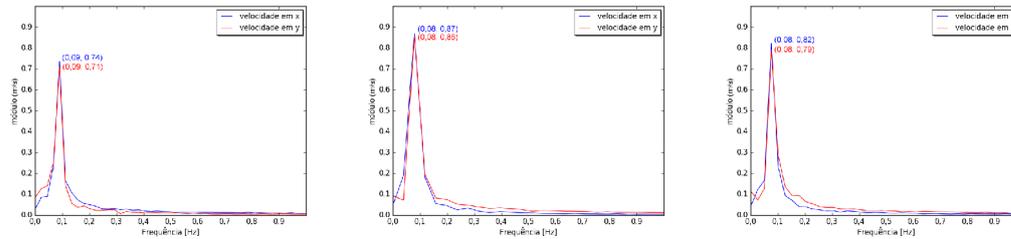
(c) Componentes x e y da posição do diffboat simulado no domínio da frequência (Hz).

Figura 4.8: Resultado da simulação do airboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas a trajetória (Figura 4.8a) e suas componentes ao longo do tempo (Figura 4.8b), a frequência e amplitude do movimento da embarcação (Figura 4.8c).

Com relação a velocidade da embarcação real, os gráficos que reproduzem as suas componentes x e y no domínio do tempo são apresentados na Figura 4.9a. De maneira similar aos gráficos da posição, os da velocidade foram convertidos do domínio do tempo para o domínio da frequência, sendo apresentados na Figura 4.9b. A frequência das componentes da velocidade são de 0,08 Hz e 0,09 Hz. O módulo da velocidade em x assumiu os seguintes valores entre 0,74 m/s e 0,87 m/s, enquanto que o módulo da velocidade em y assumiu valores entre 0,71 m/s e 0,86 m/s. Enquanto isso, as componentes da velocidade do USV simulado são apresentados na Figura 4.10a e os gráficos correspondentes no domínio da frequência estão na Figura 4.10b. Observa-se que a frequência obtida em simulação foi de 0.08 Hz e o módulo da velocidade teve valor 0,68 m/s e 0,67 m/s nos eixos x e y respectivamente. Portanto, apesar da frequência ser similar a da embarcação real, os módulos da velocidade encontram-se abaixo dos observados nos testes no mundo real. Isso pode estar relacionado ao fato das dificuldades em medir a força de propulsão do *Lutra Prop*, já que um eixo do propulsor encontrava-se parcialmente danificado. Assim, torna-se necessário realizar novos experimentos com esse USV após a substituição ou reparo do propulsor.

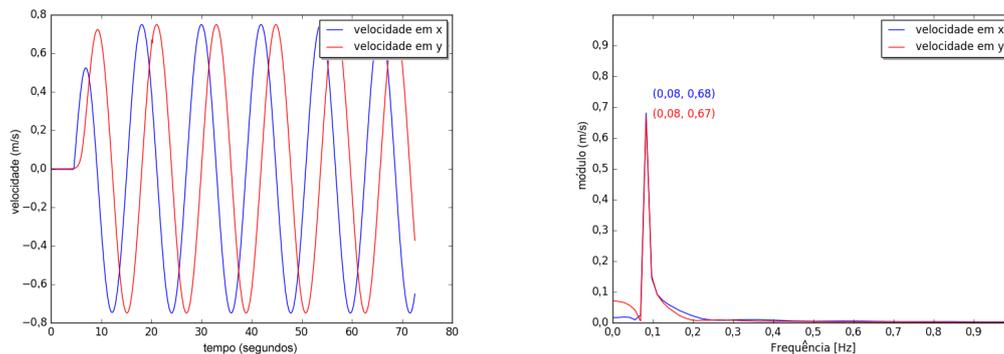


(a) Componentes x e y da velocidade do *Lutra Prop* no domínio do tempo (segundos).



(b) Componentes x e y da velocidade do *Lutra Prop* no domínio da frequência (Hz).

Figura 4.9: Resultado da velocidade do *Lutra Prop* ao longo dos três testes em campo. Em (a) são apresentados as componentes da velocidade no domínio do tempo, enquanto que em (b) são apresentados as componentes no domínio da frequência.



(a) Componentes X e Y da velocidade do diffboat simulado no domínio do tempo (segundos).

(b) Componentes X e Y da velocidade do diffboat simulado no domínio da frequência (Hz).

Figura 4.10: Resultado da simulação do diffboat utilizando os parâmetros extraídos dos testes em campo. São apresentadas as componentes da velocidade ao longo do tempo (Figura 4.10a) e as componentes da velocidade no domínio da frequência (Figura 4.10b).

4.3 Outras considerações

Durante a realização das atividades descritas ao longo desse capítulo, foram enfrentadas diferentes dificuldades, sendo algumas já pontuadas em seções anteriores. Dentre elas, tem se a pane elétrica do *Lutra Prop* devido ao excesso de corrente elétrica (ultra-

passando 80 amperes), o qual demandou a reconfiguração dos módulos ESC da embarcação, a substituição de parte do cabeamento e do fusível rompido.

Mesmo após a diminuição de potência do Lutra Prop para sanar o problema de excesso de corrente elétrica, a embarcação continuava aplicando altas rotações nas hélices, provocando uma inclinação (*pitch*) elevada e o surgimento de muitas bolhas atrás do USV. Para melhorar o desempenho e a eficiência, foi realizada uma nova diminuição de potência e substituição das hélices por uma com área maior. Após as modificações, que demandou a repetição dos testes de validação, o Lutra prop passou a consumir apenas 2,5 amperes em máxima aceleração, conseguindo se aproximar da velocidade registrada inicialmente. Com essas alterações, apesar de não ter sido realizadas medições, acredita-se que a autonomia da embarcação tenha sido ampliada.

Nas primeiras tentativas de realização dos testes de validação, percebeu-se que os comandos de PWM enviados diretamente do controle remoto eram modificados pelo software da embarcação, ou seja, os valores enviados para os módulos ESCs não eram os mesmos que haviam partido do controle remoto. Por exemplo, no teste de movimento circular (Seção 4.1.4), não era possível controlar isoladamente cada propulsor, nem manter os valores de PWM constantes ao longo de todo o teste. Portanto, não havia como garantir que as forças de propulsão permaneceriam constantes ao longo de cada teste de validação (Seções 4.1.2, 4.1.3 e 4.1.4).

Devido a esse problema, o software da embarcação foi modificado para também receber comandos PWM e repassá-los sem alterações para os módulos ESCs. Dessa forma, os comandos de PWM podiam ser enviados da GroundStation, utilizando tanto as mensagens disponíveis no MavROS quanto no MAVLINK.

Outra dificuldade encontrada foi o fato das embarcações sempre realizarem curvas, mesmo quando aceleradas apenas frontalmente. Acredita-se que a forma como a embarcação foi construída e seus componentes foram organizados, impacta diretamente na trajetória gerada. Por exemplo, a posição da bateria dentro do compartimento interno, bem como a fixação da antena sobre o casco modificam o centro de massa da embarcação. Para minimizar o impacto desse problema, antes de iniciar as coletas de dados, buscou-se posicionar os componentes de forma a reduzir esse efeito. Destaca-se que, apesar dos componentes estarem fixados no compartimento interno e na antena externa, era necessário a revisão e "regulagem", pois o deslocamento até o local de teste e a substituição das baterias afetam o centro de massa da embarcação.

Durante a realização do teste de movimento circular, percebeu-se que os valores lidos pela bússola da embarcação eram afetados pela interferência eletromagnética da aceleração dos propulsores. No caso do airboat, a média de erro de direção do compass estava em torno de 13 graus, podendo atingir em alguns casos até 20 graus. Para sanar esse tipo de problema, a primeira ação tomada foi a tentativa de isolamento eletromagnético, o que

mostrou-se insuficiente. Por fim, optou-se pela instalação de uma bússola externa a embarcação para a realização do teste de validação com movimento circular (Seção 4.1.4).

Já o teste de flutuabilidade em ondas havia sido inicialmente planejado, mas mostrou-se inviável. Por meio desse teste, esperava-se comparar os efeitos das ondas nos barcos reais e nos virtuais. Para tal, os USVs reais seriam colocados em tanques de água com geradores artificiais de ondas, suas trajetórias e rotações seriam capturadas para posterior comparação com os modelos virtuais. Apesar de se ter conseguido acesso a um tanque com geração de ondas, o sistema de captura de movimento disponível no laboratório demanda que a superfície, em que o objeto será localizado, seja não reflexiva, o que não é o caso das superfícies da água. Consequentemente, a limitação em capturar corretamente o movimento das embarcações impediu a realização desse tipo de teste.

Por fim, apesar de terem sido tomadas ações para minimizar erros e ruídos na coleta de dados, percebe-se que ainda permanecem diferenças entre os dados coletados e os simulados. Essas diferenças devem-se principalmente pelos seguintes fatores: o erro gerado pelo não alinhamento do centro de massa e do centro de flutuabilidade da embarcação, ruídos nos sensores da embarcação e simplificações nos modelos matemáticos utilizados nas simulações.

5. ESTUDOS DE CASOS

Esse capítulo demonstra os efeitos resultantes do simulador para os quatro modelos de USVs distribuídos nele, também apresenta como o modelo de simulação proposto impacta em diferentes atividades geralmente realizadas por USVs. As Seções 5.1 e 5.2 mostram dois exemplos de como o modelo de vento e de água podem afetar a trajetórias das embarcações. Já a Seção 5.3 apresenta como um barco controlado por leme pode ser afetado ao descer um rio.

A Seção 5.4 compara como cada distúrbio pode afetar a trajetória do barco diferencial. A avaliação da performance do veleiro de acordo com diferentes módulos e direções de vento pode ser verificada na Seção 5.5. Já na Seção 5.6, é avaliada a performance dos quatro USVs para três diferentes tarefas a fim de comparar a trajetória obtida de cada um deles. Além disso, essa seção também apresenta como a trajetória é modificada quando as correntes de vento e de água são ativadas na simulação.

A Seção 5.7 apresenta como a modularidade do simulador permite a realização de um experimento de *coverage path-planning* por meio da reutilização de código. Na Seção 5.8, tem-se o desempenho de um USV (com um controlador simples), enquanto ele realiza uma inspeção de ponte após a ocorrência de uma situação de desastre natural, assim a inspeção é simulada com fortes correntes de água e de vento. A última seção, descreve o experimento com um USVs navegando entre prédios numa região urbana inundada.

5.1 Embarcação em situações de distúrbios variáveis

A presente seção descreve um experimento em que um barco diferencial atravessa um rio contendo os distúrbios apresentados na Figura 3.13. Portanto, observa-se a habilidade única do simulador robótico, ao aplicar distúrbios conforme a posição da embarcação, utilizado-se um modelo hidrológico de um corpo de água. A Figura 5.1 mostra que a embarcação começa no lado direito da imagem (posição 440m no eixo horizontal e 110m no eixo vertical) e movimenta-se até o topo esquerdo da imagem.

A embarcação tem maior velocidade quando passa pela região mais estreita do canal, onde a correnteza de água é mais rápida. A medida em que a embarcação atinge regiões de águas mais abertas, a velocidade do USV diminui. Nesse teste, a embarcação percorreu 452,38 metros de distâncias ao longo de 376,98 s, o que gera uma velocidade média de 1,2 m/s e teve como maior velocidade o valor de 3,04 m/s. Assim sendo, a embarcação percorreu o trajeto movendo-se mais rápido do que observado nos testes da Seção 4.1.3. Isso porque a embarcação moveu-se a favor das correntes de vento e água, o que ajudou-a ganhar velocidade.

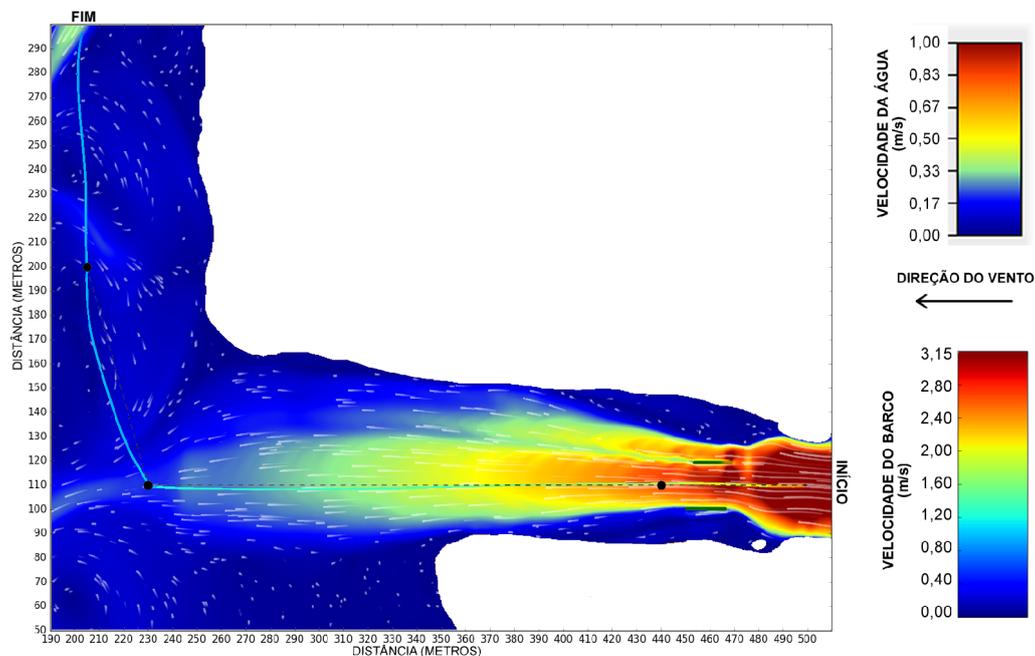


Figura 5.1: Trajetória do barco diferencial no ambiente da Figura 3.13. A cor da trajetória representa a velocidade da embarcação ao longo do caminho.

5.2 Rio com diferentes correntes de água

Essa seção apresenta como o simulador é coerente com orientações básicas de navegação em rios, ou seja, embarcações navegando rio acima devem se manter próximos a margens dos rios, onde a corrente de água é menor que no meio do canal. Por outro lado, barcos movendo-se rio abaixo devem manter-se no meio, o que permite poupar combustível, já que tem a vantagem a correnteza do rio estar a favor do seu movimento [Mig05].

Assim, a Figura 5.2 apresenta duas trajetórias diferentes percorridas pelo barco diferencial: uma com a linha em rosa, próxima a margem do rio, e outra vermelha no meio do canal do rio. Ao longo da trajetória rosa, a velocidade da corrente de água é baixa, próxima a 0,17 m/s, enquanto que, ao longo da trajetória vermelha, tem-se fortes correntezas de água. Ambas trajetórias iniciam no lado esquerdo da imagem, tendo a corrente de água oposta a direção de navegação da embarcação. Para percorrer a trajetória rosa, o barco demorou 144,9 segundos, tendo como velocidade média 1,10 m/s. Já a trajetória vermelha levou 187,2 segundos para ser percorrida, tendo como velocidade média 0,85 m/s.

A segunda parte da Figura 5.2 apresenta a velocidade da embarcação para ambas as trajetórias ao longo do tempo. Pode-se observar que a trajetória próxima a margem do rio (linha rosa) mantém uma velocidade quase uniforme entre 1,1 e 1,3 m/s. Contudo, a trajetória vermelha (centro do canal) apresenta uma clara redução na velocidade da embarcação a medida que se aproxima da região com correntes mais fortes de água. A velocidade

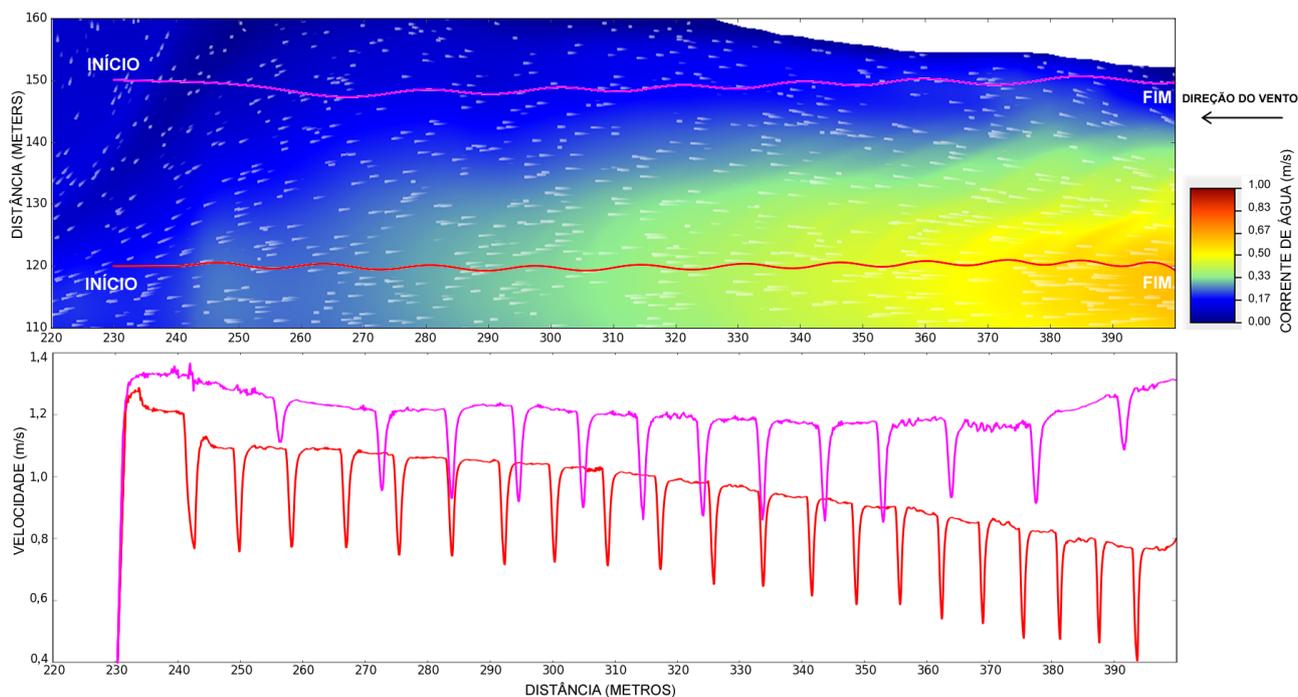


Figura 5.2: Diferentes velocidades de navegação do barco diferencial navegando rio acima, contrário a correnteza de água.

inicia em torno de 1,2 m/s, mas termina em 0,8 m/s. Ambas as trajetórias tem reduções na velocidade, provocadas pelo controle ao tentar corrigir a trajetória e os deslocamentos laterais. Porém, a embarcação próxima ao leito do rio apresenta menos reduções do que a embarcação no meio do canal, uma vez que a corrente de água possui menos turbulência e menores velocidades que no meio do rio. Além disso, nota-se na trajetória e na velocidade destacada em rosa, que o controlador atua menos para efetuar correções no trajeto. Isso é visto tanto na amplitude, quanto na frequência de quedas de velocidade na linha rosa em relação a linha vermelha, por exemplo, ocorrem 13 quedas na velocidade em rosa, enquanto que na vermelha ocorrem 21 quedas.

5.3 Rudderboat descendo rio

Outro teste que pode-se realizar, de forma a mostrar que o simulador é coerente com orientações básicas de navegação em rios, está relacionada com a manobrabilidade de embarcações com leme, as quais estejam movendo-se a favor da correnteza de água [Mig05]. Portanto, caso a embarcação navegue em velocidade similar a da correnteza de água, a manobrabilidade será prejudicada, ou seja, a embarcação terá dificuldades em evitar colisões. Isso ocorre pois a velocidade aparente de água no leme é praticamente nula, fazendo com que as forças de sustentação e arrasto tendam a zero. Isso faz com que a embarcação tenha dificuldades em realizar curvas e manobras. Para testar se o simulador

é coerente com essa orientação, foi proposto um teste em que o *rudderboat* deveria descer o rio de forma a alcançar os *waypoints* $P1(480, 120)$, $P2(476, 125)$ e $P3(460, 125)$ e desviar de um pilar localizados no cenário (ver Figura 5.5).

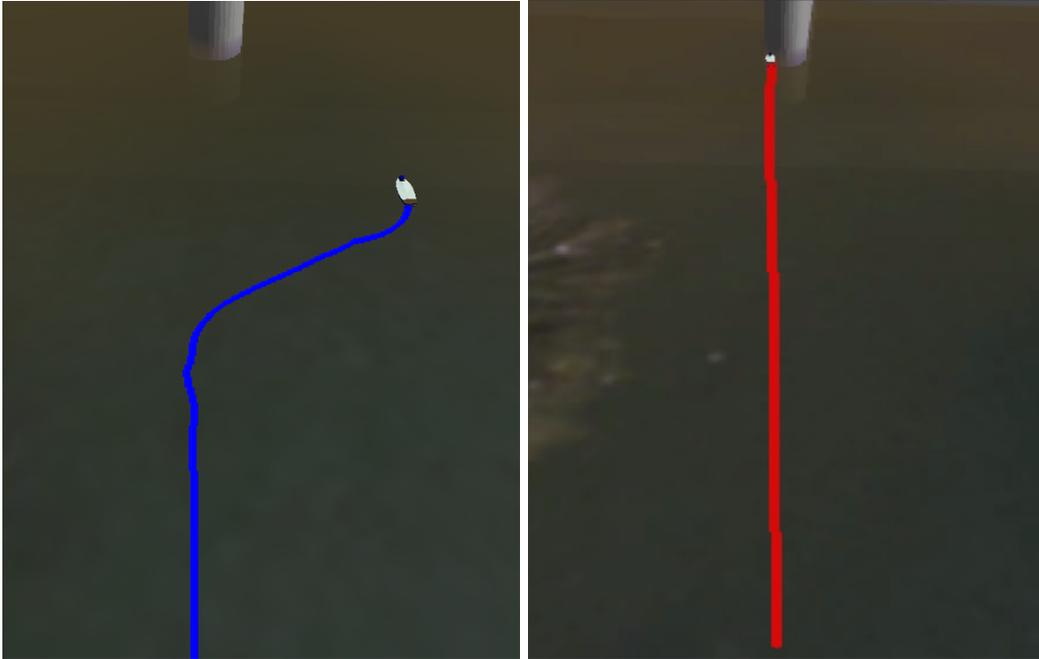


Figura 5.3: Trajetória do *rudderboat* descendo o rio sem e com correnteza de água. Trajetória em azul: a embarcação navega sem correnteza de água. Trajetória em vermelho: a embarcação navega a favor da correnteza de água com a mesma velocidade da água (módulo da velocidade da água é 1m/s; a direção/sentido é de baixo para cima).

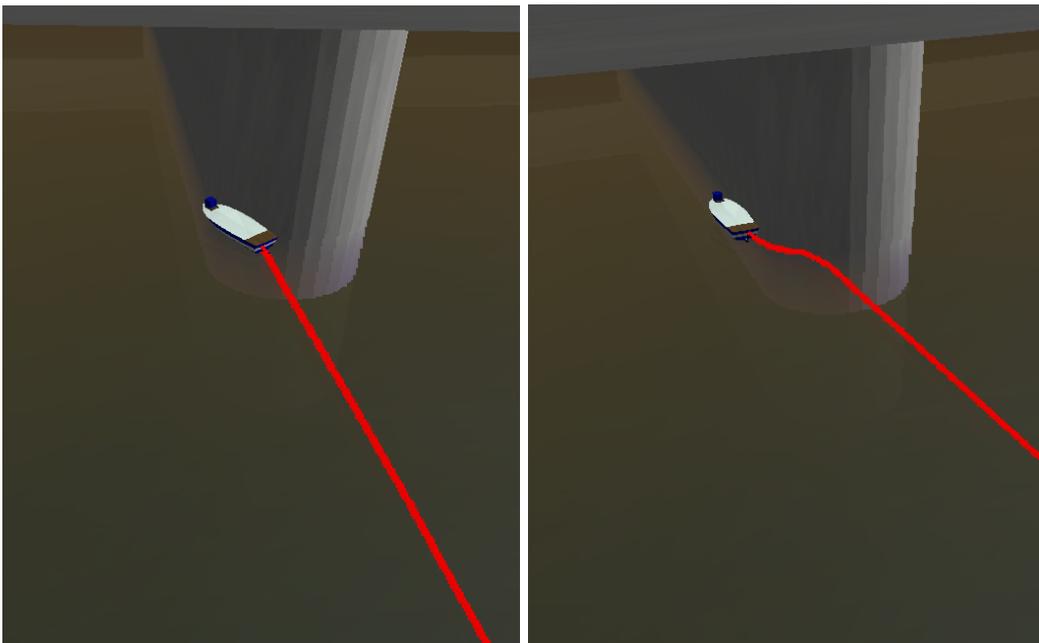


Figura 5.4: *Rudderboat* descendo o rio com a mesma velocidade da correnteza de água. Imagens apresentam a colisão da embarcação com o pilar da ponte.

Foram reproduzidas duas situações: sem correnteza de água e com uma correnteza de água a 1 m/s. A trajetória da embarcação sem correnteza de água está marcada em azul na imagem esquerda da Figura 5.3, enquanto que a trajetória da embarcação navegando a favor da correnteza pode ser vista destacada em vermelho. Observa-se que, na ausência de correnteza, a embarcação consegue manobrar de forma a evitar a colisão com o pilar da ponte. Além disso, na Figura 5.4 é apresentada o instante em que a embarcação colide com a ponte. A trajetória das embarcação ao longo do percurso pode ser observado na Figura 5.5, já a posição do leme ao longo do tempo para cada um dos testes é apresentado na Figura 5.6. No caso da embarcação navegando com correnteza de água, o algoritmo de controle utilizado no leme atinge o ponto de saturação (60°) aos 18 segundos da simulação, mantendo-se nesse valor até a colisão no instante 34 segundos. Ao colidir, o controle busca fazer uma curva a esquerda, fazendo com que o novo valor do leme torne-se -60° .

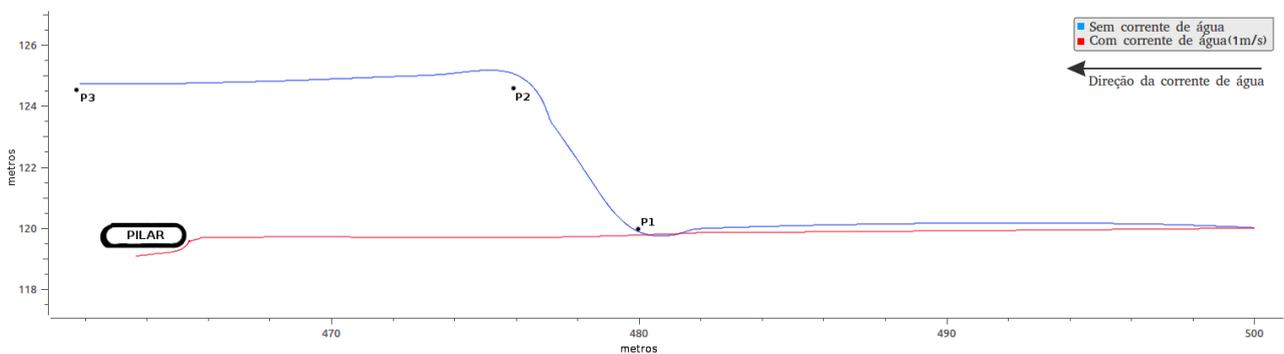


Figura 5.5: Trajetórias do *Rudderboat* ao descer o rio com corrente de água (linha azul) e sem corrente de água (linha verde). Embarcações partem lado direito da imagem.

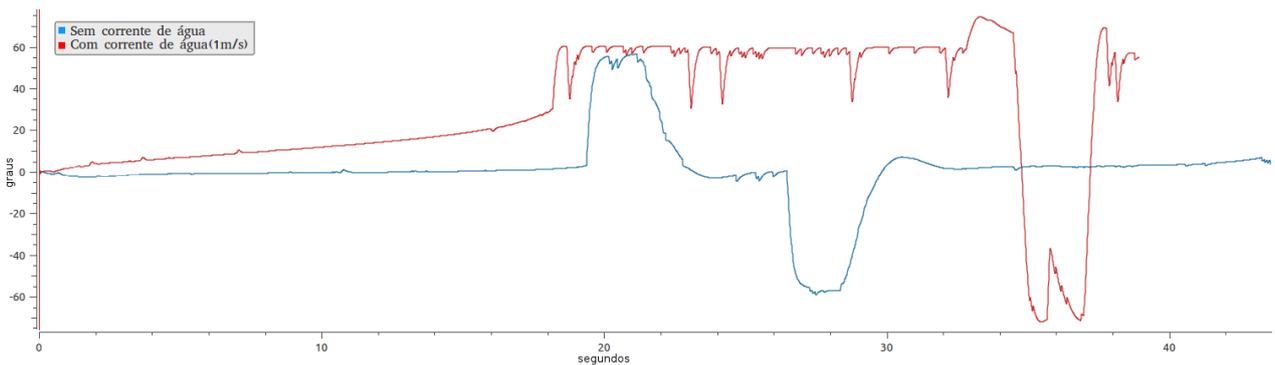


Figura 5.6: Ângulo do leme ao longo tempo para cada um dos *Rudderboats* descendo o rio.

5.4 Efeitos dos distúrbios sobre o barco diferencial

A Figura 5.7 apresenta a trajetória resultante do barco diferencial sobre ação das forças do vento e da água. Se todos os distúrbios ficam desligados, a embarcação percorre

uma linha perfeitamente reta até o destino. A imagem da figura apresenta as trajetórias para as seguintes variações: apenas a correnteza de água (linha vermelha); apenas correnteza de vento (linha rosa); ambas as correntes (linha verde). Pode-se observar que, ao receber ação do vento e da água simultaneamente, a trajetória da embarcação fica mais distorcida do que quando apenas um distúrbio está presente. Além disso, observa-se que a turbulência dos fluxos de água perturba o movimento da embarcação ao longo do percurso, o que não é observado nos modelos de USV presentes em outros simuladores.

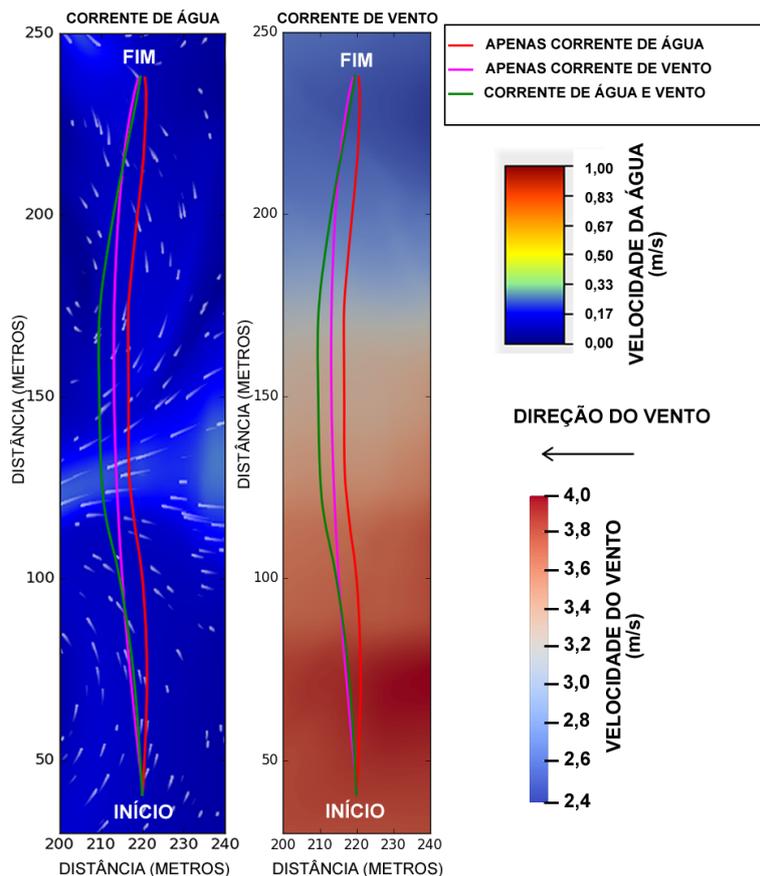


Figura 5.7: Barco diferencial navegando em linha reta. A linha vermelha apresenta a trajetória da embarcação quando afetada apenas corrente de água, a linha rosa quando afetado apenas pelo vento. A trajetória verde quando a embarcação é afetado simultaneamente pelo vento e água.

5.5 Desempenho do veleiro

Tendo como finalidade demonstrar todas as características do simulador proposto, o modelo do veleiro é particularmente representativo, já que seu desempenho está diretamente relacionado a precisão dos modelos de vento, ondas e corrente de água. Assim, o desempenho de um veleiro é geralmente descrito por meio de um diagrama polar, o qual é exemplificado na Figura 5.8a que mostra a velocidade de um *yacht* real de competição

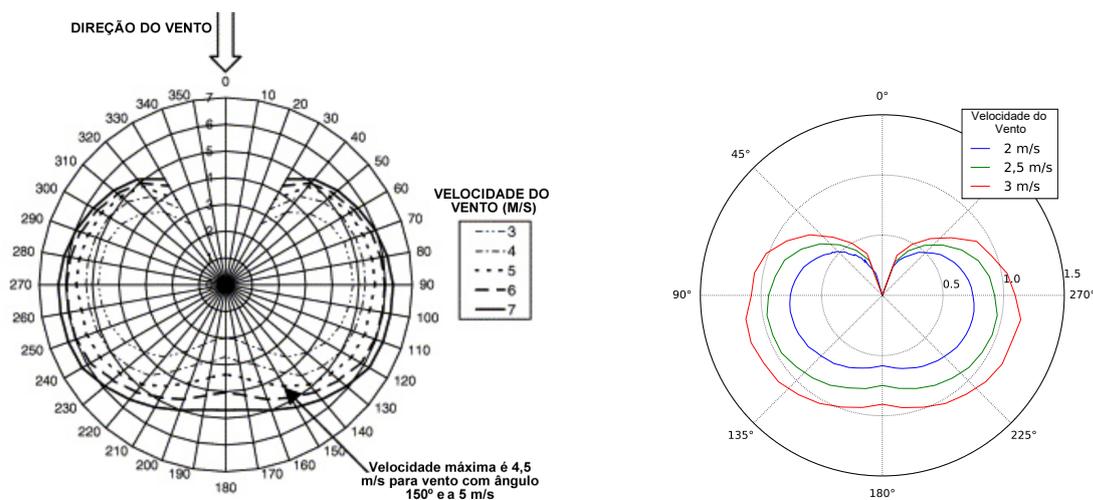
[RJS01]. Com o diagrama, pode-se perceber a velocidade máxima para diferentes condições do vento (módulo e direção), por exemplo, para um vento de 5 m/s fazendo ângulo de 150° com o *yatch*, a velocidade máxima é de 4,5 m/s.

No caso do diagrama polar da Figura 5.8b, é apresentado o desempenho do *Sailboat* modelado nesse simulador, incorporando a cinemática do plugin Foil Dynamics. O veleiro possui um perfil padrão, no qual a velocidade é zero ao tentar navegar contra o vento (0°) e não diretamente a favor do vento (180°). Além disso, os veleiros não conseguem navegar na direção de -30° and 30° , o mesmo ocorre no veleiro modelado no simulador. Essas características, similares ao da embarcação real, demonstram o realismo do modelo de simulação de veleiros.

5.6 Comparação entre os veículos

Inspirados nas competições de USV, como o *Maritime Robotx Challenge* [Rob20], o *International Robotic Sailing Championship* e o *World Robotic Sailing Championship* [WRS20], foi incluído um cenário de competição simplificado. Esses cenários, ilustrados na Figura 5.9, compreendem tarefas relevantes para testar o desempenho de USVs, como navegação em canais estreitos (cenário 1), prevenção de obstáculos (cenário 2) e missão de cobertura/mapeamento de área (cenário 3).

A Tabela 5.1 resume os três cenários apresentados na Figura 5.10, onde os quatro barcos são avaliados em relação ao tempo total e à distância percorrida para atingir a meta. O objetivo desses resultados é mostrar a complexidade de tais análises e comparações.



(a) Exemplo de diagrama polar para um veleiro real. Fonte: Richards [RJS01].

(b) Diagrama polar do *Sailboat*.

Figura 5.8: Da esquerda para a direita, diagramas polares de um veleiro real [RJS01] e do modelo de simulação do veleiro contendo uma área de vela de 2m^2 .

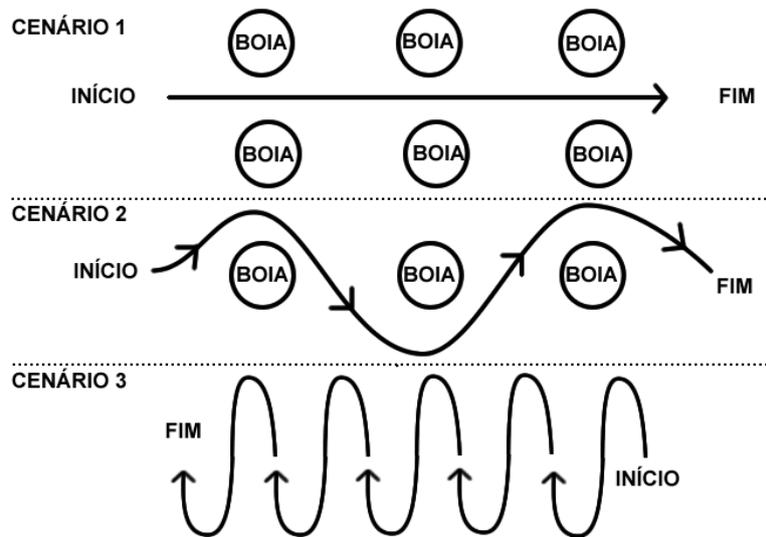


Figura 5.9: Cenários propostos para avaliação.

Apesar dos barcos possuírem a mesma estratégia de controle e o mesmo casco, existem alguns aspectos que dificultam a comparação entre as embarcações. Como exemplo disso tem-se o tipo de propulsão e os elementos de estabilização, como a quilha.

Portanto, é mais factível comparar barcos com propulsão e construção semelhantes. Embora os barcos tenham a mesma estratégia de controle com base na orientação, os parâmetros de configuração do controlador de cada USV possuem valores próprios, os quais podem afetar o desempenho da trajetória da embarcação. Por isso, um tipo de USV pode ser mais adequado para uma determinada tarefa do que outro. Isso pode ser observado nos resultados da Figura 5.10, em que o *Airboat* possui pior desempenho para realizar a tarefa de cobertura de uma área (cenário 3).

O Veleiro mostrou-se lento para os cenários testados, mas sua trajetória não foi muito afetada pelos distúrbios por causa da quilha e de seu peso. Isso é promissor para cenários de desastre no mar ou grandes massas de água, mas é necessário investigar se esse comportamento se repete para outras condições de água e vento. O barco com leme tem o melhor relação entre a distância percorrida e o tempo. Embora tenha um único motor, comparado ao barco diferencial, o leme ajuda a estabilizar o barco contra as correntes de água. Similarmente, requer-se mais investigação para confirmar o papel do leme para correntes de água mais rápidas.

O barco diferencial também tem um bom desempenho, apesar de ter feito algumas voltas extras no segundo cenário. Talvez seja necessário algum ajuste do controlador para o barco diferencial. Novamente, esses resultados não podem ser considerados literalmente, mas mostram a complexidade e a diversidade de ajustes necessários para melhorar o desempenho dos USVs sob distúrbios ambientais. Outras pesquisas podem se concentrar na análise individual e completa de cada um desses barcos com diferentes condições ambientais e métodos de GNC.

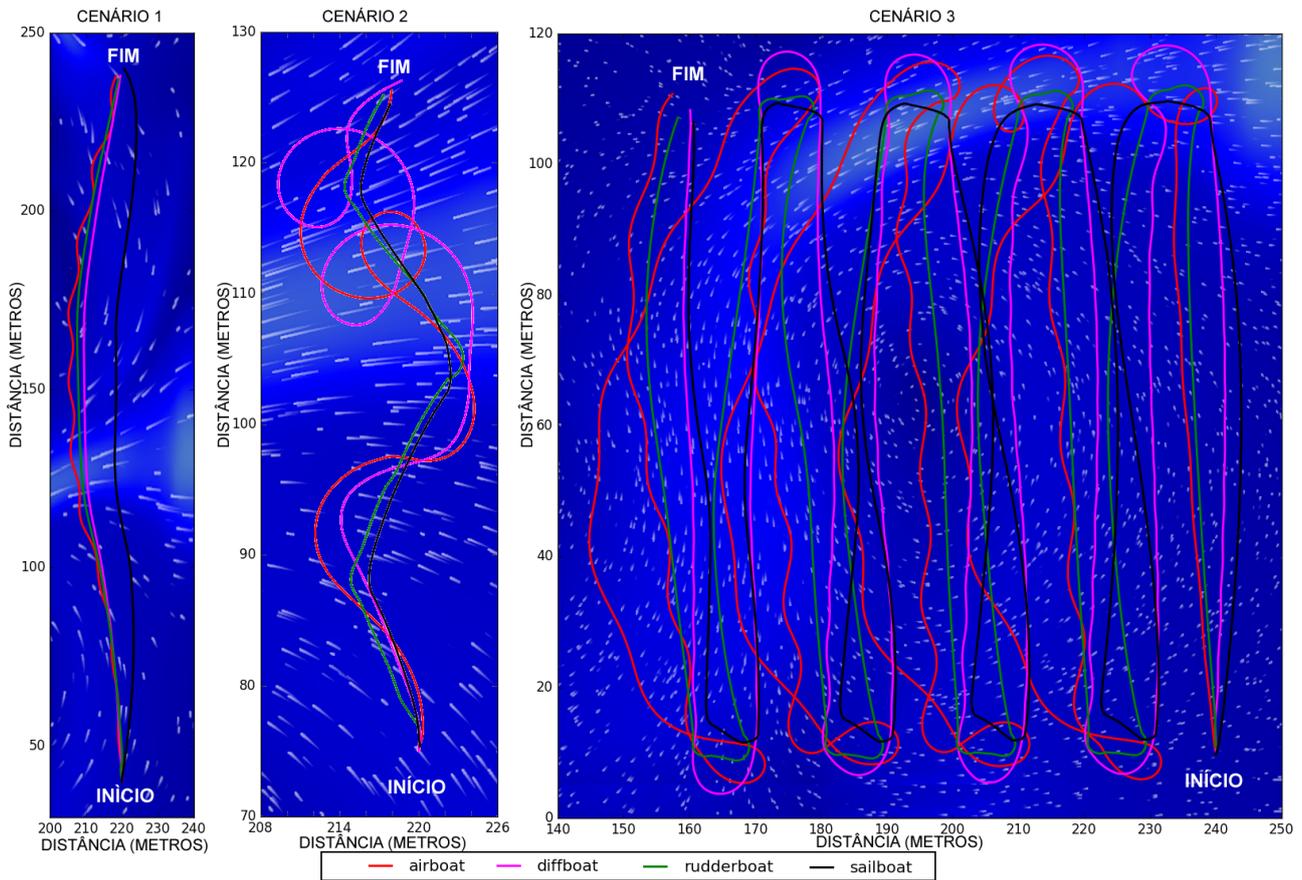


Figura 5.10: Trajetórias dos três cenários para cada um das quatro embarcações. As trajetórias do Airboat, Differential boat, Rudderboat, Sailboat são exibidas em linhas vermelhas, rosas, verdes e pretas, respectivamente.

Tabela 5.1: Estatísticas da simulação realizada com cada tipo de embarcação para todos os cenários da Figura 5.10.

Embarcação	Cenário 1		Cenário 2		Cenário 3	
	Distância (m)	Tempo (s)	Distância (m)	Tempo (s)	Distância (m)	Tempo (s)
Airboat	203	261	88	124	1177	1535
Diferencial	197	165	108	109	1043	906
Rudder boat	197	201	55	61	971	1017
Sailboat	201	326	53	93	932	1452

5.7 Planejamento de rotas de cobertura

Uma outra atividade realizada por USVs é a coleta de dados ao longo de uma região de interesse, podendo ser para a realização de batimetria, medição da qualidade de água, busca de objetos submersos. Para tal, é necessário a execução de um algoritmo de planejamento de rota de cobertura (*coverage path-planning*) que leve em consideração o ambiente a ser navegado.

No caso de realizar esse tipo de cobertura, em locais em que há a presença de distúrbios como correntes de vento e água, a direção de navegação pode ser um fator determinante na eficiência da tarefa. Nesse contexto, os estudos mais recentes em *coverage path-planning* começaram a investigar a melhor forma para realizar esse tipo de tarefa, por exemplo, Karapetyan [KBM⁺19] compara diferentes tipos de trajetórias a serem utilizadas, seja ela paralela ou perpendicular a margem do rio. Em outro trabalho, Karapetyan [KMR19] propõe um algoritmo para realizar a cobertura considerando a força da correnteza na rota a ser utilizada, a qual é sempre paralela a margem do rio. Apesar desta solução proposta considerar diversas larguras de rios, os experimentos foram realizados apenas em rios estreitos, de forma que a embarcação pode realizar a cobertura completa subindo e descendo uma única vez o rio.

Esses algoritmos ainda consideram que não deve haver nenhum tipo de obstáculo na região em que se realiza a cobertura. Se os obstáculos forem estáticos e previamente conhecidos, tais como, pedras ou pequenas ilhas, pode-se fazer a decomposição do espaço em subáreas, nas quais serão realizadas a cobertura. Já na existência de obstáculos móveis, tais como outras embarcações, carece ainda de técnicas capazes de realizar o *coverage path-planning* desviando de forma a considerar os distúrbios naturais existentes na rota, bem como respeitando as normas de navegação (como as previstas nas COLREGS [IMO72]). As COLREGS possuem as definições internacionais que as embarcações devem seguir para que não ocorram colisões tanto em cruzamentos, quanto em ultrapassagens. Essas regras contemplam também os padrões de manobras a serem seguidos, e a definição de todos os tipos de sinalização (luminosa, sonora e simbólica) para auxiliar na navegação.

Diante disso, o modelo de simulação proposto nessa tese pode se tornar uma ferramenta importante para o desenvolvimento de soluções para *coverage path-planning* na presença de obstáculos estáticos e móveis. Isso se dá ao fato da arquitetura de simulação já contar com a padronização de mensagens ROS, bem como ser compatível com o pacote de navegação *move_base* [Mar11], a qual está presente em diversos robôs.

Assim, elaborou-se um experimento em que o simulador é utilizado para a realizar a cobertura de uma região do Arroio Dilúvio. Com esse experimento, pode-se comparar a trajetória do *Differential boat* na presença de distúrbios de correntes de vento e de água, bem como realizar o desvio de uma embarcação que cruza o caminho do USV.

Para demonstrar a compatibilidade do USV_SIM com outros pacotes, o experimento fez uso do *move_base* com o planejador local ATC A* [Jur20] e do pacote *heatmap* [WC12]. O ATC A* é capaz de realizar desvio de obstáculos móveis respeitando as COLREGS, para isso ele utiliza a arquitetura e padronização de tópicos e mensagens presente no *move_base*. Sempre que o ATC A* encontrar uma embarcação no caminho, o USV irá desviar da embarcação conforme previsto nas regras da COLREGS. Já o pacote *heatmap* gera uma sequência de *waypoints* de forma que a embarcação realize um movimento

de *lawnmower*. Esses waypoints são utilizados como alvos a serem alcançados pela embarcação, sendo enviados por mensagens ROS padronizadas conforme a organização do *move_base*. Com isso, o planejador global e o planejador local escolhem a melhor forma de alcançar os *waypoints*, desviando dos obstáculos.

O diagrama da arquitetura multi-camadas do experimento é apresentado na Figura 5.11, sendo que as principais camadas são: *USV_SIM*, *move_base* e *heat_map*. O *USV_SIM* é responsável por realizar a simulação física e fornece dados de odometria e de sensores utilizados para percepção do ambiente e da embarcação (*exteroception* e *proprioception*). Já o *move_base* recebe informações do mapa, da odometria e dos sensores para planejar a rota até o objetivo (*waypoint*), o qual é definido pelo pacote *heat_map*. O *heat_map* observa o estado atual e o resultado da movimentação de forma que, ao alcançar o *waypoint*, envia um novo objetivo para o *move_base*. Além disso, o *Differential boat* é nomeado nesse experimento como *diffboat1*, sendo que ele possui a sua própria instância do *move_base*, *heat_map* e algoritmo de controle, ou seja, todas as mensagens estão dentro do espaço de nomes *diffboat1*. Para adicionar novas embarcações, basta replicar a configuração alterando o nome da embarcação, que as camadas superiores serão duplicadas, e o *USV_SIM* irá comunicar-se diretamente com cada instância dentro do espaço de nomes da nova embarcação.

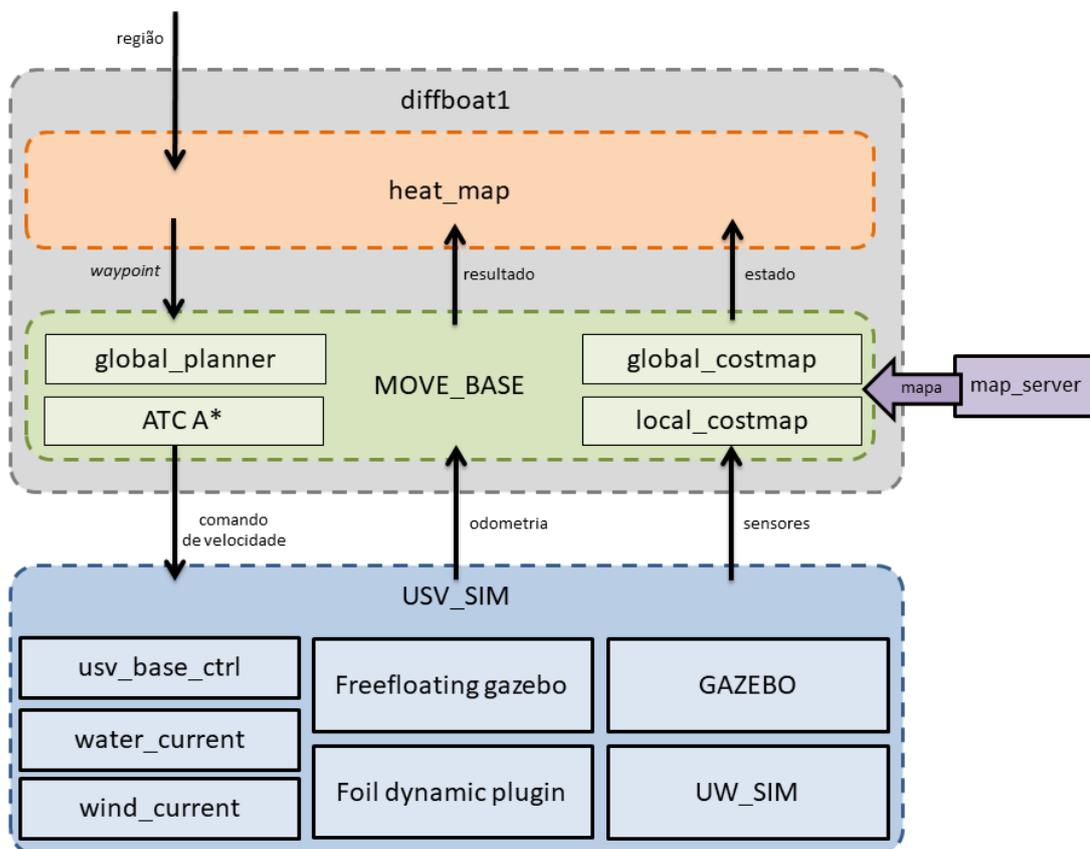


Figura 5.11: Diagrama da arquitetura multi-camadas do *USV_Sim* integrado ao pacote *move_base* e ao pacote *heatmap*.

Assim, foram conduzidos três testes em que o *Differential boat* deveria realizar a cobertura de uma determinada região do Arroio Dilúvio. No primeiro teste (Figura 5.12a) compara-se os efeitos de navegar perpendicularmente a margem do rio, com ou sem os distúrbios de correntes de vento e de água. Já no segundo teste (Figura 5.12b), além de comparar com a presença e ausência de distúrbios, uma embarcação cruza o caminho do USV perpendicularmente, exigindo que o *Differential boat* faça um desvio a direita (bom-bordo), ou seja, por trás da embarcação que desce o rio. Já no terceiro teste (Figura 5.12c), compara-se os efeitos de navegar paralelamente a margem do rio, com e sem distúrbios de vento e de água.

Nos três testes, observa-se que os distúrbios afetam o deslocamento da embarcação, já que trajetória em rosa (presença de distúrbios) diferencia-se da trajetória em preto, por não seguir em linha reta. No primeiro experimento (Figura 5.12a), o tempo e a distância aumentam de 540,9 s e 431,6 m (sem distúrbios) para 620,2 s e 501,1 m (com distúrbios) conforme a Tabela 5.2, ou seja, teve um aumento em 16,1 % e 14,6% no tempo e na distância.

Tabela 5.2: Distância e tempo para o *Differential boat* realizar a cobertura da região na presença e ausência de distúrbios de vento e correnteza de água. As consequências de haver um cruzamento de outra embarcação, provocando o desvio a direita conforme as COLREGS.

Direção	Cruza-mento	Sem distúrbios		Com distúrbios		Aumento com distúrbio	
		distância	tempo	distância	tempo	distância	tempo
perpendicular	não	431,6 m	540,9 s	501,1 m	620,0 s	16,1 %	14,6 %
perpendicular	sim	436,7 m	548,7 s	546,2 m	676,9 s	25 %	23,3 %
paralelo	não	398,0 m	505,8 s	605,8 m	756,0 s	52 %	49,4 %

No segundo teste (Figura 5.12b), na ausência de distúrbios, o USV conseguiu realizar o desvio a direita. Já com a presença dos distúrbios, mesmo o planejador tentando realizar o desvio, não foi possível. Assim, a embarcação precisou realizar uma curva completa no sentido horário para, posteriormente, seguir em direção ao *waypoint*. Similarmente ao primeiro teste, a presença de distúrbios provocou um aumento de 25 % na distância percorrida e no tempo em 23,3 % para realizar a cobertura da região.

No terceiro teste (Figura 5.12c), a distância percorrida aumentou significativamente (52 %), pois a embarcação não conseguiu mover-se em linha reta enquanto subia o rio. Já o tempo total aumentou de 505,8 s, na ausência dos distúrbios, para 756,8 s na presença dos distúrbios, ou seja, teve um aumento em de 49,4 % no tempo e na distância.

Por meio desse experimento, percebe-se a necessidade de um algoritmo de controle que considere os distúrbios naturais presentes no ambiente a ser navegado, bem como o desenvolvimento de um algoritmo de *coverage path-planning* capaz de realizar a cobertura desviando de obstáculos móveis e estáticos. Além disso, para que a cobertura da

região seja completa, o algoritmo de cobertura deve se adaptar e corrigir as rotas previamente geradas, caso um obstáculo impeça a cobertura da região naquele momento.

Considerando que a elaboração desse experimento demandou apenas a manipulação de poucos arquivos de configuração, sem exigir a programação de novas interfaces de software ou mesmos *wrappers* para integrar os algoritmos, é factível afirmar que a estrutura modular facilita a reutilização de código e agiliza o desenvolvimento de novas soluções na área da robótica.

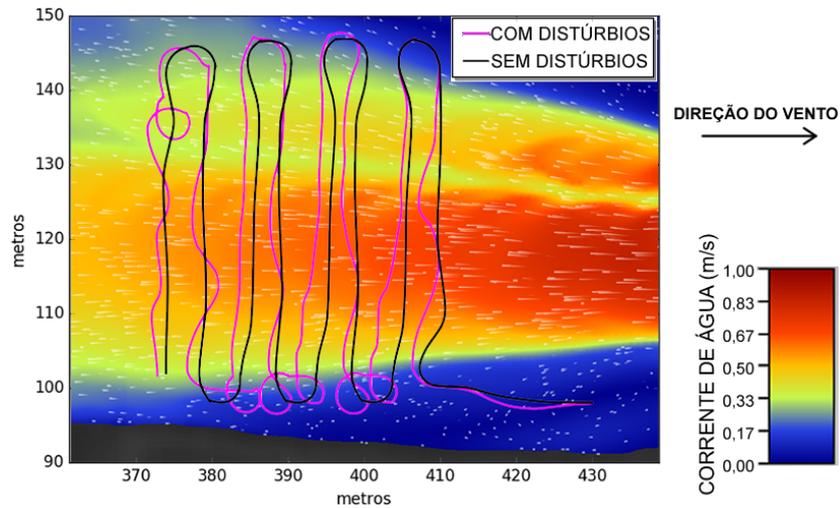
5.8 Inspeção de ponte

A inspeção de ponte é uma tarefa importante a ser executada tanto antes de um desastre (ação de mitigação), quanto após o desastre (ação de resposta). Após desastres naturais, como furacões e inundações, as condições climáticas podem impedir as inspeções tripuladas. Uma vez que colocam as equipes de emergência em risco ao se aproximarem de uma ponte que pode colapsar a qualquer instante, e que as fortes ondas, correntes de água e vento podem adernar a embarcação.

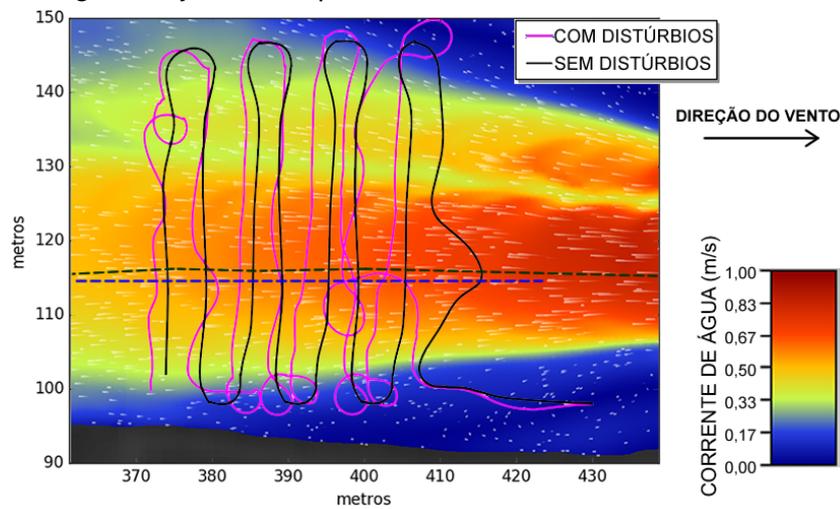
Devido a esses riscos, alguns pesquisadores propuseram o uso de veículos não tripulados para esse tipo de inspeção, incluindo o uso de USVs [MSH⁺11, HPKK15]. O principal problema é que esses veículos de pesquisa não foram devidamente testados em ambientes tão severos, especialmente ao considerar que cada cenário de desastre é único, e tem suas próprias complexidades. Por outro lado, testar um USV real em um cenário de desastre, coloca a própria embarcação em risco, pois enquanto ele ainda está sendo testada, é esperado que falhas ocorram. Uma falha, nesse caso, pode causar colisão, dano ou até perda total do USV.

Devido a essas situações, essa seção propõe um ambiente de simulação para inspeção de ponte, no qual são comparados os efeitos de distúrbios ambientais ao realizar essa tarefa. A Figura 5.13 tem uma visão detalhada da mesma ponte apresentada na Figura 3.13, localizada em 30°02'50.5"S 51°13'57,7"W, na cidade de Porto Alegre, Brasil, perto da foz do Arroio Dilúvio. Nesta figura, podemos ver que a corrente de água é mais rápida entre os dois pilares da ponte, e mais lenta nas margens do rio. O barco diferencial começou na posição superior da imagem e tem o objetivo de capturar imagens nas posições marcadas com pontos. A linha sólida preta representa a trajetória executada pelo USV, enquanto que a linha tracejada representa a trajetória esperada. A trajetória verde escura representa o movimento da embarcação na ausência de distúrbios, e os pontos com as marcas C1, C2 e C3 mostram locais de colisão com os pilares, detalhados nas capturas de tela.

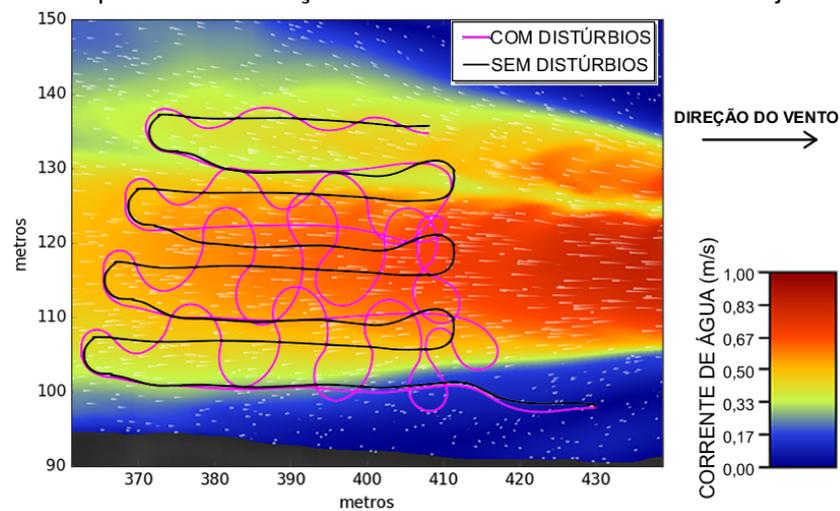
Uma das razões para o fraco desempenho da navegação do USV, está relacionada ao design do controlador. Como mencionado anteriormente, todos os quatro modelos de



(a) *Coverage path-planning* perpendicular à margem do rio. Trajetória em rosa: presença de de correntes de vento e água. Trajetória em preto: sem distúrbios.



(b) Trajetórias da embarcação ao realizar *coverage path-planning* de uma região, sendo que uma embarcação cruza o caminho descendo o rio. USV com distúrbios: trajetória em rosa. USV sem distúrbios: trajetória em preto. Embarcação cruzando o caminho do USV: trajetória tracejada.



(c) *Coverage path-planning* paralelo à margem do rio. Trajetória em rosa: presença de de correntes de vento e água. Trajetória em preto: sem distúrbios.

Figura 5.12: Experimento de *coverage path-planning* de forma a comparar os efeitos dos distúrbios e desvio de obstáculos móveis na trajetória do *Differential boat*.

barcos usam a mesma estratégia de controle baseada em orientação, no qual o objetivo é manter o USV apontando para o próximo waypoint. No entanto, em situações como esse cenário de inspeção de ponte, espera-se que o controlador diferencial do barco tenha um desempenho ruim, pois não leva em consideração a corrente da água e o vento. Assim, a embarcação poderia, por exemplo, navegar por áreas de corrente mais lenta para fazer manobras e subir o rio. Este teste mostra um possível uso do simulador proposto, ou seja, fazer experimentos do USV em ambientes hostis, validando e comparando diferentes estratégias para concluir as tarefas atribuídas ao USV.

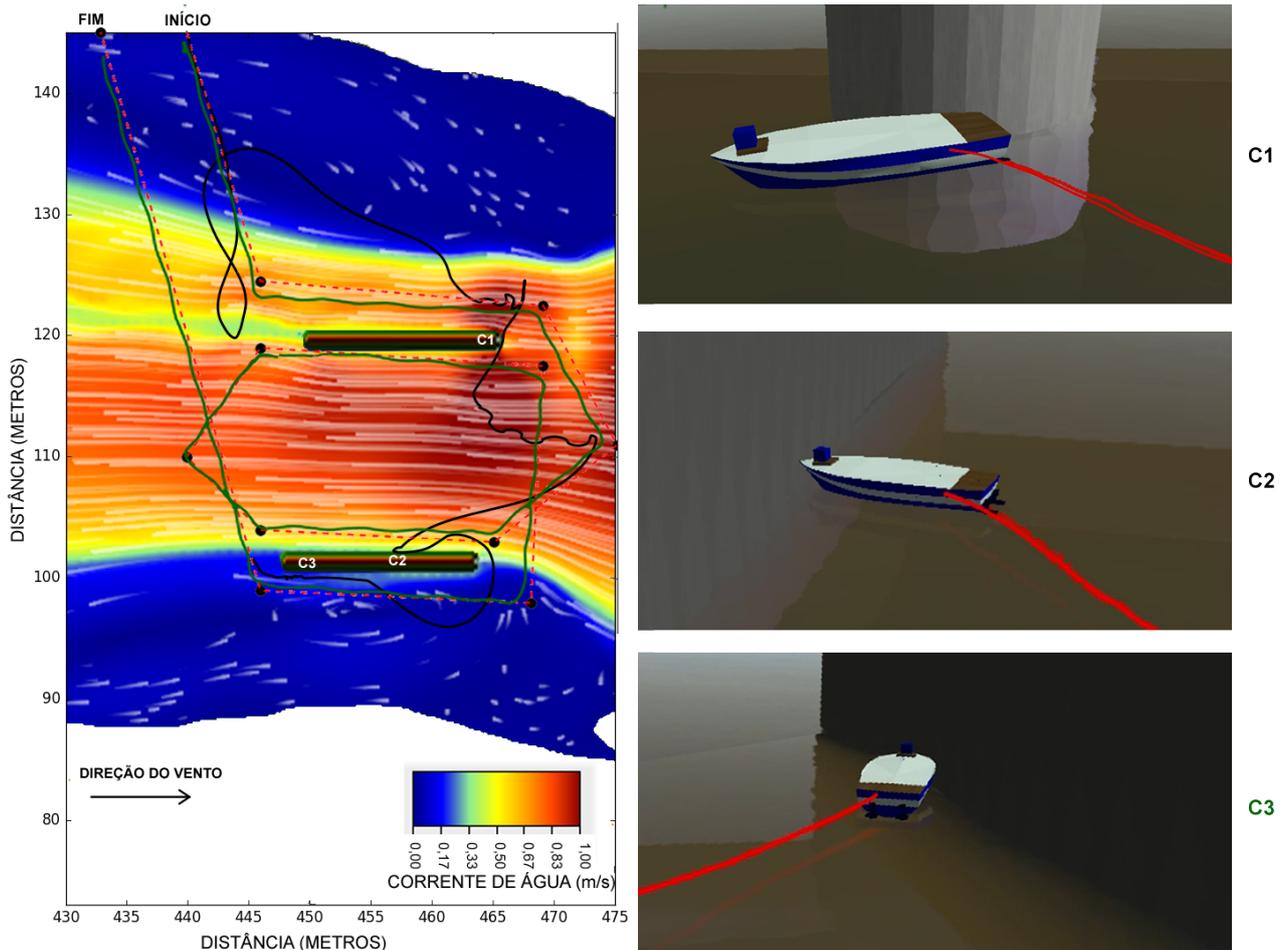


Figura 5.13: Inspeção de ponte com um barco diferencial. A linha em preto representa a trajetória da embarcação ao navegar com distúrbios, enquanto que em verde a trajetória sem distúrbios. Os pontos C1, C2 e C3 são os locais onde a embarcação colidiu ao ter sido arrastada pelos distúrbios naturais.

5.9 Inundação

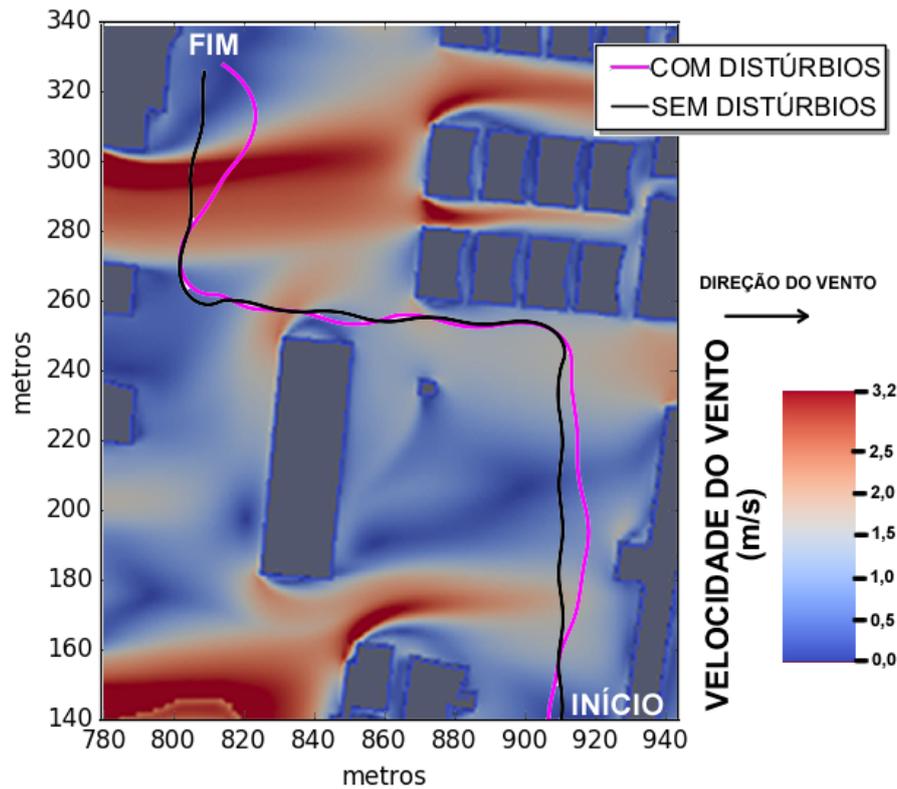
O presente experimento tem como objetivo demonstrar as possibilidades de reproduzir uma situação de inundação em área urbana, bem como a forma com que os distúrbios impactam nas trajetórias de embarcações ao se deslocarem entre as edificações. Para isso,

simulou-se no HEC-RAS o aumento de nível de água em 3,5 m numa região de Porto Alegre. Na Figura 5.14b, é apresentada a zona inundada, a velocidade da água e a trajetória da embarcações com e sem distúrbios de vento e água, sendo que, nessa imagem, a direção do fluxo de água é apresentado pelas partículas brancas. Para a mesma região foram executadas simulações de corrente de vento no simulador OpenFoam, no qual essas correntes estão direcionadas da esquerda para a direita e os módulos podem ser observados pela intensidade da cor (vermelho para 3,2 m/s e azul para 0 m/s) na Figura 5.14a.

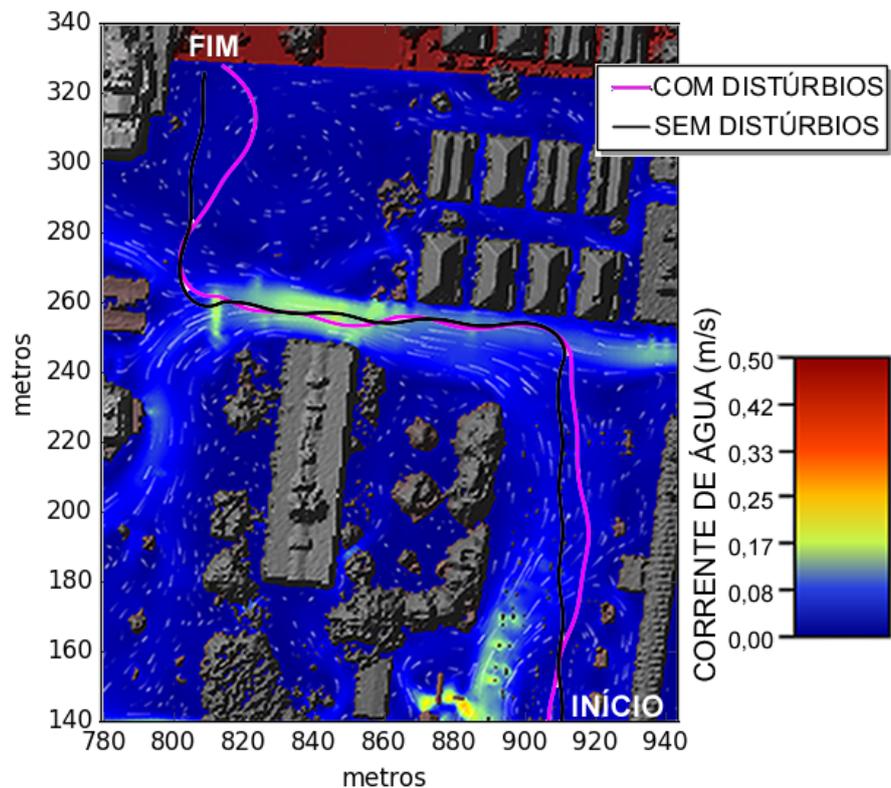
Por meio das trajetórias, observa-se que a presença de edificações podem tanto criar áreas protegidas de distúrbios quanto criar fortes correntes, já que os prédios podem canalizar o vento ou a água. Um exemplo de região, na qual o vento foi canalizado, é visualizado no entorno do ponto (810, 300) da Figura 5.14a. Diferente dos experimentos anteriores, a embarcação demorou menos tempo, mas percorreu uma distância maior quando estava sobre os efeitos dos distúrbios de vento e água. Ou seja, ao se deslocar sem correntes de vento, água e ondas, o *Airboat* levou 426,6 s para percorrer 284,7 m, enquanto que na presença dos distúrbios levou 413,0 s para uma distância total de 295,1 m. A diminuição do tempo está relacionada ao deslocamento realizado pela embarcação entre os pontos (910, 250) e (800, 260), pois, nesse trecho, a água foi canalizada pelos prédios (ver Figura 5.14b), acelerando a embarcação. Por outro lado, a distância percorrida aumentou, já que os distúrbios de vento arrastaram a embarcação perpendicularmente a sua direção de deslocamento, retirando-a de sua trajetória mais curta. Esse efeito pode ser observado nas regiões próximas aos pontos (800, 300) e (910, 180) na Figura 5.14a.

Um experimento semelhante a esse já havia sido conduzido e publicado na conferência ICINCO-2018 [PJA18], porém ela foi simulada novamente, já que as correntes de vento eram reproduzidas com um modelo LBM e a simulação de água pelas equações Diffusion Wave no HEC-RAS. Contudo, os especialistas em mecânica de fluidos indicaram as limitações do modelo LBM, tendo sido sugerido a utilização do OpenFoam ou do Incompact3d. Eles também aconselharam que o modelo de equações fosse reconfigurado do Diffusion Wave para o de Full Momentum no HEC-RAS [Bru18], o que permitiria o surgimento de vórtices nos fluxos de água. Para a realização dessa simulação, utilizou a versão 5.0.3 do HEC-RAS, sendo que a área inundada foi subdividida por uma grade de células de 2 metros de largura por 2 metros de altura. A condição inicial do nível de água foi definida como 8 metros de altura. A primeira condição de contorno foi definida com um fluxo de água (direita da Figura reffig:enchente) de vazão de $200 \text{ m}^3/\text{s}$, enquanto que a condição de contorno (esquerda da Figura reffig:enchente e conectada ao Guaíba) foi de um nível de água 8 metros. A simulação foi realizada durante 1 hora, com intervalo de cálculo de 30 segundos com número máximo de 20 iterações.

Cabe destacar que, apesar do autor dessa tese não ser especialista no uso das ferramentas QGIS, OpenFoam, HEC-RAS, foi possível criar o cenário de inundação dessa seção. No entanto, reproduzir virtualmente um cenário real demanda que dados reais co-



(a) Comparação entre a trajetória com e sem distúrbios numa inundação em área urbana. Fundo apresenta as correntes de vento.



(b) Comparação entre a trajetória com e sem distúrbios numa inundação em área urbana. Fundo apresenta as correntes de água.

Figura 5.14: Resultado da simulação do *Airboat* para mover-se numa inundação em área urbana. As trajetórias em rosa é o deslocamento realizado na presença de distúrbios, enquanto que em preto não existem distúrbios.

letados sejam utilizados como parâmetros de entrada nas ferramentas, por exemplo, uma simulação precisa e acurada do fluxo de um rio demandaria dados do nível de água, o volume de água por segundo, o estudo do perfil de velocidade da água, o que seria realizado por meio de um ADCP (*Acoustic Doppler Current Profiler*). O mesmo deve ser realizado na simulação de vento, por exemplo, um dos principais parâmetros de configuração é a velocidade no plano de entrada, que no caso desse experimento foi utilizado os valores de intensidade moderada da região de Porto Alegre, ou seja, 3,2 m/s.

Além disso, cabe ressaltar que a filtragem do MDS para a simulação das correntes de água podem gerar divergências nos obstáculos existentes, uma vez que a vegetação e pequenos objetos não estarão na simulação de vento. Isso ocorre porque a entrada de dados no HEC-RAS utiliza os arquivos de MDS, enquanto que o terreno a ser utilizado no OpenFoam foi gerado com o MDT conforme fluxograma da Figura 3.1.

Outro ponto a ser observado, na preparação para a simulação, é a coerência temporal entre os dados a serem utilizados, pois os modelos MDT, MDS e os arquivos OSM das edificações poderão não representar o mesmo instante de tempo. Por exemplo, um prédio ao estar na simulação de corrente de água, pode não estar na simulação da corrente de vento e na simulação robótica, já que os dados do MDS são anteriores a construção da edificação, enquanto que o arquivo OSM já contém dados da presença desse prédio.

Por fim, já que a simulação de fluidos em grandes cenários podem demandar alto poder de processamento e prolongar-se por horas ou dias, sugere-se a criação de uma base com os resultados dessas simulações. O compartilhamento desses resultados permitiria a sua reutilização pela comunidade científica, oferecendo uma base comum de testes, em que os distúrbios ambientais e as situações de desastres naturais poderiam estar presentes com boa acurácia e precisão.

6. CONCLUSÕES

Apesar da utilização de robôs nas ações de gerenciamento de desastres terem sido ampliadas, as plataformas robóticas ainda precisam ser melhor preparadas para utilização em condições severas. Realizar testes de novos protótipos em situações reais de desastres naturais se mostra complexo, tanto pela dificuldade de acesso a esses ambientes, quanto pelos riscos à integridade física dos pesquisadores e a eminência dos equipamentos serem perdidos ou danificados. Além disso, deve-se considerar que existem poucos locais para testes. Logo, reproduzir os ambientes com distúrbios severos em simulação, mostra-se como uma alternativa desejável, uma vez que diminui custos, evita-se riscos e permite-se repetir as mesmas características dos distúrbios naturais em diversos testes.

Neste sentido, e levando em conta que a modelagem de simulação tem limitado a confiança de que os resultados obtidos virtualmente estão próximos aos do ambiente real [Mur14], foi conduzida uma pesquisa de abordagem mista. Foram utilizados diferentes procedimentos metodológicos sequenciais, o que permitiu validar a hipótese dessa tese e responder as perguntas de pesquisa. Ou seja, a simulação pode reproduzir com precisão o movimento de USVs e dos fluídos, presentes nos ambientes reais e em desastres, por meio de um protocolo de calibração dos USVs virtuais e da utilização das ferramentas CFDs HEC-RAS e OpenFoam.

O modo como os simuladores robóticos podem ser aperfeiçoados foi respondido por meio da pesquisa descrita no Capítulo 2, na qual, a revisão detalhada dos simuladores robóticos existentes considerou os artigos científicos, os manuais e o código fonte desses. Essa revisão bibliográfica e documental identificou as principais características para a simulação de USVs, bem como os aprimoramentos necessários em cada simulador para aumentar o seu nível de realismo.

A integração com ferramentas CFDs e a forma como o modelo de simulação foi desenvolvido é apresentado no Capítulo 3. Partindo-se do Freefloating Gazebo como base, desenvolveu-se o USV_SIM de modo a contemplar as características elencadas na etapa anterior. O efeito de flutuabilidade do USV_SIM, mais realista do que o dos simuladores analisados na revisão, faz com que as ondas induzam movimentos de rotação e inclinação no barco. O módulo de vento proposto nessa tese aplica forças na embarcação considerando os obstáculos existentes no cenário, bem como de acordo com a porção do barco que está acima da água. Já o modelo de correntes de água aplica forças sobre o casco, quilha e lemes presentes na embarcação, sendo que essas forças podem variar ao longo do tempo e do espaço, o que não está disponível na maioria dos outros simuladores, pois esses aplicam uma única força constante para todo o cenário.

O presente simulador distribui quatro modelos de USV prontos para uso, bem como um cenário real. A reprodução de terrenos e reconstrução de ambientes urbanos

também diferencia-se no USV_SIM dos demais simuladores, os quais oferecem pouco ou nenhum suporte para reprodução de ambientes reais. A Seção 3.1 apresenta um protocolo de como reproduzir cenários reais nos ambientes virtuais, os quais podem ser utilizados tanto na simulação robótica no USV_SIM, quanto como entrada para a realização da simulação no OpenFoam.

A comprovação que a movimentação das embarcações virtuais se aproximam dos USVs reais, é realizado no Capítulo 4 por meio de um protocolo de validação. Com a calibração das embarcações, garante-se maior realismo, tanto do ponto de vista de suas trajetórias quanto de suas velocidades. Nesse contexto, o USV_SIM destaca-se dos simuladores anteriores por realizar a validação da dinâmica das embarcações modeladas.

Os estudos de casos do Capítulo 5 demonstram o impacto nas atividades desenvolvidas por USVs ao se moverem em cenários com distúrbios ambientais realistas. As variações de configuração, disponibilizadas e apresentadas nos estudos de caso, permitem demonstrar os diferentes graus de distúrbios a serem utilizados e exemplificam as possíveis organizações do sistema GNC (*Guidance Navigation and Control*) das embarcações.

Nas próximas seções, são apresentadas as limitações existentes e trabalhos futuros relacionados ao modelo proposto nesse tese.

6.1 Limitações

Apesar das contribuições descritas anteriormente, existem elementos nessa pesquisa que precisam ser aperfeiçoados, por exemplo, as embarcações *Rudderboat* e *Sailboat* não foram validadas com o protocolo apresentado no Capítulo 4. Essa limitação em realizar a validação dessas embarcações virtuais deve-se ao fato da falta de acesso a USVs reais com essas características para a experimentação necessária.

Outra limitação está relacionada a trajetória da embarcação nos testes, observa-se nas Figuras 5.2 e 5.10 que uma trajetória mais linear, sem tantas curvas, permitiria um deslocamento mais eficiente da embarcação. Isso se deve ao método de controle (PID), baseado apenas na direção da embarcação, que não considera distúrbios do ambiente. Conseqüentemente, aperfeiçoar ou substituir o método, permitirá a realização de rotas mais eficientes.

Apesar do protocolo da Seção 3.1 permitir a automatização de diversas etapas de modelagem de cenários, ainda demanda diversas intervenções do usuário. Por isso, permitir que exista um fluxo único integrado, poderia ampliar o uso desse protocolo. Por outro lado, o fluxo integrado tende a reduzir a liberdade de personalização e correções realizadas, conseqüentemente, a criação desse fluxo integrado não eliminaria o já apresentado.

Como apresentado no Capítulo 4, a simplificação no modelo de arrasto frontal e lateral tende a reduzir o realismo da movimentação das embarcações. A simplificação permite melhorar desempenho em simulação, já que estimar a área molhada do casco no ambiente virtual poderia ter um custo computacional elevado, por exemplo, a biblioteca de física do gazebo realiza por padrão 1000 passos (cálculos) por segundo, estimar a área do casco, nessa taxa de atualização, provocaria a redução significativa na velocidade da simulação. Por outro lado, ao não precisar reproduzir na simulação a mesma área molhada do casco do experimento real, libera-se a medição dessa área no teste em campo, o que seria uma tarefa complexa com a embarcação em movimento.

6.2 Trabalhos futuros

Como trabalho futuro, tem-se a possibilidade de modelar ondas geradas pelas embarcações, o que permitiria afetar o movimento de outras embarcações em seu entorno. O realismo da simulação pode ser ampliado de forma que essas ondas provoquem rotações no USV (*roll* e *pitch*), podendo impactar na leitura de dados de sensores que são dependentes da posição e da orientação da embarcação, por exemplo, as câmeras. A realização de testes em simulação inserindo-se ruído e erros nas medições obtidas por sensores poderia aproximar os dados obtidos das embarcações virtuais com relação aos USVs reais. Além disso, cabe o desenvolvimento de um módulo que reproduza um sistema AIS (Automatic Identification System).

Uma possibilidade de investigação é a realização de uma avaliação de UAV e UUV nos moldes da validação apresentada no Capítulo 4. Essa investigação tende a garantir uma melhor utilização de distúrbios simulados em ferramentas CFD 3D, como é o caso do OpenFoam.

Com relação a modelagem de cenários, ainda cabe desenvolver métodos para a geração automática da vegetação. Como primeira abordagem possível, tem-se a geração aleatória de árvores e arbustos. Outra possibilidade é a utilização das informações contidas nas imagens aéreas, nos dados dos arquivos MDS e OSM. Ao reproduzir essa vegetação, ainda deve-se consultar especialistas em mecânica de fluídos para identificar quais tipos e tamanhos de vegetação deverão constar na simulação de vento e de água, uma vez que elas podem restringir a passagem desses fluídos nos cenários.

Para reproduzir a interação entre as correntes de vento e o fluxo de água, cabe desenvolver simulações CFD de multifásico, ou seja, na mesma simulação estarem presentes tanto as correntes de vento, quanto as da água. Desta maneira, reproduzir desastres naturais se tornará mais realista, já que, por exemplo, ao simular um ciclone tropical, as forças do vento possibilitaria a formação de ondas e ressacas em áreas costeiras.

Apesar de já ter sido realizadas simulações de enchentes com o USV_SIM e inspeções de pontes, ainda cabe investigar e modelar alterações no simulador de forma a reproduzir cenários de desastres naturais, tais como, tsunamis, rompimentos de barragens, deslizamentos de encostas sobre rios. Dentre os elementos a serem modificados no USV_SIM, pode-se indicar a necessidade de alterar a simulação de ondas, modelar um veículo robusto e potente o suficiente para lidar com as forças de vento e correntes de água elevadas.

Os resultados apresentados na Seção 5.6 sobre a comparação dos USVs realizando tarefas diferentes sugere que barcos diferentes possuem desempenhos bem diversos em relação as atividades realizadas. Isso pode sugerir a necessidade de uma pesquisa mais aprofundada neste sentido, com o objetivo de criar uma espécie de *guideline* de aplicações dos USVs. Os resultados dessa seção também indicam que o método GNC utilizado pode ter um impacto importante no desempenho na realização das atividades. Isso pode sugerir que o método GNC mais adequado pode variar, dependendo do tipo de embarcação e do tipo de atividade a ser realizada.

No contexto de enchentes, cabe desenvolver plugins ou pacotes capazes de gerar e posicionar no ambiente detritos para serem carregados pelas correntes de água. Para determinados detritos, como galhos de árvores e sacos, permitir que eles sejam agrupados e, a medida que são carregados pelo ambiente e colidem com obstáculos, possam ser separados de forma a espalhar-se. Esse tipo de simulação, permite que seja investigado algoritmos para detecção e para evitar colisões com tais tipos de objetos.

Outra possibilidade de investigação está relacionada ao desenvolvimento de métodos de navegação simultânea para múltiplos USVs, bem como de modelos de coordenação de equipes heterogêneas de robôs, por exemplo, coordenar uma equipe formada por USVs, AUVs e veículos terrestres. Para tal, uma possível solução pode ser pela utilização de modelos BDI (*Belief Desire Intention*) da área da Inteligência Artificial.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BCK⁺07] Balakirsky, S.; Carpin, S.; Kleiner, A.; Lewis, M.; Visser, A.; Wang, J.; Ziparo, V. “Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue”, *Journal of Field Robotics*, vol. 24, Novembro 2007, pp. 943 – 967.
- [Ble94] Blendermann, W. “Parameter identification of wind loads on ships”, *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 51, Maio 1994, pp. 339 – 351.
- [Bru95] Brunner, G. W. “HEC-RAS river analysis system. hydraulic reference manual. version 1.0”, Relatório técnico, DTIC, 1995, 538p.
- [Bru18] Brunner, G. W. “Hec-ras river analysis system”, Relatório técnico, U.S. Army Corps of Engineers, 2018, 538p.
- [BSNL10] Benjamin, M. R.; Schmidt, H.; Newman, P. M.; Leonard, J. J. “Nested autonomy for unmanned marine vehicles with moos-ivp”, *Journal Field Robotics*, vol. 27, Outubro 2010, pp. 834–875.
- [Bul20] Bullet. “Bullet real-time physics simulation”. Capturado em: <https://pybullet.org/>, Junho, 2020.
- [CMBG07] Craighead, J.; Murphy, R.; Burke, J.; Goldiez, B. “A survey of commercial open source unmanned vehicle simulators”. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2007, pp. 852–857.
- [Cre10] Creswell, J. W. “Projeto de Pesquisa métodos de pesquisa qualitativo quantitativo e misto”. Artmed, 2010, 296p.
- [dBWSF17] de Brito, M.; Weber, E.; Silva Filho, L. C. “Multi-criteria analysis applied to landslide susceptibility mapping”, *Revista Brasileira de Geomorfologia*, vol. 18, Julho 2017, pp. 919–935.
- [DDC⁺13] Dias, M.; Dias, J.; Carvalho, L.; Freitas, E.; Silva Dias, P. “Changes in extreme daily rainfall for so paulo, brazil”, *Climatic Change*, vol. 116, Fevereiro 2013, pp. 705–722.
- [Edm20] Edmondson, J. “Gams”. Capturado em: <https://github.com/jredmondson/gams>, Junho, 2020.
- [ELDL11] Echeverria, G.; Lassabe, N.; Degroote, A.; Lemaignan, S. “Modular open robots simulation engine: Morse”. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 46–51.

- [Eml20] Emlid. “Emlid - reach module docs”. Capturado em: <https://doc.emlid.com.br/reach/>, Junho, 2020.
- [Eng20] of Engineers, U. A. C. “Hydrologic engineering center’s (ceiwr-hec) river analysis system (hec-ras)”. Capturado em: <http://www.hec.usace.army.mil/software/hec-ras/>, Junho, 2020.
- [Fal90] Faltinsen, O. “Sea Loads on Ships and Offshore Structures”. Cambridge University Press, 1990, 340p.
- [Fos11] Fossen, T. “Handbook of Marine Craft Hydrodynamics and Motion Control”. Wiley, 2011, 582p.
- [Fou20a] Foundation, B. “Blender”. Capturado em: <https://www.blender.org/>, Junho, 2020.
- [Fou20b] Foundation, O. “Openstreetmap”. Capturado em: <https://www.openstreetmap.org/>, Junho, 2020.
- [Fou20c] Foundation, O. S. R. “Open robotics”. Capturado em: <https://www.openrobotics.org/>, Junho, 2020.
- [Gam20] Games, E. “Unreal engine”. Capturado em: <https://www.unrealengine.com/>, Junho, 2020.
- [Gra20] Graph, O. S. “Virtual planet builder”. Capturado em: <https://github.com/openscenegraph/VirtualPlanetBuilder>, Junho, 2020.
- [GS09] Gerhardt, T. E.; Silveira, D. T. “Métodos de Pesquisa”. Editora da UFRGS, 2009, 120p.
- [GVH03] Gerkey, B.; Vaughan, R.; Howard, A. “The player/stage project: Tools for multi-robot and distributed sensor systems”. In: Proceedings of the International Conference on Advanced Robotics (ICAR), 2003, pp. 317–323.
- [HC11] Harris, A.; Conrad, J. M. “Survey of popular robotics simulators, frameworks, and toolkits”. In: Proceedings of the Proceedings of IEEE Southeastcon, 2011, pp. 243–249.
- [HCE16] Hummell, B.; Cutter, S.; Emrich, C. “Social vulnerability to natural hazards in brazil”, *International Journal of Disaster Risk Science*, vol. 7, Junho 2016, pp. 111–122.
- [HPKK15] Han, J.; Park, J.; Kim, T.; Kim, J. “Precision navigation and mapping under bridges with an unmanned surface vehicle”, *Autonomous Robots*, vol. 38, Abril 2015, pp. 349–362.

- [HSG16] Henriksen, E. H.; Schjølberg, I.; Gjersvik, T. B. “UW-MORSE: The underwater modular open robot simulation engine”. In: Proceedings of the IEEE/OES Autonomous Underwater Vehicles (AUV), 2016, pp. 261–267.
- [IBG20] IBGE. “IBGE instituto brasileiro de geografia e estatística”. Capturado em: <https://www.ibge.gov.br/>, Junho, 2020.
- [IMO72] IMO. “Convention on the international regulations for preventing collisions at sea”. Capturado em: <http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx>, Junho, 2020.
- [Inc20] Incompact3d. “Incompact3d”. Capturado em: <https://www.incompact3d.com/>, Junho, 2020.
- [Jur20] Jurak, D. A. “A COLREGS-compliant collision avoidance system for unmanned surface vehicles”, Dissertação de Mestrado, Escola Politécnica – PUCRS, Porto Alegre, RS, Brasil, 2020, 69p.
- [KBM⁺19] Karapetyan, N.; Braude, A.; Moulton, J.; Burstein, J.; White, S.; O’Kane, J.; Rekleitis, I. “Riverine coverage with an autonomous surface vehicle over known environments”. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 7.
- [KE95] Kennedy, J.; Eberhart, R. “Particle swarm optimization”. In: Proceedings of the International Conference on Neural Networks (ICNN), 1995, pp. 1942–1948.
- [Ker14] Kermorgant, O. “A dynamic simulator for underwater vehicle-manipulators”. In: Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN), 2014, pp. 25–36.
- [KH04] Koenig, N.; Howard, A. “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2004, pp. 2149–2154.
- [KK09] Khoury, H. M.; Kamat, V. R. “Evaluation of position tracking technologies for user localization in indoor construction environments”, *Automation in Construction*, vol. 18, Julho 2009, pp. 444 – 457.
- [KMR19] Karapetyan, N.; Moulton, J.; Rekleitis, I. “Meander based river coverage by an autonomous surface vehicle”. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 12.
- [LL11] Laizet, S.; Li, N. “Incompact3d: A powerful tool to tackle turbulence problems with up to $o(10^5)$ computational cores”, *International Journal for Numerical Methods in Fluids*, vol. 67, Dezembro 2011, pp. 1735 – 1757.

- [Mar11] Marder-Eppstein, E. “Move_base package”. Capturado em: http://wiki.ros.org/move_base, Junho, 2020.
- [Mig05] Miguens, A. P. “Navegação: a ciência e a arte. Volume III – Navegação eletrônica e em condições especiais”. Diretoria de Hidrografia e Navegação, 2005, 1822p.
- [MSH⁺11] Murphy, R. R.; Steimle, E.; Hall, M.; Lindemuth, M.; Trejo, D.; Hurlebaus, S.; Medina-Cetina, Z.; Slocum, D. “Robot-assisted bridge inspection”, *Journal of Intelligent & Robotic Systems*, vol. 64, Janeiro 2011, pp. 77–95.
- [MSM⁺13] Mendonça, R.; Santana, P.; Marques, F.; Lourenço, A.; Silva, J.; Barata, J. “Kelpie: A ROS-based multi-robot simulator for water surface and aerial vehicles”. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2013, pp. 3645–3650.
- [Mur14] Murphy, R. R. “Disaster robotics”. MIT press, 2014, 240p.
- [Ope20a] OpenCFD. “Openfoam the open source CFD toolbox”. Capturado em: <https://www.openfoam.com/>, Junho, 2020.
- [Ope20b] OpenCFD. “Solution and algorithm control”. Capturado em: <https://www.openfoam.com/documentation/user-guide/fvSolution.php#x24-930006.3>, Junho, 2020.
- [OSM20] OSM2World. “Osm2world”. Capturado em: <http://osm2world.org/>, Junho, 2020.
- [PJA18] Paravisi, M.; Jorge, V. A. M.; Amory, A. “Toward an accurate hydrologic urban flooding simulations for disaster robotics”. In: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2018, pp. 425–431.
- [Pla20] Platypus. “Platypus”. Capturado em: <http://senseplatypus.com>, Junho, 2020.
- [PPFS12] Prats, M.; Pérez, J.; Fernández, J. J.; Sanz, P. J. “An open source tool for simulation and supervision of underwater intervention missions”. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 2577–2582.
- [PSJ⁺19] Paravisi, M.; Santos, D. H.; Jorge, V.; Heck, G.; Gonçalves, L. M.; Amory, A. “Unmanned surface vehicle simulator with realistic environmental disturbances”, *Sensors*, vol. 1, Março 2019, pp. 444–457.
- [PUC20] PUC-Rio. “Lua”. Capturado em: <https://www.lua.org/>, Junho, 2020.

- [QGI20] QGIS. “QGIS - a free and open source geographic information system”. Capturado em: <https://qgis.org/>, Junho, 2020.
- [RJS01] Richards, P.; Johnson, A.; Stanton, A. “America’s cup downwind sails—vertical wings or horizontal parachutes?”, *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 89, Dezembro 2001, pp. 1565–1577.
- [RM15] Raskovic, B.; Mrdja, S. “Natural disasters: Prevention, risk factors and management”. Nova Science Publishers, 2015, 374p.
- [Rob20] RoboNation. “Robotx”. Capturado em: <https://robotx.org/>, Junho, 2020.
- [RSF13] Rohmer, E.; Singh, S. P. N.; Freese, M. “V-REP: A versatile and scalable robot simulation framework”. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 1321–1326.
- [San07] Santos, A. M. N. “Wavesim - ambiente de simulação para veículos aquáticos”, Dissertação de Mestrado, Faculdade de Ciências da Universidade do Porto, 2007, 104p.
- [SAU20] SAUC. “Student autonomous underwater vehicles challenge - europe”. Capturado em: <http://sauc-europe.org/>, Junho, 2020.
- [SC10] Sehgal, A.; Cernea, D. “A multi-AUV missions simulation framework for the USARSim robotics simulator”. In: Proceedings of the Mediterranean Conference on Control Automation (MED), 2010, pp. 1188–1193.
- [Sch20] Schneider, F. E. “ELROB – the european land robot trial”. Capturado em: <https://www.elrob.org/>, Junho, 2020.
- [SK16] Siciliano, B.; Khatib, O. “Handbook of Robotics”. Springer-Verlag, 2016, 2227p.
- [SST+16] Stanislas, L.; Smith, P.; Tidd, B.; Kujala, P.; Nicholson, S.; Dawson, A.; Lamont, R.; Chamberlain, W.; Barton, R.; Eich, M.; Dunbabin, M. “Bruce: An asv solution for the 2016 maritime robotx challenge”. Capturado em: http://www.robonation.org/sites/default/files/QueenslandUni_2016_RobotX_Journal.pdf, Junho, 2020.
- [TKT+00] Tadokoro, S.; Kitano, H.; Takahashi, T.; Noda, I.; Matsubara, H.; Shinjoh, A.; Koto, T.; Takeuchi, I.; Takahashi, H.; Matsuno, F.; Hatayama, M.; Nobe, J.; Shimada, S. “The robocup-rescue project: A robotic approach to the disaster mitigation problem.” In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2000, pp. 4089–4094.
- [TM14] Tosik, T.; Maehle, E. “MARS: A simulation environment for marine robotics”. In: Proceedings of the Oceans - St. John’s, 2014, pp. 1–7.

- [TTACP14] Torres-Torriti, M.; Arredondo, T.; Castillo-Pizarro, P. “Survey and comparative study of free simulation software for mobile robots”, *Robotica*, vol. -1, Julho 2014, pp. 1–32.
- [USA18] USACE. “HEC-RAS verification and validation tests”, Relatório técnico, U.S. Army Corps of Engineers, 2018, 154p.
- [USG20] USGS. “USGS science for a changing world”. Capturado em: <https://earthexplorer.usgs.gov>, Junho, 2020.
- [WC12] Woodall, W.; Carrol, M. “Moe: the autonomous lawnmower”, *ROSCON*, vol. 1, Maio 2012, pp. 13.
- [WRB⁺13] Wirtensohn, S.; Reuter, J.; Blaich, M.; Schuster, M.; Hamburger, O. “Modelling and identification of a twin hull-based autonomous surface craft”. In: Proceedings of the International Conference on Methods Models in Automation Robotics (MMAR), 2013, pp. 121–126.
- [WRS20] WRSC. “World robotic sailing championship”. Capturado em: <https://www.roboticsailing.org/>, Junho, 2020.
- [XJ14] Xiao, L.; Jouffroy, J. “Modeling and nonlinear heading control of sailing yachts”, *IEEE Journal of Oceanic Engineering*, vol. 39, Abril 2014, pp. 256–268.
- [ZZ18] Zhang, B.-J.; Zhang, S.-L. “Research on Ship Design and Optimization Based on Simulation-Based Design (SBD) Technique”. Shanghai Jiao Tong University Press, 2018, 239p.



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br