



UNIVERSIDADE FEDERAL DA BAHIA  
PROGRAMA DE PÓS GRADUAÇÃO EM MECATRÔNICA

ENG633 - SISTEMAS MECATRÔNICOS

---

**Relatório I  
Projeto Conceitual**

---

*Professor:*  
Leizer Schnitman

*Grupo:*  
Jailson dos Santos Junior  
Maxwell Francisco da Silva  
Rafael Santana Queiroz  
Rosinery Rosária de Medeiros  
Yuri de Matos Alves de Oliveira

Salvador

19 de Agosto de 2021

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Análise de Requisitos</b>	<b>6</b>
2.1	Matriz QFD . . . . .	6
2.2	Matriz Morfológica . . . . .	6
<b>3</b>	<b>Projeto Conceitual</b>	<b>10</b>
<b>4</b>	<b>Validação de projeto</b>	<b>18</b>
4.1	Módulos de simulação . . . . .	18
4.2	Missões . . . . .	21
	<b>Apêndices</b>	<b>27</b>
	<b>Referências</b>	<b>27</b>

## **Lista de Figuras**

3.1	ROVs que inspiraram o design do BROV . . . . .	10
3.2	Estrutura conceitual do BROV . . . . .	11
3.3	Propulsor do BROV . . . . .	12
3.4	Posicionamento dos thrusters . . . . .	12
3.5	Prototipação do propulsor . . . . .	13
3.6	Aquário para testes de propulsão e comunicação . . . . .	13
3.7	Projeto conceitual da <i>housing</i> da câmera e seu controlador . . . . .	14
3.8	Lanterna que será usada no BROV . . . . .	15
3.9	Vista isométrica do BROV conceitual . . . . .	15
3.10	Vista frontal do BROV conceitual . . . . .	16
3.11	Vista lateral do BROV conceitual . . . . .	16
3.12	Vista superior do BROV conceitual . . . . .	17
4.1	Estrutura de programação de um nó acústico AUVNetSim . . . . .	20
4.2	Arquitetura de comunicação subaquática . . . . .	21
4.3	Missão de submersão e emersão . . . . .	22
4.4	Missão de velocidade máxima e seguimento de trajetória . . . . .	23
4.5	Missão de mapeamento . . . . .	24
4.6	Tabela comparativa de simulators e suas funcionalidades . . . . .	25

## **Lista de Tabelas**

2.1	Matriz QFD . . . . .	8
2.2	Matriz morfológica . . . . .	9
4.1	Tabela comparativa de simulatores e suas funcionalidades . . . . .	19
.2	Matriz QFD . . . . .	26

## 1 Introdução

A comunicação subaquática sem fio tem sido um tema de vasta pesquisa nos últimos anos por conta do interesse de exploração do ambiente marinho. Enquanto no ar a maioria dos sistemas de comunicação utiliza ondas de rádio ou espectro de espalhamento e posicionamento global [1], na água os sensores acústicos são os mais utilizados. Contudo, outras tecnologias de comunicação sem fio também são aplicáveis nesse ambiente, como ondas de radiofrequência (RF) e ondas óticas [2].

Os sensores acústicos possuem a vantagem de permitir uma comunicação de longo alcance, da ordem de 20 km. Em contrapartida, possuem baixa taxa de transmissão (da ordem de kbps), alta latência (da ordem de segundos), são relativamente pesados e caros, além de serem perigosos para certas espécies marinhas [2].

A comunicação subaquática por ondas RF permite uma transição suave entre a interface água/ar, sendo útil para integração com sistemas RF terrestres. Além disso, é menos suscetível às interferências decorrentes de turbulência e turbidez marinha, se comparada com a comunicação acústica e ótica. A principal limitação da aplicação de RF na água é o baixo alcance e o alto custo decorrente da necessidade da instalação de antenas [2].

A comunicação ótica sem fio subaquática (UOWC, do inglês *Underwater Optical Wireless Communication*) possui, dentre os métodos supracitados, a maior taxa de transmissão, a menor latência e o menor custo de implementação. Por outro lado, o sinal ótico é alterado por conta de absorção e dispersão, demanda alinhamento preciso dos transceptores e possui alcance moderado (da ordem de dezenas de metros) [2].

Diante desse cenário, o objetivo geral desse trabalho é implementar, em ambiente de simulação, algoritmos de comunicação embarcados em um veículo submarino remotamente operado (ROV, do inglês *Remotely Operated Underwater Vehicle*), que deverá ser projetado para realizar uma operação de resgate de um objeto qualquer.

Os objetivos específicos do trabalho são:

1. Projetar um ROV para atuação em uma missão de resgate, com requisitos de funcionamento pré-estabelecidos.
2. Desenvolver e implementar, em ambiente simulado, algoritmos de comunicação subaquática.

3. Simular comportamento dinâmico, localização, mapeamento, navegação e comunicação do ROV.

O escopo do trabalho está restrito à entrega de três relatórios, desenhos 3D e de fabricação e algoritmos desenvolvidos. A prototipação física, assim como testes experimentais do ROV em piscina ou mar estão fora do escopo desse trabalho.

A divisão dos relatórios será feita da seguinte maneira:

1. Relatório I - Projeto Conceitual
2. Relatório II - Projeto Detalhado
3. Relatório III - Resultados de Simulação

No Relatório I, que é este documento, será apresentada inicialmente uma análise de requisitos do sistema, matriz QFD (do inglês *Quality Function Deployment*) para obtenção dos requisitos de projeto e matriz morfológica para seleção preliminar dos tipos de componentes do protótipo a ser simulado. Posteriormente serão apresentados o projeto conceitual e as estratégias para validação do protótipo em ambiente simulado.

No Relatório II será apresentado o projeto detalhado, com especificação dos componentes e sistema, juntamente com um cronograma para execução das tarefas.

No Relatório III serão mostrados os resultados da simulação do protótipo atuando na missão de resgate.

## 2 Análise de Requisitos

Dentre os dez requisitos do projeto, alguns exigem soluções envolvendo diversos sensores e técnicas específicas para seu cumprimento. Esta seção visa esclarecer a estratégia a ser adotada para algumas das funcionalidades necessárias do veículo.

### 2.1 Matriz QFD

### 2.2 Matriz Morfológica

#### Localização

Um dos requisitos do projeto é a capacidade do veículo de se localizar no ambiente. Esse requisito tem impacto em diversos outros, pois não havendo sua localização, não se pode criar um mapa ou gerar uma trajetória a ser seguida. A localização do veículo consiste em saber a pose, os seis DOF (*Degrees of Freedom*), do veículo em relação ao ambiente no qual está inserido. Dessa forma, deve-se ter inseridos no ambiente objetos conhecidos em poses conhecidas, os quais o veículo possa utilizar as detecções desses objetos para aferir sua pose no ambiente. Serão utilizados um sistema SBL (*Short BaseLine*) e arucos como esses objetos conhecidos.

O sistema SBL é um conjunto de quatro *beacons* acústicos que devem ficar emitindo um sinal acústico para o veículo, que através da trilateração, é capaz de estimar a posição em XY desse sistema. Esse sistema em conjunto com o sensor de pressão e a inferência da posição em Z provê a todo momento a posição em XYZ do veículo. Dessa forma, somente falta a rotação em XYZ, que pode ser aferida a partir do giroscópio presente na IMU.

Mesmo já sendo possível obter a pose 6DOF do veículo com essas três soluções, a detecção de Arucos, *dead-reckoning* e dados do acelerômetros também serão usados visando obtenção de uma localização mais precisa. A detecção por visão computacional de um aruco fixado ao ambiente possibilita a estimação da pose 6DOF do veículo, porém esse dado só está disponível enquanto o veículo aponta sua câmera para o aruco. O *dead-reckoning* consiste em utilizar o modelo cinemático de deslocamento do veículo com a velocidade de cada propulsor para saber o deslocamento 6DOF que deve acontecer em um certo período, contudo, essa técnica pode agregar

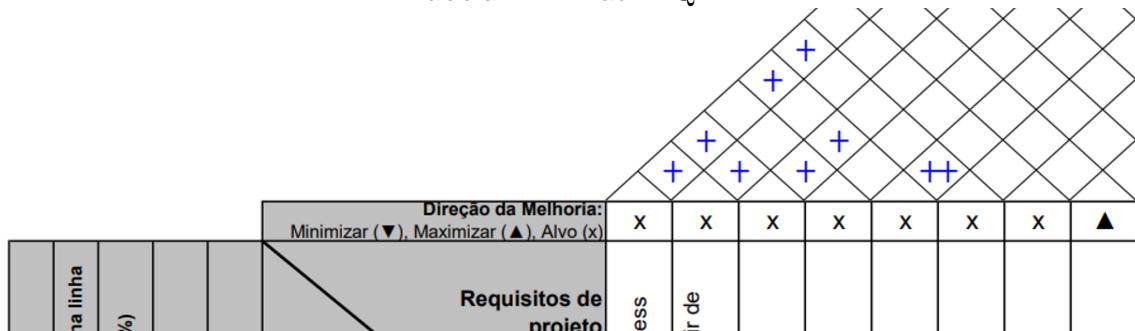
erros de modelagem, assim como de estimação do empuxo em cada propulsor. Por fim, o dado do acelerômetro integrado duas vezes provê dado de deslocamento de posição, mas como o erro do sensor também é integrado duas vezes, a técnica não é tão precisa. Dessa maneira, pelos problemas apresentados nesse parágrafo sobre a estimativa de pose com acelerômetro, arucos e *dead-reckoning*, essas serão usadas somente a partir da fusão por filtro de Kalman com as informações obtidas pelo SBL, sensor de pressão e giroscópio. O filtro de Kalman leva em consideração a precisão de cada dado durante sua fusão, então dados mais precisos tem mais peso do que dados imprecisos.

Uma possibilidade são os próprios marcos artificiais, porém precisam estar em uma pose conhecida previamente, então com a estimativa de sua pose em relação ao veículo submarino, se calcula a pose do veículo no mundo. Outra possibilidade é a triangulação de sinal, com três emissores de pulsos acústicos ou ópticos com posições previamente conhecidas, consegue-se estimar a pose do veículo. Por fim, o *dead-reckoning* também pode ser usado através de sensores iniciais e de estimadores de modelo cinematográfico e dinâmico do veículo. Esses componentes geram uma informação de deslocamento em relação a uma pose inicial, que é chamada de *dead-reckoning*, ou odometria no caso de veículos terrestres.

## Mapeamento

O mapa será gerado a partir das medições de distância realizadas com os transdutores do fundo do veículo. Cada medição virará um voxel em um mapa de ocupação 3D.

Tabela 2.1: Matriz QFD



The diagram shows a triangular QFD matrix with handwritten annotations in blue. The matrix has 'Requisitos de projeto' (Requirements) on the top row and 'Necessidades do cliente' (Customer Needs) on the left column. The diagonal line from bottom-left to top-right is crossed out with a large 'X'. Handwritten annotations include several '+' signs scattered across the matrix, with a large '+' sign at the top center.

Direção da Melhoria: Minimizar (▼), Maximizar (▲), Alvo (x)					X	X	X	X	X	X	X	X	▲
Número da linha	Máx. relacionamento na linha	Importância relativa (%)	Modelo Kano	Importância ou peso	Comunicação Wireless	Submergir e emergir de forma controlada	Seguir trajetória pré-estabelecida	Mapear leito	Localizar objeto	Localizar ROV	IHM	Suportar pressão	
1	9	16,1	M	5	Estabelecer comunicação com operador na superfície	○	○	○	▲	▲	○	▲	
2	9	9,7	E	3	Submergir e emergir de forma controlada	○	○	○	▲	▲	▲	▲	
3	3	3,2	A	1	Velocidade mínima de 2 m/s	▲	○	▲	▲	▲	▲	▲	
4	9	9,7	E	3	Seguir trajetória pré-estabelecida	○	▲	○	▲	▲	○	○	
5	9	3,2	A	1	Mapear leito submarino	▲	▲	▲	○	▲	○	▲	
6	9	12,9	M	4	Localizar objeto	○	▲	○	▲	○	○	▲	
7	9	12,9	M	4	Se deslocar até coordenada	○	○	▲	▲	○	○	▲	
8	9	12,9	M	4	Detectar sinal de socorro	○	▲	▲	▲	○	▲	▲	
9	9	16,1	M	5	Transmitir dados em tempo real	○	▲	▲	▲	▲	▲	▲	
10	9	3,2	A	1	Armazenar o caminho	▲	▲	▲	▲	▲	○	▲	
					Limite superior	-	200	29	3000	-	-	-	1230
					Alvo	-	140	8	1000	-	-	-	37
					Limite inferior	-	100	8	250	-	-	-	37
					Dificuldade organizacional								
					Máx. relacionamento na coluna	9	9	9	9	9	9	9	1
					Peso ou importância	551,6	241,9	351,6	125,8	229,0	358,1	312,9	100,0
					Importância relativa	24,3	10,7	15,5	5,5	10,1	15,8	13,8	4,4
					Ordem de atuação	1	5	3	7	6	2	4	8

Fonte: Autores

Tabela 2.2: Matriz morfológica

Requisitos	Definição do problema	Soluções			
Comunicação Wireless com Operador e Detecção de Sinal de Socorro	Transmissão de Sinal em Meio Subaquático	Acústica		Óptica	Eletromagnética
Sub/Emersão Controlada e Locomoção em 4 DOF	Medição de Profundidade	Ultrassom apontado para superfície		Sensor de pressão	Câmera e objetos pré definidos
	Sistema de Propulsão	Bomba de Porão		Motor CC	Motor Brushless
Seguir Trajetória Pré-estabelecida	Controle de Posição	Controle PID cascata		Controle multivariável com base em modelo	Controle não linear
Mapeamento de Leito	Sensores para Mapeamento	Múltiplos Ultrassom		Câmera Stereo	Sonar
Localização de Objeto	Método de Localização de Objeto	Marco Artificial Passivo		Marco Artificial Ativo	Rede Neural e Visão Computacional
Localização do Veículo Submarino	Técnicas	Trilateração (SBL)	Marcos Artificiais	Dead Reckoning	IMU
IHM	API e Métodos de Criação de GUI	Qt Creator	Swing	WebDesign	
Suportar Pressão	Estruturas Possíveis	Tubular (PVC / Aço Inox)			Monocoque

Fonte: Autores

### 3 Projeto Conceitual

O design do protótipo foi concebido a partir das definições oriundas da matriz morfológica e também de inspirações de ROVs comerciais, como o *Seafox* e o *BlueROV2*, mostrados na Figura 3.1.

Figura 3.1: ROVs que inspiraram o design do BROV: *Seafox* (à esquerda) e *BlueROV2* (à direita)



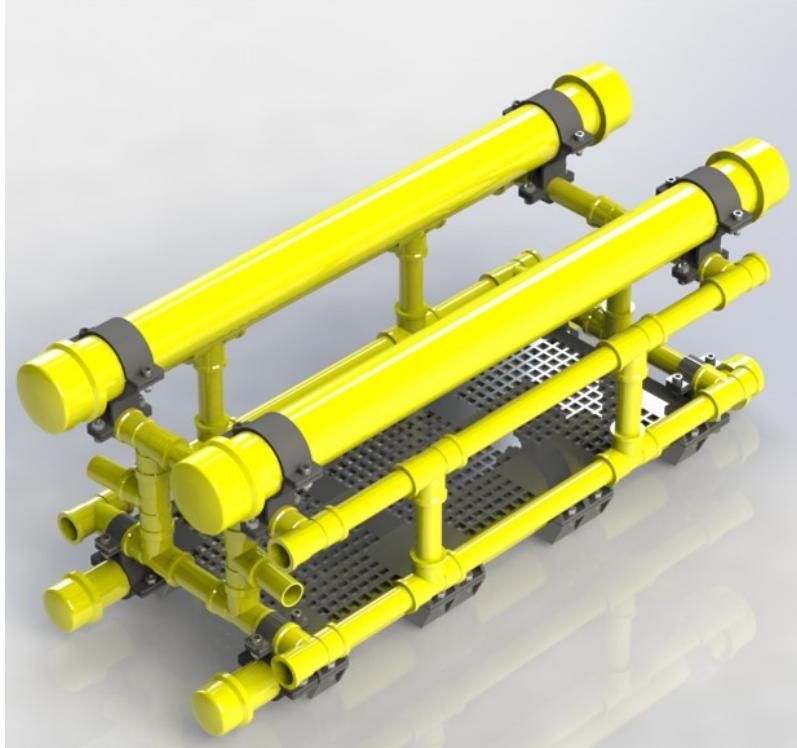
Fonte: [3] e [4]

Para escolha do material da estrutura do BROV foram levados em consideração o baixo custo e a versatilidade para futuras integrações de novos sensores. Dessa forma, optou-se por utilizar estrutura tubular, como a do *Seafox*. O design conceitual da estrutura está mostrado na Figura 3.2.

Para a fixação da estrutura tubular entre si foram utilizados conectores do tipo “tê soldável” e “joelho 90°”, itens comerciais retirados do catálogo da Tigre® [5]. Nos casos em que não foi possível a união tubular através desses conectores, optou-se por utilizar peças impressas 3D em ABS, modeladas exclusivamente para esse fim, evitando-se uso de abraçadeiras metálicas e do tipo “enforca-gato”, que viriam a prejudicar a estética do ROV. Julgou-se necessário também a utilização de uma base (impressa 3D) para apoio e fixação dos mais diversos tipos de componentes, como *housings*, fios e acessórios. Parafusos e porcas de aço inoxidável austenítico 304 foram selecionados como elementos padrão de fixação.

No design do ROV também foi levado em consideração a versatilidade no controle da flutuabilidade. Os dois maiores tubos da parte superior podem acomodar um grande volume de espuma expansível, que possui baixa densidade, da ordem de 0,1 g/cm<sup>3</sup>. Caso os tubos não resistam aos esforços de pressão na profundidade

Figura 3.2: Estrutura conceitual do BROV



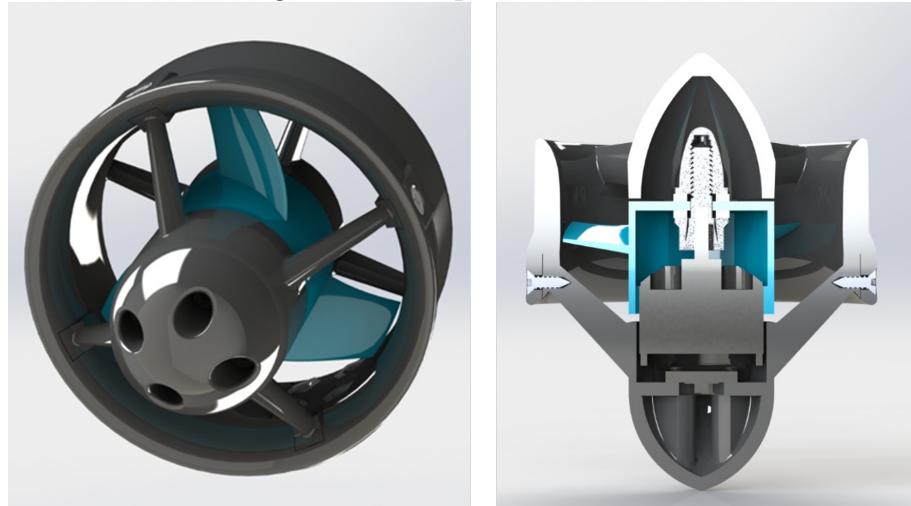
Fonte: Autores

especificada pelo projeto, a espuma poderá ser aplicada para evitar o colapso do tubo, ao mesmo tempo que favorecerá o empuxo se comparado com o preenchimento com água. De modo análogo, os tubos inferiores foram pensados para um eventual acréscimo discreto de massa – com bolas de gude, por exemplo – para ajuste da altura do centro de gravidade do protótipo. Os tubos menores, que compõem a parte central da estrutura, foram mantidos abertos para circulação de água.

O sistema de propulsão foi fortemente inspirado no *BlueROV2*, que é um ROV *open source* da empresa *BlueRobotics*. Algumas alterações em nível de escala, elementos de fixação e componentes do duto canalizador foram feitas em relação aos presentes no *BlueROV2*, de modo a tornar a montagem compatível com o motor *brushless A2212*, que será utilizado no projeto. A Figura 3.3 mostra o propulsor que será utilizado no BROV.

Os motivos pela escolha desse modelo de motor foram: não necessidade de selagem, facilidade de montagem, alto torque e baixo *lead time*. A alocação dos *thrusters*

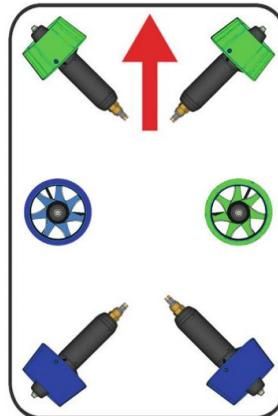
Figura 3.3: Propulsor do BROV



Fonte: Autores

– mostrada na Figura 3.4 – foi pensada de modo a obter o controle dos 3 graus de liberdade translacionais – *surge*, *sway* e *heave* – e de 2 rotacionais – *roll* e *yaw* [6].

Figura 3.4: Posicionamento dos thrusters

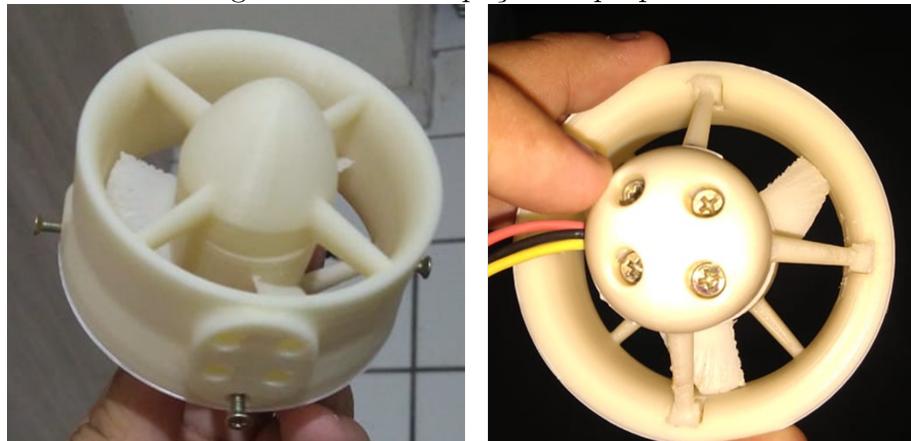


Fonte: [6]

No momento em que foi escrito esse relatório o propulsor se encontrava em fase de testes, tendo a hélice e demais acessórios sido fabricados em ABS, em uma impressora 3D, conforme indicado pela Figura 3.5. A próxima etapa a ser realizada para modelagem computacional do sistema de propulsão consiste no levantamento

da curva Corrente x Torque do motor através de testes em um aquário adquirido pela equipe, conforme indicado pela Figura 3.6, de modo que seja possível estimar o comportamento dinâmico do ROV quando submerso.

Figura 3.5: Prototipação do propulsor



Fonte: Autores

Figura 3.6: Aquário para testes de propulsão e comunicação

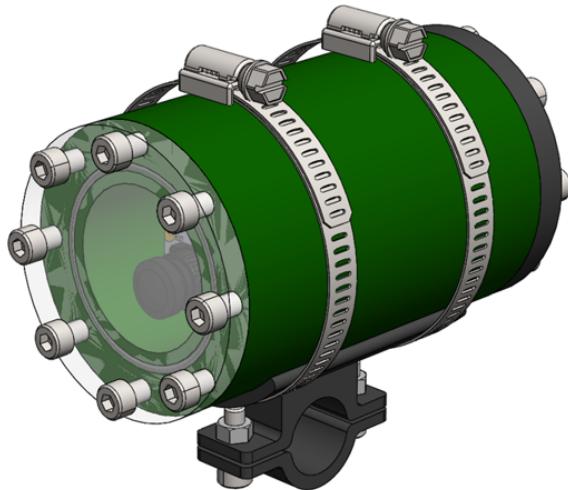


Fonte: Autores

*Housings* tiveram que ser projetadas para uma vedação confiável da câmera monocular juntamente com seu controlador, assim como dos componentes eletrônicos de controle – Arduino, IMU e bateria.

O corpo da *housing* da câmera foi concebido para ser impresso 3D, com um visor de policarbonato na parte frontal, comprimido com parafusos e vedado com *o-ring*. A fixação na estrutura se dará por meio do uso de abraçadeiras metálicas, de modo que seja permitida a sua rápida montagem e desmontagem, sem que haja nenhum tipo de perfuração. A Figura 3.7 mostra o projeto conceitual dessa *housing*. A outra *housing*, que acomodará os demais componentes eletrônicos e bateria, ainda está em desenvolvimento.

Figura 3.7: Projeto conceitual da *housing* da câmera e seu controlador



Fonte: Autores

Para prover iluminação na profundidade de 10 m, a equipe optou por utilizar uma solução de mercado já pronta: uma lanterna de pesca à prova d'água de baixo custo com fonte de alimentação própria, mostrada na Figura 3.8.

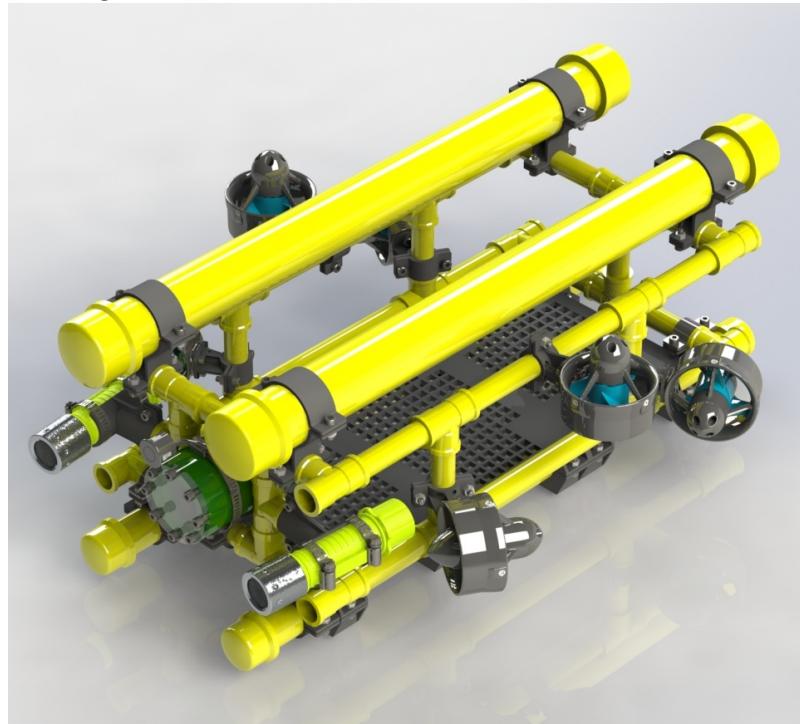
Além dos componentes mencionados acima, também serão acomodados no BROV sensores de ultrassom para mapeamento e comunicação. Todos esses itens estão representados nas vistas de montagem das Figuras 3.9, 3.10, 3.11 e 3.12.

Figura 3.8: Lanterna que será usada no BROV



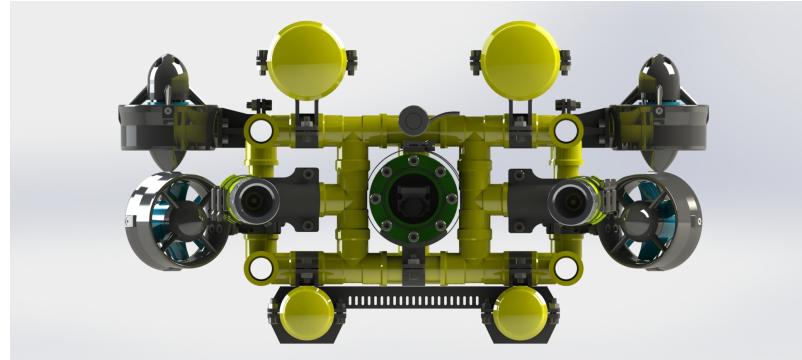
Fonte: Banggod

Figura 3.9: Vista isométrica do BROV conceitual



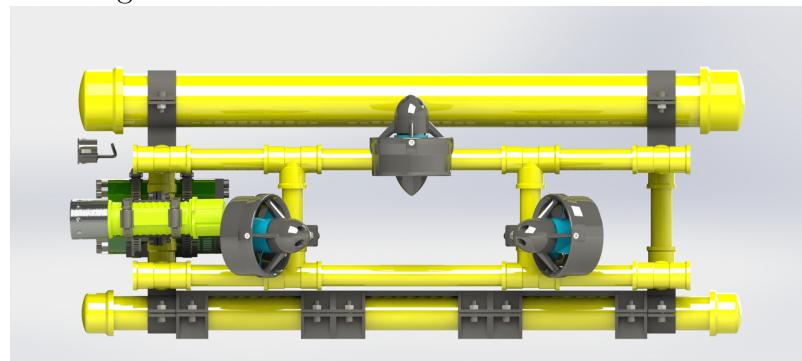
Fonte: Autores

Figura 3.10: Vista frontal do BROV conceitual



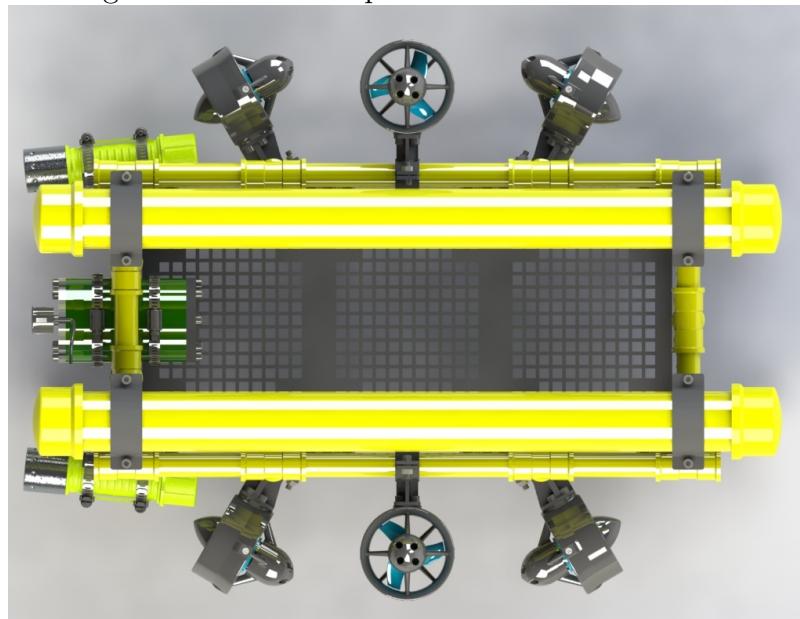
Fonte: Autores

Figura 3.11: Vista lateral do BROV conceitual



Fonte: Autores

Figura 3.12: Vista superior do BROV conceitual



Fonte: Autores

## 4 Validação de projeto

Ao longo do texto, definiu-se os requisitos e como os atingi-los, porém ainda é preciso estabelecer como ocorrerá a prova do cumprimento do que foi acordado. Para isso, nessa seção serão propostos alguns testes que são capazes de validar o cumprimento dos requisitos. Essa etapa de validação poderia ser executada somente após o desenvolvimento do protótipo, contudo, quaisquer modificações de projeto com o protótipo já finalizado acrescentaria em um grande custo de financeiro e em atraso de projeto. Dessa forma, optou-se pela validação do protótipo via simulação, o que permite testar o veículo em ambiente subaquático simulado, sem risco de dano ao veículo e evitando grandes retrabalhos por modificações no projeto. Nessa seção, será discutido como a simulação deve ser realizada para validação dos requisitos de projeto, assim como a configuração do ambiente virtual submarino.

### 4.1 Módulos de simulação

A simulação fidedigna de um veículo submarino deve contar com a simulação dos atuadores, sensores e da física de um corpo rígido debaixo d'água. Com o objetivo de obter uma simulação que incluísse todos esses tópicos, algumas soluções de *software* disponíveis serão adotas e algumas desenvolvidas, a especificação de cada uma e justificativa de escolha é o objetivo desta seção.

#### Sensores, atuadores e física

As funções de simular o comportamento de um robô no meio com seus sensores e atuadores é a tarefa de um simulador de robótica, dessa forma, é necessário se definir qual deve ser utilizado para a validação deste projeto. A tabela da Figura ?? mostra as alternativas de simuladores que possuem funcionalidades essenciais para validação de um robô submarino. Dentre as cinco alternativas apresentadas na tabela, 3 delas utilizam o simulador Gazebo [?], que são UWSim (*Underwater Simulator*), UUV (*Unmanned Underwater Vehicle Simulator*) e USVSim (*Unmanned Surface Vehicle Simulator*). Dentre esses, o USVSim seria o mais completo, pois dispoẽ de quase todas as funcionalidades apresentadas na Figura ??.

Contudo, o USVSim é desenvolvido a partir do Gazebo, então seria necessário a

Tabela 4.1: Tabela comparativa de simuladores e suas funcionalidades

Simulator	Hydrodynamics	Hydrostatics	Thruster	Fins	Pressure Sensor	GPS	DVL	Realistic Rendering	Contact Dynamics	Wind	Waves	Water Currents
UWSim	✓	✗	✓	✗	✓	✓	✓	✓, osgOcean	✓	✗	✓	✗
UUV	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗
Stonefish	✓	✓	✓	✗	✓	✓	✓	✓, custom	✓	✓	✓	✓
URSim	✓	✓	✓	✗	✓	✗	✗	✓, Unity	✗	✗	✓	✗
USVSim	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓

Fonte: [?]

instalação de duas ferramentas, sendo que a Open Robotics, empresa desenvolvedora do Gazebo, lançou o novo simulador que substituirá o Gazebo, o Ignition. Ademais, o Ignition já apresenta as funcionalidades de robótica submarina incorporadas, dessa forma, optou-se por utilizar somente o *software* Ignition. Nele é possível se carregar um ambiente submarino 3D assim como o veículo, sendo possível não somente a visualização, mas também obter dados simulados de sensores, controlar motores e simular a interação do robô com o ambiente ao seu redor.

### Comunicação e sensor de distância acústicos

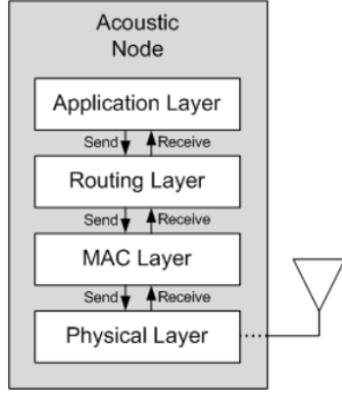
Apesar do Ignition possibilitar a simulação dos motores e sensores do veículo, a simulação da comunicação e sensor de distância acústicos é uma funcionalidade que não consta no *software*. Portanto, serão desenvolvidos *plugins* para o *software* que simulam as peculiaridades desse tipo sensores. Os *plugins* devem simular atraso da onda sonora na água, ruído e atenuação.

É extremamente caro e complexo implantar uma estrutura completa de rede subaquática com links de dados para coletar e transmitir informações de uma coluna de água, composta por nós de sensores, gateways na superfície e AUVs [7]. Pois, os dispositivos custam caros devido à baixa de proteção, são propensos a falhas por causa da corrosão e incrustação, as baterias limitadas não podem ser recarregadas com energia solar e a manutenção é complicada por estarem submersos [8, 9].

Além do mais, é difícil validar os protocolos ou algoritmos usados na rede UWSN por meio de testes, os planos de testes implantados podem não ser adequado para todos os tipos de aplicações, implementação repetida afeta os resultados experimentais e ainda é demorada [10].

Diante do exposto acerca dos desafios citados, a existência de um ambiente de simulação que replicasse o cenário subaquático real, permitiria ao engenheiro ou

Figura 4.1: Estrutura de programação de um nó acústico AUVNetSim



Fonte: [11]

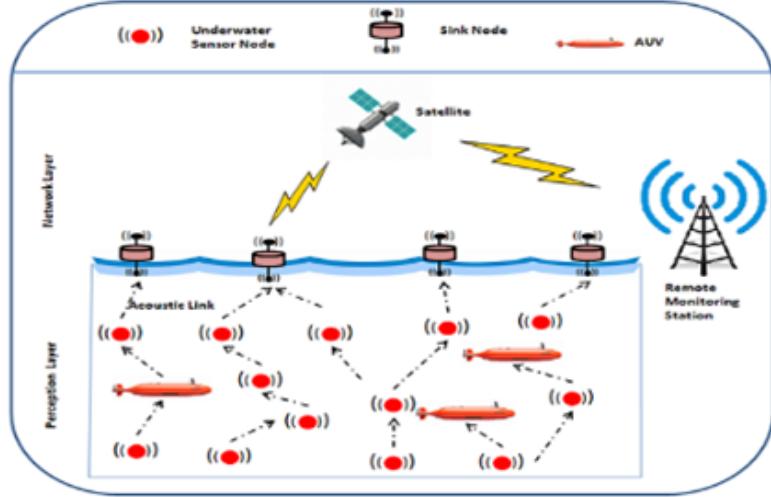
pesquisador estudar os comportamentos com base nas características físicas, por meio dos testes de seus modelos em par com a implementação em tempo real.

Para simular a comunicação, no presente trabalho será utilizado a biblioteca de simulação escrita em Python que objetiva testar algoritmos de rede acústica chamada AUVNetSim: A Simulator for Underwater Acoustic Networks produzido pelo [11] em um programa no MIT(do inglês, Massachusetts Institute of Technology) em 2008. O pacote contém uma grande variedade parâmetros e protocolos para redes acústicas subaquáticas que podem ser selecionadas ou modificados para implementar novos protocolos aproveitando a estrutura existente.

Além disso, na simulação de cada nó acústico a comunicação é realizada por troca de mensagens curtas entre as camadas que podem ser configuradas na seguinte ordem: física, MAC, roteamento e aplicação conforme é apresentado na Figura 4.1 logo abaixo.

Apesar de existirem outras propostas de ferramentas disponíveis no mercado com mesma característica e até melhores que o AUVNetSim, como Network Simulator (NS-3), SUNSET ou Aqua-Sim que permitem além da emulação, testes em tempo real [7] o que aproximaria ainda mais do sistema mais realista . Entretanto, a biblioteca foi selecionada por ser de código aberto e mesmo que ainda não testada a fundo, apresenta-se através do manual de utilização [11] que é de fácil integração e manutenção, os pré-requisitos não exigem recursos computacionais muito pesados para a sua utilização, e a curva de aprendizado é reduzida em comparação a outras

Figura 4.2: Arquitetura de comunicação subaquática



Fonte: [7]

opções e o prazo de entrega do projeto.

Assim, à adoção do AUVNetSim tem como objetivo integrá-la com Robot Operating System (ROS), com a finalidade de simular a comunicação entre os dispositivos em uma topologia de rede com a arquitetura tridimensional com AUV conforme proposta apresentada na Figura 4.2. Do mesmo modo, ainda serão estudados, avaliados e testados outros protocolos propostos na literatura com o propósito de se obter uma configuração satisfatória e que se aproxime do mundo real na simulação de como o todo.

## 4.2 Missões

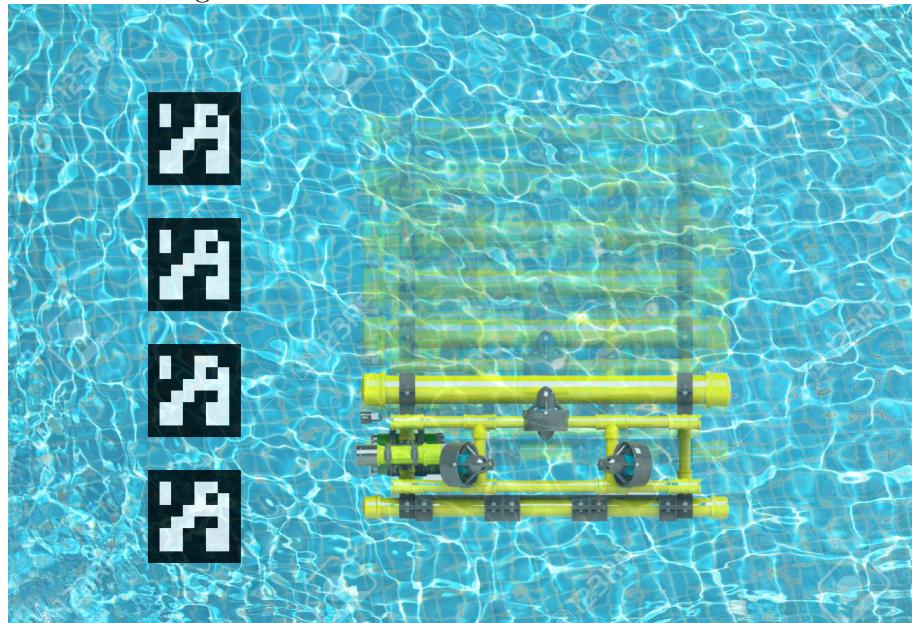
As missões são tarefas definidas para validação das funcionalidades do veículo, de forma que ao cumprir cada missão, o projeto fica validado a nível de simulação para o cumprimento dos requisitos de projeto.

### Submersão e emergência controlada

Essa missão consiste em submergir e emergir o ROV a uma determinada profundidade, como pode ser visto na Figura 4.2. Esse procedimento será executado tanto com controle manual de velocidade, como também com controle automático, no qual

o ROV se desloca ao longo de um caminho previamente estabelecido. Além disso, arucos seriam posicionados ao longo de todo o trajeto, para obter uma medição mais precisa de profundidade.

Figura 4.3: Missão de submersão e emersão



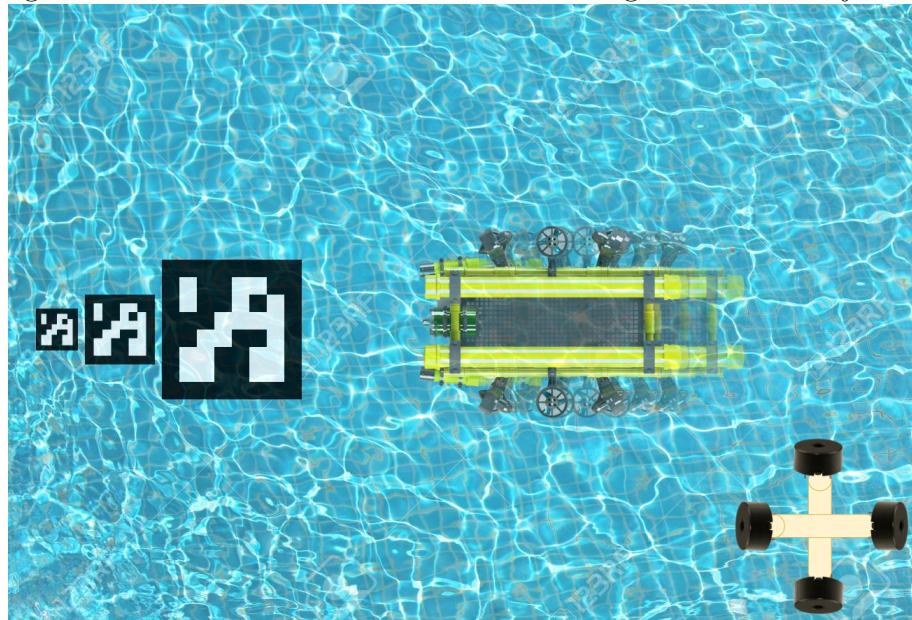
Dessa forma, os requisitos 2, 6 e 7 seriam validados, visto que para a execução dessa tarefa, o ROV deve ser capaz de ir para uma posição previamente conhecida, localizar os arucos e submergir e emergir de forma controlada.

### Teste de velocidade e trajetória circular

O requisito 3 determina a velocidade máxima que o veículo deve atingir, então é necessário que haja algum teste em que os motores do veículo fiquem no seu máximo para verificar se a velocidade determinada consegue ser atingida. O ROV foi projetado para ter mais velocidade em *surge*, de forma a se deslocar mais rápido ao se mover para frente. Dessa forma, o teste de velocidade consiste em colocar a referência de velocidade no seu valor máximo e medir a velocidade de deslocamento do ROV nesse trecho, porém essa medição é pouco precisa com a configuração de sensores atuais.

Portanto, para medir a velocidade, serão posicionados arucos de diferentes tamanhos e o sistema de SBL. A configuração do ambiente de teste pode ser observada na Figura 4.2. Os arucos tem tamanho variado pois quanto maior o aruco, mais longe deve-se estar para medir sua pose com precisão, então arucos de diferentes tamanhos servirão para ter uma boa medição em um intervalo maior de distância do veículo para o aruco.

Figura 4.4: Missão de velocidade máxima e seguimento de trajetória



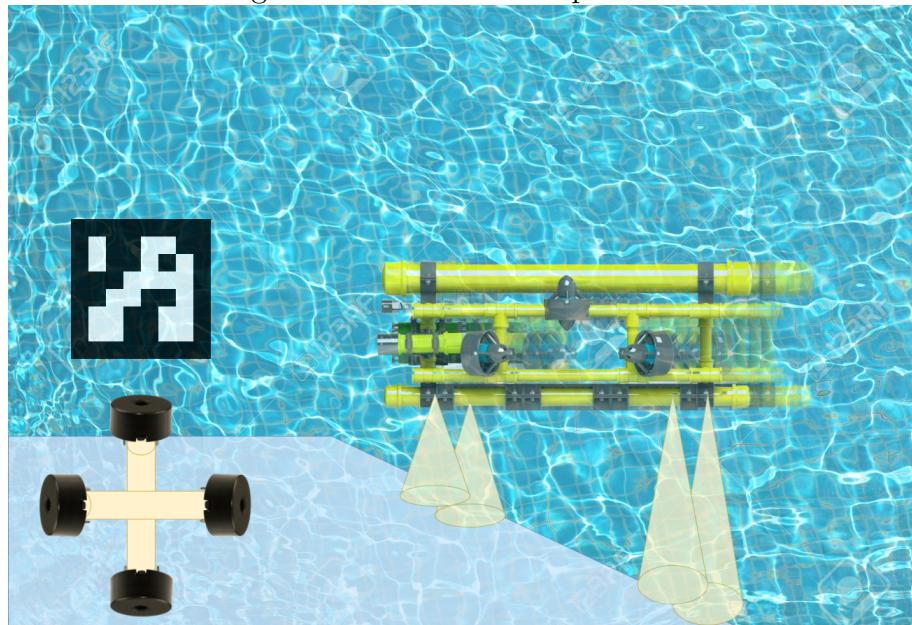
Outra missão que será executada nessa mesma configuração é a execução de uma trajetória circular, para validação dos requisitos 4, 6 e 7. Ao se executar a trajetória circular, mantendo a profundidade e mantendo os arucos no campo de visão da câmera, pode-se usar o armazenamento da trajetória para validar se o requisito de seguir uma trajetória realmente foi cumprido.

### Mapeamento de rampa

A validação do requisito 5, que se refere ao mapeamento do leito marinho, pode ser validado ao se utilizar um terreno conhecido. Então, propõe-se aqui a execução do simples mapeamento de uma rampa, caso a inclinação da rampa seja semelhante ao do mapeamento, pode-se afirmar que o sistema funciona. O mapeamento não será

muito detalhado, visto que irá se adquirir somente informação de 4 pontos, medidos com sensores de distância ultrassom abaixo do ROV. Portanto um cenário simples como uma rampa é adequado para validação. Além disso, também se necessita do aruco para posicionamento do ROV, como na Figura 4.2.

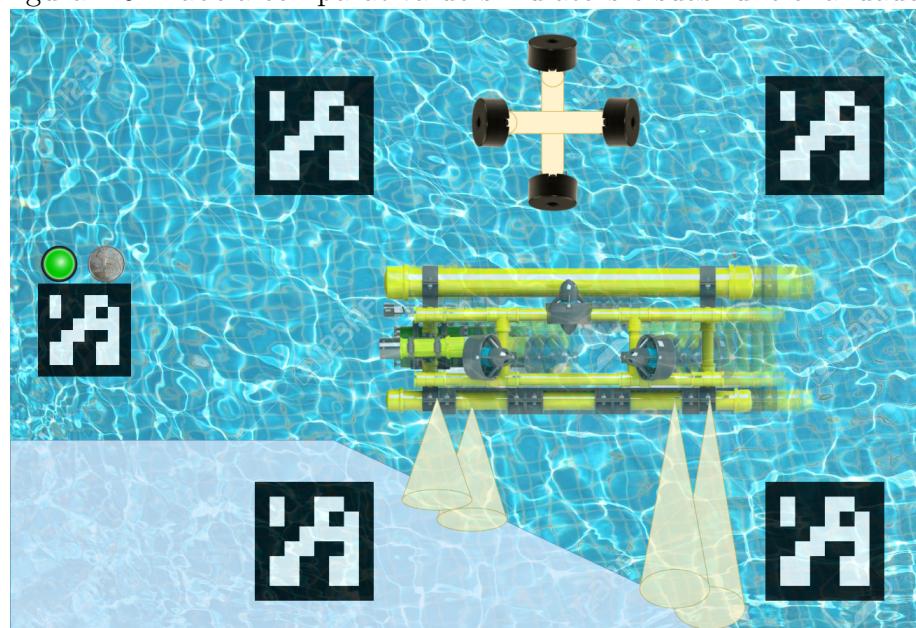
Figura 4.5: Missão de mapeamento



## Resgate da Moeda

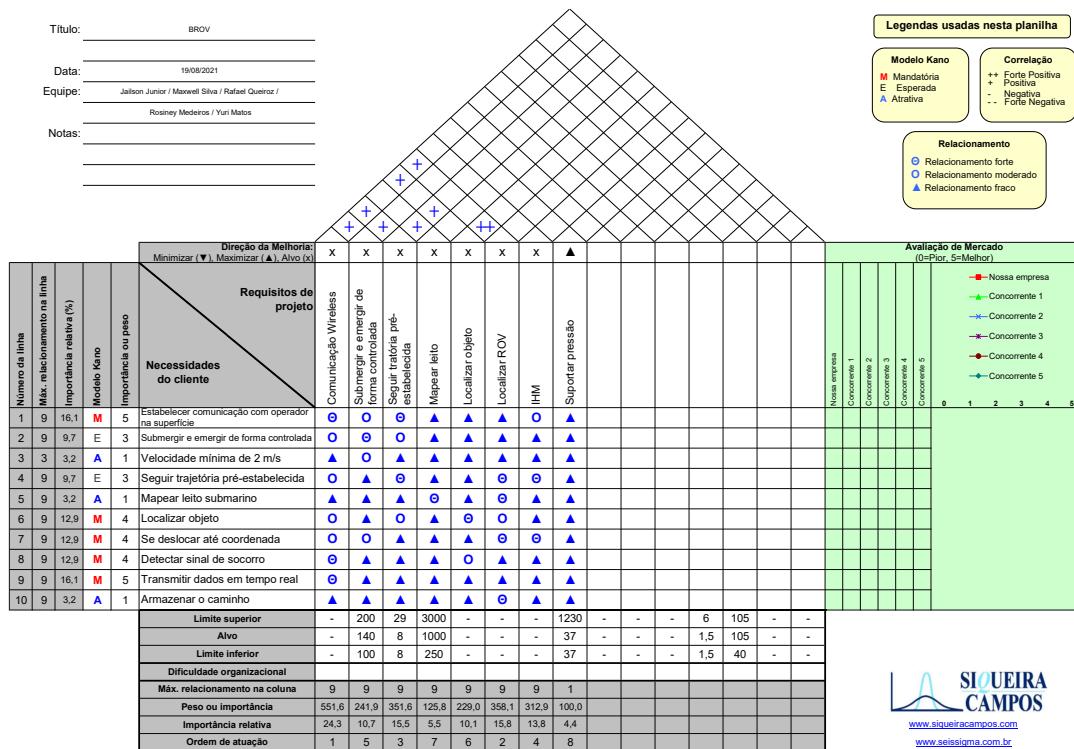
As missões apresentadas até agora cobriram a validação de todos os requisitos de sistemas que não envolvem resistência e confiabilidade, a validação que está faltando quanto a funcionalidade é a 8. Esse requisito determina que o ROV deve ser capaz de atender um pedido de socorro. Para a validação desse requisito, o BROV deve localizar a caixa de resgate, representada pelo aruco mais à esquerda da Figura 4.2. Além disso, deve esperar o led presente na caixa ascender, a partir daí, o ROV deve se aproximar e capturar a moeda com seu imã e retornar à superfície. Esse teste não só valida o requisito 8, mas como todos os citados anteriormente, a ideia é testar o sistema atendendo todos os requisitos funcionais do ROV ao mesmo tempo. Para esse teste, tanto a localização por aruco como a por SBL .

Figura 4.6: Tabela comparativa de simulators e suas funcionalidades



## A Apêndice A

Tabela A.1: Matriz QFD



Fonte: Autores

## Referências

- [1] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2014.
- [2] Obaid Ur-Rehman and Natasa Zivic. Wireless communications. *Signals and Communication Technology*, 19(1):7–21, 2018.
- [3] Carlos R Rocha, Rogério M Branco, Lais A Cruz, Marcos V Scholl, Matheus M Cezar, and Felipe D Bicca. Design Aspects Of An Open Platform For Underwater Robotics Experimental Research. (June 2016), 2014.
- [4] Blue Robotics. BlueROV2.
- [5] Tigre. Ficha Técnica - Linha Roscável, 2009.
- [6] Gianluca Antonelli. *Underwater Robots Fourth Edition*. Springer International Publishing AG, 4th edition, 2018.
- [7] Geeta Godi et al. Survey on simulators for underwater sensor networks. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(13):1719–1728, 2021.
- [8] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257–279, 2005.
- [9] A Shantaram, H Beyenal, R Veluchamy, and Z Lewandowski. The challenges of building scalable mobile underwater wireless sensor networks for aquatic applications. *Environ. Sci. Technol.*, 39:5037–5042, 2005.
- [10] Anjana P Das and Sabu M Thampi. Simulation tools for underwater sensor networks: a survey. *Network protocols and Algorithms*, 8(4), 2016.
- [11] Josep Miquel Jornet Montana. Auvnetsim: A simulator for underwater acoustics networks. Technical report, Massachusetts Institute of Technology. Sea Grant College Program, 2008.