

Sistema de Acompanhamento de Missão Para Uma Plataforma Experimental De Robótica Subaquática

Marcos Vinicius Scholl¹, Carlos Rodrigues Rocha²

¹Universidade Federal do Rio Grande (FURG) - Campus Carreiros: Av. Itália km 8
Bairro Carreiros, Rio Grande (RS) - Brasil

²Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) -
Campus Rio Grande, R. Eng. Alfredo Huch, 475 - 96201-460 - Rio Grande(RS) – Brasil

marcos.vinicius.scholl@gmail.com, carlos.rocha@riogrande.ifrs.edu.br

Abstract. The design and implementation aspects of a mission monitoring and control system for underwater unmanned vehicles are presented in this work. This system is part of a wider research project to develop an open experimental platform for underwater robotics, in order to stimulate more research studies in this area.

Resumo. Neste trabalho são apresentados aspectos de projeto e implementação de um sistema de acompanhamento de missão para veículos subaquáticos não tripulados. Este sistema é parte de um projeto de pesquisa que tem por objetivo desenvolver uma plataforma experimental aberta para robótica subaquática a fim de viabilizar um maior número de trabalhos nessa linha de pesquisa.

1. Introdução

A robótica subaquática é uma área de pesquisa que apresenta uma série de desafios, e por isso é motivadora de vários trabalhos. No Brasil, em particular, há uma motivação forte advinda da necessidade de tecnologia nacional para a exploração de petróleo e gás da camada do Pré-Sal, que se dá a grandes profundidades [PETROBRAS, 2011].

Segundo BUTTON *et al.*[2009], a autonomia é o grande desafio de médio e longo prazo da robótica subaquática. Para tanto, vários problemas devem ser resolvidos no que tange a hardware e software, de forma que o ambiente e o sistema robótico possam ser apropriadamente modelados e que haja condições do robô planejar seu movimento de forma autônoma. WHITCOMB[2000], por sua vez, indica a importância de se conduzir não apenas desenvolvimentos teóricos e simulados, mas também validações experimentais, pois o ambiente subaquático é de difícil modelagem e extremamente dinâmico. Porém, a pesquisa experimental em robótica subaquática tem vários fatores limitantes, entre eles o elevado custo dos equipamentos envolvidos, e o fato da maioria deles ter tecnologia proprietária e fechada, o que dificulta o acesso aos dados de sensores, câmeras e atuadores dos sistemas robóticos.

Por esse motivo, foi proposto um projeto de construção de uma plataforma experimental para robótica subaquática que fosse aberta e de relativo baixo custo. Para tanto, foram adotados como características do desenvolvimento do projeto o emprego de tecnologias

open source de hardware e software e a disposição de manter o projeto final *open source* também, de forma a facilitar o acesso à plataforma para os interessados em realizar pesquisa experimental em robótica subaquática. Esse projeto está em desenvolvimento por um grupo de professores/pesquisadores do Campus Rio Grande do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS-Rio Grande), alunos do curso Técnico em Automação Industrial desse Instituto e alunos do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Rio Grande (FURG).

O presente trabalho aborda um dos componentes da plataforma experimental, que é o *Sistema de Monitoramento e Controle de Missão* (ou MMCS, do inglês *Mission Monitoring/Control System*). Também chamado de Sistema de Acompanhamento de Missão, a sua função é possibilitar aos usuários do robô a sua teleoperação, bem como a visualização de informações do ambiente subaquático obtidas por sensores e câmera embarcados no sistema. Outra função do MMCS é servir de base para a implementação gradual de características autônomas para o robô.

Na próxima seção, será apresentada a motivação para o trabalho e desenvolvimentos relacionados. A seguir, define-se o modelo conceitual da plataforma, bem como as tecnologias de hardware e software adotadas. Após, descrevem-se os primeiros resultados da implementação dos componentes de software, rodando sobre a plataforma de hardware escolhida. Na conclusão, discute-se o que foi obtido até o momento e trabalhos futuros que poderão advir desse projeto.

2. Contextualização

Atualmente, robôs subaquáticos são usados na maioria das atividades em ambientes submersos, desde a pesquisa oceanográfica até a exploração de petróleo e gás *offshore*. São empregados tanto equipamentos teleoperados, comumente chamados de ROV (de *Remotely Operated Vehicle*, em inglês) quanto autônomos, ou AUV (de *Autonomous Underwater Vehicle*, em inglês). O motivo de não se utilizarem apenas sistemas autônomos deve-se ao fato de ainda existirem muitos problemas a serem resolvidos para que qualquer tipo de operação possa ser realizado de forma autônoma. Em particular, atividades de intervenção, onde o robô deve manipular objetos do meio em que se encontra, ainda têm muitos problemas não resolvidos [ANTONELLI, 2006].

Assim, a pesquisa em robótica subaquática é bastante ativa. Muitos dos desenvolvimentos, porém, são teóricos e analisados em simulações, pois esses sistemas robóticos são caros, bem como de operação onerosa. No caso da pesquisa experimental, ainda deve-se lidar com o problema das arquiteturas de hardware e software fechadas e proprietárias, o que dificulta a implementação dos experimentos. Com isso, vários projetos de sistemas robóticos subaquáticos surgiram em instituições de pesquisa de todo o mundo. Entre os trabalhos mais recentes, podem ser citados o SeaPerch [SEA, 2013], o SAUVIM [MARANI, CHOI, YUH, 2009] e o RAUVI [DE NOVI et al. 2010]. Em termos de desenvolvimentos nacionais, destacam-se o pioneirismo de HSU et al.[2000] e BARROS e SOARES[2002], e projetos recentes como o ROVFURG [CENTENO, 2007; KHUN, 2011].

A motivação para o projeto da plataforma experimental aberta, da qual o sistema de acompanhamento de missão objeto deste trabalho faz parte, surgiu da observação dos projetos supracitados e de outros e da necessidade da validação experimental da metodologia de planejamento de movimento de sistemas robóticos subaquáticos cooperativos por teoria de helicoides desenvolvida por um dos pesquisadores [ROCHA, 2012]. Este trabalho utilizou tecnologias livres para o desenvolvimento de um framework de modelagem cinemática por helicoides, que serviu de base para um estudo de como criar uma arquitetura que poderia ser utilizada não apenas em simulação, mas em um sistema para experimentação que fosse aberto e extensível. Da análise de requisitos dessa necessidade, surgiu o projeto da plataforma aberta. O acompanhamento de missão, nessa proposta, é o módulo a partir do qual tanto a teleoperação quanto a implementação das estratégias de planejamento de movimento serão feitas.

3. Projeto Conceitual

Para estruturar o desenvolvimento do projeto, empregou-se a metodologia PRODIP (Processo de Desenvolvimento Integrado de Produtos) [BACK *et al.*, 2008]. A partir desta, uma análise inicial da plataforma resultou no levantamento de requisitos e na criação de concepções de soluções que atendessem aos requisitos identificados. Destas, a solução mais adequada aos requisitos principais (como ser uma plataforma aberta/livre e de relativo baixo custo) foi selecionada.

Nesta solução, a plataforma é formada por dois componentes, o veículo subaquático não tripulado (ou UUV, do inglês *Underwater Unmanned Vehicle*) e o MMCS. A visão geral do projeto conceitual criado é apresentada na Figura 1.

Embora o conjunto seja formado pelo UUV e pelo MMCS, apenas este será abordado no restante do texto, por ser o foco do presente trabalho. Após a definição do hardware a ser empregado para a construção do MMCS, foi definida a plataforma de desenvolvimento do software de acompanhamento de missão e feito um projeto de sua estrutura, baseada no paradigma da orientação a objetos. Estes elementos serão detalhados a seguir.

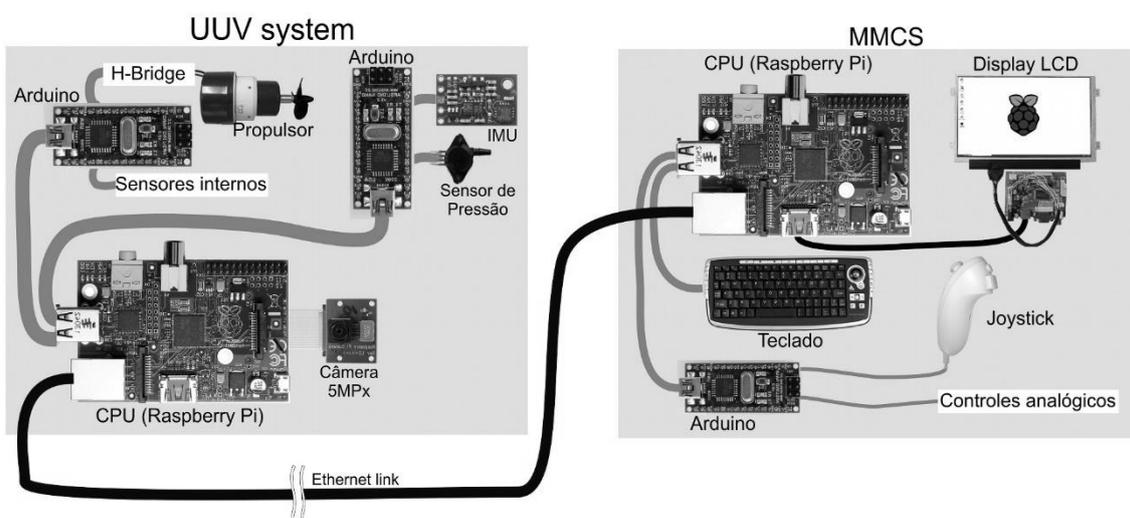


Figura 1 Visão geral do hardware da plataforma

4. Tecnologias de Hardware e Software Adotadas

Na construção do sistema de acompanhamento de missão, optou-se por projetar e construir um hardware portátil para poder ir a campo com facilidade. Diferentemente de um laptop/notebook, o hardware deve ser mais resistente às características de ambientes externos e ter controles analógicos. Em relação ao software, ele deveria ser aberto, extensível e permitir fácil acesso aos seus componentes, o que motivou o uso de software livre. Além disso, ele deve ser multiplataforma, capaz de rodar tanto no hardware construído para esse fim quanto em notebooks com sistemas operacionais conhecidos, caso se optasse por esse tipo de equipamento. Assim, foram escolhidas as tecnologias a seguir para a implementação do sistema de acompanhamento de missão.

4.1. Hardware

Arduino: Plataforma de prototipação muito popular, tem baixo custo, é aberto e tem grande base de conhecimento disponível. Assim, ele foi adotado para a construção da interface analógica, com controles como um *joystick* wii Nunchuk e outros sensores. Além disso, haverá indicadores como leds e display LCD. O modelo utilizado na construção do módulo analógico é o Arduino Nano [ARDUINO, 2013].

Raspberry Pi: Esta placa de computação de baixo custo é do tipo *System-on-a-Chip* (SoC). Ele será a Unidade Central de Processamento do sistema de acompanhamento de missão, por ter baixo consumo de energia, capacidade de processamento suficiente para as necessidades do software, disponibilidade de rede Ethernet, e dimensões reduzidas. O RPi suporta diversos sistemas operacionais, entre eles distribuições Linux, o que permite tratá-lo como um computador similar aos PCs [RASPBERRY, 2013].

Entradas/Saídas: Para a interface com o usuário, serão empregados um display LCD de 10 polegadas portátil de alta resolução com controlador de vídeo dedicado, leds indicadores de estado de operação do sistema, teclado com *trackball*, *joystick* com diversos eixos, além de alguns potenciômetros.

4.2. Software

O software da plataforma deve ser extensível e facilmente adaptável às necessidades de experimentação. Assim, optou-se pela abordagem orientada a objetos para o projeto e posterior implementação. No caso do sistema de acompanhamento de missão, era necessário definir componentes visuais, ou *widgets*, para apresentarem informações de sensores e câmera do UUV. Além desses, deveriam haver classes que gerenciassem a comunicação entre o veículo e o acompanhamento de missão, necessária para a obtenção dos dados dos sensores e envio dos comandos para teleoperação do UUV. Assim, foi definido o diagrama de classes mostrado na Figura 2, que descreve as classes inicialmente definidas e o relacionamento entre elas.

A escolha da plataforma de desenvolvimento do software levou em consideração a necessidade dele ser multiplataforma e facilmente modificável/extensível, além de orientado a objetos. Além disso, a plataforma deveria ter disponibilidade de bibliotecas matemáticas e boa base de software/conhecimento existente. Assim, optou-se pela utilização de Python [PYTHON, 2014].

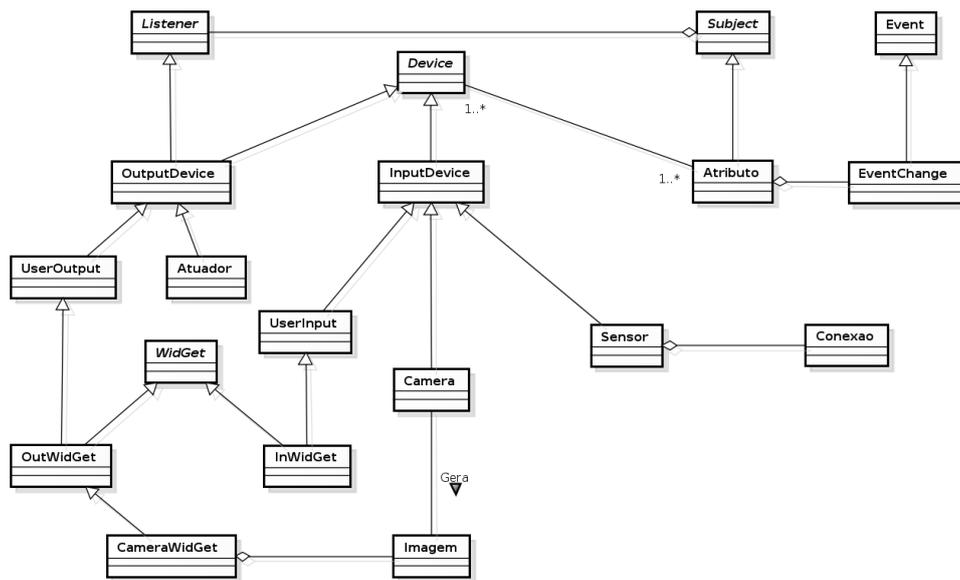


Figura 2 Diagrama de classes do sistema de acompanhamento de missão

Além de ser multiplataforma, a linguagem é bastante utilizada no meio científico tanto para o desenvolvimento quanto para a integração entre sistemas. A disponibilidade de bibliotecas numéricas/científicas facilitam o seu uso para aplicações em robótica [OLIPHANT, 2007]. Em espacial, a disponibilidade da biblioteca OpenCV é relevante para o processamento de imagens da câmera do UUV [HOWSE, 2013]. Para a interface gráfica com o usuário foi adotado o Qt, entre as diversas possibilidades existentes. A biblioteca QWT tornou disponíveis componentes visuais customizáveis como escalas, *gauges* e *displays*, além dos *widgets* presentes no Qt [RATHMANN; WILGEN, 2014].

5. Desenvolvimento da Interface Gráfica

Os componentes visuais foram definidos com base no trabalho de ROCHA[2012], que analisou a arquitetura de um sistema robótico subaquático e do subsistema de acompanhamento de missão. *Displays* textuais de dados numéricos, indicadores visuais binários (on/off, ou normal/problema), *gauges* ou escalas, bússola e horizonte artificial foram identificados como os básicos necessários para a interface. Alguns desses componentes eram parte da biblioteca QWT, necessitando de adaptações às necessidades do software. Outros componentes foram implementados. Em ambos os casos, a tarefa foi bastante simplificada pelo uso do mecanismo de herança. A principal modificação foi a atualização automática da apresentação dos dados, obtidos de provedores de dados (descendentes de *InputDevice*), que representam os sensores do UUV junto ao sistema de acompanhamento de missão.

Os componentes implementados até o momento são brevemente apresentados a seguir. A Figura 3 mostra a interface homem-máquina (aqui denominada *cockpit*) criada com o uso desses componentes, para teste de suas funcionalidades.

5.1. Gauge/Escala

É utilizado para mostrar dados numéricos entre limites mínimos e máximos. Na interface da Figura 3 (3.a), ele é utilizado para mostrar a profundidade, que é medida

através de um sensor de pressão absoluta instalado no UUV. Assim, além da informação da profundidade em si, tem-se uma noção dos limites de operação do veículo.

5.2. Bússola

A bússola apresenta a informação de direção do veículo (ou *heading*, em inglês). Ela é medida pelo magnetômetro de 3 eixos da unidade de medição inercial (ou IMU, do inglês *Inertial Measurement Unit*) instalada no veículo. Essa informação é fundamental para a navegação do veículo. A bússola é mostrada na direita do *cockpit* (Figura 3.b).

5.3. Horizonte Artificial

O horizonte artificial apresenta informações da atitude (ou postura) do veículo, mostrando a rolagem e a arfagem (ângulos dos eixos transversal e longitudinal do veículo, respectivamente, em relação ao plano horizontal) em uma única representação, como mostrado na Figura 3.c. Ele é baseado nos instrumentos usados em aviação. Os dados correspondentes a essas variáveis são obtidos por unidades de medição inercial, através de softwares de fusão sensorial como o FreeIMU [VARESANO, 2014].

5.4. Informações

Este é um componente binário, projetado para uso em indicadores de operação normal ou em exceção. Para tanto, uma instância de Informações pode definir um ícone relativo ao estado medido, que deve ter uma cor para operação normal (por *default*, verde) e outra para exceção/erro (vermelho, por *default*). Um agrupamento desses *widgets* é apresentado no canto inferior esquerdo do *cockpit* (Figura 3.d).

6. Interface Com o Usuário

O *cockpit* é a interface gráfica com o usuário do sistema de acompanhamento de missão. Nele são agrupados os *widgets*, que podem ser dispostos na tela de acordo com a necessidade do experimento, requisito desejado para o sistema (modificabilidade). Ele é na verdade uma instância da classe *MainWindow* da biblioteca Qt. Além dos componentes detalhados anteriormente, a visualização da câmera e os componentes de interação (botões, *sliders*) também compõem o *cockpit*. A Figura 3 mostra uma configuração de *cockpit* utilizando os componentes até então implementados.

7. Apresentação e Uso do Vídeo do UUV

A câmera faz parte do subsistema do UUV. A imagem por ela captada é transmitida entre os Raspberry Pi do UUV e da unidade de acompanhamento de missão por um *streaming* de vídeo através da interface de rede Ethernet que compõe o cordão umbilical do veículo. Um *widget* de visualização do vídeo está sendo finalizado, junto com os testes da implementação do sistema de streaming, cuja dificuldade reside na obtenção de baixa latência associada à melhor qualidade de imagem possível.

8. Conclusão

Neste trabalho foram apresentados aspectos do projeto e implementação de um sistema de acompanhamento de missão para veículos subaquáticos não tripulados. Além das definições dos requisitos do sistema e conceituação adotada para atendê-los, foram discutidas as implementações feitas até o momento.

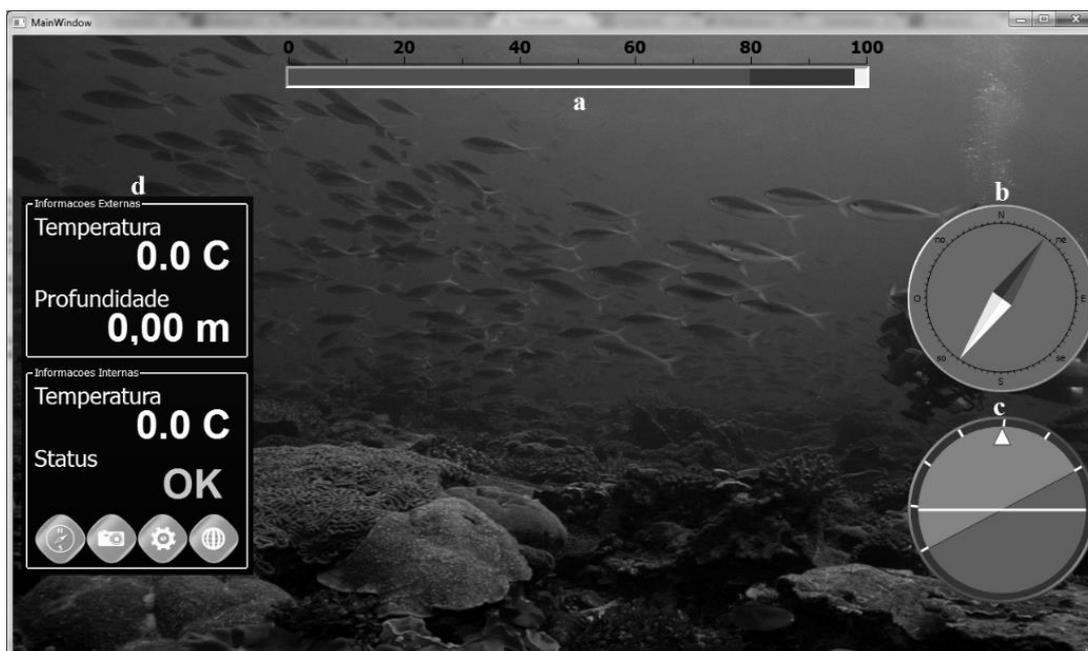


Figura 3 Cockpit do sistema de acompanhamento de missão

Este é um trabalho em andamento, com diferentes atividades sendo desenvolvidas em paralelo para a montagem do protótipo da plataforma experimental composta pelo veículo subaquático e pela unidade de acompanhamento de missão. Na presente etapa do desenvolvimento foi obtido um grande aprendizado pela equipe do projeto, como o uso de Raspberry Pi integrado a Arduinos para servirem de eletrônica embarcada de veículos e a questão do streaming de vídeo entre os RPi e as limitações envolvidas nesse processo devido ao hardware empregado, entre outros tópicos que tiveram de ser abordados. Trabalhos futuros apresentarão a evolução do projeto e os resultados da experimentação com o protótipo, bem como derivações desse trabalho.

Por fim, observa-se que a plataforma experimental será disponibilizada como hardware/software livre, com o projeto acessível para quem quiser construir ou modificar a proposta inicial, constituindo assim uma contribuição à pesquisa experimental em robótica subaquática.

Referências

- ANTONELLI, G. **Underwater Robots: Motion and Force Control of Vehicle-manipulator Systems**. Heidelberg: Springer, 2006.
- ARDUINO. **Arduino Nano**. (Online). Disponível em <<http://arduino.cc/en/Main/arduinoBoardNano>>. Acesso em 11/03/2014.
- BACK, N. et al. **Projeto Integrado de Produtos: Planejamento, Concepção e Modelagem**. São Paulo: Manole, 2008.
- BARROS, E.A. e SOARES, F.J.A. *Desenvolvimento de um robô submarino de baixo custo*. **Congresso Brasileiro de Automática**. v. 14, p 2121—2126. Natal, 2002.
- BUTTON, R. W. et al. **Survey of Missions for Unmanned Undersea Vehicles**. Arlington: RAND Corporation, 2009.

- CENTENO, M.L. **ROVFURG-II: Projeto e Construção de Um Veículo Subaquático Não Tripulado de Baixo Custo**. Dissertação de Mestrado. FURG - Pós-Graduação em Engenharia Oceânica. Rio Grande, 2007.
- DE NOVI et al. *New Approach for a Reconfigurable Autonomous Underwater Vehicle for Intervention*. **Aerospace and Electronic Systems Magazine**. v. 25, n. 11, p. 32—36. IEEE, 2010.
- HOWSE, J. **OpenCV Computer Vision with Python**. Packt Publishing, 2013.
- HSU, L. et al. *Avaliação Experimental da Modelagem e Simulação da Dinâmica de Um Veículo Submarino de Operação Remota*. **Controle e Automação**. v. 11, n. 2, p. 82—93. 2000.
- KUHN, V. N. **Controle Automático de um Veículo de Inspeção Subaquática Utilizando Sensoriamento de Baixo Custo**. Dissertação de Mestrado. FURG - Pós-Graduação em Engenharia Oceânica. Rio Grande, 2011.
- MARANI, G., CHOI, S.K. e YUH, J. *Underwater autonomous manipulation for intervention missions AUVs*. **Ocean Engineering**. v. 36, n. 1, p. 15—23. Elsevier, 2009.
- OLIPHANT, T. E. *NumPy: Python for Scientific Computing*. **Computing in Science and Engineering**, v. 9, no. 3, pp 10—20. IEEE, 2007.
- PETROBRAS. *Cada Vez Mais Fundo*. 2011. **Minisite do Pré-Sal**. Disponível em <<http://www.petrobras.com.br/minisite/presal/pt/cada-vez-mais-fundo>>. Acessado em 04/06/2011.
- PYTHON Software Foundation. **Linguagem de programação Python**. (Online) Disponível em <<http://www.python.org/>>. Acesso em 11/03/2014.
- RATHMANN, U. and WILGEN, J. **Qwt - Qt Widgets for Technical Applications**. Disponível em <<http://qwt.sourceforge.net>>. Acesso em 03/03/2014.
- RASPBERRY Pi Foundation. **Raspberry Pi**. (Online). Disponível em <<http://www.raspberrypi.org/about>>. Acesso em 15/12/2013.
- ROCHA, C.R. **Planejamento de movimento de sistemas robóticos de intervenção subaquática baseado na teoria de helicoides**. Tese de Doutorado. UFSC – Pós-graduação em Engenharia Mecânica. Florianópolis, 2012.
- SEA Perch Program. (online). 2013. Disponível em <<http://seaperch.mit.edu>>. Acessado em 11/08/2013.
- VARESANO, F. **FreeIMU: an Open Hardware Framework for Orientation and Motion Sensing**. (Online) Disponível em <<http://www.varesano.net/projects/hardware/FreeIMU>>. Acesso em 07/02/2014.
- WHITCOMB, L.L. *Underwater Robotics: Out of The Research Laboratory and Into The Field*. **Proceedings of ICRA'00**. v. 1, pp 709—716. IEEE, 2000.