

SimVR-Robotics: Uma Ferramenta de Realidade Virtual para Avaliação e Planejamento do Uso de Robôs em Cenários Offshore

SimVR-Robotics: A Virtual Reality Tool for Evaluating and Planning the Use of Robots in Offshore Scenarios

Felipe Carvalho, Philip Dunker, Thiago Motta,
Eduardo Albuquerque, Alberto Raposo

*Instituto Tecgraf, PUC-Rio
Rio de Janeiro, Brasil
{kamel,pdunker,trmotta, edualbu,
abraposo}@tecgraf.puc-rio.br*

Ismael Santos, Mauricio Galassi

*CENPES, Petrobras, Rio de Janeiro, Brasil
{ismaelh,mauricio.galassi}@petrobras.com.br*

Resumo— A indústria de Óleo & Gás tem investido no uso de robôs para a realização de diversas atividades offshore como forma de diminuir custos de produção, substituir funcionários em procedimentos que representem risco às suas vidas, viabilizar exploração em lugares remotos e, em especial, em regiões de águas profundas. O projeto e a validação de uma tarefa robótica em um ambiente virtual ajudam a estudar como inserir robôs nesses cenários de maneira eficiente e, quando comparado a testes reais, reduz custos e ajuda na mitigação dos riscos operacionais. Nesse contexto, apresentamos o SimVR-Robotics, uma ferramenta de realidade virtual para planejamento, simulação e avaliação do uso de robôs em cenários offshore. SimVR-Robotics é capaz de gerar ambientes 3D realistas e de reproduzir de maneira acurada a física de cenários com múltiplos robôs. O usuário pode montar cenários complexos a partir de uma biblioteca de modelos ou, se desejar, pode usar o editor gráfico para construir e configurar seu próprio robô. Em seguida, pode simular o cenário em tempo real. Diversos componentes foram criados para visualizar os dados da simulação, controlar os robôs e objetos da cena, e avaliar a viabilidade dos cenários projetados. Além disso, o software oferece aos usuários avançados acesso à sua API interna para controle dos diversos componentes do cenário. Como a ferramenta é baseada no ROS (Robot Operating System) a sua integração com outros simuladores externos é extremamente facilitada. Nesse artigo, discutiremos a arquitetura do SimVR-Robotics, assim como seus recursos mais importantes. Para ilustrar seus benefícios, são apresentados dois casos reais de aplicação: o estudo de trajetória de um braço robótico e o planejamento de uma operação submarina usando um ROV (*Remote Operated Vehicle* – Veículo de Operação Remota).

Palavras-chave— realidade virtual; simulação robótica; plataformas offshore.

Abstract— The Oil & Gas industry have been focusing on robots to accomplish several offshore activities as a way to diminish production costs, replace workers during hazardous operational procedures, enabling exploration in remote places and, in particular, deep-water regions. The project and validation of a robotic task on a virtual environment aids studies on how to introduce robots in these environments in an efficient way and, when compared to actual tests, significantly reduce costs and help to mitigate operational risks. In this context, we present SimVR-Robotics, a virtual reality tool to

plan, simulate and evaluate the use of robots on offshore scenarios. SimVR-Robotics is capable to recreate realistic environments and to reproduce accurately physics with multiple robots. The user can create complex scenes from a model library and simulate them in real time or, if desired, use the graphic editor to create and configure its own robot. Several components are provided to view the simulation data, control the robots and objects in the scene, and assess the feasibility of the projected scenarios. In addition, the software offers, to advanced users, access to its internal API to control the various components of the scene. As the tool is based on the ROS (Robot Operating System), it is extremely easy to integrate with other external simulators. In this article, we discuss the SimVR-Robotics architecture, as well as its most important resources. To illustrate its benefits, present two real applications: a trajectory study of a robotic arm and planning an underwater operation using an ROV (Remote Operated Vehicle).

Keywords— virtual reality; robot simulation; offshore platforms

I. INTRODUÇÃO

As recentes descobertas de reservatórios de petróleo em águas ultra profundas têm proporcionado grandes desafios para a indústria de Óleo & Gás, especialmente quando localizadas em locais remotos ou hostis. Em tais condições, a viabilidade da exploração depende da adoção de novas tecnologias voltadas para o aumento da segurança e da eficiência operacional, além da redução do contingente de pessoal embarcado (POB – people on board). A redução do POB é um aspecto importante especialmente no caso das reservas do Pré-Sal onde as Unidades Estacionárias de Produção (UEPs) estão localizadas a centenas de quilômetros da costa. Esta distância da costa eleva consideravelmente os custos de logística com o transporte de pessoal e bens perecíveis (alimentos, água e etc.).

Uma proposta de solução para este problema é aumentar o uso de dispositivos robóticos em tais cenários. No entanto, devido à dificuldade de transporte até esses locais e aos altos gastos de aluguel e reparo desses dispositivos, pequenos erros de instalação ou operações mal planejada podem resultar em perda da eficiência operacional, devido ao aumento dos custos e tempo de operação.

Para reduzir custos é necessário encontrar uma forma barata de testar, planejar e estudar a melhor maneira de se utilizar robôs nesses cenários. Nesse contexto, a simulação em ambiente virtual, por ser flexível e possuir baixo custo quando comparado a situações reais, se apresenta como uma opção vantajosa para esse problema.

Neste artigo será apresentado o simulador de robótica virtual, SimVR-Robotics (Virtual Reality Robotics Simulator), cujo objetivo é facilitar o planejamento e avaliação do uso de robôs em UEPs em diversas operações offshore.

O SimVR-Robotics possui um motor gráfico integrado, chamado de VR-Engine, baseado na engine de jogos Unity3D [1], e permite a visualização realista de modelos de cenários offshore em ambientes imersivos.

Um editor intuitivo permite que o usuário crie robôs usando uma interface drag-and-drop e configure diversos parâmetros de simulação. Esses robôs podem depois ser adicionados a uma biblioteca contendo vários modelos que foram previamente criados. Com isso, é possível criar uma cena offshore virtual para realizar os estudos necessários. Além disso, diversas ferramentas visuais permitem avaliar o desempenho de cada cena criada pelo usuário. Desta forma, uma instalação ou operação pode ser planejada de forma interativa e incremental.

O SimVR-Robotics utiliza o ROS (Robot Operating System) [2], um sistema operacional robótico open-source multiplataforma amplamente usado pela comunidade de robótica. Isso permite integrar o simulador com uma série de outras ferramentas e bibliotecas utilitárias que facilitam a construção de aplicações robóticas.

Uma dessas ferramentas é o Gazebo, um simulador projetado para reproduzir de forma acurada a dinâmica de corpos rígidos e simular os dados provenientes de sensores de robôs reais. No SimVR-Robotics, ele é responsável por calcular todas as interações físicas entre os objetos do cenário de estudo.

Até o presente várias simulações de pequenas operações já foram realizadas e diversos modelos de robôs reais como Motoman DIA10, Puma 560, SeekUR, Baxter[3] e o ROV Triton XLS 150 já foram modelados. Os resultados obtidos até o momento apontam os benefícios que o SimVR-Robotics irá trazer para o processo de robotização progressiva das UEPs da Petrobras, atendendo aos objetivos de aumento de segurança operacional e de redução dos custos.

Este artigo está dividido da seguinte maneira: primeiramente, é feita uma breve revisão sobre trabalhos relacionados, em seguida são apresentados a arquitetura do SimVR-Robotics e seus principais componentes. Na seção seguinte, são apresentados detalhes sobre o ambiente virtual e o editor de cenários. Por fim, é feito um estudo de caso de duas aplicações do SimVR-Robotics, apontando os benefícios que ele traz para o planejamento de operações offshore.

II. TRABALHOS RELACIONADOS

Diferentes frameworks de robótica e plataformas de simulação foram desenvolvidos tanto para fins de pesquisa quanto para aplicações industriais. Antes de iniciar o desenvolvimento do SimVR-Robotics em 2011, foi feito um estudo das ferramentas disponíveis no mercado lançados até então para que este projeto pudesse tanto inovar nas suas escolhas quanto fornecer as principais ferramentas e opções necessárias. Foram analisados os softwares Webots[4], Microsoft Robotics Developer Studio[5] e V-REP[6].

Webots é um software de desenvolvimento e simulação de protótipos voltados para robôs móveis. Cenários robóticos complexos podem ser modelados com diferentes robôs equipados com sensores e atuadores virtuais. Ele permite controlar os robôs por diferentes simuladores que implementam funcionalidades distintas. Sua principal desvantagem é a necessidade de uma licença paga para ter recursos completos e um certificado de renovação anual de atualização do software, motivo pelo qual não foi possível realizar uma avaliação extensa do mesmo. A capacidade de modelar um cenário personalizado é uma característica que também foi adotada no SimVR-Robotics, onde o controlador do robô, assim como no Webots, pode ser programado por simuladores externos compatíveis com ROS.

O Robotics Developer Studio (RDS), atualmente na versão RDS 4, é gratuito e destinado ao controle e simulação de robôs. Este software foi desenvolvido pela Microsoft e os conceitos mais importantes escolhidos para este projeto foram os CCR (Concurrency and Coordination Runtime), DSS (Decentralized Software Services) e VPL (Visual Programming Language). O CCR é uma biblioteca que gerencia tarefas paralelas assíncronas, permitindo a administração de múltiplos sensores em diversos robôs simultaneamente. O DSS adota um modelo de serviços orientado a estados que permite que vários serviços sejam executados em um único nó ou em nós distintos de uma rede local, criando comportamentos complexos a partir de recursos simples. O VPL é um ambiente projetado em um modelo de programação gráfica baseado em fluxo de dados, permitindo que o comportamento de um robô seja programado sem a necessidade de escrever uma linha de código sequer. Assim como no RDS, o SimVR-Robotics usa os conceitos de CCR e DSS ao fazer uso da biblioteca ROS e os componentes de seu ecossistema, enquanto o conceito de VPL foi implementado usando o conceito de máquina de estados hierárquica, fornecendo assim um mecanismo de composição visual de tarefas complexas a partir de tarefas simples e padronizadas.

V-REP é uma plataforma de experimentação de robôs virtuais que tem um ambiente de desenvolvimento integrado e é baseado em uma arquitetura de controle distribuído. Desta forma, os objetos podem ser controlados por meio de scripts, plugins, nós do ROS, uma aplicação remota ou uma solução personalizada. O V-REP possui uma versão gratuita, mas exige uma licença paga para ter todas as funcionalidades. Apesar de não ser tão modular quanto o V-REP, o SimVR-Robotics possui diversas funcionalidades semelhantes, como cinemática inversa/direta, detecção de colisão com opção de otimização, sensores de proximidade utilizados no cálculo de trajetórias e de outras operações robóticas, possibilidade de construção de um robô virtual a partir de componentes simples, visualização de dados gerados durante a simulação, entre outros.

O SimVR-Robotics tem como objetivo que usuários, especialistas em robótica ou não, possam utilizá-lo e focar seus esforços na avaliação e planejamento dos cenários criados e das operações a serem simuladas. Para isso, concentrou-se grande esforço na elaboração de interfaces gráficas que tornem a interação do usuário com o mundo virtual simples e que permitam a construção e modificação rápida de cenários. Diversas tarefas de baixo nível foram previamente automatizadas e foram criadas funcionalidades para estudo e análise qualitativa das simulações. Ao mesmo

tempo, o software foi pensando para ser flexível o bastante para que especialistas possam integrá-lo com outras ferramentas de forma a expandir as suas funcionalidades.

III. ARQUITETURA

A tendência de programação em aplicações robóticas é o uso de programas modulares para compor programas cada vez mais complexos. Isso permite integrar mais facilmente diferentes aplicações sem que seja necessário recodificar tarefas que já haviam sido programadas anteriormente. Neste contexto, uma arquitetura componentizada e flexível que possibilite uma rápida prototipagem de qualquer cenário de operação é uma ferramenta valiosa para a indústria.

Conforme mencionado anteriormente, o SimVR-Robotics faz uso do ROS, um middleware para troca de mensagens entre diferentes dispositivos e processos que cooperam para realizar uma determinada tarefa robótica. De forma análoga a um sistema operacional tradicional, o ROS fornece abstrações de hardware, controle dos dispositivos de baixo nível e uma camada de comunicação entre processos e dispositivos, provendo tópicos difundidos na rede que podem ser acessados através de mensagens ou através da execução de serviços sob demanda. No SimVR-Robotics, para aumentar a portabilidade das aplicações, foram definidas mensagens padrões para acesso a recursos comuns, como o alcance de um sensor laser, dados das câmeras ou o estado de movimentação das juntas e elos de um robô (posição, orientação e velocidade).

A Figura 1 mostra a arquitetura de componentes de alto nível do SimVR-Robotics. O componente VR-Engine é responsável pela visualização da simulação. Existem componentes para prover recursos para a interação, 3D por meio de dispositivos de manipulação, e controle de fluxo de áudio e vídeo. Os componentes ROS Bridge e Módulo de Simulação Robótica serão detalhados a seguir.

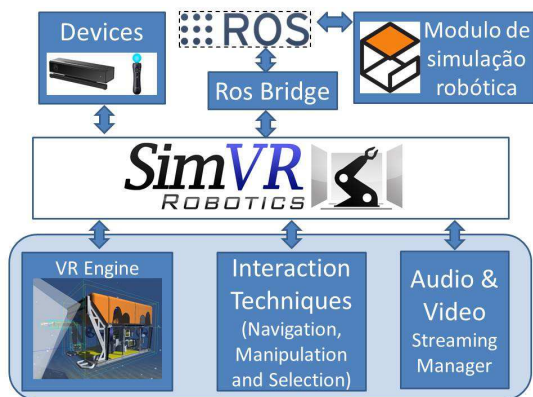


Figura 1 - Diagrama representando a arquitetura SimVR baseada em componentes

A. Ros-Bridge

O ROS-Bridge é responsável pela comunicação entre o SimVR-Robotics e o ROS. Para isso, ele implementa uma série de mensagens, tópicos e serviços customizados. Dessa forma, via ROS, é possível manter a compatibilidade entre os eventos vindos da VR-Engine com os eventos vindos de dispositivos reais, do Módulo de Simulação Robótica e de outros possíveis módulos que o usuário possa vir a escrever. Essa interação é ilustrada na figura 2.

A conexão do SimVR-Robotics com o ROS também permite que diversos algoritmos robóticos criados pelo usuário ou pela comunidade de usuários do ROS possam ser usados na simulação. É possível usar pacotes para realizar tarefas de navegação, construção de mapas de oclusão 3D, processamento de imagem, entre outros.

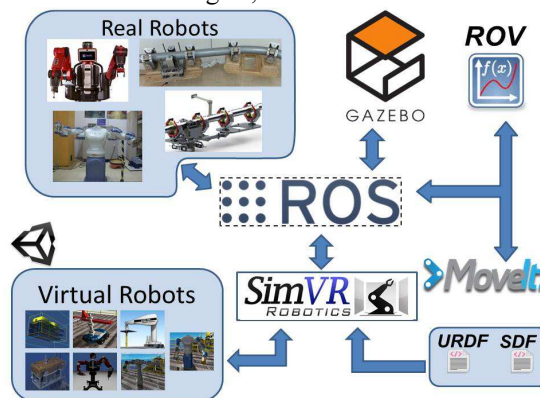


Figura 2 - Integração do ROS com SimVR-Robotics.

B. Módulo de Simulação Robótica

Para calcular o comportamento físico dos objetos de cena e dos robôs de maneira acurada e realista, o SimVR-Robotics utiliza o Gazebo[9][10], um software gratuito e de código aberto para simulação de robôs. Ele permite modelar e reproduzir com acurácia o comportamento dinâmico de um ou mais robôs, e também os dados gerados por diversos sensores como, *laser scanners*, câmeras do Kinect, câmera de vídeo, sensores de torque e odometria. A dinâmica é calculada por meio de um dos motores de física disponibilizados para o usuário: ODE (Open Dynamics Engine) [11], Bullet [12], Simbody [13] e DART [14]. Além disso, o Gazebo possui uma comunidade muito ativa e suas funcionalidades estão em constante expansão.

Cada objeto simulado é representado por uma estrutura composta de corpos rígidos ligados por juntas em que se pode aplicar forças lineares e angulares, permitindo que os mesmos interajam com o ambiente a sua volta. O usuário pode definir massa, atrito, geometrias de colisão e diversos outros parâmetros nos elementos dessa estrutura.

Os principais componentes dos corpos simulados são:

- 1) Modelos: Um modelo é um objeto na simulação que possui representação física. Ele é formado por pelo menos um elo e zero ou mais juntas.
- 2) Elos: Representam um corpo rígido e suas propriedades inerciais como massa e matriz de inércia. Além disso, eles podem possuir representações visuais e de colisão. Nessa última, são descritas geometrias de colisão, atrito, coeficiente de restituição e outras propriedades que definem como um elo interage com outros corpos rígidos.
- 3) Juntas: Conectam elos e restringem seus movimentos, formando cadeias cinemáticas e dinâmicas. Possuem atributos como limites, atrito e força elástica. Podem ser usados como motores para atuar no sistema.

O Gazebo possui integração com o ROS e está conectado ao SimVR-Robotics por meio da ROS Bridge. Um modelo de robô construído na VR-Engine é convertido em um

SDF, que é um arquivo no formato XML que provê informações a respeito do robô, seus sensores, juntas, elos, assim como sua cinemática e dinâmica associadas. O SDF, por sua vez, é lido pelo Gazebo e adicionado à simulação.

Durante a simulação, o Gazebo funciona como um servidor de física enquanto que o SimVR-Robotics atua como um de seus clientes. A cada iteração, as orientações e posições de cada modelo assim como a informação dos sensores são publicados em tópicos. Esses tópicos são lidos pelo SimVR-Robotics para atualizar a representação visual dos modelos e objetos do ambiente virtual. Esse último, por sua vez, utiliza tópicos e serviços para alterar a simulação ou interagir com os modelos simulados, podendo aplicar forças e alterar velocidade dos elos e juntas, ou modificar parâmetros da simulação, como a gravidade ou o tamanho do passo de cada iteração.

Devido a sua arquitetura de plugins, é possível ampliar as funcionalidades do Gazebo. Os plugins podem ser usados, por exemplo, para reproduzir o comportamento de novos sensores ou para acrescentar novos efeitos dinâmicos à simulação. Mais adiante, será mostrada uma aplicação do SimVR-Robotics, para planejamento de operações com ROVs, onde o Gazebo foi expandido para levar em consideração efeitos hidrodinâmicos.

IV. AMBIENTE VIRTUAL

O SimVR-Robotics foi projetado para suportar dois tipos de usuário, o operador e o desenvolvedor. O operador é o responsável por executar a simulação de tarefas específicas, como por exemplo, um *pick-and-place* usando o robô Baxter por exemplo, reparos submarinos utilizando um modelo específico de ROV ou um mapeamento 3D de um ambiente de operação usando sensores de laser do robô SeekUR. Este tipo de usuário não requer todas as funcionalidades disponíveis no SimVR-Robotics e, conseqüentemente, utiliza uma interface reduzida para facilitar sua interação.

O desenvolvedor é o encarregado de criar novas ferramentas para o operador, utilizando então todas as funcionalidades do SimVR-Robotics para criar novos robôs e cenários ou aprimorar aqueles já existentes.

O principal componente gráfico do SimVR-Robotics é o VR-Engine implementando usando a *engine* de jogos Unity3D, oferecendo uma série de facilidades para a criação de ambientes virtuais 3D, como renderização realista, programação alto nível e interface *drag-and-drop*. Mais importante ainda, o Unity3D permite que tanto suas funcionalidades quanto sua interface gráfica sejam customizadas por meio de scripts na linguagem C#.

Apesar de o Unity3D oferecer diversos recursos, eles não são suficientes para manipular e simular ambientes virtuais com robôs. Por exemplo, o motor de física original do Unity3D, não é capaz de simular robôs de maneira realista e acurada. No entanto, graças a possibilidade de extensão de seus recursos via script, foram criados uma série de funcionalidades e interfaces apropriadas para a manipulação de robôs.

Dessa forma o componente de arquitetura VR-Engine é o resultado da adição desses novos recursos ao Unity3D. Eles incluem: integração com o ROS, substituição do motor de física original do Unity3D pelos motores de física do Gazebo, construção de uma lógica hierárquica de robô baseada naquela definida pelo Gazebo, criação de interfaces

gráficas especializadas para o contexto de aplicações robóticas, e integração com diferentes dispositivos de interação 3D.

A. Descrição da Cena 3D

Toda a parte de gerenciamento dos elementos da cena é gerenciada pelo Unity3D, que pode armazenar as cenas em arquivos de descrição de cena (*scene*) ou seus elementos separadamente em arquivos do tipo *prefabs*. Uma cena guarda informações como a posição e configurações de cada objeto instanciado. Esses objetos podem ser robôs ou elementos componentes do cenário. Além disso, uma cena contém a configuração de iluminação, posicionamento de câmeras e outros parâmetros importantes.

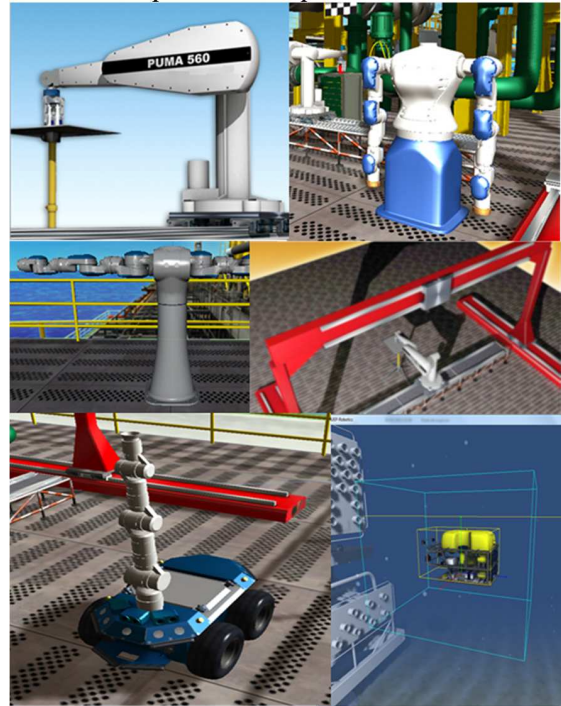


Figura 3 - Exemplos de robôs disponíveis do SimVR-Robotics.

Os robôs podem ser criados no próprio editor ou carregados a partir de arquivos URDF[15], SDF ou *prefabs*. Já existem diversos robôs virtuais previamente criados no SimVR-Robotics, como o MotoMan DIA10, Motoman SDA10, Grant Crane, SeekUR, Baxter e um ROV Triton XLS 150 (Figura 3).

B. Sensores Virtuais

Os sensores são utilizados por um robô para perceber o ambiente e capturar eventos que acontecem ao seu redor. No SimVR-Robotics, alguns sensores são simulados pelo Gazebo, enquanto que outros são reproduzidos no próprio Unity3D (sensor câmera, por exemplo). Esta integração com o Gazebo garante o aumento do número de sensores simulados a cada atualização do software.

A Figura 4 mostra a representação visual de um laser, assim como a interface customizada que representa os vetores de distância capturados.

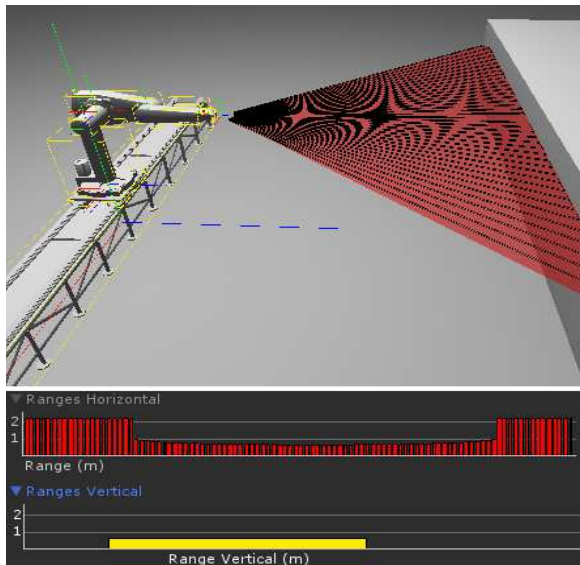


Figura 4 - Visualização dos sensores laser.

C. Ferramentas de Visualização

Para auxiliar a construção de cenários virtuais assim como a avaliação dos mesmos, o SimVR-Robotics oferece diversas ferramentas visuais.

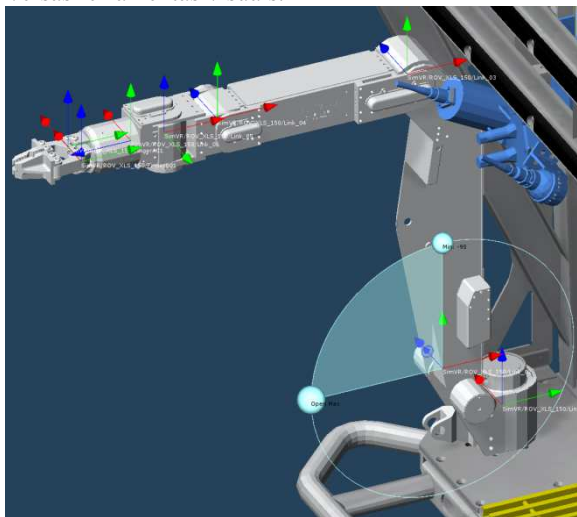


Figura 5 - Visualização dos eixos das juntas e seus limites de rotação.

É possível visualizar eixos e gizmos dos principais elementos formadores de um robô. Isso permite localizar, transladar e rotacionar os objetos em cena rapidamente e facilita, por exemplo, a identificação do eixo de rotação de uma junta ou a visualização dos seus limites de rotação (Figura 5).

O usuário tem acesso também a diversos dados de simulação de cada robô ao longo da simulação. É possível exibir gráficos de tantos elementos diferentes e simultâneos quanto desejados. Esses gráficos possuem diferentes opções de customização e podem ser configurados de acordo com as necessidades do usuário. A figura 6 exemplifica como diversos dados de duas juntas diferentes podem ser exibidos ao longo do tempo na interface.

Outras duas funcionalidades de visualização disponíveis são: *Ghostview* e *3D Point Trajectory Visualization*, que permitem a visualização da trajetória de robô de diferentes maneiras. Ao usar o *Ghostview*, o usuário observa

representações sucessivas do robô ao longo do tempo (Figura 7). Já o *3D Point Trajectory Visualization* permite a visualização da trajetória de um determinado ponto do corpo de um robô ao longo do espaço.

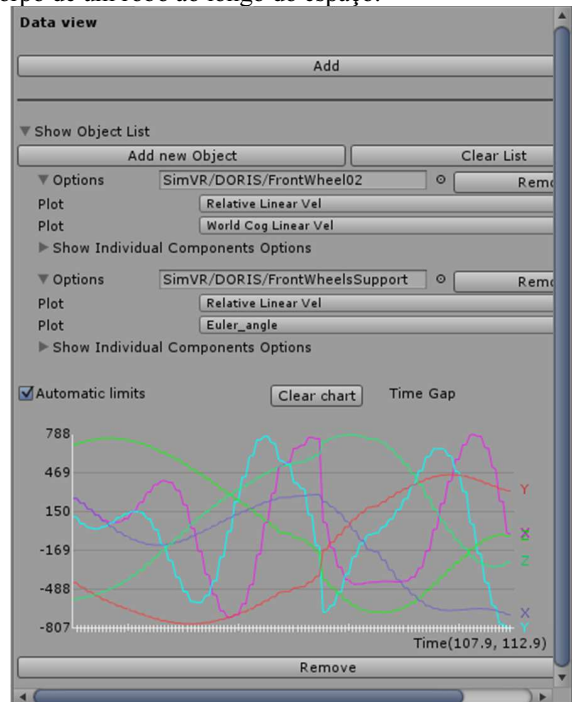


Figura 6 – Exemplo de visualização de dados por meio de gráficos customizados.

O SimVR-Robotics pode ser utilizado em ambientes do tipo desktop, bem como em ambientes imersivos, tais como CAVE, PowerWalls e outros sistemas de projeção. A visualização imersiva com recursos de estereoscopia é um recurso importante para aumentar a compreensão e o entendimento das trajetórias, permitindo a identificação e a solução de problemas de interferências e colisões.

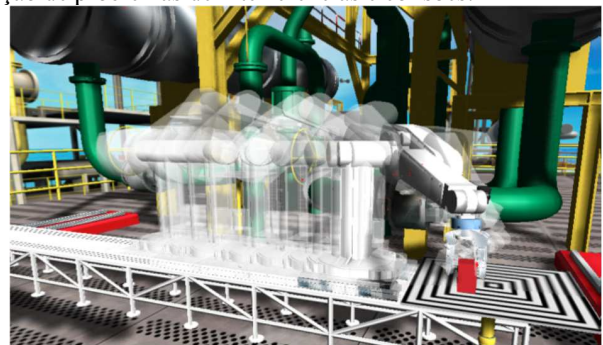


Figura 7 - Visualização da trajetória usando a técnica Ghostview

V. EDITOR DE CENÁRIOS

SimVR-Robotics possui um editor de cena para compor cenários virtuais (Figura 8). O editor gráfico fornece acesso a uma biblioteca de objetos e robôs previamente construídos que permite a composição de cenários de operação submarinos e offshore complexos. A interface *drag and drop* facilita o posicionamento dos objetos no cenário virtual 3D.

A interface padrão do Unity3D possui recursos básicos para manipulação de cenários 3D, no entanto eles não são suficientemente especializados para o contexto de aplica-

ções envolvendo robôs. Por essa razão, sua interface foi especializada, adicionando novas funcionalidades, como interfaces de configuração dos diferentes elementos de um robô, automatização de tarefas de baixo nível, nomenclatura especializada, dicas de contexto específicas a cada parâmetro de configuração, entre outros.

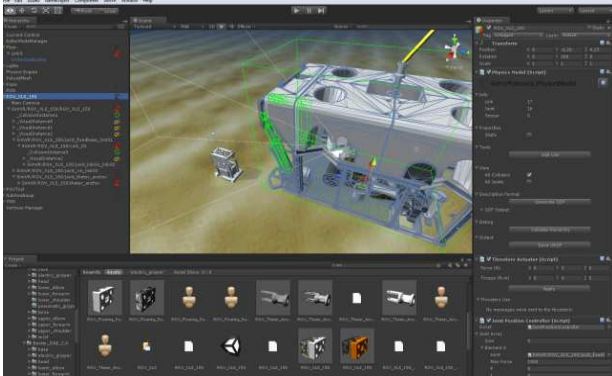


Figura 8 - Editor de cenários e robôs do SimVR-Robotics.

Foi criada também a opção de edição simultânea de diversas instâncias de um mesmo componente, reduzindo o tempo de configuração do cenário. As informações comuns foram separadas em subgrupos, que podem ser minimizados ou expandidos conforme a necessidade do usuário, deixando toda a interface mais enxuta e intuitiva.

Por meio dessas interfaces especializadas, o desenvolvedor pode verificar e alterar os principais parâmetros de um determinado elemento de um robô, como por exemplo, alterar as configurações de arrasto de um corpo rígido ou sua matriz de inércia, modificar a aparência de alguma peça ou otimizar a performance de detecção de colisão de corpos rígidos ao ainda utilizar uma caixa envolvente orientada alinhada com o objeto, *Oriented Bounding Box (OBB)* [16], como geometria de colisão desses corpos.

A estrutura de um robô no SimVR-Robotics é baseada naquela encontrada no Gazebo. Seus elementos principais são: *PhysicsModel*, a raiz de qualquer elemento que terá simulação física; *Link*, que descreve as propriedades de um corpo rígido; *Collider*, responsável pela geometria de colisão de um corpo; *Visual Instance*, a representação visual desse corpo; e *Joint*, que conecta dois elos. Eles possuem uma relação hierárquica onde um *PhysicsModel* contém um ou mais *Links*, que pode ou não conter um ou mais *Visual Instances*, *Colliders* ou *Joints*. As *Joints*, por sua vez, contêm um e somente um *Link*.

A seguir, detalhamos as funcionalidades de cada elemento no editor. Para cada uma delas foi criada uma interface gráfica customizada no Unity3D.

A. *PhysicsModel*

É o elemento raiz do elemento que será simulado fisicamente (robô ou qualquer outro objeto do cenário) e define a posição e a orientação do mesmo espaço, podendo ser estático ou não. Um objeto estático não sofrerá ação de forças ao colidir com outros objetos. A partir de sua interface pode-se visualizar a geometria de colisão associada ao objeto, visualizar o eixo de rotação de todas as suas juntas e gerar os *Links* filhos dessa raiz. A partir dela também é possível gerar arquivos SDF ou URDF da configuração do objeto para uso posterior no SimVR-Robotics ou em outras aplicações.

B. *Link*

É equivalente ao elemento elo do Gazebo. Representa as propriedades físicas de um corpo rígido que compõe o robô (a roda de um veículo, por exemplo).

A interface do *Link* permite alterar diversas características físicas, como a massa, posição do centro de gravidade, arrasto angular e linear. É possível calcular automaticamente a matriz de inércia do corpo rígido se assim desejado. Pode-se definir se esse elo colide ou não com outros elos do mesmo robô e se ele sofre efeito da gravidade. Nesta interface, também é possível adicionar novos *Colliders*, *Visual Instances*, *Joints* e *Sensores* como filhos desse *Link*.

C. *Collider*

O *Collider* define as propriedades materiais e a geometria de colisão de um *Link* de acordo com os parâmetros usados no Gazebo.

Visando fornecer flexibilidade ao desenvolvedor, diferentes tipos de geometrias foram estabelecidos. Elas podem ser definidas manualmente ou automaticamente e afetam a qualidade e performance da simulação conforme mais preciso for o cálculo de colisões. É possível escolher geometrias primitivas como esferas, cilindros, caixas ou planos, que apesar de nem sempre representarem a realidade com acurácia, produzem um aumento da performance computacional. O usuário pode escolher uma *Oriented Bounding Box*, garantindo que o programa automaticamente escolha a caixa de menor volume que envelopa a representação visual daquele corpo rígido. Também é possível escolher a própria representação visual do objeto como geometria de colisão, fornecendo assim uma maior precisão para simulação, ao custo de maior custo computacional. Por fim, o usuário pode optar por utilizar o fecho convexo associado a malha original do objeto. O fecho convexo é gerado automaticamente quando a opção é selecionada. Apesar do fecho convexo ser uma aproximação da realidade, ele oferece uma performance mais otimizada por trabalhar com uma malha menos refinada que a malha original.

Todas as opções listadas anteriormente oferecem ao usuário a possibilidade de salvar os colisores resultantes em arquivos no formato DAE (collada).

D. *Visual Instance*

Esta é a representação visual de um *Link* e não afeta de forma alguma a simulação. É apenas usada na visualização.

Sua interface permite que o usuário crie a representação visual por meio de primitivas gráficas (caixa, esfera, cilindros) ou que escolha uma malha 3D previamente criada.

E. *Joint*

É semelhante ao elemento junta do Gazebo. Ela restringe o movimento entre dois *Links*.

A interface permite ao usuário escolher entre os vários tipos de juntas implementados, como por exemplo, junta de revolução (um grau de rotação de liberdade) ou prismática (um grau de translação de liberdade). Cada junta possui parâmetros de configuração como eixo de rotação (ou translação, no caso de uma junta prismática), coeficiente de arrasto, atrito, constante de mola e limites de suas restrições (Figura 9).

A interface também permite aplicar forças e torques diretamente sobre as juntas durante a simulação.

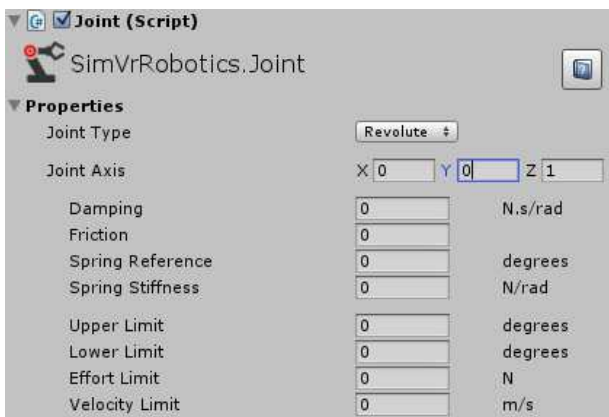


Figura 9 – Exemplo de interface gráfica customizada implementada no Unity3D.

VI. ESTUDOS DE CASO

Nesta seção apresentados dois estudos de caso desenvolvidos para avaliar e aperfeiçoar os recursos disponíveis no SimVR-Robotics. No primeiro caso apresentamos a implementação de operações de pick-and-place com um braço articulado de um tipo de ROV e o outro com o robô Baxter do laboratório de robótica do CENPES (Centro de Pesquisas da Petrobras).

No segundo estudo de caso foi criada uma nova aplicação SimROV, como uma especialização do framework SimVR-Robotics, cujo objetivo é permitir o planejamento e simulação de operações de instalações submarinas apoiadas por ROVs.

A. Pick-and-Place

Neste caso o SimVR-Robotics foi utilizado para realizar estudos de trajetória de manipuladores articulados durante a execução de uma tarefa simples de pick-and-place. Para isso, ele foi integrado ao software MoveIt [17] que contém planejadores de movimento robustos, permitindo que mesmo usuários não especialistas em robótica possam compor tarefas complexas a partir de tarefas mais simples.

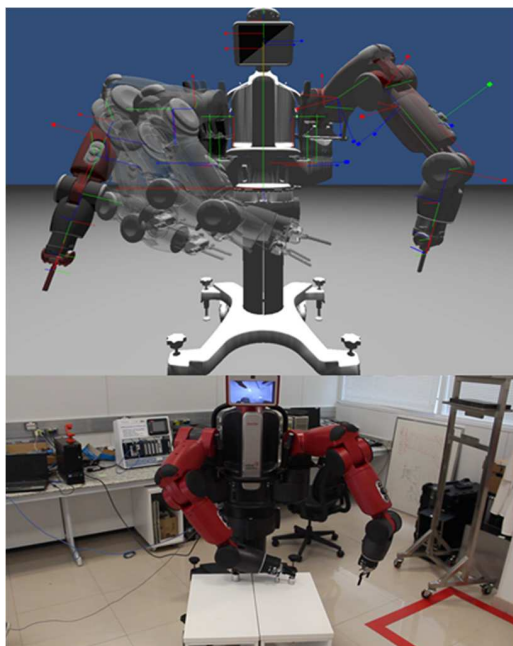


Figura 10 - Análise de trajetória de movimentos em operações de manipulação de objetos.

Dois testes foram realizados: o primeiro utilizou um braço articulado presente em um robô do tipo ROV Triton XLS 150; e o segundo usou o robô Baxter (Figura 10).

O estudo visou reproduzir num ambiente virtual operações reais de *pick-and-place*, isto é, pegar um objeto num determinado ponto e colocá-lo em outro lugar usando um manipulador robótico. Parte da tarefa exigia que o robô virtual percebesse o ambiente a sua volta e gerasse uma trajetória para seu manipulador de forma a desviar de possíveis obstáculos no caminho. Os obstáculos foram detectados por meio de sensores virtuais como *laser scanners* e câmeras, e então representados como nuvem de pontos (Figura 11).

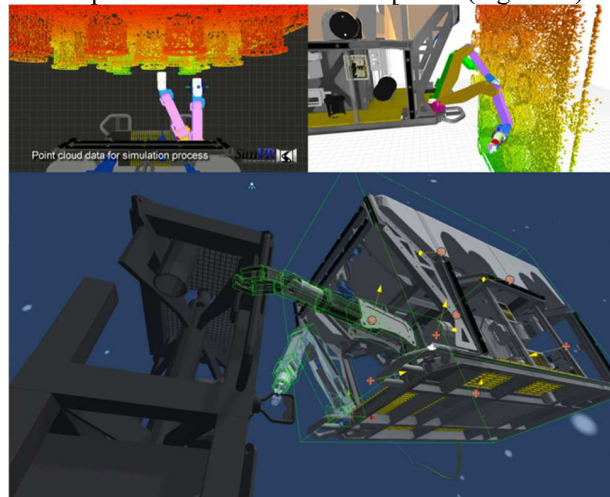


Figura 11 - Simulações de alcance de braço utilizando nuvens de pontos.

Os testes demonstraram o sucesso do SimVR-Robotics, reproduzindo os comportamentos esperados. O framework permitiu uma fácil integração com o MoveIt, demonstrando sua flexibilidade. Esse primeiro estudo confirmou a viabilidade da utilização do SimVR-Robotics para comparar o desempenho e adequabilidade de diferentes modelos de robôs e manipuladores na realização de determinadas tarefas em cenários offshore. Isto permitirá a especificação de requisitos necessários aos robôs para a realização de tarefas reais.

Alternativamente, como é possível editar o cenário virtual, esse tipo de estudo permitirá a avaliação de qual a melhor forma de se projetar uma determinada instalação de modo que seja fácil equipá-la com robôs, conforme foi feito pela indústria automobilística, que mudou o layout de suas fabricas para a robotização da linha de produção.

B. SimROV

O uso de ROVs, apesar de amplamente disseminado em operações em águas profundas, é caro e propenso a falhas. Os custos incluem aluguel, mão de obra especializada e gastos de logística dos navios de suporte a instalação de equipamentos submarinos. Uma operação mal planejada, como por exemplo, o uso de um ROV cuja a configuração não permita a realização de determinada tarefa, pode significar horas de trabalho perdido obrigando a repetição da operação com outra configuração de ROV apropriada a operação.

Portanto, o uso de um simulador especializado para auxiliar os operadores e engenheiros no planejamento da operação é extremamente valioso. Nesse segundo estudo, o SimVR-Robotics foi usado para fazer o planejamento e simulação de uma operação de instalação submarina.

O SimROV é uma especialização do framework SimVR-Robotics projetada especialmente para simulações de operações de instalações submarinas usando ROVs. Nele é possível reproduzir tarefas comuns, como inspeções de equipamentos, limpeza de conexões de equipamentos, comissionamento, instalação de equipamentos, e outros.

Graças à flexibilidade do SimVR-Robotics, foi possível estender suas funcionalidades para reproduzir corretamente as especificidades de cenários submarinos. Para adicionar efeitos hidrodinâmicos, de empuxo e de correnteza, assim como reproduzir a ação de propulsores sobre o ROV, construiu-se plugins do Gazebo baseados no pacote `freefloating_gazebo` [18] e no estudo realizado pelo artigo de Liu et al. [19] que descreve o equacionamento da dinâmica de movimento do ROV sujeito as ações de forças de correnteza e arrasto devido a massa do ROV em águas profundas.

Grande parte dos ROVs fazem uso de um cabo, chamado *tether*, ligado à uma gaiola de submersão, o TMS (Tether Management Systems). O *tether* influencia o comportamento dinâmico do ROV e pode limitar seus movimentos, caso o mesmo alcance sua extensão máxima ou caso tenha enroscado em alguma estrutura submarina. Assim, sua simulação é essencial para o planejamento de uma operação e seu comportamento também foi reproduzido no SimROV (Figura 12).



Figura 12 - Visualização de cena com TMS, Tether e ROV.

O software conseguiu reproduzir de forma satisfatória os diferentes efeitos hidrodinâmicos que atuam no ROV mostrando a viabilidade do desenvolvimento de diversas aplicações, que no momento estão sendo desenvolvidas junto com o CENPES.

SimROV permite avaliar estratégias e planejar operações submarinas. Simulações em realidade virtual podem estudar os efeitos do *tether* em determinadas operações e prever possíveis dificuldades criadas por diferentes intensidades e direções das correntezas. É possível realizar testes de interferência geométrica do corpo do ROV com as estruturas submarinas e se certificar que os manipuladores são adequados para executar uma determinada tarefa. Dessa maneira, pode-se escolher o melhor modelo de ROV para determinada operação, estudar a melhor abordagem e percurso para realizar uma operação e até mesmo planejar a construção de estruturas que facilitem e diminuam o risco de operações envolvendo ROVs.

VII. CONCLUSÃO

Nesse artigo, apresentamos a arquitetura e funcionalidades do SimVR-Robotics e como ele pode ser usado para planejar e avaliar o uso de robôs em operações offshore. Sua contribuição é a criação de uma plataforma flexível e expansível capaz de criar e reproduzir um grande número de cenários e tarefas diferentes, ao mesmo tempo que ele oferece uma interface simples e poderosa que permite que usuários não especializados em robótica possam planejar e simular tarefas e, por meio das várias ferramentas de visualização disponibilizadas, avaliar o desempenho e viabilidade de uma determinada operação submarina de acordo com a configuração do cenário de operação.

Como trabalho futuro continuaremos expandido as funcionalidades do software, visando reproduzir um número maior de sensores e aumentar a coleção de robôs já construídos, além da construção de simuladores mais robustos.

Além disso, será permitido aos usuários não especializados a montagem de cenários cada vez mais complexos. Para isso, será criada uma interface de programação visual onde o usuário poderá concatenar uma série de tarefas simples e padronizadas construindo tarefas complexas através de uma lógica de composição. Por exemplo, exemplos de tarefas simples podem ser: levar um robô de um ponto A a um ponto B, posicionar o efetuador de um manipulador num ponto C e girar o manipulador um determinado número de vezes. Quando combinadas, estas tarefas realizam a tarefa de manipulação de uma válvula.

Os formalismos teóricos para a implementação da lógica de composição ainda estão sendo avaliados. A princípio estamos avaliando o uso de Redes de Petri ou o uso de Maquinas de Estado Hierárquicas.

Graças à simulação física, o operador do software poderá averiguar se determinadas tarefas são possíveis ou não em cenários específicos, usando-se determinado robô.

AGRADECIMENTOS

Os autores gostariam de agradecer à Petrobras pelo apoio neste projeto. Os autores também gostariam de agradecer o apoio prestado pelas organizações de fomento à pesquisa do CNPq e FAPERJ. Alberto Raposo agradece também ao CNPq pelo auxílio individual.

REFERÊNCIAS

- [1] Technologies, Unity3D. s.d. <http://unity3d.com/>.
- [2] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System." ICRA Workshop on Open Source Software. 2009.
- [3] E. Guizzo, and E. Ackerman, "The rise of the robot worker." Spectrum, IEEE 49.10 (2012): 34-41.
- [4] O. Michel, "Webots: Symbiosis between virtual and real mobile robots", Proc. First Int. Conf. Virtual Worlds, VW'98, pp.254 - 263 1998
- [5] Microsoft Robot Studios - J. Jackson, "Microsoft robotics studio: A technical introduction," in IEEE Robotics and Automation Magazine, Dec. 2007, <http://msdn.microsoft.com/en-us/robotics>.
- [6] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, Virtual Robot Experimentation Platform V-REP: A Versatile 3D Robot Simulator. In: Ando, N., Balakirsky, S., Hemker, T., Reggiani, M., von Stryk, O. (eds.) SIMPAR 2010. LNCS, vol. 6472, pp. 51-62. Springer, Heidelberg (2010).
- [7] C. Connolly, "Technology and applications of ABB RobotStudio." Industrial Robot: An International Journal 36.6 (2009): 540-545.

- [8] S. Edwards, and C. Lewis, Rosindustrial: applying the robot operating system (ros) to industrial applications. IEEE International Conference on Robotics and Automation, ECHORD Workshop, St Paul, Minnesota, May 2012.
- [9] Gazebo Project, Open Source Robotics Foundation (OSRF) <http://www.gazebosim.org>
- [10] N. Koenig, and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator." Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 3. IEEE, 2004.
- [11] R. Smith, "Open dynamics engine." (2005): 3-2.
- [12] E. Coumans, "Bullet physics engine." Open Source Software: <http://bulletphysics.org> (2010)
- [13] M. A. Sherman, A. Seth, and S. L. Delp, "Simbody: multibody dynamics for biomedical research." *Procedia Iutam* 2 (2011): 241-261.
- [14] K. Liu, "Dynamic Animation and Robotics Toolkit." (2014).
- [15] W. Garage, "Universal Robot Description Format (URDF)." (2009).
- [16] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection." *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.* ACM, 1996.
- [17] S. Chitta, I. Sucas, and S. Cousins, "Moveit![ROS topics]." *Robotics & Automation Magazine, IEEE* 19.1 (2012): 18-19.
- [18] O. Kermorgant, "A dynamic simulator for underwater vehicle-manipulators." *Simulation, Modeling, and Programming for Autonomous Robots.* Springer International Publishing, 2014. 25-36.
- [19] L. Hsu, et al. "Avaliação experimental da modelagem e simulação da dinâmica de um veículo submarino de operação remota." *Revista Controle e Automação* 11.2 (2000): 82-93.