

UTILIZAÇÃO DO ROBOT OPERATING SYSTEM (ROS) EM CONJUNTO COM O SIMULADOR V-REP NO ENSINO DA ROBÓTICA

AMANDA TONEL LIMA*, ISABELLA FERREIRA DE OLIVEIRA*, GUILHERME BRITO RODRIGUES*, JACÓ DIAS DOMINGUES†, FILIPE AUGUSTO SANTOS ROCHA†, GUSTAVO MEDEIROS FREITAS†

**Departamento de Engenharia de Controle e Automação
Escola de Minas, Universidade Federal de Ouro Preto
Ouro Preto, MG, Brasil*

†*Instituto Tecnológico Vale - Mineração
Ouro Preto, MG, Brasil*

Emails: amanda.lima@pq.itv.org, isabella.oliveira@pq.itv.org,
guilherme.brito@aluno.ufop.edu.br, jaco.domingues@itv.org, filipe.rocha@itv.org,
gustavo.medeiros.freitas@itv.org

Abstract— The establishment of theoretical concepts through practical experiences enrich the teaching and learning process. However, the low budgets to support the needs of robotics subjects at Brazilian public universities and the high cost of robotic devices may limit the students' practical experience. Therefore, this paper presents an alternative to minimize this problem, proposing the use of ROS and V-REP to simulate the interaction with real robotic devices. We applied the methodology in the undergraduate courses Elements of Robotics and Mobile Robotics, offered at the Federal University of Ouro Preto, supporting the theoretical lectures. Questionnaires were proposed to the students, whose answers confirmed their acceptance of the teaching method.

Keywords— Robotic, ROS, Educational Practices, V-REP.

Resumo— O estabelecimento de conceitos teóricos por meio de experiências práticas enriquecem o processo de ensino e aprendizagem. Entretanto, o baixo orçamento para suprir as necessidades das disciplinas de robótica nas universidades públicas brasileiras e o elevado custo dos dispositivos robóticos podem limitar a experiência prática dos estudantes. Diante disso, este trabalho consiste em apresentar uma alternativa para minimizar esse problema, propondo a utilização do ROS e do V-REP a fim de simular a interação com dispositivos robóticos reais. A metodologia deste trabalho foi inserida nas disciplinas de graduação Elementos de Robótica e Robótica Móvel, oferecidas na Universidade Federal de Ouro Preto, dando suporte às aulas teóricas. Questionários foram propostos aos estudantes, cujas respostas confirmaram a aceitação do método de ensino.

Palavras-chave— Robótica, ROS, Práticas Educacionais, V-REP.

1 Introdução

Sistemas robóticos são largamente utilizados na indústria com o intuito de tornar os sistemas de produção mais robustos e flexíveis. Além disso, robôs têm sido aplicados com sucesso em diversas outras áreas nos últimos anos.

Considerando a importância da robótica nos dias atuais, a formação de profissionais capacitados a atuarem nessa área é de suma importância para o desenvolvimento econômico e social da nação. Além disto, por se tratar de uma área multidisciplinar, seu ensino se torna mais efetivo quando os conceitos teóricos são associados à prática (Cocota et al., 2013).

Equipamentos robóticos de alto desempenho geralmente apresentam custos elevados. Aliando isso à realidade financeira das universidades públicas do Brasil, grande parte das instituições não consegue adquirir equipamentos a fim de enriquecer a experiência didática. O ensino da robótica acaba se restringindo aos conceitos teóricos, tendo como resultado egressos com pouca ou nenhuma experiência prática.

Uma alternativa de ensino está na utilização de ferramentas computacionais que simulam a in-

teração com dispositivos robóticos reais. Existem atualmente várias soluções gratuitas e de boa qualidade que viabilizam esse contato.

Hoss et al. (2009) desenvolvem em seu trabalho o simulador de um robô articulado com cinco graus de liberdade. O usuário pode interagir com o dispositivo por meio de uma interface gráfica ou utilizando a linguagem SCORBASE, sendo esta a mesma empregada para a programação do robô real. Apesar do simulador proporcionar a familiarização e conhecimento do robô de forma virtual, a interação com o robô é restrita por permitir apenas um tipo de linguagem de programação.

Em Cañas et al. (2014) é exposta uma abordagem do ensino da robótica utilizando o simulador Gazebo e o *software* JdeRobot, ambos de código aberto. Por meio da simulação de um robô móvel, os resultados mostraram que as ferramentas são úteis para as práticas estudantis. Apesar do simulador Gazebo ser difundido na comunidade acadêmica, por ser o simulador padrão utilizado no ROS, o V-REP apresenta uma visualização gráfica amigável ao usuário, onde todas as propriedades dos objetos podem ser acessadas e modificadas, neste ponto o Gazebo apresenta desvantagens por demandar maior conhecimento do

usuário para criar qualquer configuração.

Kerschbaumer et al. (2014) sugerem a utilização do simulador V-REP (*Virtual Robot Experimentation Platform*) no ensino de sistemas de controle aplicado a robôs autônomos. Para isso, foi implementado um controlador *fuzzy* para um robô hexápode seguidor de linha. Apesar de terem alcançado o objetivo, o método de interação com o robô é restrito.

O ensino de técnicas de programação de robôs e a utilização de simuladores proporcionam o estudo dos conceitos teóricos aplicados à prática. Tendo isso em vista, este projeto propõe a utilização do *Robot Operating System* (ROS) em conjunto com o simulador V-REP para o ensino da robótica. Essa metodologia é aplicada nas disciplinas de graduação CAT181 - Elementos de Robótica e CAT343 - Robótica Móvel na Universidade Federal de Ouro Preto (UFOP).

2 Softwares utilizados

Esta seção apresenta o ROS e o simulador V-REP, descrevendo suas funcionalidades e como a conexão entre esses *softwares* os tornam uma ferramenta adequada para a simulação robótica.

2.1 Robot Operating System (ROS)

O ROS é um meta sistema operacional de código aberto para o desenvolvimento de aplicações robóticas. Esse ambiente consiste em uma coleção de ferramentas, bibliotecas e convenções utilizadas para criar sistemas robóticos complexos e robustos para uma grande variedade de plataformas robóticas (Foundation, 2018).

Esse *software* oferece funcionalidades de um sistema operacional, como abstração de *hardware*, controle de dispositivos a baixo nível, troca de mensagens entre processos e gerenciamento de pacotes (O’Kane, 2013). Os programas/processos são chamados de nós, podendo ser programados em C++ ou Python. Um sistema robótico é composto geralmente por vários nós, sendo que a estrutura modular do ROS permite a execução destes de forma paralela, comunicando entre si através de mensagens em estruturas conhecidas como tópicos (Quigley et al., 2009).

Quando comparado a outras plataformas, o ROS se destaca por apresentar um elevado número de componentes e suportar o maior número de robôs (Magyar et al., 2015), além de contar com uma ampla e crescente comunidade de usuários contribuindo para a sua expansão e usufruindo de uma de suas maiores vantagens, que é o compartilhamento e o reuso de códigos (Staranowicz e Mariottini, 2011).

2.2 Virtual Robot Experimentation Platform (V-REP)

Simuladores desempenham um papel importante no estudo da robótica ao viabilizar uma prototipagem rápida e barata, permitindo que a eficiência e segurança de novos algoritmos sejam testadas. O projeto e a validação de um robô ou de um algoritmo robótico é uma tarefa complexa, que exige um rigoroso controle de qualidade e uma extensa série de testes. Tendo isso em vista, simuladores podem ajudar a realizar simulações físicas trabalhosas e visualizações 3D, economizando tempo e recursos.

Plataformas de simulação como Open HRP, Gazebo e Webots oferecem uma funcionalidade eficiente, porém suportam uma pequena quantidade de meios de programação. Além disso, seus modelos de simulação e controladores são parcialmente portáteis, muitas vezes sendo necessário por exemplo uma recompilação do controlador (Rohmer et al., 2013).

Por sua vez o V-REP é um simulador comercial, desenvolvido pela Coppelia Robotics, que pode ser obtido gratuitamente na sua versão educacional.

O V-REP é um simulador robótico versátil e ideal para aplicações com multi-robôs, é focado na diversidade de linguagens de programação, na portabilidade e na facilidade de reutilização de códigos, baseado em uma arquitetura de controle distribuída, onde cada objeto ou modelo em cena pode ser comandado individualmente.

Esta plataforma oferece facilidade na criação das cenas, possuindo nativamente muitos modelos que podem ser facilmente inseridos, desde modelos de infraestrutura como móveis, paredes e diversos modelos de robôs. Inclusive permite inserir dispositivos criados pelo usuário.

A interface do V-REP (Figura 1) é composta primariamente por uma cena indicada pela letra “a”, onde os objetos são inseridos. Cada objeto pode ser composto por vários elementos ligados entre si de forma hierárquica, conforme indicado pela letra “b”, sendo possível acessá-los para modificações nas propriedades ou reorganização. A parte superior (Letra “c”) inclui a barra de ferramentas e controle da simulação.

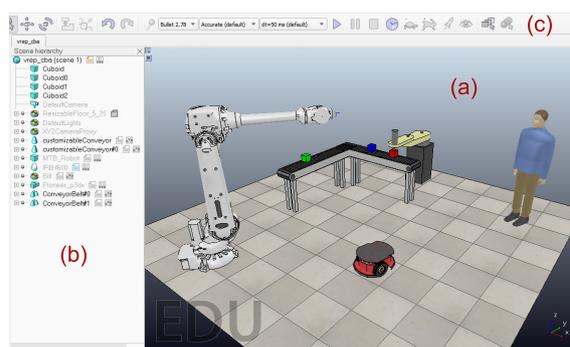


Figura 1: Interface do V-REP.

2.3 Integração ROS e V-REP

Para a execução das práticas propostas neste artigo, é necessário comunicar o ROS com o V-REP. Essa integração é realizada através da ferramenta *RosInterface* que pode ser encontrada em um repositório online¹. Essa integração permite que o usuário construa algoritmos para comandar robôs virtuais através do ROS.

3 Estudos de caso

Dois estudos de caso são feitos afim de validar a metodologia proposta. Esses são aplicados no formato de trabalho prático e, em comum, requerem que os alunos apliquem em simulação conceitos teóricos das disciplinas e desenvolvam seus conhecimentos de programação em ROS. Esta seção descreve as tarefas propostas.

3.1 Controle cinemático do manipulador robótico ABB IRB 4600

O primeiro estudo de caso é aplicado na disciplina Elementos de Robótica do curso de Engenharia de Controle e Automação da UFOP. O trabalho consiste em comandar um manipulador robótico através do ROS, com o objetivo de empurrar e derubar as cinco caixas presentes no cenário simulado², evitando contato com os demais elementos (Figura 2). O dispositivo utilizado é o IRB 4600 da ABB, um robô antropomórfico Industrial.

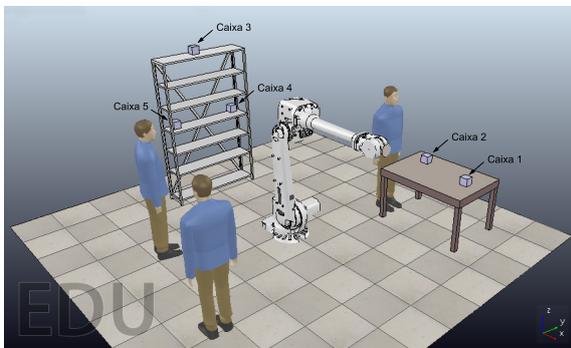


Figura 2: Cena do ABB IRB 4600 no V-REP.

A implementação do comando cinemático do braço robótico é proposta. A cinemática descreve a relação analítica entre o estado das variáveis das juntas com a pose (posição e orientação) de um corpo no espaço, se referindo ao movimento de um mecanismo robótico sem considerar as forças/torques que o causaram (Khatib e Siciliano, 2008).

A cinemática pode ser estudada de forma direta ou inversa. A forma direta consiste em ex-

¹A ferramenta *RosInterface* por ser obtida em: <https://goo.gl/Vrcncg>.

²Os arquivos necessários ao Trabalho Prático de Elementos de Robótica podem ser encontrados em: <https://goo.gl/BWHZgs>

pressar a pose do efetuador com respeito a um sistema de referência arbitrário em função das variáveis das juntas. Por sua vez, a cinemática inversa é a determinação das variáveis de junta a fim de levar o efetuador a uma determinada pose (Siciliano et al., 2010).

A resolução da cinemática inversa de posição e orientação de manipuladores antropomórficos envolve cálculos não lineares. Logo, abordagens simplificadas para a resolução desse problema são desejadas a fim de se tornarem mais adequadas e receptivas a iniciantes da área.

Um primeiro método de simplificação é por meio do desacoplamento cinemático, permitindo calcular separadamente as cinemáticas inversa de posição e orientação. Essa técnica pode ser aplicada em manipuladores robóticos que possuem arquitetura de punho esférico, em que os sistemas de coordenadas das três últimas juntas do robô podem ser fixadas no mesmo ponto (Kvernberg, 2015). Tendo em vista também a tarefa proposta, não é necessário incutir orientações ao punho do manipulador, apenas posições.

Para a resolução do problema, é sugerido aos alunos a utilização de uma abordagem geométrica, pois possibilita um maior entendimento da arquitetura do braço robótico usado. Esse método faz uso de cálculos trigonométricos e decompõe a geometria espacial do manipulador em vários problemas planares.

A construção de um procedimento para a resolução da cinemática é derivada da estrutura do manipulador. Cada junta liga dois elos consecutivos, e um sistema de coordenadas O_i , composto por três vetores unitários ortogonais $\hat{x}_i, \hat{y}_i, \hat{z}_i \in \mathbb{R}^3$, é fixado convenientemente em cada elo i do manipulador, sendo o sistema referencial O_0 fixado à base (Figura 3). Assim, cada junta i liga o elo i ao elo $i+1$ (ou o elo $i-1$ ao elo i , de acordo com a convenção adotada) (Siciliano et al., 2010).

Na Figura 3 é demonstrada a representação do manipulador robótico em questão após a fixação dos sistemas de coordenadas.

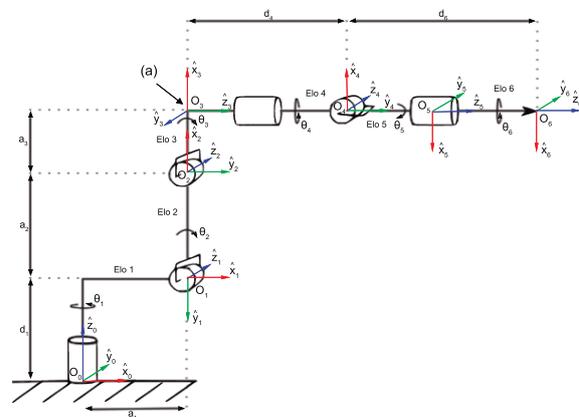


Figura 3: Modelo do braço robótico IRB 4600. Fonte: Adaptado de Kvernberg (2015).

Os valores de a_i e $d_i \in \mathbb{R}$ são obtidos no manual fornecido pela ABB Robotics (2016) e podem ser vistos na Tabela 1.

Tabela 1: Dimensões do IRB 4600.

Parâmetro	Valor [m]
a_1	0.175
a_2	0.900
a_3	0.175
d_1	0.495
d_4	0.960
d_6	0.135

Após a representação do robô e a fixação dos sistemas de coordenadas, é possível calcular os ângulos das juntas em razão de uma pose final. O cálculo das variáveis de junta utilizando a função atan2 determina o quadrante correto dentre as quatro possíveis soluções para o problema da cinemática inversa de posição do manipulador robótico em questão.

A equação da primeira variável θ_1 (Equação 1.1) é definida a partir da projeção do manipulador no plano $\hat{x}_0\hat{y}_0$ e do uso de relações trigonométricas, como mostrado na Figura 4, sendo x_p e y_p referentes à posição da origem do centro de coordenadas do punho do robô.

$$\theta_1 = \text{atan2}(y_p, x_p) \quad (1.1)$$

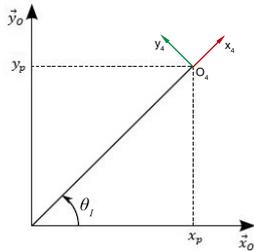


Figura 4: Projeção do manipulador no plano $\hat{x}_0\hat{y}_0$.

Fonte: Adaptado de Kvernberg (2015).

O IRB 4600 possui um deslocamento no cotovelo, indicado pela letra “a” na Figura 3, que deve ser considerado nos cálculos de θ_2 e θ_3 , como observado na Figura 5, onde $\vec{\lambda} \subset x_0y_0$ e $|\vec{\lambda}| = 1$.

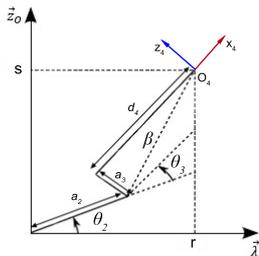


Figura 5: Projeção do manipulador no plano λz_0
Fonte: Adaptado de Kvernberg (2015).

Os valores das juntas θ_2 e θ_3 podem ser encontrados pela lei dos cossenos, calculados nas Equações 1.2 e 1.7.

$$\theta_3 = \text{atan2}(\sqrt{1 - D^2}, D) - \frac{\pi}{2} + \text{atan2}(a_3, d_4) \quad (1.2)$$

onde

$$D = \frac{r^2 + s^2 - a_2^2 - \beta^2}{2a_2\beta} \quad (1.3)$$

$$s = z_p - d_1 \quad (1.4)$$

$$r = \sqrt{((x_p - a_1) \cos \theta_1)^2 + ((y_p - a_1) \sin \theta_1)^2} \quad (1.5)$$

$$\beta = \sqrt{a_3^2 + d_4^2} \quad (1.6)$$

sendo s (Equação 1.4) e r (Equação 1.5) as distâncias do sistema de referência do segundo elo (O_1) ao centro do punho no plano λz_0 , e β (Equação 1.6) a distância do sistema de referência do terceiro elo (O_2) ao centro do punho no plano λz_0 . Dado θ_3 , é possível encontrar o θ_2 pela seguinte equação:

$$\begin{aligned} \theta_2 = & -\text{atan2}(s, r) + \text{atan2}(\beta \sin(-\theta_3 - \frac{\pi}{2} \\ & + \text{atan2}(a_3, d_4)), a_2 + \beta \cos(-\theta_3 - \frac{\pi}{2} \\ & + \text{atan2}(a_3, d_4))) + \frac{\pi}{2} \end{aligned} \quad (1.7)$$

As outras três variáveis de junta, θ_4 , θ_5 e θ_6 , correspondem à orientação do efetuador, cuja resolução não é demonstrada neste artigo, visto que apenas a posição é exigida na tarefa proposta. A resolução completa da cinemática inversa de orientação para um manipulador antropomórfico pode ser encontrada em Kvernberg (2015).

Após a definição das equações das juntas do robô em simulação, essas são incluídas no código de comando do ROS. O nó recebe os valores atuais de posição angular das juntas e aplica nelas um *setpoint* de posição angular, calculado a partir das coordenadas de posição das caixas com respeito à base do robô, obtidas no V-REP. Assim, é possível programar os movimentos do IRB 4600 da definição de pontos sequenciais, que são alcançados à medida em que os pontos anteriores são atingidos. A estratégia adotada para realizar a tarefa de derrubar as caixas fica a critério dos alunos.

O algoritmo a seguir (Algoritmo 1) exemplifica uma forma de implementação do nó para derrubar as caixas. Na inicialização do ROS é feita a importação dos pacotes e mensagens necessárias. As posições angulares calculadas são publicadas em um tópico do ROS, que é subscrito pelo simulador V-REP. O simulador por sua vez publica as posições atuais das juntas em outro tópico do ROS, que é subscrito pelo nó de comando. A função *juntaPosAtual* é chamada cada vez que uma nova mensagem chega nesse tópico.

Algoritmo 1: Algoritmo para derrubar as caixas.

```
1: procedure derruba_caixa()
2:   Init ROS
3:   Function coordCaixa(k)
4:   Function cinematicaInv(x,y,z)
5:   Function juntaPosAtual()
6:   for  $k = 1$  to  $K$  do
7:      $x, y, z \leftarrow \text{coordCaixa}(k)$ 
8:      $\theta_1, \theta_2, \theta_3 \leftarrow \text{cinematicaInv}(x, y, z)$ 
9:     publicaROS( $\theta_1, \theta_2, \theta_3$ )
10:    while juntaPosAtual()  $\neq \theta_1, \theta_2, \theta_3$  do
11:      Aguarda
12:    end while
13:  end for
14: end procedure
```

O controle cinemático no espaço das juntas é feito por meio de um controlador PID presente em cada junta do manipulador robótico, cujos parâmetros já são pré-definidos pelo simulador. Assim, a trajetória de cada junta é controlada independentemente, conforme apresentado na malha de controle a seguir (Figura 6).

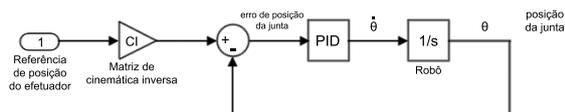


Figura 6: Malha de controle das juntas do manipulador robótico IRB 4600.

Com a realização deste estudo de caso utilizando o ambiente de simulação criado, os alunos são capazes de compreender melhor os conceitos teóricos de manipuladores robóticos apresentados em sala, além de adquirir noções de programação de robôs e se familiarizarem com o ROS e o V-REP.

3.2 Competição entre alunos para o ensino da robótica móvel

O segundo estudo de caso que foi implementado na disciplina de graduação CAT343 - Robótica Móvel do curso de Engenharia de Controle e Automação da UFOP, tem como proposta uma competição entre alunos que devem programar robôs móveis virtuais para realizar uma determinada tarefa.

A partir de um cenário previamente criado no V-REP³ (Figura 7), os alunos devem programar o robô para detectar caixas e posteriormente derrubá-las de uma plataforma suspensa, sem que o robô saia da arena. As caixas possuem pontuações distintas de acordo com a sua posição. A

³Os arquivos necessários ao Trabalho Prático de Robótica Móvel podem ser encontrados em: <https://goo.gl/CqvTKF>.

equipe que marcar mais pontos ou derrubar o oponente vence a partida.

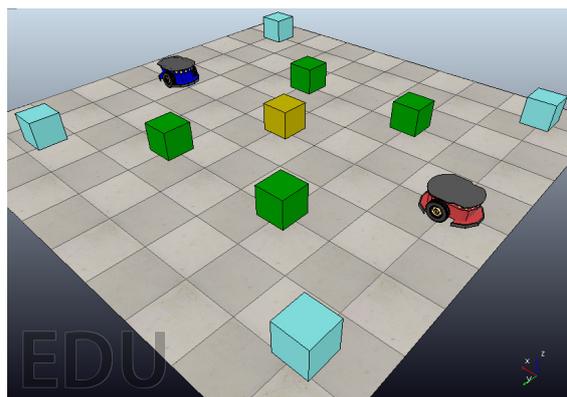


Figura 7: Cena no V-REP para aprendizagem da robótica móvel.

O robô móvel em questão é o Pioneer 3-DX, largamente utilizado na comunidade acadêmica. Esse possui movimentação diferencial de duas rodas e 16 sensores ultrassônicos (Figura 8), sendo oito voltados para frente e oito voltados para trás (MobileRobots, 2012). Além disso, o robô permite que o seu comando seja realizado através do ROS, tornando-o ideal para a metodologia em estudo.

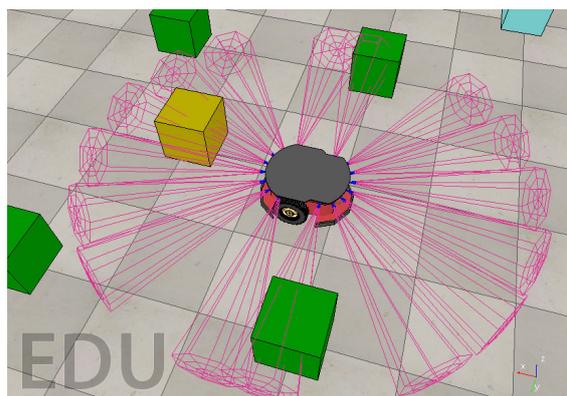


Figura 8: Visualização dos sensores ultrassônicos do Pioneer 3-DX.

A fim de implementar o algoritmo de operação autônoma, os alunos precisam calcular a transformada homogênea de maneira que as caixas detectadas sejam referenciadas com respeito ao sistema de coordenadas do robô a partir dos dados disponibilizados pelo V-REP. Essa transformada representa a posição do sensor para o sistema de coordenadas do robô. As informações dos sensores são apresentadas por meio de um *array* com 16 posições, onde cada posição equivale ao valor do sensor, correspondente à menor distância com respeito a um obstáculo detectado.

A transformada homogênea do sensor é computada utilizando sua matriz de rotação e seu respectivo vetor de posição com respeito ao sistema de coordenadas do robô. Como pode ser visto na

Figura 9, tem-se as coordenadas do ponto mais próximo da caixa detectada ($x_c; y_c$), do sensor ultrassônico i ($x_i; z_i$), do robô ($x_0; z_0$) e a distância (d) percebida pelo sensor ultrassônico com respeito à caixa.

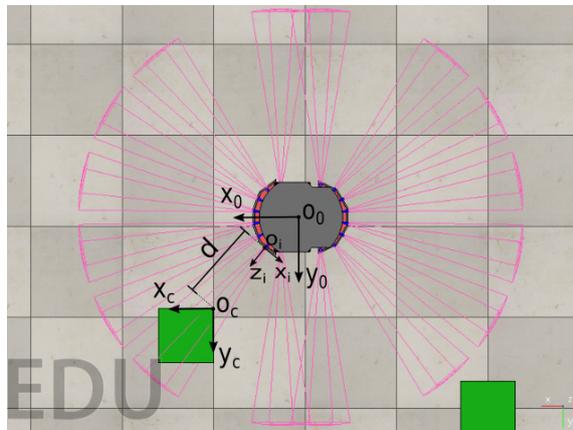


Figura 9: Sistemas de coordenadas do Pioneer 3-DX, da caixa e do sensor ultrassônico.

Deve-se analisar o *array* para verificar se algum sensor detectou um objeto, para então calcular a posição da caixa com respeito ao sistema do robô utilizando a respectiva matriz de transformada homogênea, extrair essa distância e por fim descobrir as coordenadas do ponto mais próximo da caixa com respeito ao centro do robô (Equação 2):

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & \vec{p}_3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ d \\ 1 \end{bmatrix}, \quad (2)$$

onde x_c , y_c e z_c são as coordenadas do ponto mais próximo da caixa detectada com respeito ao centro do robô, $R \in SO(3)$ (Grupo Especial Ortonormal de dimensão 3) a matriz de rotação do sensor i , $\vec{p}_{3 \times 1}$ seu vetor de posição e d , a distância detectada pelo sensor.

O Algoritmo 2 é a implementação de um nó para calcular as coordenadas da caixa com respeito ao centro do robô. Primeiramente é realizada a inicialização do ROS para a importação dos pacotes e mensagens necessárias para a execução do nó. Enquanto o nó do ROS estiver sendo executado, o algoritmo será processado. No *array dist* que corresponde as informações dos sensores ultrassônicos são coletadas a menor distância absoluta e seu respectivo índice no vetor. Esses dados são armazenados nas variáveis d e i nessa ordem. Então, o valor armazenado em d é acondicionado no vetor coluna *vet_sensor*. Caso nenhum obstáculo seja detectado pelos sensores, os dados

no vetor d são iguais a 1, portanto não são calculadas as coordenadas do ponto da caixa. Caso contrário, são calculadas as coordenadas do ponto mais próximo da caixa com respeito ao robô de acordo com a Equação 2. Por fim, estas coordenadas são publicadas em um tópico do ROS que é então subscrito pelo V-REP.

Algoritmo 2: Algoritmo para calcular as coordenadas da caixa.

```

1: procedure coord_caixa()
2:   Init ROS
3:   while ROS esta sendo executado do
4:      $d \leftarrow \text{minimo abs dist}[0...15]$ 
5:      $i \leftarrow \text{arg do minimo abs } d[0...15]$ 
6:      $\text{vet\_sensor} \leftarrow [0, 0, \text{dist}, 1]$ 
7:     if  $\text{dist} = 1$  then
8:        $x_c, y_c \leftarrow 0$ 
9:     else
10:       $\text{vet\_caixa} \leftarrow (\text{Th}[i]) x (\text{vet\_sensor})$ 
11:       $x_c \leftarrow \text{vet\_caixa}[0]$ 
12:       $y_c \leftarrow \text{vet\_caixa}[1]$ 
13:    end if
14:    PublicaRos( $x_c, y_c$ )
15:  end while
16: end procedure

```

Para que o robô consiga perceber se está prestes a sair da arena ou não, são inseridos quatro sensores de distância na parte inferior do dispositivo apontados para baixo. Quando o sensor não detecta a arena, o seu valor retornado é igual a 0.

Conhecendo a posição da caixa com respeito ao centro do robô, cada equipe deve desenvolver um código de navegação autônoma, adotando a estratégia que considerar mais eficiente.

A competição é realizada utilizando três computadores de forma a demonstrar a conectividade simplificada de diversos computadores operando o ROS (Figura 10). Um computador central executa o ROS master interfaceado com o simulador V-REP a cena da competição. Nos outros dois computadores, cada grupo executa seus códigos responsáveis pela movimentação dos robôs.

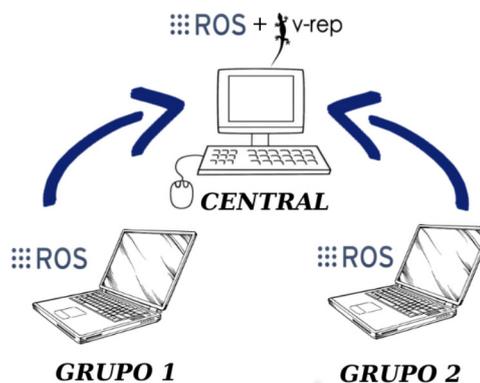


Figura 10: Configuração da competição em múltiplas máquinas.

Este Trabalho permite aos alunos aprender a integração de informações dos sensores para a programação eficiente de robôs móveis, colocar em prática os conceitos teóricos da robótica, além de aprender ferramentas computacionais que são usadas por profissionais. Na Figura 11 pode ser vista a competição sendo realizada.



Figura 11: Competição realizada na disciplina CAT343-Robótica Móvel.

4 Avaliação da metodologia

A metodologia descrita neste artigo, aplicada nas disciplinas CAT181-Elementos de Robótica e CAT343-Robótica Móvel, do curso de graduação em Engenharia de Controle e Automação da UFOP, possui o objetivo de apoiar as aulas tradicionais de robótica, oferecendo aos alunos a possibilidade de lidar na prática com os conteúdos ensinados em sala de aula.

Ao término das disciplinas, os estudantes foram submetidos a um questionário de avaliação do ambiente de simulação V-REP/ROS para ensino de robótica com a finalidade de mensurar a aceitação e aprendizado. Foram avaliadas as contribuições dessas ferramentas para uma maior absorção do conteúdo teórico e relevância de algumas características específicas como modularidade do ROS.

O questionário utiliza perguntas com escalas específicas propostas por Rensis Likert (1932), com o objetivo de obter a opinião mais detalhada dos entrevistados, podendo indicar até que ponto eles concordam ou discordam de uma afirmação. As opções de resposta com escalas específicas apresentam menos erros de medição se comparado com questões de concordo e discordo, muito popular em pesquisas baseadas em questionários (Sarís et al., 2010).

Os participantes da pesquisa foram divididos de acordo com a disciplina que cursaram durante o semestre letivo, onde 48 alunos cursaram Elementos de Robótica (ER), 19 cursaram Robótica Móvel (RM) e 7 cursaram ambas as disciplinas (AD) simultaneamente.

4.1 Resultados

A partir das respostas dos estudantes ao questionário proposto, foram obtidos os resultados apre-

sentados nas Tabelas 2, 3 e 4.

Tabela 2: Respostas do questionário proposto para os alunos que cursaram as disciplinas ER, RM ou AD.

O Ambiente de Simulação V-REP/ROS contém recursos que despertam a sua atenção.			
	ER	RM	AD
Concordo Totalmente	57,1%	57,0%	71,4%
Concordo	28,6%	29,0%	28,6%
Indiferente	14,3%	14,0%	0,00%
Discordo	0,00%	0,00%	0,00%
Discordo Totalmente	0,00%	0,00%	0,00%

Tabela 3: Respostas do questionário proposto para os alunos que cursaram as disciplinas ER, RM ou AD.

O Ambiente de Simulação V-REP/ROS facilita a realização das atividades práticas e a absorção do conteúdo teórico.			
	ER	RM	AD
Concordo Totalmente	14,3%	86,0%	85,7%
Concordo	71,4%	14,0%	14,3%
Indiferente	14,3%	0,00%	0,00%
Discordo	0,00%	0,00%	0,00%
Discordo Totalmente	0,00%	0,00%	0,00%

Tabela 4: Respostas do questionário proposto para os alunos que cursaram as disciplinas ER, RM ou AD.

A modularidade do ROS facilita o trabalho em equipe e o desenvolvimento das aplicações.			
	ER	RM	AD
Concordo Totalmente	14,6%	57,1%	71,4%
Concordo	71,4%	42,9%	28,6%
Indiferente	14,0%	0,00%	0,00%
Discordo	0,00%	0,00%	0,00%
Discordo Totalmente	0,00%	0,00%	0,00%

Na disciplina de Robótica Móvel, o ambiente V-REP/ROS foi utilizado em tempo integral durante as aulas expositivas, onde previa -se que os estudantes já possuíssem o conhecimento teórico adquirido durante a disciplina Elementos de Robótica, na qual por apresentar um extenso conteúdo teórico, o ambiente V-REP/ROS foi utilizado somente em algumas atividades práticas. Diante disso as afirmações propostas apresentaram menor nível de aceitação pelos estudantes que cursaram exclusivamente esta disciplina.

Apesar da recomendação que os alunos cursassem ER antes de RM, nota-se um maior nível de aceitação diante das afirmações propostas pelos estudantes que cursaram ambas as disciplinas, que simultaneamente obtiveram o primeiro contato com os conteúdos de robótica e aplicação no

ambiente V-REP/ROS, seguidos pelos alunos que cursaram apenas Robótica Móvel.

Com o objetivo de permitir que os estudantes expressassem sua opinião e dessem sugestões, foi incluído no questionário um campo com resposta discursiva de preenchimento não obrigatório. De modo geral, os alunos avaliaram positivamente o uso de simuladores no ensino de robótica, apresentando como principal desafio as dificuldades em programação por uma parcela dos alunos.

5 Conclusão

A metodologia descrita neste trabalho possui a finalidade de tornar mais efetivo o processo de ensino e aprendizagem de robótica nas universidades, apresentando ferramentas computacionais que permitem solidificar o conhecimento dos estudantes.

Diante dos resultados obtidos, fica evidente que o uso de simuladores na aplicação dos conceitos de robótica desperta o interesse dos alunos, permitindo consolidar os conceitos teóricos lecionados em sala de aula. Com isso, é possível observar que a aplicação desta metodologia pode trazer benefícios significativos ao processo de aprendizagem dos estudantes, buscando uma alternativa para as dificuldades das universidades públicas brasileiras na aquisição de dispositivos robóticos.

Para trabalhos futuros, é proposta uma extensão das atividades práticas, abrangendo demais técnicas de análise cinemática, incluindo a implementação da representação por Denavit-Hartenberg. Ainda, a realização de um estudo de viabilidade da criação de disciplinas de introdução ao *Robot Operating System* (ROS), que aborde as linguagens de programação utilizadas, a fim de facilitar a absorção do conteúdo pelos estudantes.

Referências

- ABB Robotics (2016). IRB 4600 Industrial Robot Data Sheet.
- Cañas, J. M., Martn, L. e Vega, J. (2014). Innovating in robotics education with gazebo simulator and jderobot framework, *XII Congreso Universitario de Innovacion Educativa en las Enseñanzas Tecnicas*.
- Cocota, J. A. N., Barbosa, R. C., Moreira, A. D. e Lage, V. (2013). Desenvolvimento de um robô antropomórfico com punho esférico para práticas de robótica com alunos de graduação, *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics* **1**(1).
- Foundation, O. S. R. (2018). Ros - powering the world's robots.
URL: <http://www.ros.org/>
- Hoss, A., Hounsell, M. e Leal, A. B. (2009). Virbot4u: Um simulador de robô usando x3d, *I Simpósio de Computação Aplicada* pp. 1–15.
- Kerschbaumer, R., Lima, C. R. E. e Simão, J. M. (2014). O simulador de robôs v-rep no ensino das técnicas de controle de robôs autônomos.
- Khatib, O. e Siciliano, B. (2008). *Springer Handbook of Robotics*, Springer.
- Kvernberg, P. (2015). *Developing force control scenarios on abb irb 4600 with camera capture of dynamic motions*, Master's thesis, Norwegian University of Science and Technology.
- Magyar, G., Sinčák, P. e Krizsán, Z. (2015). Comparison study of robotic middleware for robotic applications, *Emergent Trends in Robotics and Intelligent Systems*, Springer, pp. 121–128.
- MobileRobots, A. (2012). Pioneer p3-dx, Website. <http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx>.
- O'Kane, J. M. (2013). *A gentle introduction to ROS*.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. e Ng, A. Y. (2009). Ros: an open-source robot operating system, *ICRA workshop on open source software*, Vol. 3, Kobe, Japan, p. 5.
- Rohmer, E., Singh, S. P. N. e Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework, *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326.
- Saris, W., Revilla, M., Krosnick, J. e Shaefter, E. (2010). Comparing questions with agree/disagree response options to questions with item-specific response options, *Survey Research Methods (2010) Vol.4, No.1* pp. 71–79.
- Siciliano, B., Sciavicco, L., Villani, L. e Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*, Advanced Textbooks in Control and Signal Processing, Springer London.
- Staranowicz, A. e Mariottini, G. L. (2011). A survey and comparison of commercial and open-source robotic simulator software, *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*, PETRA '11, ACM, New York, NY, USA, pp. 56:1–56:8.