DOI: 10.48011/asba.v2i1.1697

Proposição de arquitetura para ensino de automação industrial: Integrando ambiente virtual 3D, ferramenta de modelagem matemática e ambiente real

Pedro Henrique Pinto Cavalcante *
Lucas Gabriel Cosmo Morais ** **Raphael van der Linden** ***
Ademar Virgolino da Silva Netto ****

Centro de Energias Alternativas e Renováveis - CEAR, Universidade Federal da Paraíba - UFPB, João Pessoa - PB - Brasil

* *e-mail: pedro.cavalcante@cear.ufpb.br*
** *e-mail: lucas.morais@cear.ufpb.br*
*** *e-mail: raphael.linden@cear.ufpb.br*
**** *e-mail: ademar@cear.ufpb.br*

Abstract: Together with the new needs and teaching technologies, this study proposes a modular architecture for integration between real and virtual environments. The objective was to develop the platform for application on subjects in the industrial automation area, relying on free and open source tools as a way to expand access for future users. The architecture was promoted through CoppeliaSim and Modelica language's three-dimensional and mathematical modeling characteristics, combined with communication through the Robot Operating System - ROS. To validate the structure, the case study carried out dealt with the elaboration of a factory cell, in which a robotic manipulator NS-16-1.65 performed pick and place tasks with blocks on a conveyor belt. In this context, it is expected that the use of the presented architecture will be useful and will benefit the new teaching and learning relationships, fostering innovation and increasing the opportunities for access to this type of technology.

Resumo: Em conjunto às novas necessidades e tecnologias de ensino, este estudo traz como proposta uma arquitetura modular para integração entre ambiente real e virtual. Teve-se como objetivo desenvolver a plataforma para a aplicação em disciplinas da área de automação industrial, apoiando-se em ferramentas gratuitas e *open source* como forma de ampliar o acesso para futuros usuários. A arquitetura foi promovida por meio das características de modelagem tridimensional e matemática do CoppeliaSim e da linguagem Modelica, aliados à comunicação através do Sistema Operacional de Robôs - ROS. Para validação da estrutura, o estudo de caso realizado tratou da elaboração de uma célula fabril, na qual um manipulador robótico NS-16-1.65 executou tarefas de *pick and place* de blocos sobre uma esteira. Nesse contexto, espera-se que o uso da arquitetura apresentada seja útil e beneficie as novas relações de ensino e aprendizagem, fomentando a inovação e aumentando as oportunidades de acesso a este tipo de tecnologia.

Keywords: Teaching; Industrial Automation; Integration Architecture; Mixed Reality; Modularity.

Palavras-chaves: Ensino; Automação Industrial; Arquitetura de Integração; Realidade Mista; Modularidade.

1. INTRODUÇÃO

Tem se tornado cada vez mais comum o surgimento de iniciativas de ensino baseadas em técnicas de realidade virtual, como uma forma de fomentar as relações de ensino e aprendizagem e também de se adequar às recentes necessidades da área da educação. A tendência de alteração dessas relações para métodos mais interativos e dinâmicos teve como maior contribuição o avanço de tecnologias baseadas em plataformas digitais e dos diferentes recursos e possibilidades que estas geraram. A importância de se ter novas práticas para a formação de futuros profissionais é fundamental quando se pretende estar atualizado com os novos desafios e habilidades que o mercado de trabalho necessita. Além disso, a ampliação dos recursos de ensino, para além da sala de aula, possibilita driblar possíveis momentos de crise, dando continuidade a realização dos mecanismos de educação.

Nesse sentido, algumas práticas e experiências de sucesso no ensino com tecnologias virtuais e mistas podem ser citadas como forma de exemplificar as contribuições desse tipo de abordagem, sendo uma delas o caso de Milano et al. (2008) que, com a implementação de um laboratório

virtual de sistemas de potência, a partir de um pacote de *software open-source* e gratuito, possibilitou a melhoria nos níveis de aprovação, aprendizagem e comprometimento de alunos e usuários da ferramenta. Em outro projeto, com aplicações por realidade virtual, Kucera et al. (2018) despertaram o interesse e a popularização de áreas da automação e mecatrônica entre futuros estudantes do ensino superior. Tendo isto em consideração, evidencia-se também que outros significativos benefícios podem ser alcançados pelo uso de tecnologias virtuais na educação: a virtualização de ambientes possibilita a redução de custos com a compra de equipamentos e de manutenção, otimiza o processo de aprendizado em treinamentos, minimiza o uso de espaços físicos, além de ser um motor para incentivar a inovação das relações de ensino e aprendizagem (Martin and Bohuslava, 2018).

Voltando-se para áreas de ensino de cunho tecnológico, como a área de automação industrial, há uma demanda por um conjunto de ferramentas que proporcionem o desenvolvimento do ensino por métodos que implementem sistemas mais interativos, com redução de custo e que visem a segurança. Entre os mais diversos tipos de *softwares* para simulação e modelagem que geralmente se encontram disponíveis nos laboratórios de automação, não há a disponibilidade (que seja gratuita e de código livre) de um ambiente multifuncional capaz de realizar a representação de espaços e processos industriais. Esforços, como realizado por Zata et al. (2016), para propor e implementar sistemas que virtualizam estes processos para ampliar os recursos dos laboratórios são realmente válidos, no entanto, deve-se também atentar para o emprego de ferramentas gratuitas, uma vez que pretende-se ampliar este acesso.

Diante disto, este trabalho tratou de definir uma arquitetura modular que combinasse funcionalidades, tais como: modelagem tridimensional de equipamentos, simulação com interação do usuário e modelagem matemática, projetando a criação de um ambiente de realidade mista com a implementação de estudo de caso de uma célula fabril, contendo uma esteira e um manipulador robótico.

O presente está dividido da seguinte forma: na seção 2 são apresentados trabalhos relacionados ao tema; na seção 3 discute-se o conceito de realidade mista e os limites do escopo do trabalho; na seção 4 descreve-se as tecnologias utilizadas no estudo; na seção 5 é apresentada a arquitetura proposta e seu funcionamento; na seção 6 descreve-se as perspectivas de uso da arquitetura; a seção 7 acompanha um estudo de caso com manipulador robótico em uma célula fabril; finalmente seguido pelas conclusões na seção 8.

2. TRABALHOS RELACIONADOS

Há disponível na literatura alguns casos de desenvolvimento de ambientes virtuais para ensino de áreas comuns à automação industrial. Schaf and Pereira (2010) propuseram a estrutura de um ambiente de ensino baseado em um Ambiente Virtual de Aprendizagem (AVA), com uma interface social 3D e um laboratório virtual controlado por *middleware*. Assim como Callaghan et al. (2012), que desenvolveram um ambiente para aprendizado remoto construído em 3D baseando-se em técnicas de mundos virtuais e jogos, com a possibilidade de simulação de experimentos de engenharia elétrica. Tais trabalhos concentraram-se na elaboração de ambientes de aprendizagem com a viabilidade de acesso remoto e a colaboração entre seus usuários.

Tendo em vista a ênfase em modularidade objetivada no início deste trabalho, tomou-se como direção a possibilidade de integração entre ambientes e equipamentos em diferentes configurações. Ou seja, estabeleceu-se uma arquitetura que proporcionasse aos usuários o uso de componentes somente virtuais, ou virtuais e reais, ou somente reais. Esta capacidade de múltiplas configurações de ambiente e conexão teve como intermediador um Controlador Lógico Programável (CLP). De forma equivalente procederam Martin and Bohuslava (2018), os quais apresentaram uma ferramenta de realidade aumentada que, por meio de um sistema modular e desenvolvido em uma plataforma *open source*, objetivava o ensino de automação industrial. O trabalho incorporou um equipamento didático com sensores e atuadores controlados por CLP.

Lima et al. (2012) relataram a experiência do ensino de automação industrial em laboratório brasileiro, com a utilização de três células fabris (robôs articulados e esteiras). O estudo tratou do processo de integração entre o *hardware* presente no laboratório e ferramentas de *software* tais como: *Visual Object Net++* e *Promodel*. A estrutura foi utilizada para o ensino de engenharia, onde os estudantes puderam desenvolver, de modo empírico, a otimização de *layouts* de células fabris, aliando modelagem de processos por meio de teoria de redes de Petri e *software* de análises numéricas.

Em seu trabalho, Brito et al. (2018) combinaram duas ferramentas comerciais: *Simulink* (MATLAB) e *AutoDesk Inventor* para o controle de plantas industriais virtuais, promovendo um laboratório virtual. O trabalho realizado teve como um dos principais objetivos proporcionar uma experiência de laboratório para melhorar as habilidades dos alunos em programação de CLPs. Diante disso, a abordagem do presente trabalho, diferentemente deste anterior, fundamentou-se na universalização de acesso à tecnologia, utilizando-se de uma série de ferramentas gratuitas ou educacionais para reduzir custo e viabilizar seu uso.

3. REALIDADE MISTA

A realidade mista é uma extensão dos conceitos de realidade virtual e realidade aumentada. Esta produz um meio de interação para objetos físicos e digitais coexistirem e interagirem em tempo real, podendo ser entendida como uma fusão entre mundo real e virtual (Kucera et al., 2018). A implementação da realidade mista frequentemente é abordada pelo aspecto da realidade virtual, o que permite a flexibilização em diversas aplicações (como na área de robótica), diminuindo a necessidade de componentes físicos e de custo de equipamentos. Assim, uma série de experiências podem ser realizadas, evitando-se restrições de cunho físico e espacial (Hönig et al., 2015).

A proposição de ambiente modular deste trabalho é conectada, em alguns aspectos, ao conceito de realidade mista, uma vez que visa a integração de elementos virtuais 3D com elementos reais disponibilizados em laboratório de ensino de automação. Como abordagem inicial do estudo, optou-se tão somente pela confirmação da viabilidade e

validação na integração destes dois ambientes. Cabe ressaltar ainda que, a característica de imersão do usuário no ambiente virtual vai além do escopo deste estudo.

4. TECNOLOGIAS UTILIZADAS

As tecnologias utilizadas na proposta de integração, como já mencionadas, focalizaram-se em ferramentas gratuitas ou de versões educacionais, como forma de ampliação de acesso para os usuários.

A integração entre os ambientes baseou-se no *framework Robot Operating System* (ROS). O ROS é uma ferramenta *open source* capaz de realizar o gerenciamento e controle de robôs. Dentro do ROS, a execução de programas em Python ou C++ funcionam como nós desta arquitetura. Os nós podem realizar uma série de comunicações por meio de tópicos, assinando ou publicando dados para estes. A troca de dados entre os nós pode ser realizada de acordo com diferentes formatos de mensagens, o que flexibiliza o desenvolvimento de aplicações para robôs (Quigley et al., 2009).

Para desenvolvimento de modelagem e simulação 3D requerida na arquitetura, adotou-se o CoppeliaSim, desenvolvido pela Coppelia Robotics. Comparado a outras ferramentas similares, como o Gazebo e o ARGoS, o CoppeliaSim levou vantagem por possuir uma interface mais intuitiva, ter uma biblioteca maior e melhor organizada de componentes, possibilidade de controle dos elementos de forma individual através de diversas linguagens de programação (C/C++, Python, Java, scripts Lua) além da interface com o ROS. (Pitonakova et al., 2018)

CoppeliaSim é oferecido em três versões: a primeira, CoppeliaSim Player, trata de uma versão gratuita, porém, com funcionalidades limitadas. Já CoppeliaSim Pro é a versão comercial do *software*, com todas as funcionalidades disponíveis. A última, e que foi utilizada neste estudo, CoppeliaSim Pro Edu, é a versão educacional da ferramenta, distribuída gratuitamente para fins educacionais.

Como ferramenta de modelagem matemática da arquitetura selecionou-se uma linguagem que pode realizar a modelagem de múltiplos domínios físicos (mecânicos, pneumáticos, elétricos, hidráulicos), Modelica. Essa linguagem, juntamente com a ferramenta OpenModelica, foi a mais adequada para utilização no processo de modelagem do ambiente. Os parâmetros para esta escolha basearam-se no requerimento de ferramentas gratuitas, *open-source*, de fácil manipulação e com a possibilidade de ser customizável, adequando-se aos requerimentos do projeto. Tal linguagem é baseada em equações e orientada a objeto, permitindo o encapsulamento de código, a reusabilidade de modelos e o uso de hierarquia entre componentes. Modelica tem sido desenvolvida em um esforço internacional, para ser de fácil utilização e ter um variado acervo de bibliotecas (Seabra and Machado, 2009).

Modelica já foi utilizada para o ensino de dinâmica e cinemática de sistemas mecânicos, como realizado por Seabra and Machado (2009). Além disso, a linguagem é recorrentemente utilizada como uma ferramenta para modelagem de equipamentos robóticos, em que se necessita modelar desde componentes mecânicos à elétricos (Dwiputra et al., 2014). Para manipulação desta linguagem no presente trabalho, fez-se o uso do ambiente computacional OpenModelica (em sua versão 1.14.1). Uma plataforma multifuncional capaz de modelar, simular e otimizar modelos baseados em linguagem Modelica (Fritzson et al., 2019).

5. MODULARIDADE E INTEGRAÇÃO DA ARQUITETURA

A arquitetura proposta é organizada de forma a proporcionar a modularidade em sua utilização. Os *softwares* de simulação definidos neste trabalho podem estar associados de tal forma que simulações de ambientes reais e virtuais podem trabalhar de maneira conjunta ou separada, de acordo com a necessidade do projeto. Por exemplo, pode-se optar por trabalhar em uma abordagem somente virtual, utilizando-se das características de representação gráfica 3D do CoppeliaSim e da modelagem de multi-domínios da linguagem Modelica. Também é possível integrar o comportamento de sensores conectados ao CLP para prover dados do ambiente real (do laboratório de ensino) para serem representados em simulações virtuais. Com isso, a plataforma viabiliza o uso de tecnologias para o ensino de disciplinas da área de automação industrial.

A estrutura foi pensada e elaborada para ser trabalhada dentro de uma rede local, tomando como proveito a capacidade que o ROS tem de gerenciar nós em uma estrutura de múltiplas máquinas. Sendo assim, pretendeu-se minimizar os custos computacionais das simulações de cada *software* envolvido na arquitetura, distribuindo tais simulações em no mínimo duas máquinas da mesma rede.

Para a integração do OpenModelica na arquitetura, e consequentemente, da linguagem Modelica, utilizou-se de duas bibliotecas disponibilizadas no repositório GitHub, as quais atuam para servir como ponte de comunicação entre o ROS e OpenModelica. Durante a simulação em tempo real do OpenModelica, dados trafegam por meio de um soquete TCP/IP: O nó *modros_node* (escrito em C++), dentro do sistema ROS, atua como um servidor, enquanto um bloco *ROS_Sampler* dentro do OpenModelica serve de cliente. Este bloco, disponibilizado na biblioteca *ROS_Bridge*, se utiliza de um método de chamada de função externa (escrita em C) para enviar e receber dados para o nó presente no ROS.

Já a integração da ferramenta CoppeliaSim, foi proporcionada por meio de outra interface com ROS, sendo uma forma de acesso para aplicações ou *hardwares* externos a este *framework* (Rohmer et al., 2013). As funcionalidades do *ROS Interface* puderam ser ativadas por um *plugin* (*libsimExtROSInterface*), o que permitiu que o CoppeliaSim atuasse como um nó ROS, denominado de *sim_ros_interface*.

O princípio de comunicação entre o ROS e o CLP ocorre em duas etapas, uma vez que a diferença de arquitetura dos dois sistemas não permite uma integração direta entre ambos. Para proporcionar tal integração, elaborou-se um programa em Python, com funcionalidades de *publisher* e *subscriber*, como sendo um nó responsável pelas informações do CLP. Este mesmo programa se utilizou de comunicação por soquete TCP/IP para promover a compatibilização entre os sistemas. O CLP, inserido na rede local do laboratório, funciona como um servidor. A partir

disto, o programa traduz os dados recebido do CLP e os vincula ao nó específico. Com a informação disponibilizada em um tópico no ROS, outros programas têm acesso aos dados do ambiente real. O contrário acontece da mesma forma, a informação vinda de um tópico é traduzida e assim enviada ao CLP.

A integração entre o controlador e o ROS, via soquete, foi construída através de blocos presentes na programação Ladder, os quais também foram utilizados para a construção do sistema de memória e comunicação TCP/IP. A rede de memórias, parte essencial da integração entre os sistemas, foi projetada a partir de blocos e funções para o recebimento dos dados vindos através do soquete ou da rede Profibus, em especial os vindos do manipulador.

A comunicação via soquete é nativa do CLP, de modo que existem blocos Ladder específicos para esse procedimento. A função destes blocos é a de receber os dados através de sequências de *bytes*, com tamanho determinado pelos parâmetros do bloco. Este tamanho deveria permanecer constante durante toda a execução. Devido esta necessidade, foi determinado um padrão de organização com os dados divididos em grupos de quatro *bytes*, mesmo método utilizado no manipulador robótico, o que facilitou a integração com este.

Em relação a comunicação entre o CLP e o manipulador robótico, o controlador pode receber e enviar informações deste, visto que foi implementado a interface de comunicação entre os sistemas. Com essa integração e troca de dados, o CLP é capaz de comandar funções do braço, como enviar posições e velocidades de movimentação, permitindo assim a leitura pelo manipulador para execução de rotinas de movimentação.

Para comunicação com o ROS, os blocos de memória e de comunicação TCP/IP foram montados de forma que pudessem ser facilmente modificados, possibilitando a alteração do número de *bytes* ou quantidade de variáveis que devem ser armazenadas, dependendo da necessidade. Além disso, foi criada uma função que permite a troca rápida entre as funções de receber ou enviar mensagens, dado que o CLP não consegue com a formatação existente executar as duas funções ao mesmo tempo.

Os dois protocolos de comunicação da Figura 1, PROFIBUS e ASI, entre o CLP e equipamentos do laboratório, poderiam ser substituídos por outros protocolos. Dada a construção da arquitetura, que tem como base o ROS, independentemente do protocolo escolhido a integração aconteceria de uma mesma maneira. Uma alternativa poderia ser o protocolo OPC UA (*OLE for Process Control Unified Architecture*), que usa um protocolo binário otimizado baseado em TCP, este que já é utilizado na comunicação do ROS e do CLP. Dessa forma, sua adoção demandaria poucas adaptações.

6. USO DA ESTRUTURA PARA O ENSINO

Inicialmente, o potencial que se objetiva com a inserção desta estrutura modular nos processos de aprendizagem é de que alunos de disciplinas de controle e automação industrial, ou áreas comuns, tenham à disponibilidade uma combinação de ferramentas para pôr em prática todo o arcabouço de conhecimento absorvido da forma de ensino

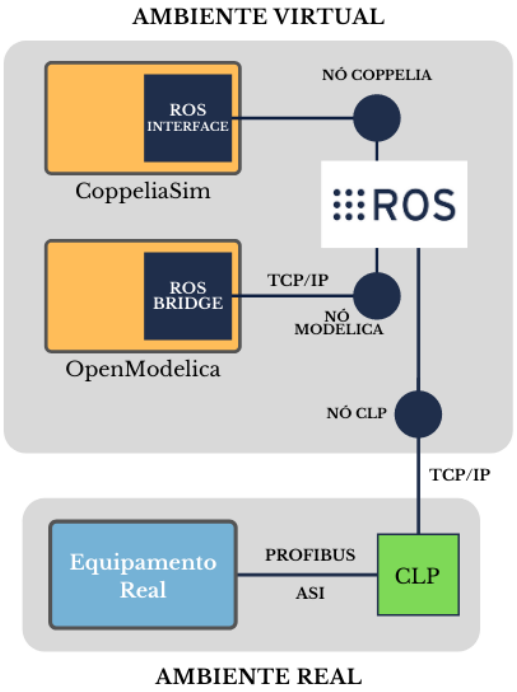

Figura 1. Arquitetura baseada em modularidade

tradicional. A arquitetura surge como um meio acessível e alternativo de implementar um ambiente industrial, sem dispor de recursos materiais ou *softwares* pagos e especializados.

Com as características desta arquitetura, um experimento possível de ser implementado, até mesmo nos mais simples laboratórios de automação, seria a simulação de um atuador pneumático, combinando comportamento e representação tridimensional com dados coletados de um atuador pneumático real. Em outra perspectiva, a realização de arranjos mais complexos também são factíveis diante do uso desta arquitetura. Tomando por base o trabalho realizado por Lima et al. (2012), com a utilização de três células fabris reais (robôs articulados e esteiras), conseguiria-se representar a mesma combinação e disposição de equipamentos sem a necessidade de dispor destes fisicamente, ampliando o acesso à tal tecnologia.

O conjunto de funcionalidades da arquitetura proposta é capaz, também, de ser manipulada para ser expandida também à outras áreas de ensino. Em uma dessas funcionalidades, devido a capacidade de representação de múltiplos domínios da linguagem Modelica e suas bibliotecas, seria possível trazer benefícios também para outras áreas do conhecimento, assim como foi alcançado por Seabra and Machado (2009) no ensino de dinâmica e cinemática.

7. ESTUDO DE CASO: MANIPULADOR ROBÓTICO EM CÉLULA FABRIL

Para a validação da arquitetura proposta, realizou-se o estudo e implementação de uma célula fabril, contendo um manipulador robótico de 6 graus de liberdade (NS-16-1.65, do fabricante COMAU), para execução de tarefas de *pick and place*. Tal tarefa consiste de o manipulador robótico

pegar objetos, um por vez, e posicioná-los no ambiente de acordo com uma programação. O braço robótico descrito faz parte de uma série de equipamentos disponíveis no laboratório deste estudo e que já foi utilizado em outros estudos.

7.1 Virtualização da Célula

Para realizar a representação virtual do NS-16-1.65, adaptou-se as peças do manipulador para o CoppeliaSim, por meio de modelos em *Computer Aided Design* (CAD), disponibilizados no site do fabricante. Sete peças em CAD foram unidas por meio de seis juntas, formando uma estrutura hierárquica e possibilitando a movimentação em cada eixo.

Uma pequena planta foi elaborada, com o braço robótico, uma esteira contendo etapas de detecção por sensores e quatro mesas. Tanto a esteira quanto as mesas, utilizadas na montagem da planta, fazem parte do conjunto de peças disponibilizadas por padrão no simulador do CoppeliaSim. A inserção do conjunto de quatro sensores sobre a esteira foi disposta de modo a estarem paralelos uns aos outros e, para uma melhor identificação, ganharam cores distintas, sendo eles: sensor laser (azul claro), sensor capacitivo (azul escuro), sensor ultrassônico (rosa) e sensor indutivo (verde).

Além dos componentes mencionados, blocos de quatro tipos foram criados para interagirem com a rede de sensores. As classes dos blocos foram dos mesmos tipos e cores correspondentes ao que foi descrito para os sensores. Elaborou-se também uma interface para que o usuário gerenciasse o aparecimento de blocos, no início da mesa (lado direito da Figura 2), durante a simulação virtual da célula fabril.

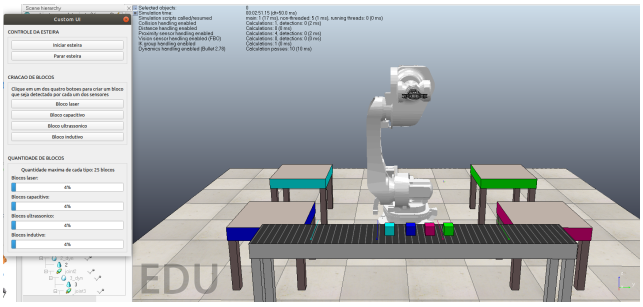

Figura 2. Cena no CoppeliaSim com manipulador e componentes da célula fabril

7.2 Modelagem da Esteira

Para validar a integração e troca de informações entre as simulações virtuais, teve-se como um dos principais objetivos modelar o comportamento da esteira externamente ao ambiente do CoppeliaSim. Utilizou-se, portanto, da linguagem Modelica para representar a ações que a esteira deveria tomar frente ao sinais obtidos pelos sensores no ambiente virtual 3D. Foi adotada uma abordagem simplificada para modelar a física da esteira, assim como realizado por del Pozo et al. (2012). No entanto, com a mesma linguagem, também é possível abordar o modelo da esteira utilizando-se de componentes mecânicos, como: engrenagens, rolamentos e motores, assim como no mundo real.

Aproveitando-se do editor gráfico e textual OMEdit (presente no ambiente OpenModelica) criou-se uma biblioteca denominada *MODEST*. Alguns blocos dentro deste pacote (ou biblioteca) são expostos na Figura 3. Nesta figura, em (2), visualiza-se um controlador baseado em estados, que coordena as ações da esteira de acordo com os dados recebidos do bloco de comunicação *Ros_Sampler* (4). Já em (3) é exibido o bloco de representação da esteira, que por meio de um booleano vindo do controlador pode ser acionada ou não. A modelagem da movimentação da esteira é baseada em equacionamento cinemático de uma partícula sobre esta. Além disso, dados de posição, velocidade e movimentação são entregues na saída deste bloco.

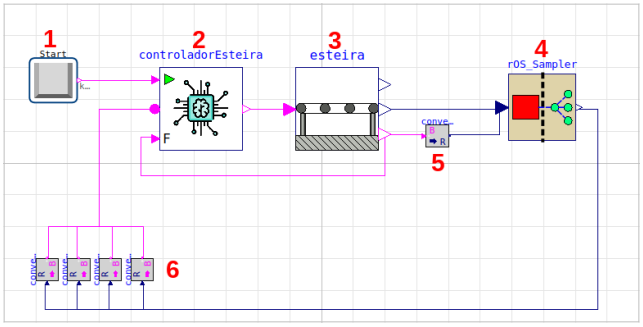

Figura 3. Representação gráfica do modelo da esteira associado ao controlador e sistema de comunicação.

Além da elaboração de uma biblioteca própria, fez o uso de algumas bibliotecas disponibilizadas no sistema de repositórios GitHub, foram elas: *ROS_Bridge*, *Modelica_DeviceDrivers* e *Modelica_Synchronous*. A biblioteca *ROS_Bridge*, em especial, proporcionou a comunicação da simulação no OpenModelica com o sistema de nó do ROS, atribuindo à simulação um endereço IP para onde enviar os dados da esteira e de onde receber dados dos sensores.

7.3 Integração de componentes e execução da tarefa

A integração e transporte dos dados entre os *softwares* da estrutura foi organizada pelo ROS. A simulação da célula foi realizada em 3 máquinas, em uma disposição *multiple machines*, tendo como sistemas operacionais o Ubuntu 18.04. Assim, com a simulação do CoppeliaSim e do OpenModelica sendo executada em tempo real, foi possível implementar um ambiente que integra o comportamento da esteira com as ações do manipulador robótico.

Optou-se, inicialmente, por vincular o controle do braço a um nó do ROS, escrito em Python. Esse nó (*controle_braco*) possui quatro rotinas pré-estabelecidas para cada um dos tipos de bloco detectados pelos sensores. Rotinas estas que dizem respeito às posições dos blocos na esteira e as posições de cada bloco em cada uma das mesas correspondentes.

Assim como no braço real, as posições utilizadas são as da cinemática direta, ou seja, a posição cartesiana e a orientação do efetuador, no caso, da garra. Entretanto, para alcançar essas posições no espaço operacional, é necessário obter as variáveis das seis juntas do braço robótico. Um algoritmo de cinemática inversa trata de

determinar essas variáveis de juntas que correspondem a uma determinada posição e orientação do efetuador no espaço operacional do braço (Cruz et al., 2007). O CoppeliaSim conta com um módulo que executa esses cálculos da cinemática inversa e define as variáveis das juntas diretamente, de forma que basta o nó de controle do braço enviar, via ROS, a posição e orientação do efetuador desejada que o módulo de cálculo da cinemática inversa do CoppeliaSim irá setar as juntas aos seus valores correspondentes para que aquela posição da garra seja atingida. (CoppeliaRobotics, 2019)

A seguir descreve-se o fluxo de trabalho e informações durante o processo de simulação em tempo real na arquitetura: o OpenModelica inicia o movimento da esteira com um valor de velocidade pré-selecionado, por exemplo 0,1m/s. O início da movimentação da esteira com a velocidade especificada é enviada via TCP/IP para o nó *modros_node* e a informação é atribuída ao tópico *model_values*. Os sinais detectados pelos sensores na esteira são atribuídas ao tópico *control_values*, que voltam ao ambiente da esteira e é administrado pelo *controladorEsteira*. Quando o sensor detecta a presença do bloco, o OpenModelica recebe a informação e para a esteira. O nó responsável pelo controle do braço entra em ação e executa uma série de movimentos pré-definidos para recolher o bloco da esteira (Figura 4.b) e recolocá-lo na mesa de cor correspondente (Figura 4.d). Depois de removido o bloco do alcance do sensor, o OpenModelica reinicia o movimento da esteira.

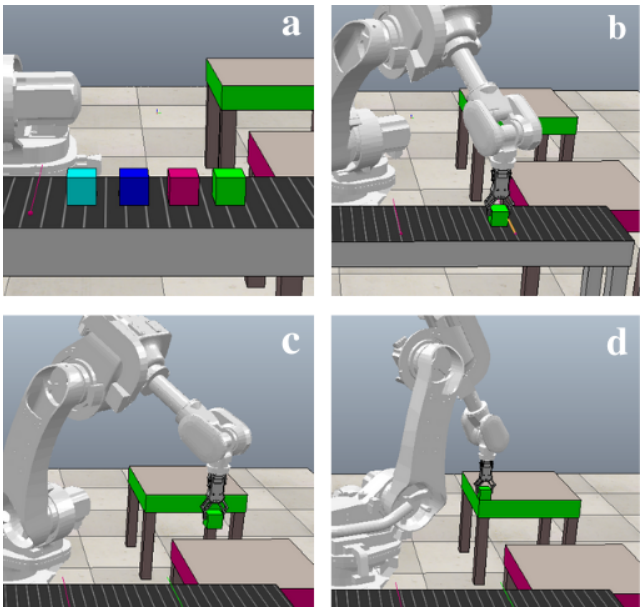

Figura 4. Blocos e etapas do pick and place

Para ilustrar a comunicação entre tópicos e nós durante uma simulação, tem-se o diagrama da Figura 5. O nó *sim_ros_interface* é o nó no qual o CoppeliaSim trabalha, já o nó *controle_braco* é um programa em Python responsável por controlar os movimentos do braço robótico a partir de informações dos sensores presentes na esteira. O nó leitor é o programa Python responsável pela conversão dos dados provenientes do CoppeliaSim para um formato adequado para interpretação pelo Modelica, e vice-versa.

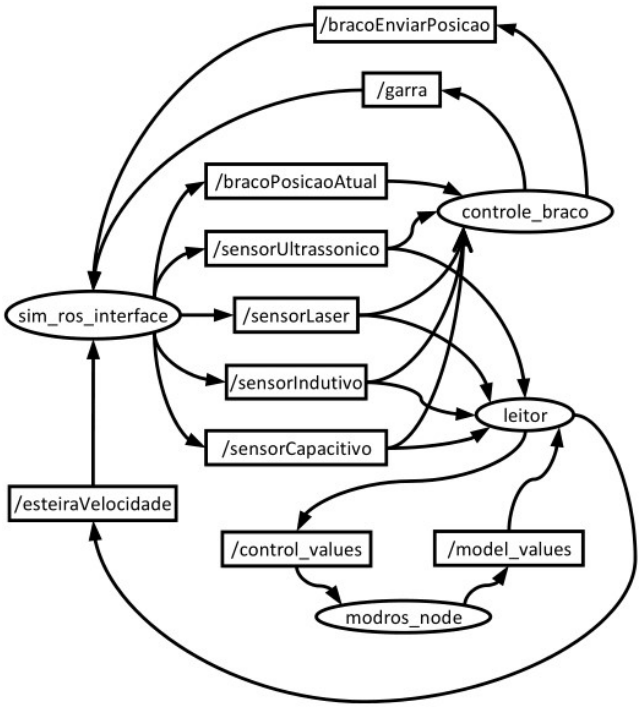

Figura 5. Rede de nós e tópicos durante a simulação da célula

Nos primeiros testes da célula, verificou-se problemas de latência provocados pela rede durante o processo de percepção do bloco pelo sensor e o envio do comando de parada para a representação virtual da esteira no CoppeliaSim. Isto ocasionou um mal posicionamento do bloco, fora do alcance da garra do braço ou até mesmo fora do alcance do sensor, o que fez a esteira reiniciar o movimento. No entanto, foi realizado uma série de otimizações nos programas para atenuar problemas desta natureza, bem como optou-se por utilizar apenas rede cabeada para melhorar o desempenho do processo de comunicação dentro da rede local.

8. CONCLUSÃO E TRABALHOS FUTUROS

Conseguiu-se realizar a integração entre os módulos da arquitetura proposta, uma vez que foi realizado algumas simulações da célula fabril do estudo de caso descrito no decorrer deste trabalho. Embora ainda necessite de ajustes e uma validação final, já desenvolveu-se um meio de comunicação efetiva entre o ROS com o manipulador robótico real NS-16-1.65, sendo necessária uma etapa de testes com todo o sistema em laboratório. A fase de conclusão entre a integração com o ambiente real foi afetada pelo período de isolamento social mundial vivido durante o desenvolvimento deste estudo. Sendo assim, o próximo passo a ser executado, será a união dos módulos já construídos com o CLP e o manipulador robótico, de forma que as próximas atividades que envolvem este trabalho não demandam tanto esforço.

O sistema desenvolvido apresenta-se como uma ferramenta de grande margem de utilização, dada a sua construção modular, o que permite executar diferentes tipos de trabalhos sem depender do uso total de seus módulos. Isso torna-se ainda mais evidente principalmente em períodos

como o do presente estudo, onde não é possível ter acesso ao laboratório (devido às regras de isolamento social), mas tem-se potencial de desenvolver técnicas de estudo somente virtuais.

Com isso, as práticas evidenciadas com o uso de tal arquitetura, como a representação virtual do ambiente do laboratório e de seus componentes, a simulação de processos industriais e a experimentação prática pelos alunos de todo este contexto, tornam útil e beneficiam as relações de ensino e aprendizagem recentes. Em trabalhos futuros, projeta-se um refinamento dos processos de comunicação entre os módulos, visando-se a diminuição da latência na troca de informações dentro da arquitetura. Pretende-se também atribuir inteligência artificial à célula fabril já descrita neste estudo, buscando otimizá-la e desenvolver outras práticas com a arquitetura aqui proposta.

AGRADECIMENTOS

Este trabalho é parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Os autores agradecem ao CNPq e a Universidade Federal da Paraíba (UFPB) pelo apoio no programa de bolsas em Iniciação Científica.

REFERÊNCIAS

Brito, J., Toledo, P., and Alayón, S. (2018). Virtual laboratory for automation combining inventor 3d models and simulink control models: Virtual laboratory for automation. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, 555–562.

Callaghan, M.J., McCusker, K., Losada, J.L., Harkin, J., and Wilson, S. (2012). Using game-based learning in virtual worlds to teach electronic and electrical engineering. *IEEE Transactions on Industrial Informatics*, 9(1), 575–584.

CoppeliaRobotics (2019). *CoppeliaSim User Manual*. https://www.coppeliarobotics.com/helpFiles/index.html.

Cruz, D.F.M.d. et al. (2007). *Implementação da cinemática inversa de robôs redundantes operando em ambientes confinados no projeto roboturb*. Dissertação de mestrado, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

del Pozo, A.S., Escaño, J.M., and Bordons, C. (2012). Simulator for control and automation using an interactive and configurable 3d virtual environment. In *2012 Proceedings of SICE Annual Conference (SICE)*, 2268–2273.

Dwiputra, R., Zakharov, A., Chakirov, R., and Prassler, E. (2014). Modelica model for the youbot manipulator. In *Proceedings of the 10 th International Modelica Conference*, 096, 1205–1212.

Fritzson, P., Pop, A., Asghar, A., Bachmann, B., Braun, W., Braun, R., Buffoni, L., Casella, F., Castro, R., Danós, A., et al. (2019). The openmodelica integrated modeling, simulation, and optimization environment. In *Proceedings of The American Modelica Conference 2018, USA*, 154, 206–219.

Hönig, W., Milanes, C., Scaria, L., Phan, T., Bolas, M., and Ayanian, N. (2015). Mixed reality for robotics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5382–5387.

Kucera, E., Haffner, O., and Leskovskỳ, R. (2018). Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine. In *2018 Cybernetics & Informatics (K&I)*, 1–5.

Lima, F., Prado, Á.C., Massote, A.A., and Leonardi, F. (2012). An experience of teaching industrial automation for industrial engineering undergraduate students. In *Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Hong Kong*, 1–6.

Martin, J. and Bohuslava, J. (2018). Augmented reality as an instrument for teaching industrial automation. In *2018 Cybernetics & Informatics (K&I)*, 1–5.

Milano, F., Vanfretti, L., and Morataya, J.C. (2008). An open source power system virtual laboratory: The psat case and experience. *IEEE Transactions on Education*, 51(1), 17–23.

Pitonakova, L., Giuliani, M., Pipe, A., and Winfield, A. (2018). Feature and performance comparison of the v-rep, gazebo and argos robot simulators. In *Annual Conference Towards Autonomous Robotic Systems*, 357–368.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 5.

Rohmer, E., Singh, S.P., and Freese, M. (2013). Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proceedings of The International Conference on Intelligent Robots and Systems*, 1321–1326.

Schaf, F.M. and Pereira, C.E. (2010). Ambiente de realidade mista 3d colaborativo: Mrcs-carlab3d. In *XVIII Congresso Brasileiro de Automática (CBA 2010), Bonito, MS*.

Seabra, E. and Machado, J.M. (2009). Teaching kinematics and dynamics of multibody mechanical systems using the object oriented language modelica. *International Journal of Online Engineering (iJOE)*, 5(2), 33–38.

Zata, N.M., van Niekerk, T.I., and Fernandes, J.M. (2016). A process control learning factory with a plant simulation integrated to industry standard control hardware. In *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, 1–8.