

DESENVOLVIMENTO DE UMA PLATAFORMA PARA O APRENDIZADO EM ROBÓTICA COM BASE NO ROBÔ SCARA

Victor Rafael Bezerra Maciel – victor.maciel@ee.ufcg.edu.br

Raphael Santana Galdino – raphael.galdino@ee.ufcg.edu.br

Rafael Bezerra Correia Lima – rafael.lima@dee.ufcg.edu.br

Alexsandro Ferreira de Barros Júnior – alexsandro.junior@ee.ufcg.edu.br

Universidade Federal de Campina Grande

R. Aprígio Veloso, 882 – Universitário

58429-900 – Campina Grande – Paraíba

Resumo: A necessidade por redução de custos e aumento na eficiência são uma realidade no setor produtivo, sendo a robótica uma área cada vez mais fundamental na indústria 4.0. A metodologia focando no modelo para o desenvolvimento de um projeto, o Model-Based Design (MBD), e o uso de simulações computacionais são alguns modos de testar, gerenciar e desenvolver sistemas complexos. Neste cenário, torna-se de extrema importância que o ensino em engenharia explore novas ferramentas de ensino. Desta forma, neste trabalho é abordado o emprego do MBD ao desenvolvimento de um robô Scara, criando-se uma plataforma para ensino. São apresentados a modelagem da cinemática do robô, além de aspectos relacionados a simulação e projeto utilizando a metodologia Model-Based Design com a biblioteca Simscape do Simulink. Ao fim do projeto, comparativos entre o modelo e o robô construído foram feitos, além da criação de uma interface gráfica no ambiente App Designer.

Palavras-chave: Robótica. Model-Based Design. Simscape. MATLAB.

1. INTRODUÇÃO

A globalização impõe novos desafios à indústria, pois acirra ainda mais a competição, aumenta a pressão por produtos mais baratos e individuais, que usam tecnologias cada vez mais complexas. A solução que a indústria 4.0 propõe é um sistema de produção mais flexível, com altos níveis de conectividade e automação (RUSSWURM, 2014). Tendo em vista este cenário, é incontestável que a formação em engenharia deve abordar as tendências de modernização dos sistemas de produção, objetivando formar e capacitar profissionais preparados para lidarem com os novos paradigmas, sobretudo no que diz respeito ao projeto, desenvolvimento e implementação de sistemas robóticos.

Sistemas robóticos são cada vez mais empregados no ambiente industrial, pois são capazes de reduzir custos e aumentarem os níveis de produtividade. Contudo, robôs são máquinas complexas e muitas vezes caras, tornando muitas vezes inviável a construção de protótipos para testes e para o ensino em universidades e escolas de engenharia. Deste modo, o emprego de modelos computacionais, simulação e plataformas de ensino são de extrema importância para a formação do engenheiro em conformidade com o perfil do profissional da indústria 4.0.

Uma metodologia que pode ser aplicada para o desenvolvimento de novas tecnologias, incluindo robôs, é o Model-Based Design (MBD). O MBD é uma abordagem centrada no modelo para o desenvolvimento de sistemas dinâmicos e complexos. Por não necessitar de protótipos para o teste e implementação de um produto, esta abordagem reduz consideravelmente os custos, podendo ainda ser utilizado para fins didáticos (AARENSTRUP, 2015).

O objetivo deste trabalho é descrever o desenvolvimento de uma plataforma de ensino em robótica baseado no robô de montagem *Scara* sob a perspectiva do *Model-Based Design*, com o uso de visão computacional. Para tanto, empregou-se as ferramentas disponíveis no ambiente *Simulink*, com ênfase na *toolbox Simscape*. O *Simscape* é uma biblioteca de blocos para modelagem de sistemas dinâmicos que fornece simulações por meio de redes físicas.

O conteúdo deste artigo está dividido da seguinte forma. A seção 2 descreve a constituição do robô *Scara* construído, detalhando os componentes utilizados. A seção 3 introduz conceitos de visão computacional. A seção 4 conceitua brevemente *Model-Based Design*, citando ainda a modelagem cinemática do braço e apresentando a biblioteca *Simscape* e a interface gráfica da plataforma. A seção 5 analisa os resultados obtidos experimentalmente, comparando-os com os resultados verificados no modelo. A seção 6 contém as considerações finais e conclusões do trabalho, além dos agradecimentos.

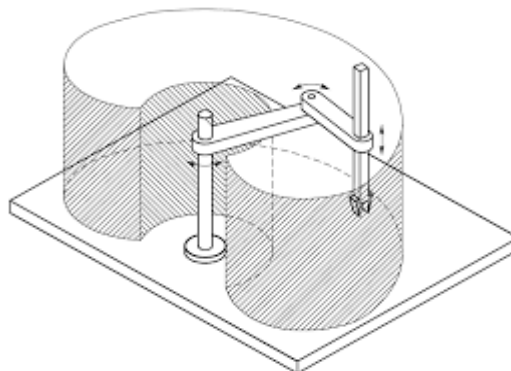
2. CONSTITUIÇÃO DO ROBÔ

2.1 Robô *Scara*

Scara é o acrônimo em inglês para *Selective Compliance Assembly Robot Arm* e significa braço robótico para montagem de conformidade seletiva. Rápidos e compactos, robôs *Scara* são versáteis, apresentando bom desempenho em operações de montagem de componentes de pequenas dimensões e transferência de peças entre células (MAKINO, 1982).

Em geral, este tipo do robô apresenta duas juntas de rotação dispostas em paralelo, obtendo-se um movimento em um plano, podendo ainda dispor de uma junta prismática perpendicular a esse plano. Por tais características, equipamentos desta espécie possuem um volume de trabalho aproximadamente cilíndrico, conforme ilustrado na Figura 1.

Figura 1 – Volume de trabalho típico de um robô *Scara*.



Fonte: Venditelli (2016)

2.2 Características físicas

A estrutura física do robô *Scara* construído assemelha-se à apresentada na Figura 1. A coluna central do braço é composta por um tripé de aço, fixada ao primeiro elo com o auxílio de um suporte plástico fornecido pelo fabricante dos motores. Cada elo é feito de duas placas de fibra de vidro, cada uma com 17 cm de comprimento e 2,5 cm de largura. Além disto, estão presentes nas juntas de rotação atuadores AX-12A, que são abordados na subseção 2.3. Em razão de aspectos construtivos do robô, os motores possuem limitações quanto ao seu movimento, ficando a primeira e a segunda junta restritas a rotacionar 102° e 150°, respectivamente.

O robô contém ainda uma câmera USB *Dlink DSB C-120*, posicionada acima do braço utilizando um suporte em L invertido em aço, com 25 cm de comprimento e 52 cm de altura. É mostrada na Figura 2 a estrutura física do robô *Scara*.

Figura 2 – Estrutura física do robô



Fonte: Autoria própria

2.3 Motores AX-12A

O servo motor AX-12A é um atuador inteligente, que integra um redutor de velocidade, um motor de precisão, um circuito de controle e uma interface de comunicação, em um único conjunto. De acordo com o manual fornecido pela ROBOTICS, empresa sul coreana que fabrica este dispositivo, o motor AX-12A possui um peso de 54,6g, com dimensões de 32 mm x 50 mm x 40 mm e possuindo uma resolução de $0,29^\circ$, devido aos 1024 níveis de controle, variando o ângulo de 0° a 300° . Por ser leve e pequeno, é adequado para aplicações robóticas simples, sobretudo àquelas ligadas ao ensino da robótica.

Para comunicação, é estabelecida uma conexão serial TTL do tipo *Half-Duplex*, em que um motor AX-12A identificado por um ID único atua como um servidor, que recebe comandos e retorna respostas ao controlador externo (PC). O dispositivo utilizado para conectar os motores *Dynamixel* presentes no braço robótico foi o *USB2Dynamixel* a partir da porta USB de um computador, dispondo os motores em série. (ROBOTICS, 2019)

3. VISÃO COMPUTACIONAL E PROCESSAMENTO DE IMAGEM

Visão computacional é o processo de determinação e descrição de um espaço tridimensional no qual um agente está inserido, extraindo-se informações a partir de uma ou mais imagens, capturadas por uma ou mais câmeras, de modo a obter um modelo computacional do ambiente. Visão computacional pode ser entendida ainda como uma tentativa de fazer o processo inverso da formação de uma imagem, isto é, reconstruir suas propriedades e formas, além de obter parâmetros como iluminação e distribuição de cores (SZELISKI, 2010).

3.1 Transformada de Hough

A transformada de Hough é uma ferramenta comumente utilizada no ramo da visão computacional para detectar em uma imagem curvas facilmente parametrizáveis, como retas, círculos e elipses. Esta transformada fornece a relação entre o domínio da imagem e o domínio

dos parâmetros, que consiste em uma matriz de acumuladores, com dimensão igual ao número de parâmetros a serem determinados.

Círculos são curvas parametrizáveis e, portanto, podem ser facilmente encontrados em uma imagem utilizando a transformada de Hough. Qualquer círculo pode ser descrito em termos de uma equação paramétrica que depende somente de três parâmetros: as coordenadas x e y do seu centro (a,b) e o seu raio R . A equação é dada por:

$$(x - a)^2 + (y - b)^2 = R^2 \quad (1)$$

Há um conjunto de algoritmos para a detecção de círculos que empregam a transformada de Hough, tomando apenas implementações diferentes para isto. Contudo, há três etapas essenciais que são comuns: o cálculo da matriz de acumuladores, a estimação do centro do círculo e a estimação do raio do círculo.

Na primeira etapa, pixels do primeiro plano da imagem com gradiente elevado são aptos a incrementar a matriz de acumuladores, no qual cada um destes pixels forma um padrão circular, de raio fixo, centrado em um pixel de borda. Na fase de estimação do centro do círculo, as coordenadas do centro são obtidas aproximadamente buscando-se observar os máximos da matriz de acumuladores. Isso ocorre pois os pixels de gradiente elevado tendem a se acumular na região que corresponde ao centro do círculo a ser determinado. A última etapa consiste em estimar os raios dos círculos presentes na imagem. É um processo distinto ao estágio da captura do centro do círculo.

3.2 *Toolbox Image Processing* e a função *imfindcircles*

Na *toolbox Image Processing* estão presentes diversos algoritmos, transformadas e funções que permitem o processamento, análise e visualização de imagens, facilitando e automatizando tarefas e processos comuns no processo digital de imagens. Uma das funcionalidades presentes neste pacote é a função *imfindcircles*, que utiliza a transformada de Hough para encontrar círculos em uma imagem digital.

A função *imfindcircles* tem como argumentos de saída *centers* e *radii*, que são as coordenadas dos círculos encontrados e os seus respectivos raios. Como argumentos de entrada, tem-se a imagem *Img*, o intervalo do raio dos círculos procurados chamado *radiusRange*. Há ainda pares de argumentos de entrada adicionais (*Name*, *Value*), que alteram especificações do algoritmo. Estes especificadores são descritos no Quadro 2. A sintaxe da função é: $[centers, radii] = imfindcircles(Img, radiusRange, Name, Value)$.

Quadro 1 – Especificadores de *imfindcircles*

Name	Descrição	Value
'ObjectPolarity'	Indica se os círculos são mais claros ou escuros que o fundo da imagem.	- 'bright' (padrão) - 'dark'
'Method'	Indica a técnica utilizada na matriz de acumuladores.	- 'PhaseCode' (padrão) - 'TwoStage'
'Sensitivity'	Indica a sensibilidade para encontrar círculos mais obscuros ou fracos.	- Valor entre 0 e 1 - 0.85 (padrão)
'EdgeThreshold'	Indica o limite do gradiente para determinar as bordas da imagem.	- Valor entre 0 e 1 - Valor calculado usando <i>graythresh</i> .

Fonte: Autoria própria

4. MODEL-BASED DESIGN

4.1 Introdução teórica ao *Model-Based Design*

O *Model-Based Design* (MBD) é uma abordagem centrada no modelo para o desenvolvimento de sistemas de controle, processamento de sinais, comunicações e outros sistemas dinâmicos. O modelo inclui todos os componentes relevantes para o comportamento do sistema, algoritmos, lógica de controle e propriedade intelectual (AARENSTRUP, 2015).

Antes da utilização do *Model-Based Design*, para toda nova tecnologia, seria necessário a criação de um protótipo físico para realização de testes, com a finalidade de verificar e validar a tecnologia. O uso do MBD visa principalmente substituir os protótipos de plantas com valores elevados, tornando dessa forma o projeto mais viável, reduzindo custos, acelerando a detecção de erros e validando requisitos.

Alguns cientistas e engenheiros ainda têm uma reação negativa sobre o *Model-Based Design*, porque eles assumem que a simulação e modelagem está próximo da teoria porém distante da prática. Contudo, modelos são boas representações da realidade, e, para o ensino da engenharia são fundamentais, pois facilitam o entendimento e tornam didático o ensino de sistemas complexos (PATERNO, 1999).

Além de reduzir custos, o MBD apresenta outras vantagens, dentre elas: gerenciar sistemas complexos, automatizar tarefas complicadas e propensas a erros, explorar rapidamente novas ideias, criar uma linguagem comum que promova a comunicação e colaboração, capturar e reter propriedade intelectual, melhorar a qualidade do produto e reduzir os riscos (AARENSTRUP, 2015). Além disto, muitas empresas e projetistas buscam acelerar o desenvolvimento dos seus algoritmos, valendo-se do ambiente de simulação para modelar os sistemas de controle e a planta (NXP, 2017).

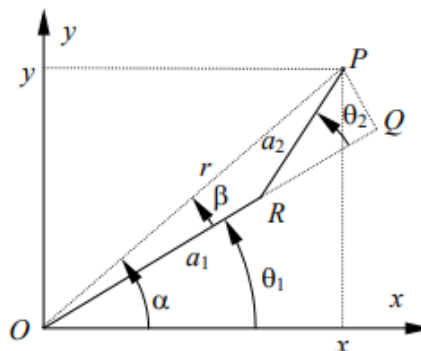
4.2 Modelagem cinemática

Para encontrar o modelo matemático do robô Scara é utilizado um robô com dois graus de liberdade como mostrado na Figura 3. Existem dois processos para determinação dos parâmetros: Cinemática Direta e Cinemática Inversa. A cinemática direta, tem como objetivo encontrar os valores cartesianos, x e y , quando se tem os ângulos θ_1 e θ_2 , as Equações (2) e (3) mostram essa relação:

$$x = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \quad (2)$$

$$y = a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \quad (3)$$

Figura 3 – Cinemática direta com dois graus de liberdade



Fonte: Apostila de robótica

Já a cinemática inversa, por sua vez, é o processo oposto, isto é, encontrar os valores dos ângulos θ_1 e θ_2 em função dos valores de x e y . Aplicando a lei dos cossenos na junta 2 do braço encontrado na Figura 3 e isolando θ_2 , obtém-se a Equação (4), dada por:

$$\theta_2 = \pm \arccos\left(\frac{x^2 + y^2 - a_1^2 - a_2^2}{2 a_1 a_2}\right) \quad (4)$$

O sinal de mais e menos em θ_2 indica que existem duas posições do braço que alcançam o mesmo resultado, sendo um deles com o cotovelo do braço flexionado para cima e com o outro, para baixo. Deste modo, o ângulo θ_2 é positivo quando o cotovelo está para baixo e negativo quando o cotovelo está para cima. Além disto, é possível demonstrar que o ângulo θ_1 pode ser encontrado em termos do ângulo θ_2 e das coordenadas x e y , sendo dado pela Equação (5):

$$\theta_1 = \tan^{-1}\left(\frac{y(a_1 + a_2 \cos \theta_2) - x a_2 \sin \theta_2}{x(a_1 + a_2 \cos \theta_2) - y a_2 \sin \theta_2}\right) \quad (5)$$

4.3 Simscape

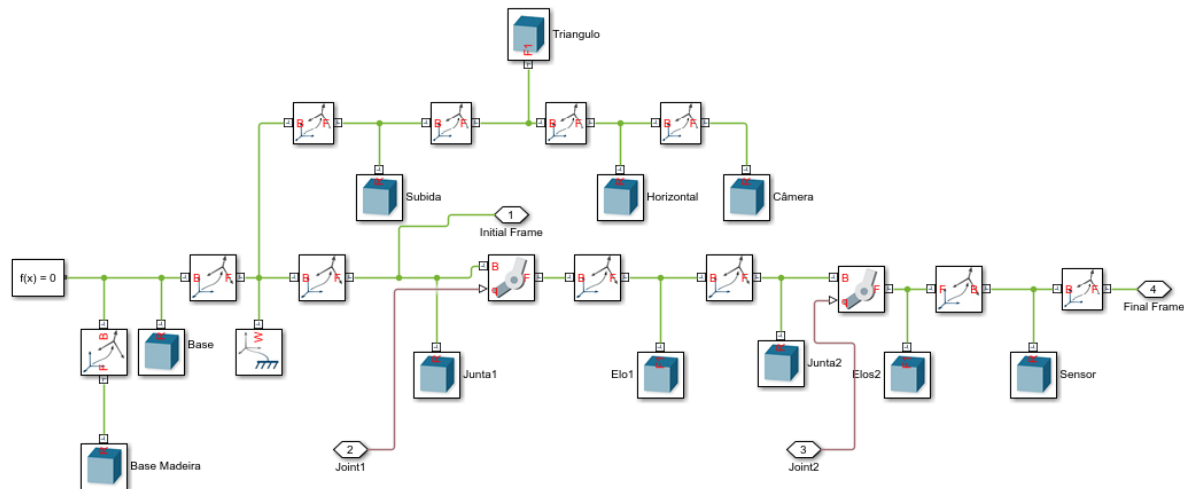
Simscape é um conjunto de bibliotecas de blocos e recursos para modelagem de sistemas físicos no ambiente *Simulink*. Enquanto blocos tradicionais do *Simulink* representam operações matemáticas, os blocos do *Simscape* representam por si só diagramas dos modelos matemáticos do sistema em projeto. Esta diferença se deve a abordagem utilizando redes físicas, na qual cada sistema é representado por elementos funcionais que trocam energia entre si, de forma não direcional, por meio de portas conectadas. A conexão é feita de modo a se assemelhar às conexões entre componentes físicos reais (MATHWORKS, 2019).

Devido à facilidade e variedade de elementos que o *Simscape* fornece, empregá-lo para modelar problemas é relativamente simples, sendo possível ainda utilizá-lo como uma ferramenta auxiliar para o ensino de engenharia.

Para a modelagem do robô *Scara*, foi utilizada a biblioteca *Multibody*, que faz parte do *Simscape*. Ela fornece um ambiente de simulação de sistemas com múltiplos corpos em 3D, como suspensões de automóveis, equipamentos de construção e robôs. De acordo com a montagem feita, o *Simscape* formula e resolve as equações mecânicas que descrevem os corpos envolvidos, levando em conta os parâmetros fornecidos para a simulação.

Para a construção do modelo, foram usados os blocos *Solid*, que correspondem aos elos do braço e os blocos *Revolute Joint*, que representam as juntas de revolução. Além disso, foi necessário o emprego de um bloco de referencial inercial, o *World Frame*, e alguns blocos de transformação de referências, o bloco *Rigid Transform*, que é responsável por rotacionar e/ou transladar a referência. A Figura 4 apresenta o modelo desenvolvido. Alguns outros blocos encontrados nesta figura são os blocos *Connection Port*, que servem para fazer a conexão do modelo com os demais conjuntos de blocos da simulação, e um bloco *Solver Configuration*, que é o bloco responsável por especificar os parâmetros do modelo antes da simulação de cada rede física presente no modelo.

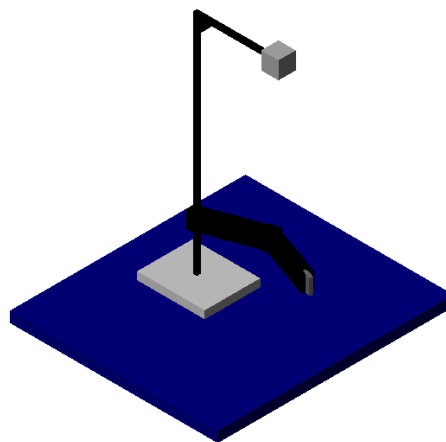
Figura 4 – Modelo no *Simscape* do robô *Scara*



Fonte: Autoria própria

Com o modelo desenvolvido, foi possível avaliar se a modelagem cinemática de fato descrevia adequadamente o braço robótico, além de que tornou fácil explorar ideais para movimentação e trabalho do robô, tornando o ensino de robótica mais didático. A Figura 5 mostra o modelo na simulação.

Figura 5 – Modelo do robô *Scara* no *Simscape*



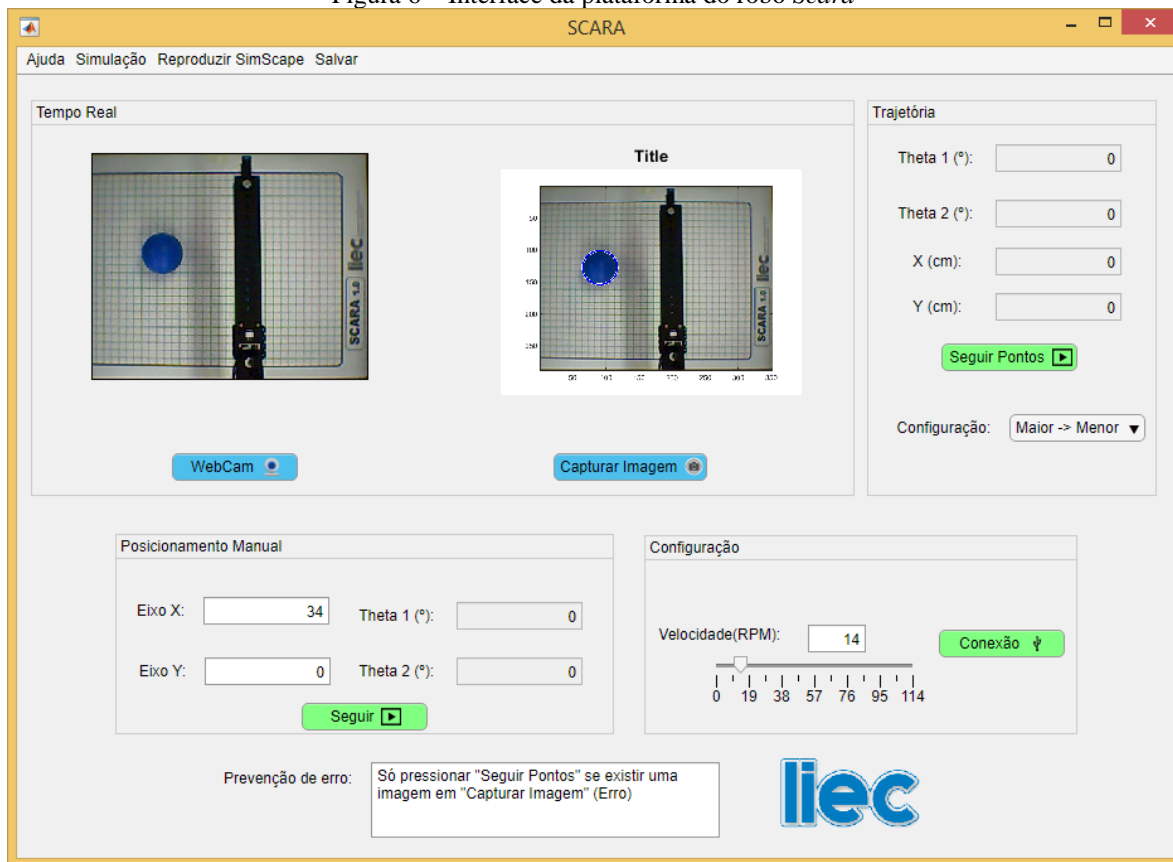
Fonte: Autoria própria

4.4 Interface gráfica

A interface humano-computador como sendo a parte de um sistema interativo responsável por mostrar ao usuário o estado do sistema, traduzir ações do usuário em pedidos de processamento (funcionalidades), e mostrar os resultados de forma adequada, além de coordenar a interação (CYBIS, 2000).

A interface criada para comandar o robô *Scara* foi embasada na definição de Cybis, tendo em vista ser tão didática quanto possível, permitindo a sua utilização para o ensino e estudo sobre o robô. A interface produzida é apresentada na Figura 6, e foi feita com o auxílio da ferramenta *App Designer*, do MATLAB.

Figura 6 – Interface da plataforma do robô Scara



Fonte: Autoria própria

5. ANÁLISE DE RESULTADOS

Os resultados obtidos com a simulação mostram que o modelo cinemático adotado descreve de forma bastante satisfatória o robô *Scara*, conforme pode ser visto na Tabela 1. A adoção do *Model-Based Design* com auxílio da biblioteca *Simscape* para modelagem e simulação no ambiente *Simulink* permitiu prever erros e limitações mecânicas do robô antes mesmo da sua fabricação, reduzindo o tempo e o material gasto no projeto.

Tabela 1 – Comparação entre os valores de ângulo do robô físico real com os valores de ângulo da simulação

x (cm)	y (cm)	θ_1 real (°)	θ_2 real (°)	θ_1 simulação (°)	θ_2 simulação (°)	ErroX (%)	ErroY (%)
10,00	5,00	105,90	291,20	105,80	291,60	1.057	0.776
-4,00	12,00	190,30	285,60	190,30	286,30	0.222	1.727
20,00	-7,00	181,80	46,63	182,20	47,10	0.894	2.368
-7,50	-7,50	86,80	5,865	86,82	6,355	1.450	4.50
17,50	-19,50	141,60	70,09	141,50	70,82	1.045	0.123
-15,00	-15,00	66,28	46,63	66,40	47,21	0.069	1.353
-2,00	20,00	191,80	257,50	192,00	257,50	2.859	0.051

Fonte: Autoria própria

Analisando os valores experimentais obtidos é possível notar que o resultado é satisfatório, mostrando que a modelagem cinemática descreve adequadamente a realidade.

CONSIDERAÇÕES FINAIS

O principal objetivo deste trabalho foi alcançado, tendo em vista que foi desenvolvida uma plataforma para o ensino de robótica, mais especificamente sobre o robô *Scara*. Por meio de uma interface gráfica feita no ambiente de desenvolvimento *App Designer*, o usuário pode acessar facilmente a simulação, criada utilizando a biblioteca *Simscape*, com foco na importância da abordagem de projeto baseada em modelo, o *Model-Based Design* (MBD). O usuário ainda é capaz de visualizar animações que ilustram o comportamento cinemático do robô e checar o modelo empregado, o que torna a plataforma ainda mais educativa.

Com o objetivo de ser aplicado ao ensino, esse robô poderá ser implementado nos laboratórios de robótica. Como a plataforma permite que o modelo seja acessado com facilidade, pode-se abordar a simulação para verificar a modelagem cinemática. Após isto, é possível comparar as respostas obtidas no robô *Scara* real com o resultado obtido no *SimScape*. Aspectos ligados a cinemática, MBD e programação podem ser discutidos, levando à sala de aula uma forma mais prática de abordar temas que são de extrema importância na robótica.

Em trabalhos futuros, pretende-se aprimorar o robô para ele seja capaz de realizar atividades como o recolhimento e organização de peças, etapa esta qual será empregado processamento de imagem descrito, adicionando outro atuador na extremidade do elo mais distante da base. Essa melhoria visa tornar a plataforma mais didática e próxima de uma aplicação real da indústria, encorajando e desmitificando o uso da robótica para os alunos.

Agradecimentos

Agradecemos a Universidade Federal de Campina Grande (UFCG) e ao Laboratório de Instrumentação Eletrônica e Controle (LIEC) pelo apoio prestado.

REFERÊNCIAS

AARENSTRUP, R. **Managing Model-Based Design**. CreateSpace Independent Publishing [Platform, 2015].

CYBIS, Walter de Abreu; NASCIMENTO JUNIOR, W. B.. Modelagem do Conhecimento Ergonômico para Avaliação da Usabilidade de Objetos de Interação. In: **Workshop on Human Factors in Computer Systems, 2000, Gramado. Proceedings of the 3rd Workshop on Human Factors in Computer Systems**. Porto Alegre: Sociedade Brasileira de Computação, 2000.

CARRARA, Valdemir. **Apostila de Robótica**. Universidade Braz Cubas.

MAKINO, Hiroshi. ASSEMBLY ROBOT US4341502. Depósito: 24 mar. 1980. Concessão: 27 jul. 1982.

NXP. **Model-based design toolbox enabling fast prototyping and design**, 2017. Disponível em: <<https://www.nxp.com/docs/en/supportinginformation/MBDT-GeneralPresentation.pdf>>. Acesso em 24 de abril de 2019.

PATERNÒ, Fabio. **Model-based design and evaluation of interactive applications**. Springer, 1999.

ROBOTICS. **ROBOTIS e-Manual v1.29.00.** Disponível em:
<http://support.robotis.com/en/product/actuator/dynamixel/ax_series/dxl_ax_actuator.htm#Actuator_Address_28> Acesso em 28 de abril de 2019.

ROMANO, V. F. ; DUTRA, M. S. . Introdução à Robótica Industrial. In: Vitor Ferreira Romano. (Org.). **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processos**. 1ed. São Paulo: Edgard Blücher, 2002, v.1, p. 1-19.

RUSSWURM, S. **Industrie 4.0 – from vision to reality**. SIEMENS Industry Sector – Background Information, 2014. Disponível em:
<<http://www.siemens.com/press/pool/de/events/2014/industry/2014-04-hannovermesse/background-indutrie40-e.pdf>>. Acesso em: 27 de abril de 2015.

SZELISKI, Richard. Introduction. In: _____ Computer Vision: Algorithms and Applications. Springer, 2011, p. 1-28.

The MathWorks, Inc. **Image Processing Toolbox User's Guide**. Disponível em:
<https://www.mathworks.com/help/pdf_doc/images/images_tb.pdf>. Acesso em: 18 de abril de 2019.

The MathWorks, Inc. **Simscape User's Guide**. Disponível em:
<https://ww2.mathworks.cn/help/pdf_doc/physmod/simscape/simscape_ug.pdf>. Acesso em: 24 de abril de 2019.

DEVELOPMENT OF A PLATFORM FOR LEARNING IN ROBOTICS BASED ON SCARA ROBOT

Abstract: *The needs to reduce costs and increase efficiency are the reality in the productive sector, being robotic field increasingly fundamental in the industry 4.0. The methodology focused on the design model, Model-Based Design (MBD), and the use of computational simulations are some ways of testing, conducting and developing complex systems. In this scenario, it becomes extreme importance for engineering teaching to explore new teaching instruments. In this way, this work addresses the use of MBD with the development of a Scara robot, creating a platform for teaching. The modeling of the kinematics of the robot, in addition to aspects related to simulation and design using the Model-Based Design methodology with Simulink's Simulink library. At the end of the project, comparatives between the model and the built robot were made, and a graphical interface was created in the App Designer environment.*

Key-words: *Robotic. Model-Based Design. Simscape, MATLAB.*