



Páginas 82-97

Gustavo Copini Decol

Universidade do Vale do Itajaí
gustavo.copini@gmail.com

Paulo Roberto Zem

Universidade do Vale do Itajaí
paulo-f-z@outlook.com

Alejandro Rafael Garcia Ramirez

Universidade do Vale do Itajaí
ramirez@univali.br



PORTO ALEGRE

**RIO GRANDE DO SUL
BRASIL**

Recebido em: agosto de 2020

Aprovado em: março de 2021

Desenvolvimento de um Ambiente Computacional para a Programação, Geração de Movimentos e a Análise Cinemática de um Robô Didático

Computational Environment for Programming, Motion Generation and Kinematic Analysis of a Didactic Robot

Resumo

O Laboratório de Automação e Robótica da universidade possui um kit robótico denominado Robix, formado por um corpo robótico montável, uma unidade controladora e um software de programação chamado Rascal. O software permite a programação dos movimentos, mas apresenta limitações, pois a geração de trajetórias e a análise cinemática não são recursos com os quais o acadêmico consiga interagir de forma didática. Assim, desenvolveu-se um ambiente computacional, usando o Matlab, para a programação, geração de trajetórias e a análise cinemática do Robix. A interface gráfica desenvolvida visou a aplicabilidade dos cálculos, sendo possível modificar e aperfeiçoar os algoritmos pelos acadêmicos. Também, de modo a tornar a solução independente da unidade de controle original do Robix, e visando à flexibilização da conexão com o computador, foi implementada uma interface via USB, que interage com a plataforma Arduino. Os algoritmos e recursos foram testados com resultados preliminares satisfatórios no aspecto educacional, constatando-se o interesse dos alunos através da observação.

Palavras-chave: Cinemática; Geração de Trajetórias; Robótica.

Abstract

The laboratory of automation and robotics at the university has the robotic kit Robix, which is formed by a mountable body structure, a controller unit, and the software Rascal. The software Rascal allows programming of movements but has some limitations because the generation of motion and kinematic analysis are no resources with which academics can interact in a didactic way. Thus, it was developed a computational solution to program the Robix movements, also featuring the kinematics analysis. The graphical user interface was developed in Matlab including the kinematic analysis and the trajectories generation by academics. Also, was implemented an interface with the robot, via USB, using the Arduino MEGA platform to become independent of the original control unit. All algorithms and resources were tested in the classroom, and the preliminary results were satisfactory also verifying the students' interest through observation.

Keywords: Kinematics; Trajectories Generation; Robotics.

1. Introdução

Os robôs são dispositivos poderosos na indústria. Eles são capazes de realizar diversas tarefas e operações, são precisos e não requerem elementos de segurança e de conforto que os seres humanos necessitam (NIKU, 2013).

Os robôs industriais são utilizados para substituir a operação manual em tarefas repetitivas, para executar tarefas que exigem níveis de precisão elevados e para executar tarefas, em geral, que contenham algum nível de insalubridade ou que ofereçam risco ao ser humano (CRUZ, 2007). As aplicações típicas dos robôs industriais incluem pintura, soldagem, montagem, movimentação de cargas, inspeção de produtos e realização de testes, as quais são realizadas com precisão, velocidade e robustez elevadas.

De acordo com UNECE (The United Nations Economic Commission for Europe), citado por Cruz (2007) a robótica era, em termos globais, uma indústria de 8 bilhões de dólares em 2007, com crescimento esperado para 22,61 bilhões em 2010. Como cita a Robotic Industries Association (RIA, 2014), um total de 22.591 robôs, no valor de 1,39 bilhões de dólares, foi enviado para as empresas norte-americanas em 2013, batendo o recorde anterior de 20.328 robôs no valor de 1,29 bilhões dólares enviados em 2012. Segundo RIA (2014) em 2013 cerca de 228 mil robôs estiveram em uso em fábricas dos Estados Unidos, colocando os EUA em segundo lugar, perdendo apenas para o Japão.

Hoje, devido às grandes mudanças tecnológicas e impulsos de aprendizado na área, como machine learning ou aprendizado de máquina, que é uma tecnologia onde os computadores aprendem (HURWITZ; KIRSCH, 2018) e, também, a inteligência artificial, o valor de mercado dos robôs praticamente triplicou nos últimos anos. Combinando os valores de 2019 até 2021, para robôs e seus serviços, estima-se um valor aproximado de 37 bilhões de dólares (SHAW, 2019).

Além de seu impacto econômico, vale ressaltar a importância da robótica nas mais diversas áreas do conhecimento. Segundo a Organização Pan-Americana de Saúde (OPAS) até o dia 22 de julho de 2020 mais de 600 mil mortos já foram confirmados devido ao Covid-19. Desde então diversos projetos e desafios na área de robótica foram lançados para combater à pandemia, desde técnicas de tratamento de pacientes infectados pelo vírus de forma menos invasiva e segura para os profissionais, até a utilização de robôs para localizar e eliminar micro-organismos, em ambientes infectados, por meio de luz ultravioleta (MURRAY, 2020).

No cenário atual, o ambiente escolar ficar atento às diversas mudanças e, dentre elas, pode-se destacar a inclusão das novas tecnologias lançadas ao mercado. As novas tecnologias podem se tornar também recursos pedagógicos lúdicos e auxiliar significativamente no processo de construção do conhecimento (FRANCISCO Jr.; VASQUES; FRANCISCO, 2010). A robótica emerge como uma dessas tecnologias, tendo suas aplicações relacionadas à área em que se pretende atuar, seja focada no robô móvel ou no manipulador industrial.

Nos ambientes de ensino, em particular, é notória a procura por ferramentas inovadoras, destacando-se a área da Robótica Educacional, que possibilita mostrar na prática os conceitos teóricos e competências, como o raciocínio lógico e a interação e resolução de problemas, de forma lúdica para o aluno (COSTA, 2012).

No Dicionário Interativo da Educação Brasileira (2013), o termo Robótica Educacional é utilizado para “caracterizar ambientes de aprendizagem que reúnem materiais de sucata ou kits de montagem compostos por peças diversas, motores e sensores controláveis por computador e softwares”. A tecnologia, quando aliada a conteúdos lúdicos, possibilitaria tornar o ensino-aprendizagem mais atraente e participativo.

Para Alves et. al (2012), o uso das tecnologias educacionais, por exemplo, baseadas na utilização de kits didáticos no processo de ensino da robótica, permite transformar o ambiente educacional, tornando-o mais desafiador, criativo e dinâmico ao viabilizar a construção do conhecimento baseado em experiências práticas. Nesse contexto, a robótica se torna um eixo motivador na construção e exploração de ambientes interativos no processo de ensino e aprendizado (FERNANDES, 2013).

Este trabalho está alinhado à área da Robótica Educacional. A solução adotada consistiu em desenvolver um ambiente computacional didático para o ensino da Robótica, implementado no Matlab, usando um hardware baseado no Arduino Mega. O ambiente define, através de uma programação própria, as tarefas a serem desenvolvidas por um robô didático baseado no sistema Robix (ROBIX, 1992). Do ponto de vista de software, foram implementadas três estratégias para a geração de movimentos e foi realizada a análise da singularidade. Adotou-se a notação de Denavit-Hartenberg (doravante identificada como D-H) para informar a configuração cinemática do robô. Por sua vez, o hardware recebe, por meio da porta USB (Universal Serial Bus), os movimentos que devem ser executados pelos atuadores acoplados ao robô.

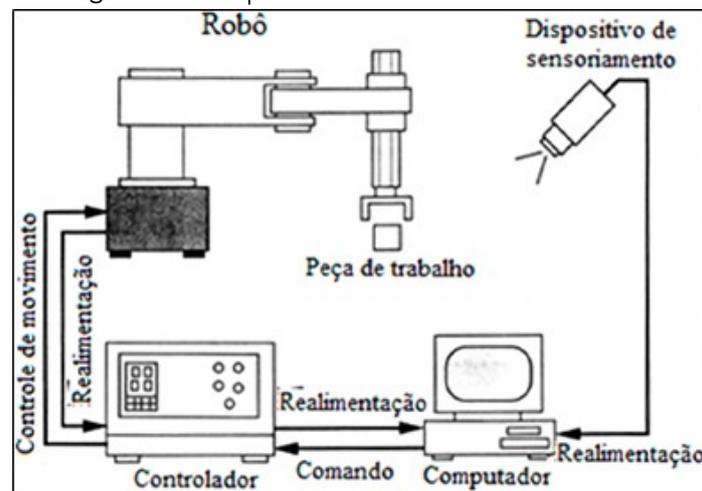
Os benefícios da solução proposta incluem a flexibilização das conexões, tornando possível a utilização de notebook, e a flexibilização da análise matemática, a partir da aplicação dos cálculos cinemáticos e de trajetórias, assim como a flexibilidade da solução adotada possibilita a inclusão de novos algoritmos. O ambiente desenvolvido permite modificações e melhorias, podendo também ser aplicado em um ambiente industrial.

2. Revisão bibliográfica

Um sistema robótico consiste em um manipulador mecânico, um atuador final, um controlador, um computador, e, em algumas aplicações, se incorpora um dispositivo de sensoriamento externo, como ilustrado na Figura 1.

Para (POLONSKII, 1996, p. 14) “Os robôs possuem quatro partes essenciais: uma base fixa, um braço articulado, uma unidade de controle e um dispositivo de programação”. Já Niku (2013) descreve que um robô dispõe dos seguintes elementos integrados: Manipulador; Atuador final; Atuadores; Sensores; Controlador; Processador e Software.

Figura 1 - Componentes de um sistema robótico



Fonte: Adaptado de Tsai (1999)

O manipulador é o corpo principal do robô. Consiste em articulações e outros elementos estruturais. O atuador está ligado à última articulação do manipulador, também chamada de mão do robô e tem como função a execução das tarefas necessárias. Os atuadores são os elementos de força, os ‘músculos’ dos manipuladores, que tem como função movimentar as articulações e, conseqüentemente, os elos do robô (ROSÁRIO, 2005). Os atuadores podem ser servomotores elétricos, motores de passo e atuadores pneumáticos ou atuadores hidráulicos, por exemplo.

Por sua vez, os sensores são elementos que coletam os dados relacionados ao robô, ou se comunicam com o ambiente externo. O controlador do robô usa essas informações para determinar as coordenadas de cada articulação, ou elo, para assim poder determinar como comandará os atuadores para executar o movimento desejado (NIKU, 2013).

O controlador recebe os dados do computador, controla o acionamento dos atuadores e coordena os movimentos a partir da informação de realimentação sensorial. Para Rosário (2005), o processador é o

cérebro do robô, faz os cálculos de movimento de articulações, determina a velocidade de cada articulação para atingir o local e a velocidade no atuador final que se deseja e supervisiona as ações do controlador e dos sensores.

Em relação a softwares para robôs são utilizados três grupos. O primeiro é o sistema operacional que opera o processador. O segundo é um software que calcula os movimentos necessários de cada articulação com base na cinemática do robô. O terceiro é um conjunto de rotinas orientadas para as tarefas a serem realizadas pelo robô, como montagem, carregamento de máquinas e manuseio de materiais (ROSÁRIO, 2005).

3. Trabalhos relacionados

Uma solução pioneira, apresentada em Cardoso (2005), teve como foco o desenvolvimento de um framework para facilitar o desenvolvimento de aplicações na área da robótica. O trabalho descreve métodos de simulação de robôs, em particular, para simular células de trabalho. Nessa proposta, construiu-

se um framework denominado fwWorkCell, utilizando metodologias de padronização de linguagem, a definição modular de processos e a reusabilidade. Com o intuito de testar o framework, foi desenvolvida a aplicação robótica fwAsimov, a qual teve como objetivo comprovar sua eficiência, facilitando, acelerando e padronizando o desenvolvimento de aplicações para robôs industriais.

O sistema fwAsimov, utilizado no trabalho citado previamente, foi criado pelo Núcleo de Informática Educacional do SENAI-RS. É formado por um simulador de célula robotizada não-imersivo (ASIMOV, 1999). O ambiente de simulação do ASIMOV é tridimensional e possui uma série de opções para visualização da célula, possibilitando a programação e o controle do ambiente, junto com a visualização dos movimentos dos robôs.

O VIRbot4u é um simulador educacional, com alusão ao nome "Virtual Robot for You" que usa como referência o robô manipulador ER-4pc, de cinco graus de liberdade. O VIRbot4u foi desenvolvido em linguagem Java, com a utilização do player xj3d browser. Conforme descrição dos autores é uma aplicação multiplataforma de código aberto (HOSS; HOUNSELL; LEAL, 2009). Entre suas características principais, possibilita que o usuário interaja diretamente com o robô através do teclado ou usando o mouse. Permite também a análise cinemática do robô ER-4pc, a inclusão de objetos no cenário e a programação do robô, através de comandos preestabelecidos.

Outro trabalho importante é a plataforma de simulação robótica denominada S-Educ, a qual foi desenvolvida especialmente para aplicação em robótica educacional e surge como uma alternativa frente aos elevados preços dos kits de robótica. Nesse ambiente, o aluno tem acesso a diferentes modelos de robôs e sensores virtuais, que interagem com os outros elementos do ambiente simulado para desempenhar a atividade proposta, tal como o desafio de andar em um labirinto (FERNANDES, 2013).

Com o passar dos anos o foco na Robótica Educacional tornou-se ainda mais visado, aumentando assim a quantidade de projetos relacionados na área. No trabalho de Rafael et al. (2019) foi produzido um simulador denominado Gazebo, juntamente com um framework ROS (Robot Operating System) (ALAJLAN, KOUBÂA, 2016) para facilitar o controle físico do robô AlphaBot, assim como a aquisição de imagens da câmera, o controle dos motores de *pan* e *tilt*, a leitura dos sensores e controle das rodas.

Para aplicações virtuais e mais acessíveis, devido ao alto custo de certos kits robóticos, Maggi et al. (2019) propuseram um simulador WEB 3D para o robô Pololu 3Pi, tornando-o viável em ambientes educacionais e

remotos. Além disso, o simulador foi totalmente validado em experimentos de análise de movimento e medição de tempo das tarefas para que a execução fosse coerente com a robô físico (MAGGI et al., 2019).

O kit Curumim destaca-se entre os poucos kits desenvolvidos no País. Foi produzido pela empresa brasileira XBot de 2008 a 2017, e seu objetivo era trabalhar a lógica de raciocínio, controle e programação do robô. O kit é formado por uma plataforma robótica e um ambiente computacional para programação de robôs com o objetivo de promover o desenvolvimento educacional e aprendizado de conceitos nas áreas de lógica digital, controle, programação e robótica para alunos dos cursos técnicos. O kit Curumin também foi projetado como plataforma para pesquisa e desenvolvimento nas áreas de computação e mecatrônica para alunos de graduação e pós-graduação (XBOT, 2017).

Todos os trabalhos consultados contribuíram, de alguma forma, nesta pesquisa. Por exemplo, o trabalho do Núcleo de Informática Educacional do SENAI-RS apresentou um ambiente de simulação, possibilitando a programação e a visualização dos movimentos do robô (ASIMOV, 1999). Por sua vez, a solução apresentada em Cardoso (2005) apresenta uma estrutura de suporte e descreve métodos de simulação de robôs. Por outro lado, o VIRbot4u (HOSS; HOUNSELL; LEAL, 2009) apresenta elementos de integração hardware-software, de código aberto, possibilitando também a análise cinemática do robô ER-4pc. As características encontradas nestes trabalhos foram elementos chaves na pesquisa aqui apresentada.

Em outra linha de análise, o ambiente de simulação robótica S-Educ, possibilita usar diferentes modelos de robôs e sensores virtuais, que interagem com outros elementos do ambiente simulado para desempenhar a atividade proposta (FERNANDES, 2013). E o trabalho de Maggi et al. (2019) explora as técnicas de simulação em 3D na WEB. Na pesquisa aqui apresentada a animação é realizada em duas dimensões, sendo interessante abordar a simulação 3D em trabalhos futuros. Por sua vez, o trabalho apresentado em Rafael et al. (2019), explora o ROS (Sistema Operacional de Robôs), para facilitar o controle físico do robô AlphaBot. Essa abordagem será usada em trabalhos futuros para a integração com novos sensores.

Analisando os trabalhos relacionados observa-se um diferencial desta proposta, que consiste em explorar a geração de movimentos de um robô, permitindo ao aluno interagir com diversas técnicas de geração de trajetórias. Essa característica não foi encontrada nos trabalhos relacionados.

Os estudos recentes de (KLEVIS et al, 2020) e (YI et al, 2020) valorizam a temática aqui abordada, demonstrando a aplicação recente da geração de trajetórias no projeto dos movimentos de uma prótese transumeral, procurando padrões de movimentação mais natural, e na geração eficiente de perfis de velocidade, aceleração e jerk em robôs manipuladores, respectivamente.

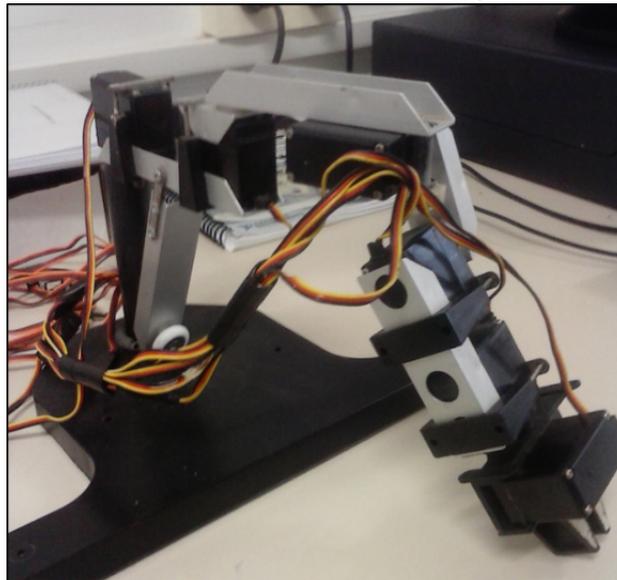
Salienta-se que hoje existem diversos kits robóticos com preços acessíveis e variados para ambientes de aprendizagem, como citam Dav-cev, Koceska e Koceski (2019), dentre eles: o Lego Mindstorm, Boe-Bot, o MiniSkybot, VEX,Fable, Linorobot, Makeblock Ultimate Robot Kit, Speechi, BIOLOID, etc. Embora cada um possui suas características próprias e focos diferentes na aprendizagem, o fato é que todos têm um objetivo em comum, que é possibilitar a maior integração dos alunos com a robótica.

4. Materiais e métodos

Do ponto de vista de hardware, nesta pesquisa, foi utilizada a Plataforma Arduino (ALVES, 2012), em particular, a placa Arduino MEGA, que possibilitou o acionamento dos seis servomotores usados na aplicação. A montagem física do robô foi implementada usando o kit Robix usando a configuração ilustrada na Figura 2.

O Robix é uma solução integrada de hardware e software, que possibilita a aplicação da robótica de forma lúdica. Do ponto de vista mecânico, o sistema é formado por articulações rotacionais básicas, elos e servomotores, que podem ser montados em diversas configurações. Do ponto de vista de programação, utiliza a linguagem Rascal que implementa comandos básicos de um robô. Os desenvolvedores disponibilizam também uma interface para interagir com o sistema (ROBIX, 1992). Na página do projeto podem ser encontrados vídeos de diferentes projetos e as técnicas de montagem, assim como os scripts de programação.

Figura 2 – O Robix e a configuração definida para este trabalho

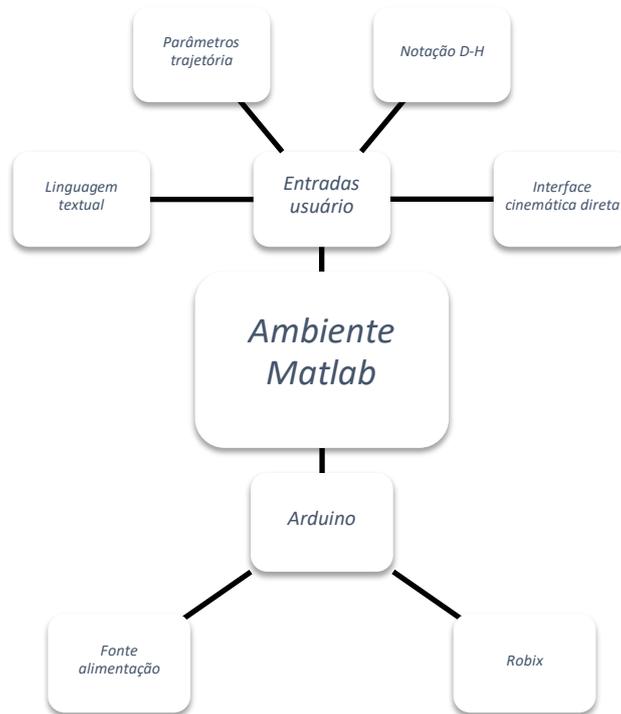


Fonte: Autores

A Figura 3 ilustra o fluxo de informações no sistema de forma básica. Inicialmente, os valores de entrada para o funcionamento devem ser informados através da interface principal do sistema, (que foi concebida usando a ferramenta GUIDE), tais como os parâmetros de Denavit-Hartenberg, a escolha do método de

geração de trajetórias e os parâmetros para o método selecionado. Estas informações são processadas pelo ambiente, resultando em comandos de movimentação, os quais são enviados para os eixos do manipulador com o auxílio do processador embarcado na placa Arduino MEGA.

Figura 3 - Fluxo de informações no sistema

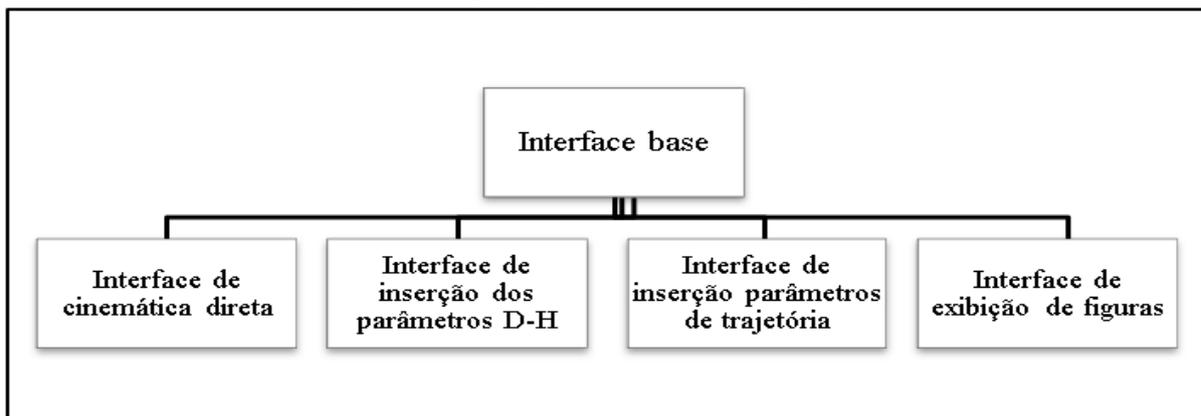


Fonte: Autores

As interfaces gráficas que fazem parte do sistema desenvolvido são divididas em quatro segmentos, interligados e acessados por uma interface

base, conforme ilustra a Figura 4. Para cada segmento da interface gráfica foi definida uma funcionalidade, descrita a seguir.

Figura 4 - Organograma das interfaces gráficas

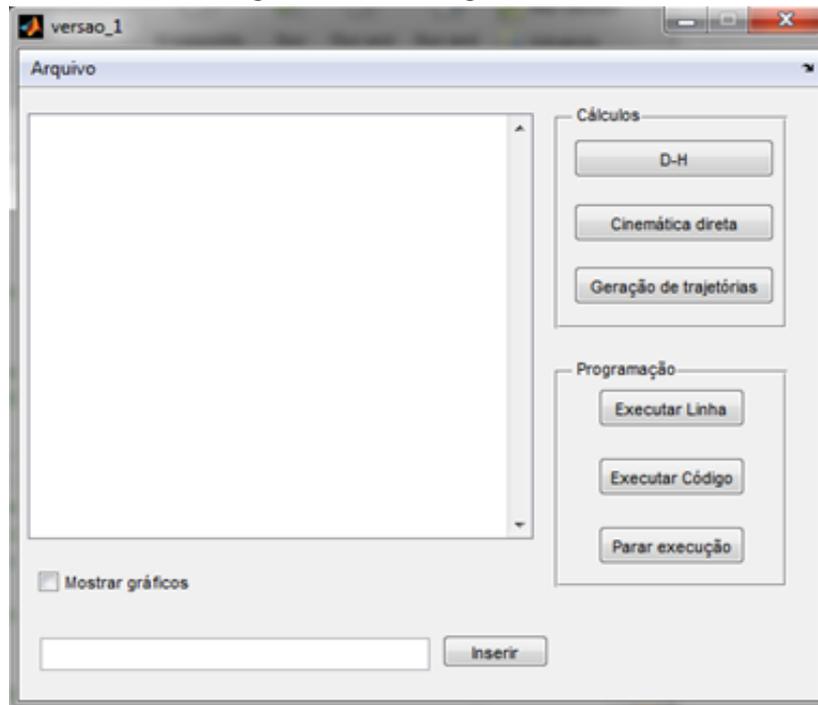


Fonte: Autores

A Interface base, ilustrada na Figura 5, possui como entradas a programação em linguagem textual. Suas funções são: receber os dados provenientes do usuário,

executar as rotinas e linhas de programação e permitir o acesso às outras interfaces.

Figura 5 - Interface gráfica base

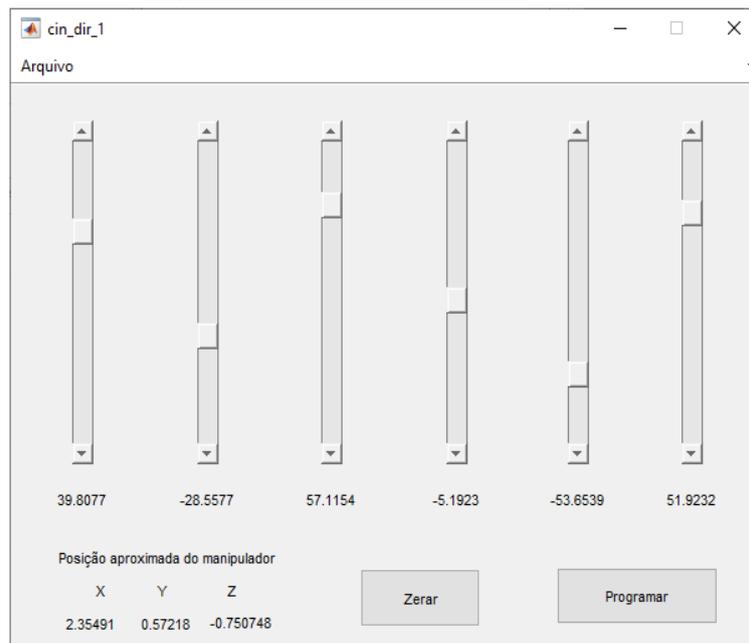


Fonte: Autores

A interface de cinemática direta possui como entradas a posição atual dos servomotores, definidas através de barra de rolagens acionada pelo usuário, como ilustra a Figura 6. Suas funções são: comandar o acionamento angular de cada servomotor, chamar o

algoritmo de geração de trajetória, executar o algoritmo de cinemática direta, possibilitar a programação textual dos movimentos e mostrar a posição da extremidade do robô.

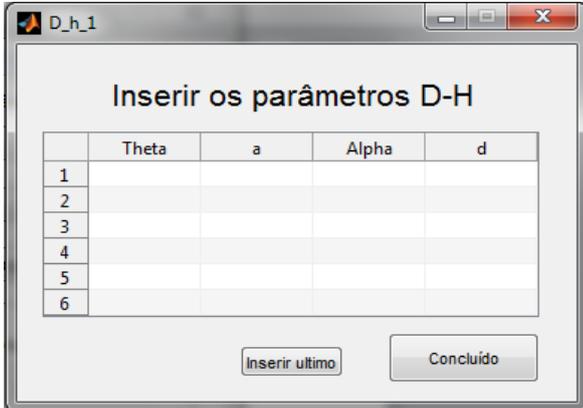
Figura 6 -Interface da cinemática direta



Fonte: Autores

A interface para a inserção dos parâmetros D-H, ilustrada na Figura 7, possui como entradas os parâmetros da representação D-H (inseridos pelo usuário através de uma tabela). Sua função é gravar os parâmetros da notação D-H, para posterior uso no cálculo cinemático direto e inverso.

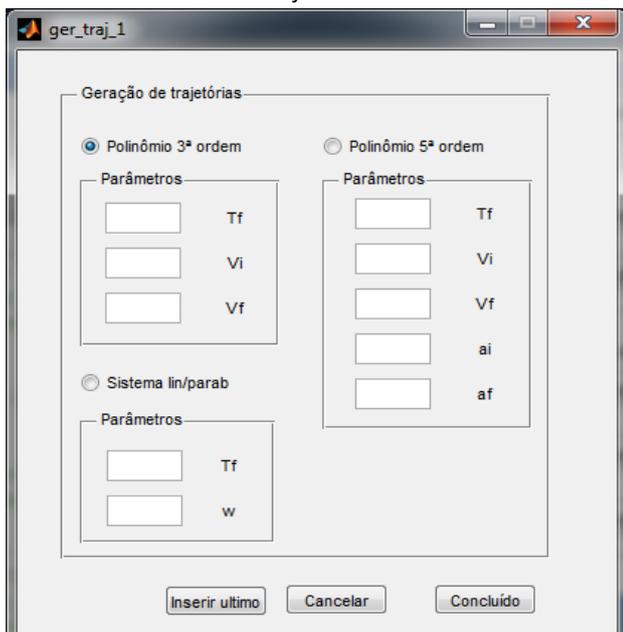
Figura 7 - Interface dos parâmetros D-H



Fonte: Autores

A interface para a inserção dos parâmetros de trajetória, ilustrada na Figura 8, possui como entradas: a definição, pelo usuário, do tipo de trajetória, os parâmetros da trajetória escolhida (V_i , V_f , a_i , a_f , w) e o tempo de realização do movimento (T_f). Sua função é passar estas informações ao algoritmo responsável pela geração dos movimentos. A geração de trajetórias é realizada por polinômios de terceiro grau, polinômios de quinto grau e por segmentos lineares combinados com segmentos parabólicos.

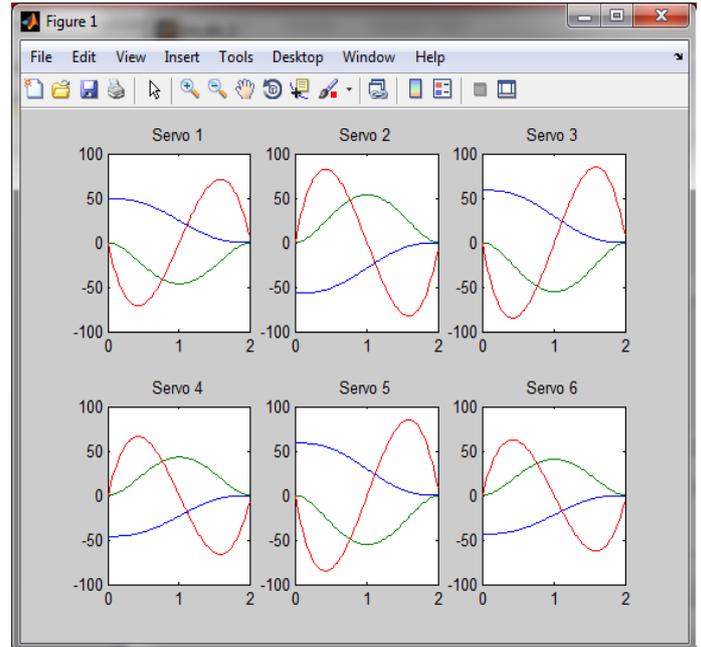
Figura 8 - Interface de inserção dos parâmetros de trajetória



Fonte: Autores

A interface de exibição de figuras possui como entradas o perfil das trajetórias de cada servomotor, fornecidas pelo algoritmo de geração de trajetória. A função desta interface é exibir os gráficos de trajetória dos seis servos do Robix, mostrando as posições, velocidades e acelerações, como ilustra a Figura 9. Na Figura 9 se observa o resultado usando a estratégia polinomial de quinta ordem considerando as seis articulações.

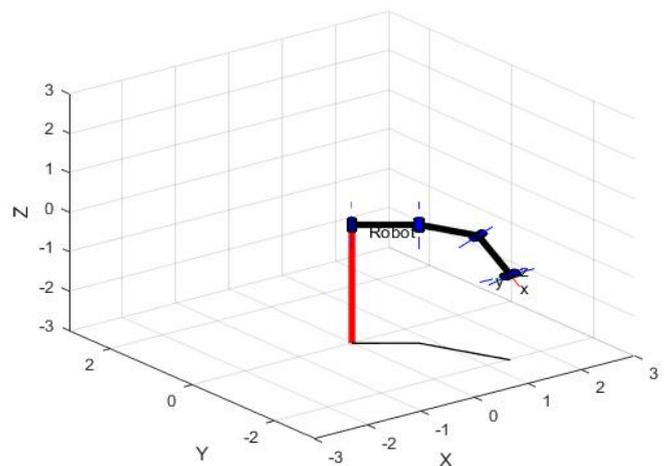
Figura 9 - Interface de exibição de figuras



Fonte: Autores

Também é possível observar animar os movimentos do robô, antes de serem executados no robô físico, como ilustra a Figura 10. A simulação é realizada usando um toolbox desenvolvido para o Matlab (CORKE, 2017).

Figura 10 - Animação dos movimentos



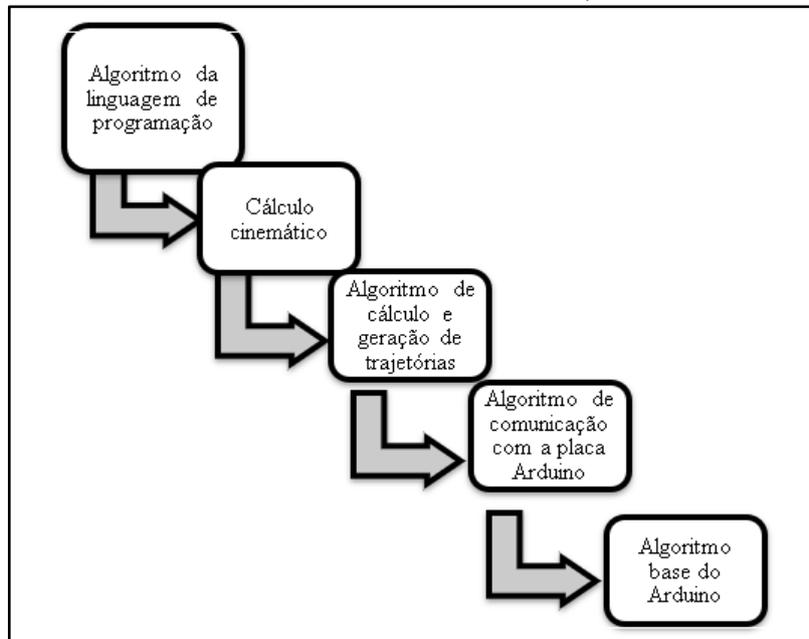
Fonte: Autores

5. Software

O ambiente desenvolvido implementa algoritmos que realizam os cálculos relativos à sua função, atuando nas interfaces gráficas e nos servomotores, de modo a

exibir os resultados dos movimentos e executar as tarefas desejadas. Os algoritmos são divididos em segmentos que possuem uma ordem de execução definida, como ilustra a Figura 11.

Figura 11 - Fluxograma dos algoritmos implementados



Fonte: Autores

A linguagem de programação adotada para o robô é similar a linguagem V+, que é uma linguagem de programação de primeira geração, que foi vendida comercialmente juntamente com os manipuladores robóticos. Em particular, foram implementadas as seguintes sintaxes:

```
MOVE 90,-90, 80,-80,0,10
```

MOVE indica a utilização da cinemática direta, movendo diretamente cada articulação de forma absoluta. No exemplo, a primeira articulação seria acionada até alcançar o ângulo de 90 graus, a segunda alcançaria o ângulo de -90 graus, a terceira alcançaria o ângulo de 80 graus, e assim sucessivamente para as outras articulações.

```
MOVES X100 Y200 Z50
```

MOVES indica a utilização da cinemática inversa, informando a posição do efetuador no espaço cartesiano. No exemplo, o efetuador deverá ser deslocado para alcançar 100 mm no eixo X, 200 mm no eixo Y e 50 mm no eixo Z.

O algoritmo, para efetuar o cálculo cinemático direto, possui como entradas os parâmetros da representação D-H: a posição inicial e a posição final de cada articulação. Tem como função realizar o cálculo cinemático direto, determinando a posição do efetuador a partir dos parâmetros cinemáticos.

Por sua vez, o algoritmo de cálculo cinemático inverso possui como entradas os parâmetros da representação D-H, e a posição desejada para o efetuador. Ele tem como função determinar as coordenadas de cada articulação usando um procedimento de cálculo iterativo, visando que o efetuador alcance a posição desejada em um dado instante de tempo.

O algoritmo de geração de trajetórias possui como entradas a posição inicial e final do movimento de cada articulação, o tipo de trajetória a ser gerada, a duração do movimento, assim como as velocidades iniciais e finais do movimento, as acelerações iniciais e finais do movimento, e as velocidades ou acelerações máximas, dependendo da estratégia selecionada para a geração de trajetória.

O algoritmo de comunicação com a placa Arduino MEGA possui como entradas as coordenadas angulares desejadas para cada articulação, as quais são

informadas pelo algoritmo de geração de trajetória, assim como o tempo total da execução do movimento. Sua função é enviar, através da porta USB, estas informações ao processador.

O algoritmo base do Arduino Mega (firmware), desenvolvido usando a IDE do Arduino, possui como entradas as coordenadas da trajetória de cada servomotor e o tempo de realização do movimento. Sua função é receber estas informações e transformá-las em

pulsos que caracterizam a Modulação por Largura de Pulso (PWM), possibilitando que os servomotores, executem o movimento desejado.

Para a montagem da estrutura do robô, como ilustrado na Figura 2, foi utilizada como referência o robô PUMA 562 (NIKU, 2013). Para a configuração selecionada foram definidos os parâmetros de Denavit-Hartenberg, como mostrado na Tabela 1.

Tabela 1 - Parâmetros de Denavit-Hartenberg para a configuração ilustrada.

θ	d	a	α
θ_1	0	95mm	0
θ_2	15mm	65mm	-90°
θ_3	0	38mm	0
θ_4	10mm	0	-90°
θ_5	120mm	0	0

Fonte: Autores

O sexto movimento foi implementado através de uma garra de dois dedos, que realiza o movimento de abrir e fechar, para pegar e liberar objetos.

A representação (standard) de Denavit-Hartenberg resulta na obtenção de uma matriz de transformação homogênea, de ordem 4, a qual reflete o sistema de coordenadas de um elo (i) em relação ao sistema de coordenadas do elo anterior ($i-1$).

A Equação 1 ilustra a representação da notação de Denavit-Hartenberg em uma matriz de transformação homogênea, São parâmetros desta representação: q , d , $a(r)$, α (NIKU, 2013).

Equação 1 - Matriz de transformação homogênea usada na notação de Denavit-Hartenberg.

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fonte: Rosário (2005)

A partir da matriz de transformação homogênea resultante, para cada articulação, podem ser extraídas

as posições e orientações relativas do sistema coordenado (i) em relação ao sistema ($i-1$) (ROSÁRIO, 2005). E aplicando-se sucessivas transformações entre o sistema de coordenadas da base do robô e o sistema de coordenadas do efetuador (geradas pelo produto das matrizes, nessa ordem), podem ser obtidas as coordenadas do elemento terminal do robô (garra ou ferramenta), expressas matematicamente em relação ao sistema de coordenadas fixo na sua base, como ilustra a Equação 2, onde n representa a quantidade de articulações, ou movimentos do robô.

Equação 2 – Procedimento para obter a matriz do elemento terminal ou efetuador

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \dots {}^{n-1}T_n.$$

Fonte: Autores

A Tabela 2 ilustra o índice de siglas, de modo a facilitar a identificação de parâmetros citados no texto, assim como sua função dentro do software.

Tabela 2 – Glossário de termos

Sigla	Significado
D-H	Representação Denavit-Hartenberg
Theta (θ)	Parâmetro angular da representação D-H
a (r)	Parâmetro linear da representação D-H
Alpha (α)	Parâmetro angular da representação D-H
d	Parâmetro linear da representação D-H
V_i	Velocidade inicial do movimento
V_f	Velocidade final do movimento
a_i	Aceleração inicial do movimento
a_f	Aceleração final do movimento
w	Velocidade máxima do movimento
T_f	Tempo para a realização do movimento

Fonte: Autores

6. Resultados

Para validar os recursos oferecidos pelo ambiente computacional desenvolvido foram inseridas linhas de programação na interface base (ver Tabelas 3 e 4),

usando a cinemática direta e a cinemática inversa, respectivamente, considerando o Robix na configuração escolhida e os parâmetros de D-H ilustrados na Tabela 1.

Tabela 3 - Linhas de comando.

Código	Posição medida (mm)	Posição desejada	Erro (mm)
MOVE 0,0,0,0,0,0	X 318 Y10.09 Z14,91	X320 Y10 Z15	X2 Y0.09 Z0.09
MOVE-45,10,20,-	X252.26 Y-182.42 Z-	X250 Y-180 Z-18	X2.26 Y2.42
MOVE0,0,0,0,0,0	X318 Y10.09 Z14.91	X320 Y10 Z15	X2 Y0.09 Z0.09
MOVE 45,-45,-10,-	X282.39 Y77.23	X280 Y75 Z62	X2.39 Y2.23

Fonte: Autores

As posições escolhidas encontram-se dentro da faixa de operação dos servomotores. Foram analisadas a posição do efetuator obtida pela cinemática direta e a posição real do robô, que foi medida utilizando-se de uma trena, e comparando-as.

Analisando os resultados da Tabela 3, percebe-se que o cálculo cinemático direto possui erros de pequena magnitude, ao comparar a posição desejada

com o resultado da medição. Os erros são atribuídos às folgas nas articulações do manipulador, causando uma propagação dessa folga no efetuator. O maior erro encontrado foi de 2.39 mm, no eixo X do quarto experimento.

Em um segundo momento foi analisada à cinemática inversa do manipulador, solicitando-lhe alcançar as posições usadas no teste anterior, mas

utilizando o comando MOVES. Assim foi comparada a posição exibida pelo cálculo cinemático direto com as posições medidas neste experimento. A cinemática inversa foi determinada usando o método recursivo descrito em (CRUZ, 2007).

Observa-se na Tabela 4 que a cinemática inversa não foi resolvida em duas das configurações usadas, pois onde se pretendia posicionar o efetuador havia articulações alinhadas, caracterizando uma

singularidade. Nesses casos, o determinante da matriz jacobiana é zero, não sendo possível inverter essa matriz. As linhas dois e quatro foram executadas normalmente. Pode-se observar que, quando resolvida, a cinemática inversa converge para um resultado próximo do desejado. O maior erro mensurado neste experimento foi de 0.47 mm, no eixo Y do segundo experimento.

Tabela 4 - Resultados da análise cinemática inversa

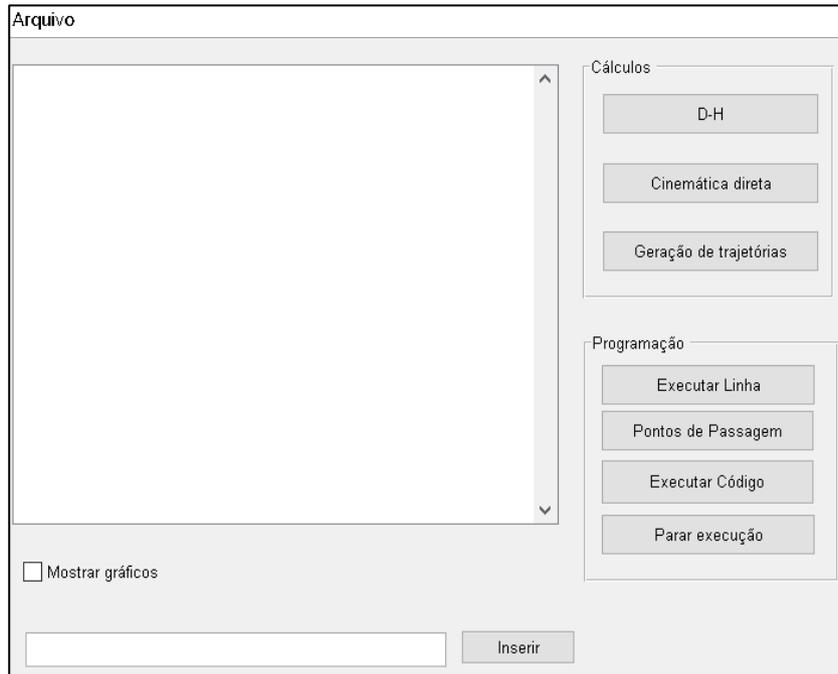
Código	Posição medida (mm)	Posição desejada	Erro (mm)
MOVES X320 Y10 Z15	Não resolveu	X320 Y10 Z15	-
MOVESX 250Y-180 Z-	X249.76 Y-179.53 Z-	X250 Y-180 Z-18	X0.24 Y0.47
MOVES X320 Y10 Z15	Não resolveu	X320 Y10 Z15	-
MOVES X280 Y75 Z62	X280.04 Y74.53	X280 Y75 Z62	X0.04 Y0.47

Fonte: Autores

Posteriormente, a interface gráfica base foi modificada, como ilustra a Figura 12, para incorporar

uma técnica de interpolação por pontos de passagem (NIKU, 2013).

Figura 12 – Interface base modificada

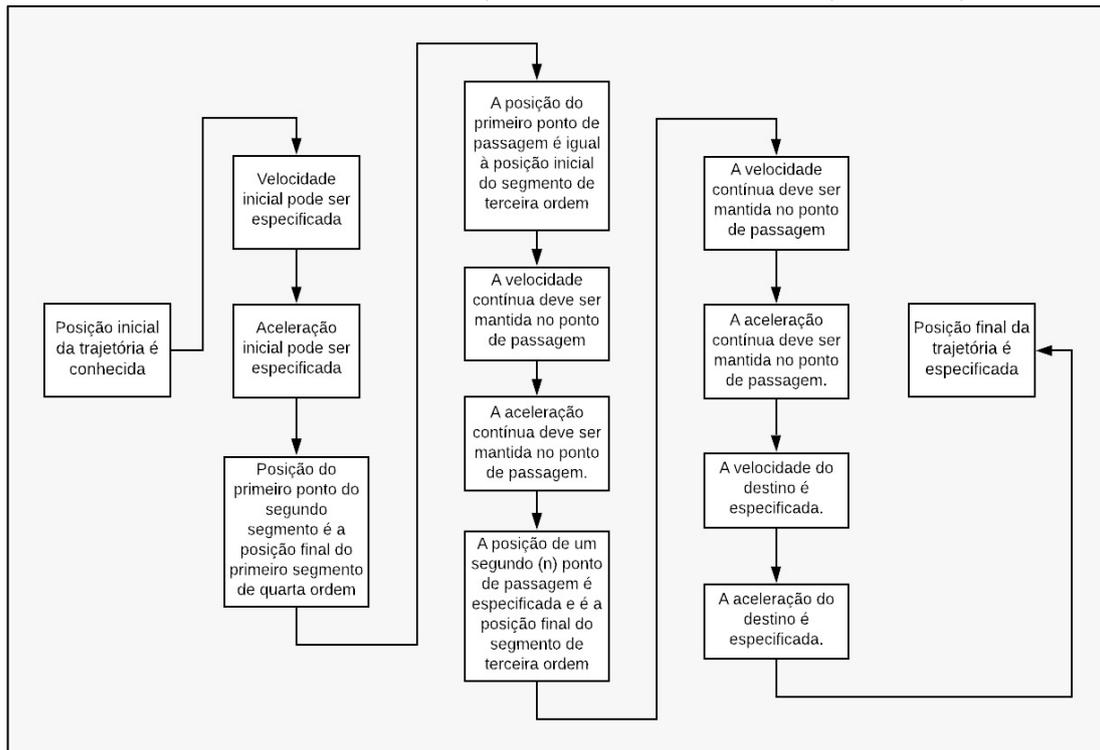


Fonte: Autores

Esta técnica permite que o movimento do manipulador seja contínuo, sem ter que passar pelos estados de repouso (parada) a cada trecho do percurso. A través dos pontos de passagem, são combinados o polinômio cúbico, de quarta ordem e de quinta ordem,

definindo-se os parâmetros iniciais e obtendo os coeficientes que possibilitam a continuidade nos movimentos. A Figura 13 ilustra o diagrama de blocos da implementação, como descrito em (NIKU, 2013).

Figura 13 - Diagrama de blocos da implementação da técnica dos pontos de passagem

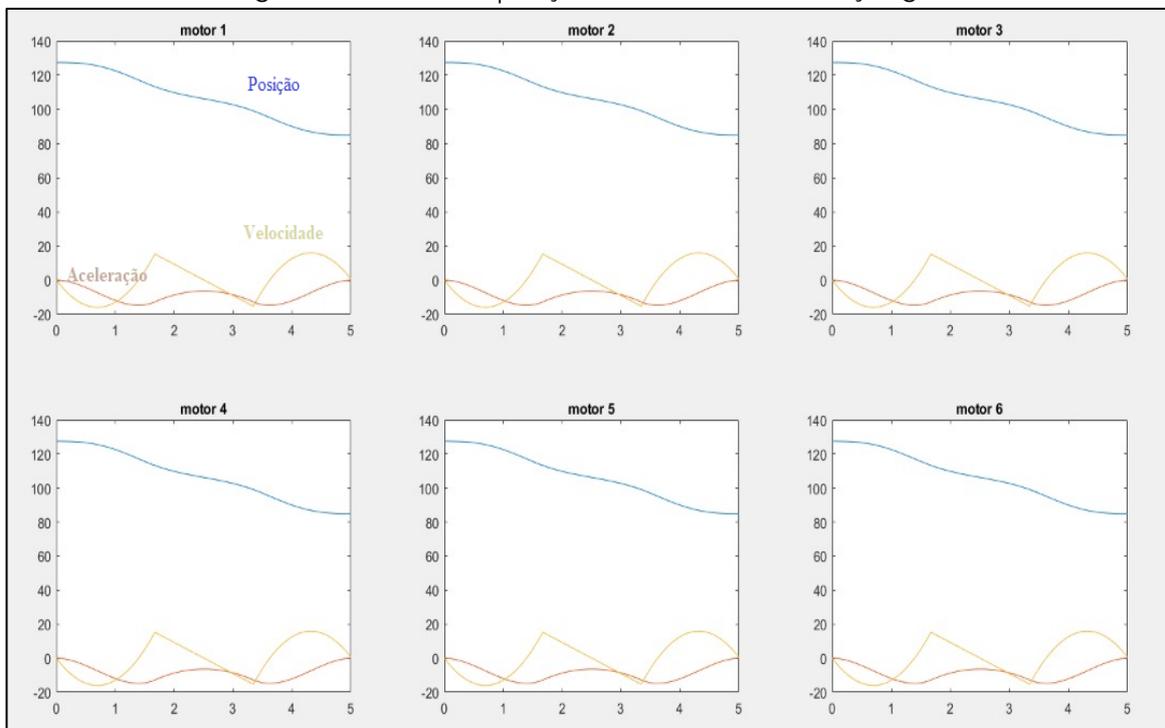


Fonte: Niku (2013)

Na Figura 14 são mostradas as curvas de posição, velocidade e aceleração determinadas para cada um

dos servomotores da estrutura usando a estratégia dos pontos de passagem.

Figura 14 - Curvas de posição, velocidade e aceleração geradas



Fonte: Autores

Analisando a Figura 14, observa-se que a posição final do primeiro segmento da trajetória se torna a posição inicial do segmento seguinte. De forma similar acontece com as velocidades e acelerações, e assim, sucessivamente, nos seguintes segmentos do movimento planejado.

A implementação prática do movimento consistiu em uma atividade de Pick and Place (Pegar e Largar), que é uma tarefa comum realizada por robôs na indústria (NIKU, 2013). Através das definições das

posições, velocidades e acelerações foi possível traçar um caminho contínuo ao longo do tempo, através dos pontos de passagem do manipulador no espaço de trabalho, sem paradas de repouso.

A Tabela 5 apresenta a comparação entre as funcionalidades do Rascal, que é o software de programação do Robix, e do ambiente desenvolvido nesta pesquisa, evidenciando as melhorias que foram obtidas através da implementação da solução aqui apresentada.

Tabela 5 - Comparativo entre Rascal e o ambiente desenvolvido

Característica	Rascal	Ambiente desenvolvido
Geração de trajetória	Apenas a técnica trapezoidal.	Polinômio de terceira, quinta ordem, técnica trapezoidal e técnica de interpolação por pontos de passagem.
Cálculo cinemático direto e inverso	Possui, mas não é possível modificar os algoritmos.	Calcula a cinemática direta e inversa utilizando a metodologia de D-H, e permite a análise da singularidade.
Incorporação de novas funcionalidades	Não contempla.	Os códigos disponibilizados em Matlab facilitam a inclusão de novas funcionalidades.

Fonte: Autores

O código deste projeto está disponível em:
<https://github.com/gutoodc2/LaboratorioRobotica>

7. Conclusões

No presente trabalho foi descrito um ambiente computacional desenvolvido para o Matlab, descrevendo a programação, geração de movimentos e a análise cinemática de um robô didático disponível no Laboratório de Robótica da universidade. As etapas do desenvolvimento visaram definir o escopo da solução proposta, detalhando suas funcionalidades e, conseqüentemente, o processo de elaboração do sistema.

Para a montagem do Robix foi escolhida uma configuração inspirada no robô PUMA 562. Do ponto de vista do hardware, foi usado o Arduino MEGA no lugar da placa controladora original do Robix. Isto permitiu customizar o acionamento dos servos e flexibilizar a programação usando o Matlab.

Do ponto de vista da programação, foram implementados os algoritmos para efetuar o cálculo cinemático direto e inverso, assim como para gerar o perfil de posição, velocidade e aceleração dos movimentos usando diversas estratégias. Buscou-se que as interfaces gráficas criadas fossem intuitivas, flexibilizando a interação com o usuário.

Nos testes realizados verificou-se que, na análise da cinemática, houve pequenos erros de posicionamento, devido a folgas existentes nas articulações, as quais foram identificadas através da inspeção no manipulador robótico. Na cinemática inversa foi possível validar as soluções possíveis e àquelas em que a singularidade deveria ser tratada. Estes resultados são importantes e são explorados nas atividades práticas.

Trabalhos futuros estarão direcionados a abordar o estudo dos aspectos de usabilidade da solução, assim

como incorporar novas funcionalidades no ambiente desenvolvido, tal como a análise da orientação do efetuador, a inclusão de novos sensores e a integração com o ROS (Sistema Operacional de Robôs). Cabe salientar que o sistema descrito é usado nas aulas práticas da disciplina de Introdução à Robótica dos cursos das Engenharias da universidade há seis meses, propiciando a aproximação da teoria e a prática dos conceitos envolvidos, almejando beneficiar a aprendizagem. Salienta-se que embora ainda não foi realizada uma avaliação formal da aprendizagem, foi constatado o interesse dos alunos sobre a temática através de observação. Os resultados aqui apresentados podem ser considerados iniciais, precisando serem validados pelos estudantes por um período maior (dois semestres), visando também a implementação de melhorias no sistema.

Agradecimentos

Agradecemos a Universidade do Vale de Itajaí (UNIVALI) por criar as condições necessárias para a realização desta pesquisa e ao Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação PIBITI, do CNPq.

Referências

ALAJLAN, M., KOUBÂA, A. Writing Global Path Planners Plugins in ROS: A Tutorial. In *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Ed. Anis Koubâa. 2016. DOI <https://doi.org/10.1007/978-3-319-26054-9>

ALVES, Rafael Machado et al. Uso do Hardware Livre Arduino em Ambientes de Ensino-aprendizagem. *Jornada de Atualização em Informática na Educação*, v. 1, n. 1, p. 162-187, 2012.

ASIMOV Simulador de Robótica e Células de Produção. Manual do Usuário. SENAI-RS, Porto Alegre, 1999.

CARDOSO, Fernando Henrique. fwWorkCell: framework para a construção de simuladores didáticos de células de trabalho. 2005. Dissertação (mestrado) – Programa de pós-graduação em Computação, Universidade Federal do Rio Grande do Sul. 2005.

CORKE, P. Robotics Vision and Control. Fundamental Algorithms In MATLAB®. Second Edition, 2017. 580 p.

COSTA, Maria Teresa Ferreira. O uso de recursos educativos abertos (rea): benefícios para alunos e professores. O repositório de acesso aberto de Portugal | Use of open educational resources (oer): benefits for students and teachers. *The Portuguese open access repository*. Liinc em Revista, v.8, n.2, setembro, 2012, Rio de Janeiro, p. 402-412.

CRUZ, Daniel Fontan Maia da. Implementação da cinemática inversa de robôs redundantes operando em ambientes confinados no projeto roboturb. 2007. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/90064/245604.pdf?sequence=>. Acesso em: 22 jul. 2020.

DAVCEV, K.; KOCESKA, N.; KOCESKI, S. A review of robotic kits used for education purposes. In *X International Conference of Information Technology and Development of Education ITRO*, Zrenjanin, Republic of Serbia, p. 152-155, 2019.

DIEB. Dicionário Interativo da Educação Brasileira, 2013. Disponível em: <https://www.educabrasil.com.br/?q=Rob%C3%B3tica+Educacional>. Acesso em: 12 jul. 2020.

FERNANDES, Carla da Costa. S-Educ: Um Simulador de Ambiente de Robótica Educacional em Plataforma Virtual. 2013. Tese (Doutorado) – Programa de Pós-graduação em Engenharia Elétrica e Computação, Universidade Federal do Rio Grande Do Norte, 2013.

FRANCISCO Jr, Nacim Miguel; VASQUES, Carla K.; FRANCISCO, Thiago Henrique Almino. Robótica educacional e a produção científica na base de dados da capes. *Revista Electrónica de Investigación y Docencia (REID)*, n. 4, 2010. p. 35-53.

HOSS, A.; HOUNSELL, M. da Silva; LEAL, A. B. VirBot4u: Um Simulador de Robô usando X3D. In *proceeding of: Simpósio de Computação Aplicada – SCA*, p. 1- 11. 2009.

HURWITZ, Judith; KIRSCH, Daniel. *Machine Learning for Dummies*. IBM Limited Edition. John Wiley & Sons, Inc. 2018.

KLEVIS, Aliaj et al. Replicating dynamic humerus motion using an industrial robot. *PLoS ONE*. 2020, Vol. 15 Issue 11, p1-23. 23p.

MAGGI, L. O. et al. 3dipi: An open-source web-based 3d simulator for pololu's 3pi platform. In IEEE. 21st Symposium on Virtual and Augmented Reality (SVR), p.52–58. 2019.

MURRAY, A. Coronavírus: os robôs usados para eliminar vírus em hospitais. Disponível em:

<<https://www.bbc.com/portuguese/geral-52038851>>. Acesso em: 27 julho.2020

NIKU, S. B. Introdução à Robótica. LTC, 2013. p. 404.

POLONSKII, Mikhail M. Introdução a robótica e mecatrônica. Caxias do Sul: EducS, 1996. p. 142.

RAFAEL, A. et al. Development of an alphabot2 simulator for rpi camera and infrared sensors. In Iberian Robotics conference, p. 502–514. Springer, 2019.

RIA (Robotic Industries Association). North American Robotics Shipments Grow in 2013 While New Orders Contract.

Disponível em:< http://www.robotics.org/content-detail.cfm/Industrial-Robotics-News/North-American-Robotics-Shipments-Grow-in-2013-While-New-Orders-Contract/content_id/4648> Acesso em: 09 jun. 2014

ROBIX. Rascal™ Robot Construction Set. 1992. Disponível em:< <http://www.robix.com/default.html> >. Acesso em: 15 jan. 2021.

ROSÁRIO, João Maurício. Princípios de Mecatrônica. 2.ed. Pearson Prentice Hall: São Paulo, 2005. p. 356.

SHAW, Keith. World Robotics Report: Global Sales of Robots Hit \$16.5B in 2018. Robotics Business Review.

September 18, 2019. Disponível em: < <https://www.roboticsbusinessreview.com/research/world-robotics-report-global-sales-of-robots-hit-16-5b-in-2018/>>. Acesso em: 14 janeiro 2021.

TSAI, Lung-wen. Robot analysis: the mechanics of serial and parallel manipulators. Nova York: John Wiley & Sons, 1999. p. 520.

XBOT. 2017. Disponível em: <<http://www.xbot.com.br/educacional/curumim/>>. Acesso em: 05 maio 2020.

YI, Fang et al. An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints. In Mechanism and Machine Theory. 2020. Elsevier Ltd. DOI: 10.1016/j.mechmachtheory.2020.103957