

# Módulo Gerador de Trajetórias Aplicado no Eixo de um Servomecanismo

Gustavo C. Decol  
Laboratório de Robótica  
Universidade do Vale do Itajaí  
Itajaí, Santa Catarina, Brasil  
gustavo.copini@gmail.com

Alejandro R. G. Ramirez  
Laboratório de Robótica  
Universidade do Vale do Itajaí  
Itajaí, Santa Catarina, Brasil  
ramirez@edu.univali.br

## ABSTRACT

Several sectors, and especially the industrial environment, have become increasingly automated. Their processes end up need more sophisticated applications and, through a feedback control system, they seek to obtain a better performance, as in the case of servomechanisms. In addition, in certain applications, it is necessary to generate movements to provide a desired behavior in the mechanical axis, simplifying the operation of the controller. In this sense, trajectory generating algorithms assist in the configuration and implementation of the position, speed and acceleration profile with unique characteristics for each situation. Considering the relevance of this theme, this work describes the development of a trajectory generating module, responsible for the generation of the movements of a servomechanism. The trajectory generation algorithms, known as cubic polynomial, fifth order polynomial and trapezoidal, were studied and implemented. A hardware based on the Arduino Mega, a power driver and a servomotor, was designed to validate these algorithms. The position, speed and acceleration profiles generated for a one axis servosystem and the feedback obtained by an encoder, allowed to validate the proposed strategy. It is hoped that the designed platform would become an important resource in a discipline of Robotics.

## KEYWORDS

Robótica, Servomecanismo, Trajetória, Controle

## 1 INTRODUÇÃO

Um servomecanismo é um dispositivo que, através de sistema de controle realimentado, procura responder ao desempenho desejado, a partir da entrada de referência especificada [9]. As aplicações e as necessidades deste tipo de sistemas são diversas. Com eles pode-se mover equipamentos pesados, acionar esteiras, rotacionar eixos de motores com precisão, dentre outros usos [10]. Muitos dos conceitos aqui apresentados se aplicam em diversos tipos de sistemas, tais como carros autônomos e veículos aéreos não tripulados, por exemplo.

A modo de exemplo de aplicação de um sistema servomecânico, pode-se citar um simulador de voo. O projeto desse tipo de sistemas abrange os mais diversos aspectos, desde a concepção da estrutura mecânica do simulador, sua modelagem, a geração dos movimentos, o controle realimentado e interação com softwares através de protocolos de comunicação [13].

Outro exemplo de aplicação é o controle da velocidade de uma esteira em um processo industrial. Neste caso requer-se um aciona-

mento preciso do movimento, já que, variações ocasionarão divergências nos dispositivos que atuam ou operem sob ela. Em aplicações mais críticas, pode-se citar manipuladores utilizados em operações médicas [15].

O planejamento de trajetórias é um subconjunto de um problema maior, que é o planejamento do movimento [9]. A hierarquia típica de planejamento de movimento inicia com o planejamento das tarefas, onde se definem os objetivos de alto nível, tal como “pegar o objeto que se encontra em determinada posição espacial”. Na sequência, planeja-se o caminho, ou percurso, formado pelos conjuntos de coordenadas (definidas no espaço operacional ou nas juntas) entre um ponto inicial e o destino. Um caminho geralmente é formado por um conjunto de pontos de passagem conectados. Por sua vez, o planejamento de trajetória se encarrega de gerar o perfil do caminho de acordo com as restrições de posição, velocidade e aceleração. Por último, o seguimento de trajetória possibilita que um sistema de controle realimentado execute a trajetória de forma precisa.

Este trabalho é motivado pela necessidade e as implicações desta temática, criando uma plataforma de hardware e software, na qual diferentes técnicas para a geração de trajetórias podem ser avaliadas em uma junta rotativa. Nas seções a seguir são descritas a problemática, o desenvolvimento e os resultados obtidos. Por último, na seção 6 são tecidas as conclusões deste trabalho.

## 2 TRABALHOS CORRELATOS

A robótica se originou com o manipulador serial e, por muito tempo, foi o único tipo de estrutura existente na indústria. De modo que as técnicas de análise cinemática e dinâmica de robôs manipuladores foram inicialmente desenvolvidas especificamente para essa classe de estruturas [9]. Posteriormente, surgem os manipuladores paralelos, precisos e podendo suportar cargas elevadas, sendo empregados nas últimas décadas em diversas aplicações [14].

No estudo prévio realizado, e apresentado em [13], a geração de movimentos na plataforma paralela, simuladora de voo, 6UDESC foi realizada a partir de um módulo CNC (Controle Numérico Computadorizado) que utiliza uma técnica conhecida como trajetória trapezoidal, onde os movimentos são formados por segmentos lineares combinados com parábolas e a velocidade segue um padrão trapezoidal [9]. Embora simples e com baixo custo computacional, esta técnica apresentou como desvantagem a descontinuidade ocasionada nas acelerações, a qual se acentua nos momentos em que há inversão do movimento e também nos momentos de início e parada.

Essas descontinuidades se manifestaram na estrutura através de vibrações. Para contornar essa situação, os movimentos de início e parada foram suavizados, incluindo um tempo maior nessas transições.

Além do método trapezoidal, existem outros, tal como o polinomial de terceira ordem e o de quinta ordem [1, 9]. Os polinômios também podem ser combinados, tal como descrito em [6], que usa polinômios de terceira e quinta ordem. Também podem ser utilizados polinômios de ordem superior, como em [4], em troca de um maior custo computacional, ou, técnicas mais complexas, tal como a Double S Velocity Profile (DSVP), implementada em [15].

Cada uma dessas técnicas imprime características específicas nas curvas de posição, velocidade e aceleração, proporcionando comportamentos diferentes em relação à eficiência do movimento, suavidade e custo computacional.

Os problemas observados na plataforma 6UDESC motivaram este estudo. Procura-se estudar e implementar uma nova proposta para a geração dos movimentos no simulador de voo, de modo a evitar vibrações indesejáveis na estrutura mecânica, visando também otimizar o percurso e a velocidade dos movimentos. Neste trabalho, foram estudados e aplicados três algoritmos geradores de trajetórias que, juntamente com o hardware desenvolvido, permitem observar o impacto na velocidade dos movimentos.

### 3 GERAÇÃO DE TRAJETÓRIAS

A geração de trajetórias consiste em gerar o histórico dos movimentos (coordenadas) dos eixos de um servomecanismo ao longo do tempo, definindo um perfil de posição, velocidade e aceleração [7, 10, 11].

Uma das primeiras opções de projeto consiste em definir se gerar a trajetória no espaço de junta ou no espaço de tarefa, ou também chamado de espaço operacional. No espaço de tarefa as coordenadas do percurso (posição e orientação da extremidade do robô), e a interpolação entre estas, são expressas no espaço cartesiano. Por sua vez, gerar a trajetória no espaço das juntas significa que os pontos de passagem, e a interpolação dos movimentos entre estes, são expressas diretamente nas coordenadas da junta (ângulos ou deslocamentos, dependendo do tipo de articulação [14].

As trajetórias geradas no espaço da tarefa permitem que o movimento da extremidade do robô seja previsível, sendo também mais simples de lidar com os obstáculos e as colisões. Porém, para gerar uma trajetória de espaço de tarefa é necessário resolver a cinemática inversa, aumentando o custo computacional [10], além de ter que controlar outros problemas relacionados a este processo, tal como a singularidade [9]. Neste trabalho foi adaptada a geração de trajetórias no espaço de juntas.

As operações matemáticas envolvidas neste processo podem ser realizadas através de polinômios [8]. Desta forma, é possível definir um polinômio de ordem  $n$ , o qual pode ser combinado com outros para satisfazer as restrições do movimento através dos pontos de passagem [9]. As Figuras 1 a 3, ilustram as técnicas polinomial cúbica, de quinta ordem e trapezoidal de velocidade, respectivamente.

As técnicas polinomiais mais comuns na prática são a polinômio cúbico (3ª ordem), Figura 1, e o Quintic (5ª ordem), Figura 2. As

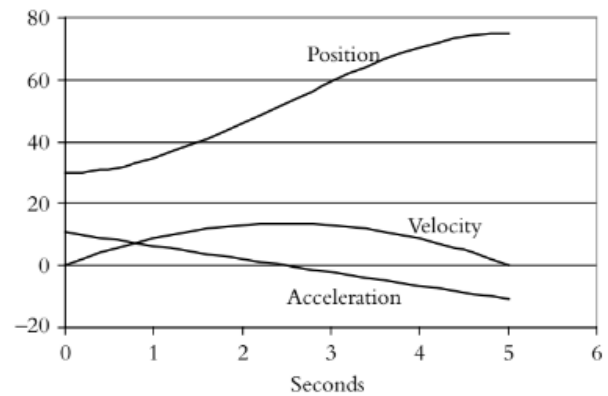


Figure 1: Curvas do polinômio de terceira ordem [9].

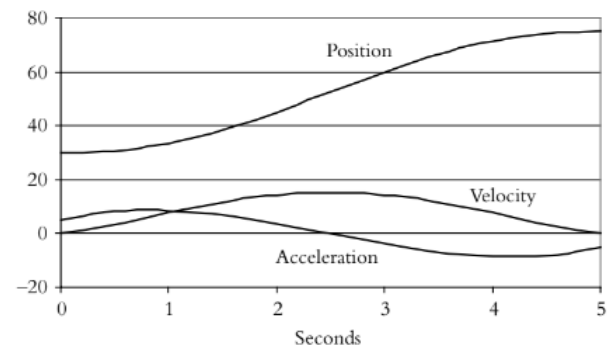


Figure 2: Curvas do polinômio de quinta ordem [9].

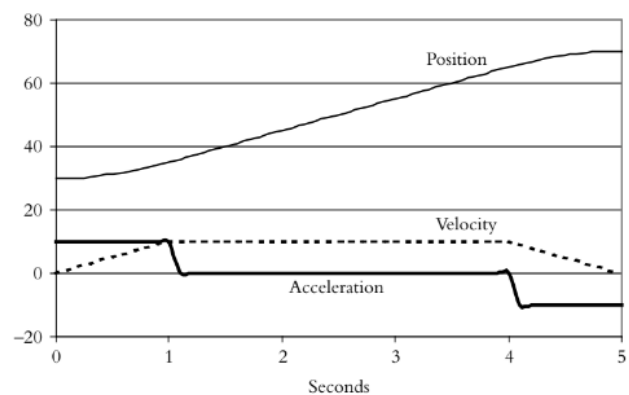


Figure 3: Curvas da técnica trapezoidal [9].

características e quantidade de parâmetros usados em cada técnica variam de acordo com a ordem do polinômio empregado. O polinômio cúbico requer 4 condições de limite: posição e velocidade no início e fim do movimento. Por esse motivo, a aceleração, embora ofereça um comportamento linear, apresenta descontinuidades no início e fim do movimento, Figura 1.

Por sua vez, o polinômio de quinta ordem requer 6 condições de contorno: posição, velocidade e aceleração, definidas no início e fim do movimento. Essas restrições permitem que a posição, velocidade e aceleração não apresentem descontinuidades, facilitando a tarefa do controlador, Figura 2.

Em aplicações onde é necessário o controle simultâneo de posição, de velocidade e aceleração, uma estratégia polinomial de quinta ordem possuirá a ordem mínima necessária para atender as necessidades de desempenho e de controle dessas aplicações [8].

As trajetórias de velocidade trapezoidal possuem trechos de aceleração constante, aceleração nula e desaceleração constante. Esse comportamento conduz a um perfil de velocidade trapezoidal, Figura 3. Por sua vez, a curva de posição apresenta um trecho linear, com segmentos parabólicos no início e fim do movimento. Essa parametrização torna relativamente simples a implementação desta técnica. Assim como ajustar e validar requisitos como posição, velocidade e limites de aceleração [9].

Trajetórias polinomiais são úteis para costurar continuamente segmentos com velocidade e aceleração nula ou diferente de zero, porque os perfis de aceleração são suaves (não apresentam descontinuidades), ao contrário das trajetórias de velocidade trapezoidal. No entanto, sua validação é mais complexa porque, em vez de ajustar diretamente as velocidades e acelerações máximas, são definidas as condições de limite, podendo ser ultrapassados os limites de velocidade ou aceleração nos segmentos da trajetória.

## 4 MATERIAIS E MÉTODOS

O projeto do módulo gerador de Trajetórias abrange o desenvolvimento de hardware da plataforma utilizada para a validação, os algoritmos (firmware e scripts do Matlab) e a interface desenvolvida para facilitar a interação com o sistema. As seguintes seções descrevem os elementos lógicos e físicos do sistema.

### 4.1 Estrutura Física

A plataforma desenvolvida para a validação das estratégias é mostrada na Figura 4. Ela possui as seguintes componentes: componentes: um kit de microcontrolador, um motor de corrente contínua com um encoder acoplado e uma ponte H para sintetizar a etapa de potência. Por sua vez, o microcontrolador estabelece uma comunicação com um computador via porta USB, possibilitando a troca de informações.

Os componentes utilizados são:

- Arduino MEGA 2560 R3;
- Ponte H L298N;
- Motor DC, de 6v e 210 RPM, com encoder acoplado.

### 4.2 Comunicação

A comunicação entre os algoritmos desenvolvidos no Matlab e o microcontrolador da aplicação possibilita a troca de informações como hardware, e o envio e o retorno dos valores de velocidades desejados e lidos, respectivamente. A Figura 5 ilustra o esquema de comunicação entre o microcontrolador e o Matlab.

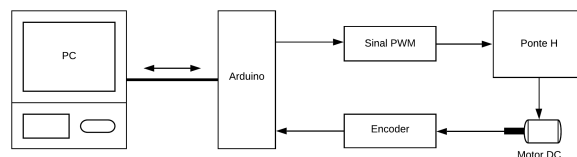


Figure 4: Diagrama da implementação estrutural do projeto. Autoral.

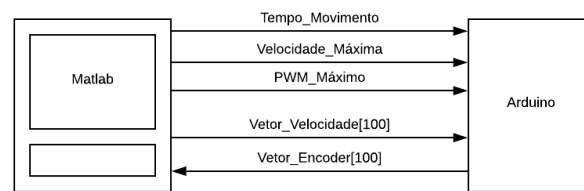


Figure 5: Diagrama da comunicação entre o algoritmo em Matlab e o Arduino. Autoral

### 4.3 Etapa de Potência

O motor utilizado no projeto necessita uma corrente de 0.13A para sua operação. E as postas do microcontrolador da aplicação fornecem até 40mA. É preciso então um elo de potência entre o microcontrolador e o motor interface [12]. A Ponte H é um circuito que possibilita comandar os movimentos de um motor de corrente contínua através de um microcontrolador, controlando a direção do movimento e também sua velocidade, Figura 6.

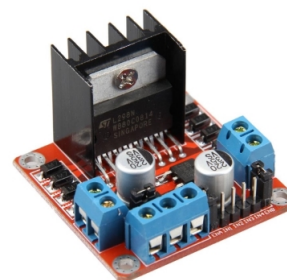


Figure 6: Driver motor ponte H L298n [5].

### 4.4 Leitura do Encoder

O motor de corrente contínua utilizado nesta implementação possui um encoder acoplado que permite determinar o ângulo de giro, o sentido de rotação do eixo e sua velocidade, Figura 7. Neste projeto, somente as características de velocidade foram exploradas, portanto, serão abordadas apenas as técnicas relacionadas a esta funcionalidade.

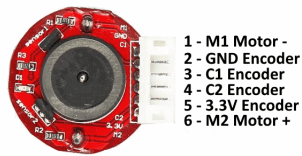


Figure 7: Encoder presente no motor utilizado [3].

#### 4.5 Software

Os algoritmos para a geração de trajetórias foram desenvolvidos no Matlab, Figura 9. E o fluxo do firmware (desenvolvido em C++) é ilustrado na Figura 8.

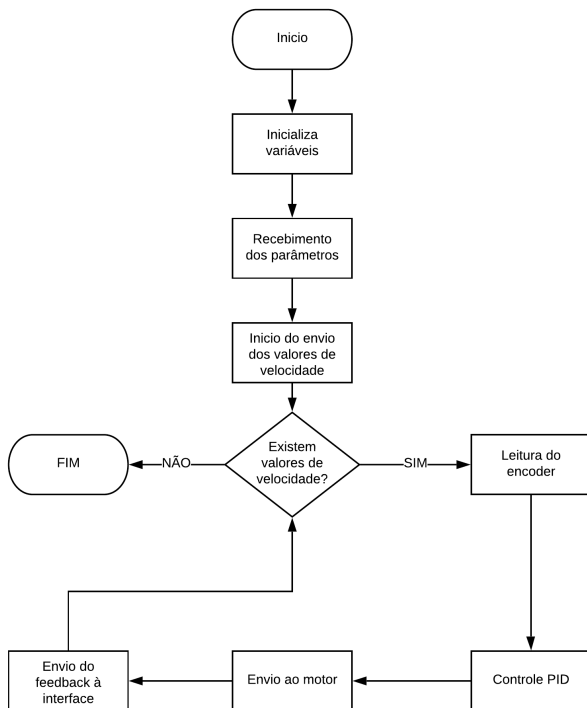


Figure 8: Fluxo do firmware. Autoral.

Cabe salientar que para aproximar os valores de velocidade desejados com os obtidos, a partir da leitura do encoder, foi implementado o controle PID para reduzir os erros e sanar problemas dinâmicos como o atrito estático no eixo do servomecanismo. O algoritmo de controle foi sintetizado no microcontrolador da aplicação, usando as equações da estrutura paralela, Figura 10. Os ganhos do controlador são definidos pelos parâmetros  $K_p$ ,  $T_i$  e  $T_d$ .

A leitura da velocidade do motor foi realizada utilizando a função denominada pulseIn(). Esta função retorna, em microssegundos, o tempo em que um sinal fica em estado HIGH e em estado LOW no pino designado. Somando estes dois valores é possível determinar o período, em microssegundos, e usando uma regra de três pode ser obtida a velocidade do giro.

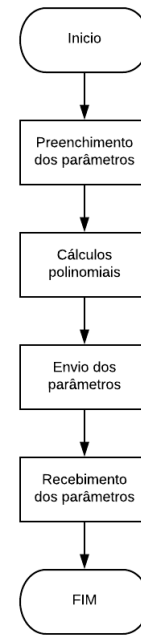


Figure 9: Fluxo do algoritmo do Matlab. Autoral.

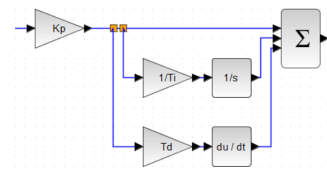


Figure 10: Estrutura paralela do PID. Adaptado de [10]

Cabe salientar que os códigos da implementação podem ser acessados no repositório criando para o Github, disponível em [2].

#### 4.6 Interface

Para facilitar a interação com os algoritmos de geração de trajetória foi desenvolvida uma interface gráfica na plataforma Matlab utilizando a ferramenta App Designer, Figura 11. O desenvolvimento desta interface tornou-se necessário para facilitar a interação com os algoritmos desenvolvidos, permitindo modificar os parâmetros das curvas diretamente no código. Desta forma, a interação com as técnicas, a visualização dos resultados e das curvas torna-se direta e intuitiva.

### 5 RESULTADOS OBTIDOS

O módulo desenvolvido em sua totalidade é mostrado na Figura 12. Na implementação observou-se a influência do atrito. As curvas apresentavam um comportamento similar ao conhecido como “zona morta” para valores baixos de velocidade, tanto no início quanto no fim do movimento. Este problema foi contornado com a inclusão

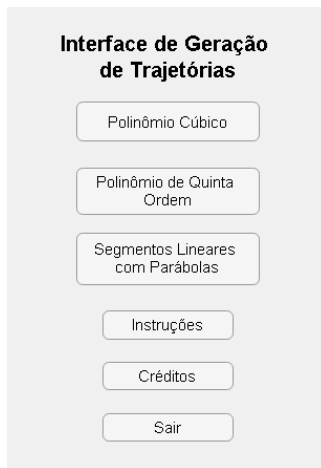


Figure 11: Tela principal da interface do módulo gerador de trajetórias. Autoral.

da técnica de controle Proporcional, Integral e Derivativa, ou PID apresentada previamente, apesar da mesma não ser a ideal para tratar de sistemas não lineares [10].

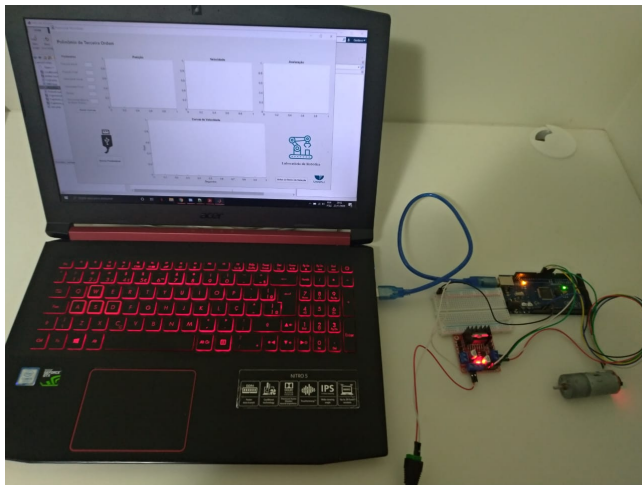


Figure 12: Hardware usado para a validação dos movimentos. Autoral.

As Figuras 13 a 15 ilustram o perfil desejado para as posições, velocidades e acelerações para cada uma das estratégias estudadas, mostrando separadamente a curva de velocidade atual e desejada. Todas as curvas apresentaram diferentes comportamentos de acordo com suas características principais e parâmetros iniciais (padronizados).

Os polinômios de terceira e de quinta ordem apresentados nas Figuras 13 e 14 obtiveram comportamentos semelhantes como as oscilações no início e no fim de seus movimentos, porém, mantendo-se sempre próximas ao eixo da curva desejada de velocidade.

Em relação ao método trapezoidal, Figura 15, são demonstradas as mesmas oscilações no início e no fim dos seus movimentos além

das alterações no momento de linearidade de sua curva, apresentando também um maior *offset* consequentemente.

Observando a proximidade entre o perfil desejado e o atual, o melhor comportamento foi obtido com a estratégia polinomial de quinta ordem, como esperado. Em todos os casos, a resposta de velocidade apresenta oscilações cuja atenuação será objeto de trabalhos futuros.

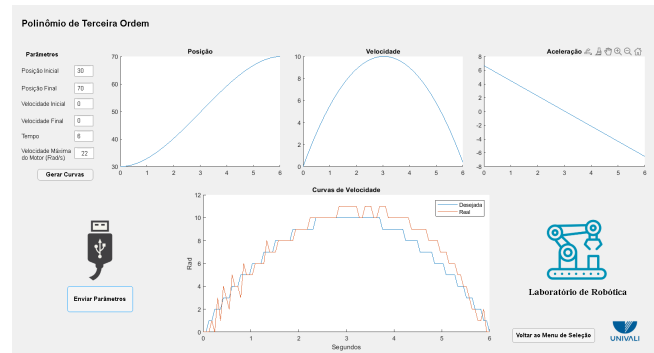


Figure 13: Tela da interface do polinômio cúbico. Autoral.

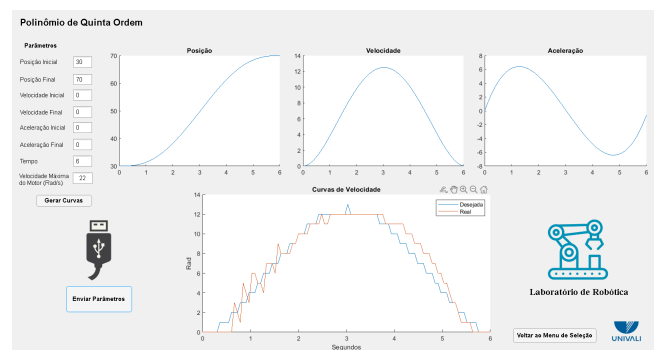


Figure 14: Tela da interface do polinômio de quinta ordem. Autoral.

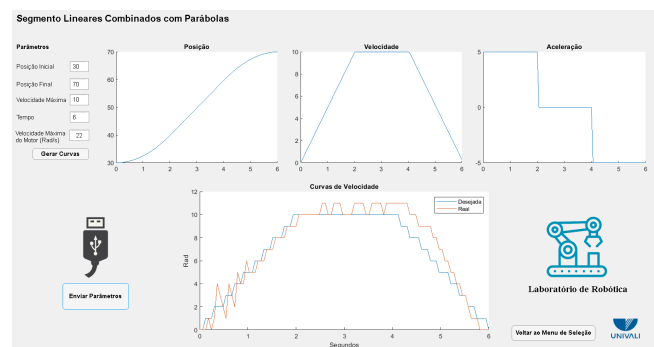


Figure 15: Tela da interface do método trapezoidal. Autoral.

As curvas de velocidade da trajetória real e a desejada, obtidas para cada um dos algoritmos estudados, permitem validar a solução

proposta neste projeto como importante recurso didático a ser utilizado na disciplina de Robótica.

## 6 CONSIDERAÇÕES FINAIS

No desenvolvimento deste projeto buscou-se entender os principais fundamentos e características dos sistemas robóticos e, em particular, dos servomecanismos. Foi também necessário o estudo das técnicas de controle para estas aplicações. No artigo são descritas três técnicas de geração de trajetórias que são utilizadas em projetos da área.

A implementação dos algoritmos desenvolvidos resultaram em curvas de posição, velocidade e aceleração fiéis às apresentadas na literatura especializada. A validação prática das técnicas abordadas foi realizada com o auxílio de uma plataforma integrada de hardware e software desenvolvida para este fim. Esta validação aconteceu em nível de simulação e em nível de hardware. Em todos os casos foi necessário compensar o atrito estático, ocasionado pela relação de redução do motor, e, para tanto, foi implementada uma estratégia de controle PID. A interface gráfica desenvolvida para o Matlab foi projetada para possibilitar a interação dos alunos com a plataforma desenvolvida.

Trabalhos futuros poderão abordar outras técnicas de geração de trajetória, além do estudo de soluções para mitigar as oscilações observadas. Também se prevê a utilização da plataforma na disciplina de Robótica, com o objetivo de avaliar a usabilidade da solução proposta.

## REFERENCES

- [1] H. Asada and J.J.E. Slotine. 1986. *Robot Analysis and Control*. Wiley. <https://books.google.com.br/books?id=KUG1VGkL3loC>
- [2] Gustavo Copini Decol. 2020. Repositório Módulo Gerador de Trajetórias Aplicado no Eixo de um Servomecanismo. <https://github.com/gutoodc2/MGT.git>.
- [3] Arduino e Cia. 2016. Como usar motor DC com encoder no Arduino. <https://www.arduinoocia.com.br/motor-dc-com-encoder-arduino>
- [4] Shuang Fang, Xianghua Ma, Jiarui Qu, Sijing Zhang, Na Lu, and Xin Zhao. 2019. Trajectory Planning for Seven-DOF Robotic Arm Based on Seventh Degree Polynomial. In *Chinese Intelligent Systems Conference*. Springer, 286–294.
- [5] FILIPEFLOP. 2020. Driver Motor Ponte H L298n. <https://www.filipeflop.com/produto/driver-motor-ponte-h-l298n/>
- [6] ZHU Hongjun. 2017. MOTION TRAJECTORY PLANNING AND SIMULATION OF 6-DOF MANIPULATOR ARM ROBOT. *Academic Journal of Manufacturing Engineering* 15, 3 (2017).
- [7] Walter Fetter Lages. 2019. Geração de Trajetórias. <http://www.ece.ufrgs.br/~fetter/eng10026/traj.pdf>
- [8] José Adenilson Gonçalves Luz Junior. 2016. *Modelagem e simulação de controle de trajetória para robôs manipuladores com três graus de liberdade com elementos rígidos*. B.S. thesis. Universidade Tecnológica Federal do Paraná.
- [9] S.B. Niku. 2010. *Introduction to Robotics: Analysis, Control, Applications*. Wiley. <https://books.google.com.br/books?id=2V4aGvIGt7IC>
- [10] N.S. Nise and J.P. Matsuura. 2017. *Engenharia De Sistemas De Controle*. LTC. <https://books.google.com.br/books?id=LYnlsWACAAJ>
- [11] Luiz Eduardo Nicolini do Patrocínio Nunes. 2007. Geração e otimização de trajetórias de um manipulador robótico utilizando algoritmos genéticos. (2007).
- [12] Luis Fernando Patsko. 2006. Tutorial Montagem da ponte H. *Maxwell Bohr-Instrumentação Eletrônica* (2006).
- [13] Alejandro RAMIREZ, Alexandre REIS, Marcelo FERREIRA, Altino CORDEIRO NETO, and Amanda HAASE. 2012. Automação de Movimentos para Simulador de Voo em Realidade Virtual Imersiva. 1, 1 (2012), 6. <http://www.revistas.udesc.br/index.php/hfd/article/view/2880>
- [14] J.M. Rosario. 2012. *Robótica Industrial I Modelagem, Utilização e Programação*. BARAUNA - UM LIVRO. <https://books.google.com.br/books?id=cbzXOdbjvYoC>
- [15] Qiao Wang, Zhanbin Wang, and Mei Shuai. 2020. Trajectory planning for a 6-DoF manipulator used for orthopaedic surgery. *International Journal of Intelligent Robotics and Applications* (2020), 1–13.