



Planejamento de trajetória de um braço robótico por interface homem-máquina

Path planning of a robotic arm by human-machine interface

Jhonatas Luthierry Barbosa dos Santos, Marcio Mendonça[†],

RESUMO

O projeto tem cunho educacional, permitindo uma interação dos alunos que cursam a disciplina de robótica objetivando ser uma ferramenta de melhor visualização do controle de trajetória de um braço robótico por interface homem máquina. Em um primeiro momento houve um estudo de casos a partir de simulações através do Matlab R2021b para definir qual seria a opção ideal de controle da trajetória do braço robótico, entre os polinômios de terceira e quinta ordem encontrados na literatura, duas das mais conhecidas e aplicadas, chegando à conclusão que o de quinta ordem atendia melhor a proposta. Após foi programado na linguagem Python através do Raspberry pi 3 um escopo para controle PWM das juntas do braço porém sem conclusão por falta de recursos necessários para os testes, por conta do afastamento do aluno ao laboratório, e por fim os resultados obtidos foram considerados satisfatórios levando em consideração as dificuldades já citadas. Este trabalho se encerra com uma conclusão e sugestão de futuro trabalho.

Palavras-chave: robótica, planejamento e controle de trajetória, braço robótico

ABSTRACT

The project has an educational nature, allowing interaction of students who are taking the robotics course, aiming to be a tool for better visualization of the trajectory control of a robotic arm through a human-machine interface. At first, there was a case study based on simulations using Matlab R2021b to define the ideal option for controlling the trajectory of the robotic arm, among the third and fifth-order polynomials found in the literature, two of the best known and most applied, concluding that the fifth-order best suited the proposal. Afterward, a scope for PWM control of the arm joints was programmed in Python language through Raspberry pi 3, but without completion due to lack of resources necessary for the tests, due to the removal of the student to the laboratory, and finally, the results obtained were considered satisfactory, taking into account the difficulties mentioned above. This work ends with a conclusion and a suggestion for future work.,

Keywords: *robotics, trajectory planning and control, robotic arm*

1 INTRODUÇÃO

Este projeto de pesquisa tem a proposta de unir a robótica e a inteligência artificial com o cunho educacional, objetivando a proporcionar um protótipo para a disciplina de robótica a princípio. Para continuação desse projeto poderia usar uma solução da cinemática inversa por meio de redes neurais (HAYKIN, 2000), (DALMEDICO et al., 2018) e/ou redes neurais convulsionais aplicadas em visão computacional para reconhecimento de objetos em um braço manipulador autônomo (WILLIAMS et al., 2019). Desse modo, seria plausível um novo projeto sequente de um protótipo que além da contribuição

* Eng. Elétrica, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, PR, Brasil; JhonatasSantos@alunos.utfpr.edu.br



inicial., poderia enriquecer como uma aplicação de técnicas computacionais, as quais poderiam enriquecer com a disciplina de sistemas inteligentes, ambas do curso de engenharia de controle e automação.

Em sua primeira versão houve a tentativa de desenvolvimento de um braço robótico programado em linguagem de Python e controlada por um microcomputador Raspberry pi 3 por controle PWM (*Pulse-Width Modulation*), que é uma técnica utilizada para permitir o controle de energia fornecida a equipamentos elétricos e assim controlar a velocidade dos servos motores em sua trajetória, aceleração e desaceleração. Todo esse processo é executado por uma interface ao usuário os ângulos desejados, $Posição_{inicial}$ e $Posição_{Final}$ com alguma estratégia de planejamento de trajetória, como por exemplo polinômio de 5 grau (NIKU, 2011).

Por conta da pandemia global da COVID-19 o projeto sofreu atrasos em sua execução, pelo fato do afastamento do aluno do campus de origem e a dificuldade de acesso a materiais e equipamentos extremamente necessários para o desenvolvimento e testes. Além disso houve troca de orientado, devido a desistência do aluno Rafael Flauzino. Deste modo devido a essas dificuldades não foi possível cumprir o planejamento proposto de forma integral, e sim somente parcialmente. Entretanto, possibilitou maior conhecimento de fundamentos de robótica (NIKU, 2011), (SICILIANO; KHATIB, 2008), ainda não cursada, na linguagem Python (STEVENS; BOUCHER, 2015) e arquitetura microcontrolador utilizado, raspberry pi 3 (RASPBerryPI, 2016).

Ainda devido a pandemia, as orientações só puderam ser feitas por meio de ‘lives’, o que dificultou ainda mais o entendimento de todas as etapas do projeto.

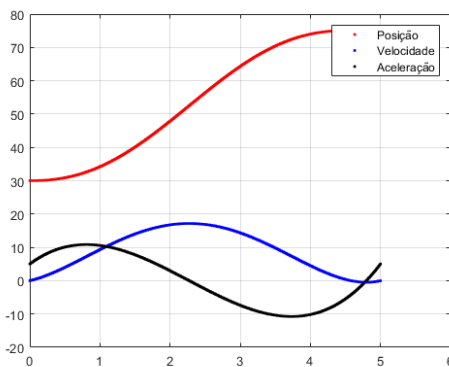
2 MÉTODO (OU PROCEDIMENTOS OPERACIONAIS DA PESQUISA)

Como base para as simulações foi usado livro (NIKU, 2011, p.184), usando assim os exemplos 5.1 e 5.3 para os polinômios de terceira e quinta ordem, respectivamente, para o planejamento de trajetória. O polinômio de terceira ordem mesmo sendo interessante há limitações que podem danificar os dispositivos atuadores do robô, com à ausência da desaceleração, onde foi feito uma simulação pelo MATLAB R2021b.

O método de acordo com a proposta montar uma interface IHM. Interface Homem Máquina por meio de um display *touch screen* acoplada em um raspberry pi 3, e empregando programação em Python realizar um planejamento de trajetória. Para entrada de dados no ambiente de simulação do Matlab, Figura 1 (entrada de dados do polinômio de ordem cinco), com os respectivos resultados cinemáticos de cada junta do robô, mostrado na figura 2 com polinômios de ordem três e cinco, ressaltando a vantagem do controle de aceleração da segunda estratégia, dentre outras presentes na literatura (NIKU, 2011). No ambiente MatLab os dados para cada junta do robô são inseridos em uma janela de dados como por exemplo a da figura 1

O polinômio de quinta ordem, dentre outras técnicas de planejamento de trajetórias é uma opção pelo fato de adicionar a aceleração e desaceleração, dando segurança aos dispositivos atuadores e tornando um movimento harmônico na trajetória. Pode-se visualizar as variáveis angulares na simulação e a inserção numéricos de entrada, a qual deveria ser inspiração da IHM (Interface Home Máquina do projeto), como mostra a Figura 1.

Figura 1 – IHM simulação polinômio de quinta ordem, entrada dados MatLab



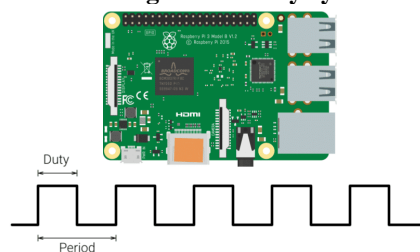
```
Digite o valor do Angulo inicial: 30
Digite o valor do Angulo final: 75
Digite o valor da aceleração inicial: 5
Digite o valor da aceleração final: -5
Informe o tempo em segundos: 5
```

Fonte: MATLAB (2021)

A análise dos resultados cinemáticos das simulações mostra que a aceleração foi o dobro da segunda estratégia

O diagrama de blocos representa as etapas do projeto onde primeiro foi executado as simulações das juntas no MATLAB R2021B e o escopo da programação Python através do *Raspberry pi 3*, não foi possível a execução de testes pois práticos, como já citado na introdução, a ausência de equipamentos como por exemplo o osciloscópio é indispensável para continuidade, e houve a queima do display LCD para a criação da interface homem-máquina.

Figura 2 – Duty cycle



Fonte: embarcados (2017)

Para controle das juntas do braço robótico através dos servos motores, foram utilizados para a simulação e testes através da IDE do Raspberry pi 3, Mu versão 1.0. Os primeiros testes foram executados a partir de um protoboard e leds, aumentando e diminuindo o seu brilho, para verificar o funcionamento do controle PWM



onde modula o ciclo ativo (duty cycle) de uma onda quadrada, como representado na Figura 1, assim o controlador entrega uma serie de pulsos, gerados em intervalos de igual duração, que pode ser variado, quanto mais largo o pulso, maior a quantidade de corrente fornecida a carga, foram adicionadas as bibliotecas para que as etapas seguintes executassem com êxito;

```
import RPi.GPIO as GPIO
import time as delay
```

A biblioteca *RPi.GPIO*, foi usada para habilitar um conjunto de pinos responsável por fazer a comunicação de entrada e saída de sinais digitais, o Raspberry pi 3 é composto por 40 pinos, Figura 4, na qual esses pinos auxiliam no desenvolvimento do protótipo.

Figura 4 – Duty cycle

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I ² C)		DC Power 5v	04
05	GPIO03 (SCL1, I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Fonte: Própria (2017)

A biblioteca *time* foi usado no pino 12 onde está o PWM, para definir o tempo de atraso da frequência do led que representa o servo motor do braço robótico onde o duty cycle inicial é igual a zero indo até 100hz;

```
GPIO.setup(12,GPIO.OUT)
p = GPIO.PWM(12,100)
p.start(0)
```

Em seguida foi adicionado o *while* para que o código entre em um loop infinito e possamos fazer os ajustes necessários antes de começar os testes no braço robótico, o range para aceleração do braço foi programado para iniciar em 0% e ir até 100% ao passo de 5 no delay de 100ms

```
while(1):
    for duty in range (0,101,5):
        p.ChangeDutyCycle(duty)
        delay.sleep(0.1)
```

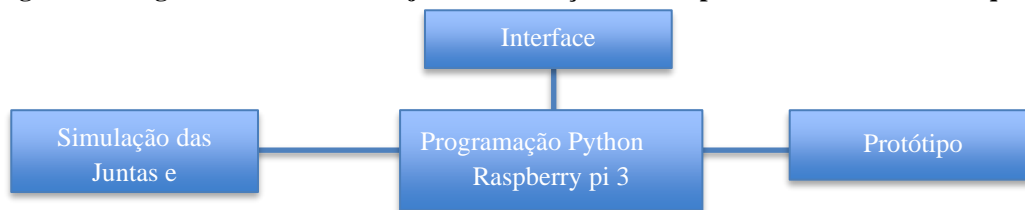
Para desaceleração do movimento o range (faixa) de 100% a 0% ao passo de -5



```
for duty in range (100,-1,-5):  
    p.ChangeDutyCycle(duty)  
    delay.sleep (0.1)  
except KeyboardInterrupt:  
    pass  
p.stop()  
GPIO.cleanup ()
```

Dessa forma foi programado a aceleração, desaceleração e o distancia da trajetória de um dos servos motores, para o restante a programação é semelhante. A figura 5 mostra uma visão macro do projeto em forma de blocos.

Figura 5 - Diagrama de blocos – Trajetória do braço robótico por interface homem-máquina

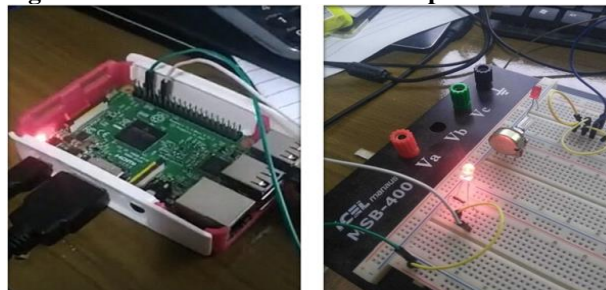


Fonte: Autoria própria (2021).

3 RESULTADOS

Para executar os testes, com recursos próprios devido à distância supracitada, uma fonte de alimentação chaveada ATX, pois a alimentação própria do raspberry pi 3 não conseguiria fornecer a corrente mínima necessária para alimentação do controlador, e dos atuadores, servo motores. Assim, os testes iniciais de controle do PWM foram satisfatórios como desejado assim conseguindo fazer o controle do brilho do led modulando o ciclo ativo da onda quadrada, e variando, com um pulso maior, como podemos ver na figura 6.

Figura 6 – Controle de luminosidade por PWM



Fonte: Autoria própria (2021).

A representação do duty cycle até a frequência máxima programada se dá no tempo de 2 segundos em um Step igual a 5, variando a frequência de 5hz até atingir a frequência máxima desejada, podendo assim simular uma possível aceleração e desaceleração das juntas do braço robótico por PWM, representado no Quadro 1



Quadro 1 – Representação dos passos do duty cycle até a frequência máxima programada

Tempo (segundos)	Passos (Step) = 5	Frequência (Hz)
0.10	5	5
0.20	10	10
0.30	15	15
0.40	20	20
2.0	100	100

Fonte: Autoria própria (2021).

4 CONCLUSÃO

Os resultados obtidos ainda que iniciais, foram satisfatórios, considerando os problemas relatados. Entretanto, foi uma base para continuidade do projeto, não foi possível apresentar uma mais de resultados, pois como já explicado anteriormente, o afastamento dos laboratórios afetou nos materiais disponíveis como por exemplo o osciloscópio, fonte externa, drive, entre outros. Essenciais para o desenvolvimento do projeto como planejado. Porém um passo inicial foi executado, funcionando uma primeira versão do protótipo objetivado. Entretanto, foi possível uma maelhor intyeração com a linguagem Python na execução da geração de pulsos PWM, que posteriormente deverão apicados a cada atuador das respectivas juntas do robô, como servomotores por exemplo. Ressalta-se que sem drives, não foi possível aplicar pulsos nos motores do protótipo.

Futuros trabalhos endereçam a realização da proposta de forma completa.

AGRADECIMENTOS

Agradeço a fundação Araucária pelo suporte financeiro para execução do projeto, Jhonatas Luthierry barbosa dos Santos.

REFERÊNCIAS

- DALMEDICO, J. F. et al. **Artificial Neural Networks Applied in the Solution of the Inverse Kinematics Problem of a 3D Manipulator Arm**. Proceedings of the International Joint Conference on Neural Networks. **Anais...**Rio de Janeiro, RJ, Brazil: IEEE, 2018
- HAYKIN, S. **Redes Neurais, Princípios e Prática**. 2. ed. [s.l.] Bookman, 2000.
- NIKU, S. B. **Introduction to Robotics: Analysis, Control, Applications**. 2. ed. Hoboken, New Jersey, USA: John Wiley & Sons, 2011. v. 53
- RASPBERRYPI. Raspberry Pi 3 Model B+ Datasheet. **Datasheet**, p. 5, 2016.
- SICILIANO, B.; KHATIB, O. Robotics in Agriculture and Forestry. In: BILLINGSLEY, J.; VISALA, A.; DUNN, M. (Eds.). . **Springer handbook of robotics**. 1. ed. Berlin: Springer, Berlin, Heidelberg, 2008. p. 1065–1077.