XXXI Congresso de Iniciação Científica Unicamp 2022 2023

IMPLEMENTAÇÃO DE UMA REDE NEURAL MLP PARA CORREÇÃO DE TRAJETÓRIA DE MECANISMOS HBOT E COREXY

Palavras-Chave: POSICIONAMENTO, CINEMÁTICA, DINÂMICA

Autores/as:

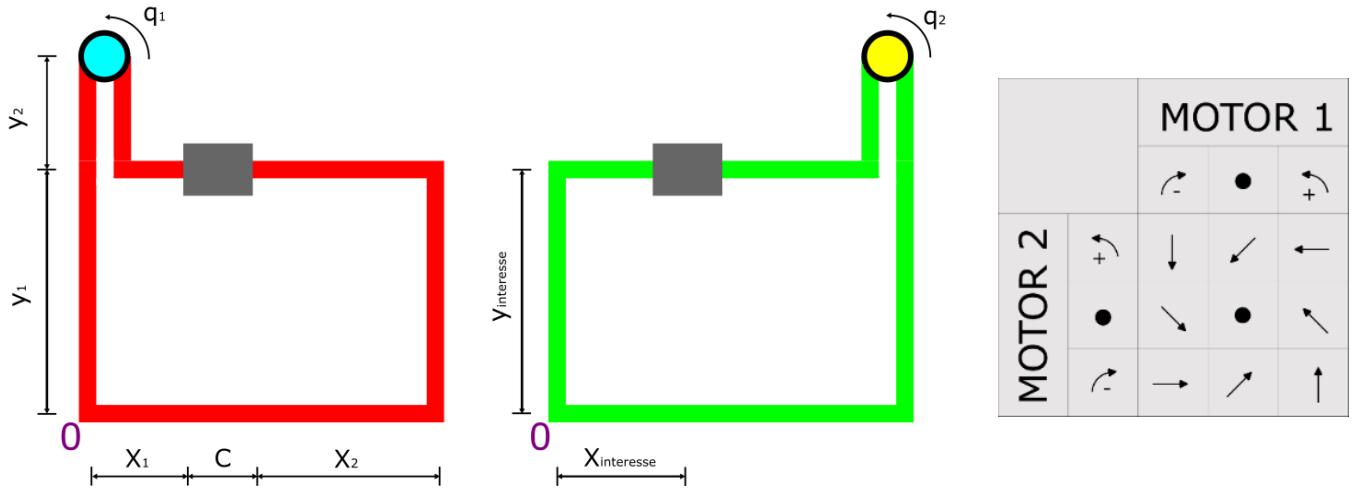
PEDRO ESCH DE CAMPOS GOMES MACEDO – FCA, UNICAMP

Prof. Dr. EDUARDO PAIVA OKABE (orientador) – FCA, UNICAMP

INTRODUÇÃO:

O HBOT e COREXY são mecanismo similares de locomoção 2D muito utilizado devido a sua simplicidade, estando presente em impressoras 3D e outros equipamentos, em especial na indústria. Esses mecanismos são compostos em essência por dois circuitos parcialmente sobrepostos que promovem o controle de deslocamentos nos eixos X e Y de acordo com a rotação de dois motores, um em cada circuito.

Figura 1: Circuitos do sistema e controle direcional.



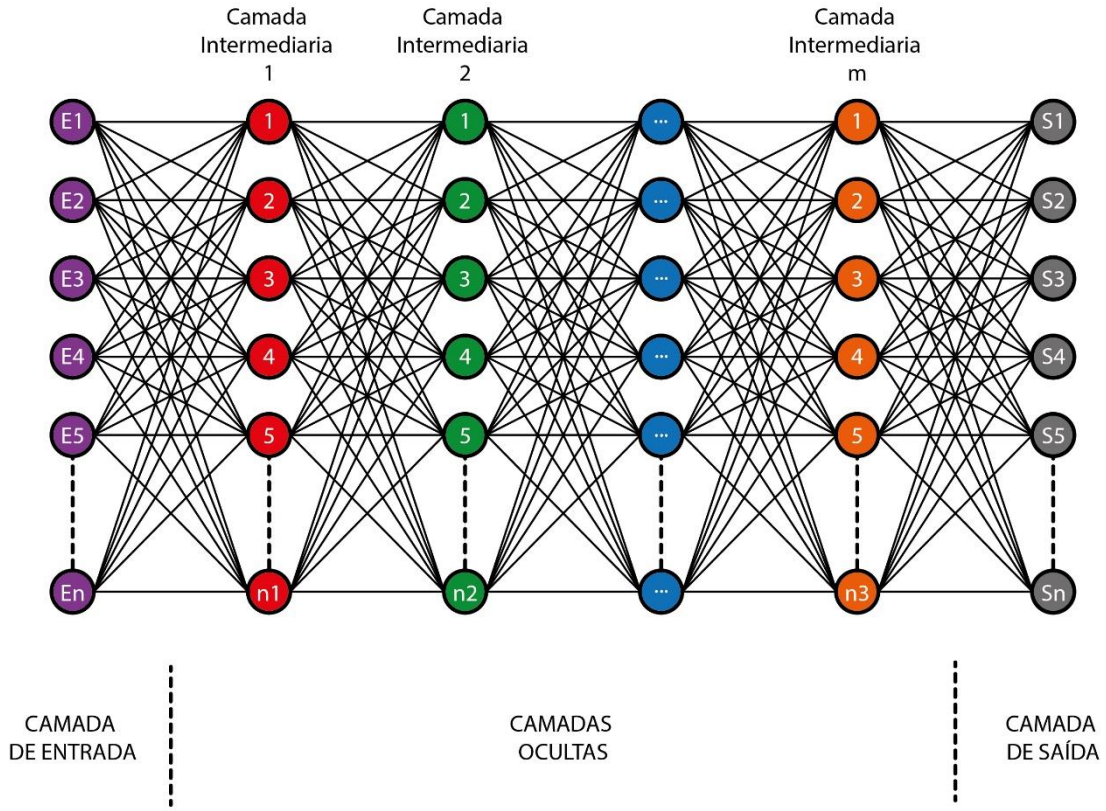
Fonte: Autoria Própria.

Contudo apesar da simplicidade do sistema há uma série de problemas que podem causar falhas de posicionamentos, como: deformidades na estrutura, desalinhamento e atrito. O projeto tem como intuído desenvolver uma solução para esse problema.

METODOLOGIA:

A fim de tornar possível a correção do posicionamento, é desenvolvida e treinada uma Rede Neural MLP (Perceptron Multi-Camadas). Esse tipo de rede utiliza, como o nome já indica, múltiplas camadas neurais para traduzir a entrada em uma saída apropriada, nesse caso, conforme ilustrado na figura 2, cada perceptron (neurônio) na camada oculta é conectada a todos os perceptons imediatamente anteriores e posteriores, e a calibragem da rede é feita com o ajuste dos “pesos” de cada ligação, que são refinados a cada iteração na etapa de treinamento.

Figura 2: Esquema de uma Rede Neural MLP.



Fonte: Autoria Própria.

Contudo é necessário entender a cinemática e dinâmica do mecanismo para gerar os parâmetros que serão utilizados na programação, como a criação da nuvem de dados utilizada como parâmetro de entrada. Portanto, para trabalhar sob esse sistema foi necessária a parametrização e compreensão dos fatores envolvidos, utilizando das equações dos circuitos dispostos na figura 1.

Após essa formulação inicial, também foi analisado um circuito distorcido, no qual há ângulos indesejados afetando a movimentação, de maneira que podemos comparar a diferença de posicionamento e treinar a rede para realizar essa correção, sendo então adicionada a analise dinâmica.

RESULTADOS E DISCUSSÃO:

As equações dos circuitos verde e vermelho são equivalentes, e sendo esse um sistema fechado o resultado das equações será naturalmente zero.

$$f_0 = x_1 + C + x_2 - X = 0 \quad (1)$$

$$f_1 = Y - y_2 - y_1 = 0 \quad (2)$$

Unindo essas equações com a relação da rotação dos motores q_1 e q_2 aos deslocamentos x e y dispostos a seguir (equações 3 e 4), seremos capazes de parametrizar a posição do corpo de interesse com base na rotação do motor.

$$R_1 q_1 - y_1 - x_2 = 0 \quad (3)$$

$$R_2 q_2 + y_1 + x_1 = 0 \quad (4)$$

Realizando algumas operações matemáticas simples é possível obter os valores de x_1, x_2, y_1 e y_2, os quais são usados para, considerando que o ponto de interesse está localizado no centro da estrutura em movimento, podemos então encontrar o valor de x_{int} e y_{int} baseado nos valores de R_1 e R_2.

$$x_{int} = \frac{X - R_2 q_2 - R_1 q_1}{2}$$

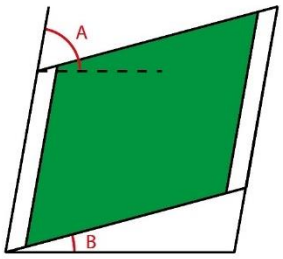
$$y_{int} = \frac{-X + C - R_2 q_2 + R_1 q_1}{2}$$

Também é feita analise inversa, utilizando x_{int} e y_{int} como primarias e q_1 e q_2 como secundarias, sendo necessário para fazer gerar os dados de treinamento da rede.

$$q_1 = \frac{y_{int} + X - \frac{C}{2} - x_{int}}{R_1}$$

$$q_2 = \frac{-y_{int} - x_{int} + \frac{C}{2}}{R_2}$$

Figura 3: Sistema angulado.



Fonte: Autoria Própria

Juntamente a isso é calculado os parâmetros de velocidades e acelerações das variáveis secundarias que serão trabalhadas na dinâmica do mecanismo. Em seguida é feita a mesma análise do modelo distorcido. Gerando novas equações de circuito e por consequência uma nova parametrização.

$$f_0 = 0 = y_1 \cos(A) + x_1 \cos(B) + C \cos(B) + x_2 \cos(B) - y_3 \cos(A) - L$$

$$f_1 = 0 = y_1 \sin(A) + x_1 \sin(B) + C \sin(B) + x_2 \sin(B) - y_3 \sin(A)$$

$$x_{int} = \frac{(X - C - R_1 q_1 - R_2 q_2)\cos(B) + C\cos(B) + (-X + C + R_1 q_1 - R_2 q_2)\cos(A)}{2}$$

$$y_{int} = \frac{(X - C - R_1 q_1 - R_2 q_2)\sin(B) + C\sin(B) + (-X + C + R_1 q_1 - R_2 q_2)\sin(A)}{2}$$

.

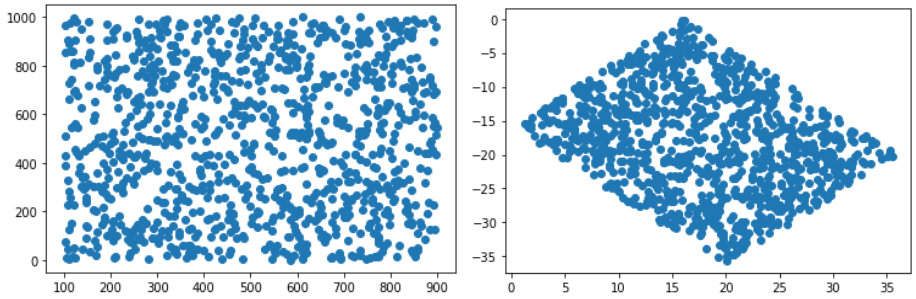
Aqui o processo inverso também é feito, obtendo a parametrização de q_1 e q_2 utilizando x_{int} e y_{int} como variáveis primárias.

$$q_1 = \frac{X - \frac{C}{2} - \frac{x_{int}\tan(A) - y_{int}}{\cos(B)\tan(A) - \sin(B)} + \frac{x_{int}\tan(B) - y_{int}}{\sin(A)\tan(B) - \sin(A)}}{R_1}$$

$$q_2 = \frac{-\frac{C}{2} + \frac{x_{int}\tan(A) - y_{int}}{\cos(B)\tan(A) - \sin(B)} + \frac{x_{int}\tan(B) - y_{int}}{\sin(A)\tan(B) - \sin(A)}}{R_2}$$

Tudo isso é então levado ao Python para gerar as nuvens de dados que serão utilizadas, nesse caso foi trabalhado com 1000 pontos, contudo a ampliação da nuvem utilizada para o treinamento melhorará os resultados obtidos apesar de aumentar o tempo de processamento.

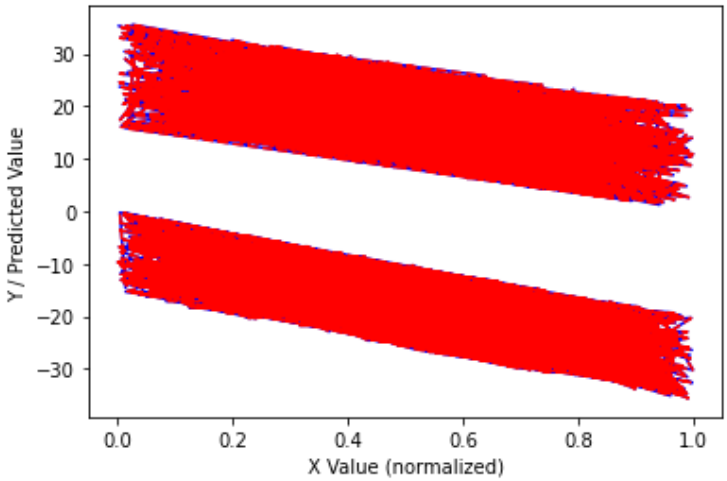
Figura 4: Nuvem de pontos do circuito inicial, a esquerda de x_{int} e y_{int} e a direita de q_1 e q_2.



Fonte: Autoria Própria.

A fim de melhorar os resultados também foi feita uma normalização nos parâmetros de entrada, colocando-os em uma escala de 0 a 1. Sendo então gerado um gráfico para averiguar a qualidade da resposta da rede neural, e uma tabela comparando a melhora com o uso da normalização. Tudo isso é feito para então unir a versão inicial e a angulada, promovendo assim a efetiva correção de posicionamento

Figura 5: Comparação da previsão em vermelho com o os valores de treinamento em azul.



Fonte: Autoria Própria

Tabela 1: Comparação dos erros da rede neural obtidos sem e com a normalização.

SEM NORMALIZAÇÃO				COM NORMALIZAÇÃO DA ENTRADA			
Erro q1	**5,1851%**	**Erro x de interesse**	**5,0147%**	**Erro q1**	**0,0508%**	**Erro x de interesse**	**0,0807%**
Erro q2	**1,1009%**	**Erro y de interesse**	**13,6855%**	**Erro q2**	**0,1017%**	**Erro y de interesse**	**0,2008%**

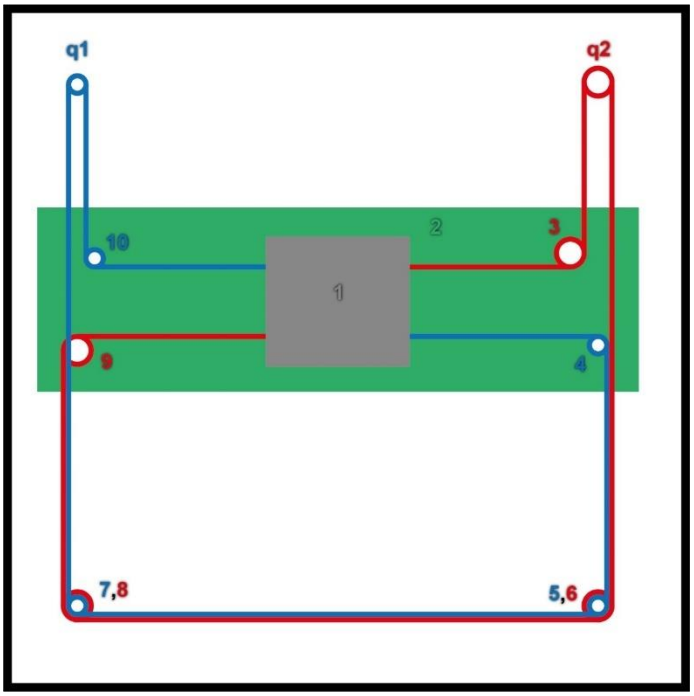
Fonte: Autoria Própria.

Por fim é então feita a analise dinâmica que considera os seguintes corpos, dispostos pela numeração na figura 6., dos quais da numeração três ao dez são polias. Sendo então feita a analise dinâmica, contando com a energia potencial (V), tendo as correias sido simplificadas para serem analisadas como molas, energia cinética (T), forças não conservativas, utilizadas para se obter os Q_j^{nc}, e finalmente a montagem da equação de Lagrange que utiliza ao todo 10 variáveis primarias na sua análise, x_{int}, y_{int}, θ_3, θ_4, θ_5, θ_6, θ_7, θ_8, θ_9 e θ_{10}. Utilizando das equações:

$$L = T - V$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_j^{nc}$$

Figura 6: Analise dinâmica do sistema.



Fonte: Autoria Própria.

CONCLUSÕES:

Com o uso de uma ampla bagagem matemática, para desenvolvimento da cinemática e dinâmica do mecanismo junto de conhecimentos de programação e desenvolvimento de aprendizado de máquina foi possível gerar os resultados desejados pela rede neural pelo treinamento da rede com dados teóricos que foram utilizados para averiguar a precisão da resposta fornecida. Sendo contatada a viabilidade da sua inserção em um sistema real para correção do posicionamento, o que pode gerar grande melhoria no controle desses mecanismos dentro das industrias, considerando o baixo erro constatado no uso da rede (inferior a 1%). Sendo ainda possível melhorar os resultados simplesmente expandindo a nuvem de dados fornecida para o treinamento, tendo sido utilizados apenas 1000 pontos para o treinamento.

BIBLIOGRAFIA

MAXIMIANO, Gabriel. **Desenvolvimento de sistema de movimentação corexy para impressora 3d**. Disponível em: <https://docplayer.com.br/146058204-Desenvolvimento-de-sistema-de-movimentacao-corexy-para-impressora-3d.html>. Acesso em: 09/03/2023.

MOYER, Ilan. **Core[X,Y].** Disponível em: <http://www.corexy.com/>. Acessado em 09/03/2023.

TENSORFLOW. **TensorFlow 2 início rápido para iniciantes**. Disponível em: <https://www.tensorflow.org/tutorials/quickstart/beginner?hl=pt-br>. Acessado em: 09/03/2023.

DOUGHTY, Samuel. **Mechanics of Machines**. Reimpressão. Editora Lulu Enterprises Incorporated, 2005.

BAELDUNG. **Normalizing Inputs for an Artificial Neural Network**. Disponível em: <https://www.baeldung.com/cs/normalizing-inputs-artificial-neural-network>. Acessado em: 09/03/2023.