



DE SANTA CATARINA *CAMPUS JOINVILLE* DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA *CAMPUS JOINVILLE*

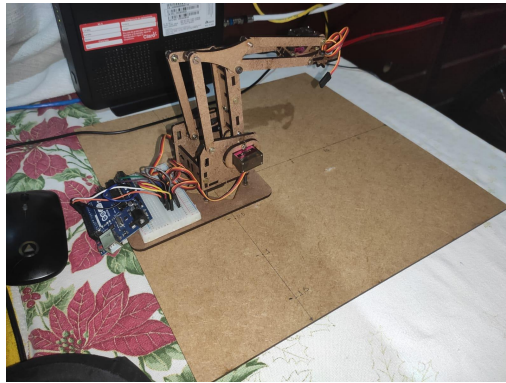
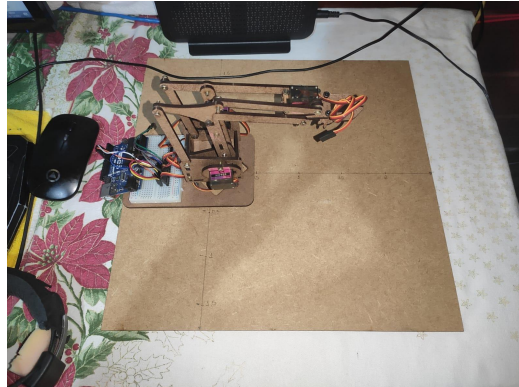
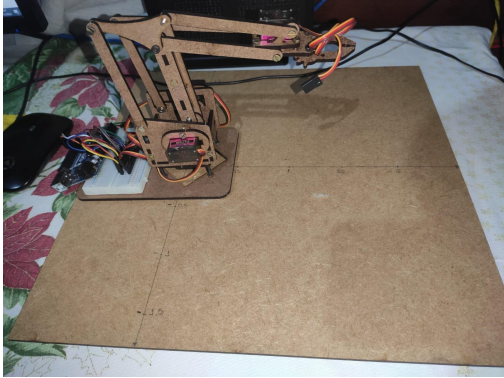
CURSO: Engenharia Elétrica
PROFESSOR: MICHAEL KLUG
UNIDADE CURRICULAR: Robótica
ALUNO: Elias Anzini Junior, Patrick Gabriel

MÓDULO: VIII

Relatório de Robótica

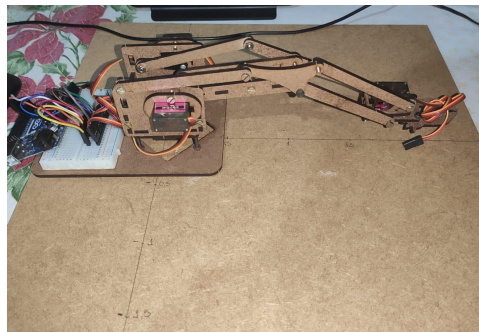
Joinville, 2022

I. Montagem do Robô



II. Definição dos parâmetros de Denavit

Os parâmetros de denavit-hartenberg foram definidos considerando a posição inicial abaixo:



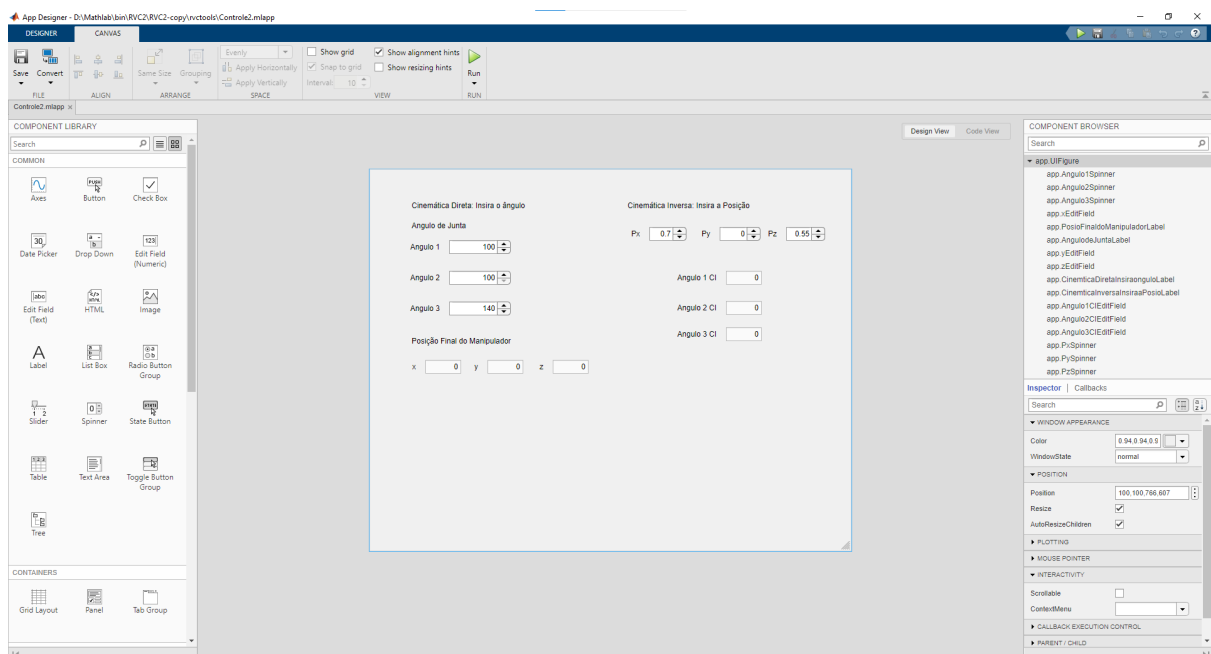
j	theta	d	a	alpha	offset
1	q1	0.45	0	1.5708	0
2	q2	0	0.9	0	0
3	q3	0	1.1	0	0

III. Programação

A programação foi realizada toda no app design do matlab. O guide sugerido apresentou muita lentidão entre os comandos enquanto o app design teve uma resposta mais rápida.

Foi empregada a biblioteca de comunicação com arduino “Matlab Support Package for Arduino Hardware” e também a biblioteca “Robotics Toolbox for Matlab” para criação de um robô virtual e execução da cinemática direta e inversa.

App Design: Dentro do app é possível utilizar as ferramentas de texto e número para interação com o usuário. Através das callbacks para cada interação é chamada a função cinemática direta ou a cinemática inversa.



Exemplo de Callback Cinemática Direta: Quando alterado o ângulo da junta pelo usuário, o servo se movimenta para a posição e envia a variável para a função CinematicaDireta, nela ocorre o cálculo e retorno dos valores x, y e z.

```
% Value changed function: Angulo1Spinner
function Angulo1SpinnerValueChanged(app, event)
    global a
    value1 = (app.Angulo1Spinner.Value);
    % Atualiza Label do Angulo
    a.servoWrite(6,value1);
    CinematicaDireta;
    app.xEditField.Value=x;
    app.yEditField.Value=y;
    app.zEditField.Value=z;
```

Arquivo Cinemática Direta: Sempre quando algum ângulo da junta é alterado ele recebe os valores e calcula a cinemática direta. Foi necessário realizar a correção dos ângulos (-100, -100, -140) para considerar a posição inicial como 0. Para a função funcionar é necessário ter instalado e executado a biblioteca Robotics Toolbox - Peter Corke.

```
thetal = ((app.Angulo1Spinner.Value-100)*pi/180);
theta2 = ((app.Angulo2Spinner.Value-100)*pi/180);
theta3 = ((app.Angulo3Spinner.Value-140)*pi/180);

L(1)= Link([0 0.45 0 pi/2]);
L(2)= Link([0 0 0.9 0]);
L(3)= Link([0 0 1.1 0]);

rob=SerialLink(L)

MT = rob.fkine([thetal theta2 theta3])

x = MT.t(1,1);
y = MT.t(2,1);
z = MT.t(3,1);
```

Exemplo de Callback Cinemática Inversa: Quando alterado a posição do manipulador é executada a função CinematicalInversa.

```
% Value changed function: PxSpinner
function PxSpinnerValueChanged(app, event)
    global a
    CinematicalInversa;
```

Arquivo Cinemática Inversa: Os valores de x, y e z são passados para matriz transposta e através da função robo.ikine é realizado o cálculo da cinemática inversa. O problema é que existem várias posições que a função não está conseguindo retornar os ângulos da juntas, ocorrendo erro no posicionamento.

```
- px=app.PxSpinner.Value;
- py=app.PySpinner.Value;
- pz=app.PzSpinner.Value;

- L(1)= Link([0 0.45 0 pi/2]);
- L(2)= Link([0 0 0.9 0]);
- L(3)= Link([0 0 1.1 0]);

- robo=SerialLink(L)

- T = [ 1 0 0 px;
        0 1 0 py;
        0 0 1 pz;
        0 0 0 1]

- try
-     q = robo.ikine(T, 'mask', [1 1 1 0 0 0])
-     app.Angulo1CIEditField.Value=(q(1)*180/pi)+100;
-     app.Angulo2CIEditField.Value=(q(2)*180/pi)+100;
-     app.Angulo3CIEditField.Value=(q(3)*180/pi)+140;
-     A1 =app.Angulo1CIEditField.Value
-     A2 =app.Angulo2CIEditField.Value
-     A3 =app.Angulo3CIEditField.Value
-     a.servoWrite(6,A1);
-     a.servoWrite(5,A2);
-     a.servoWrite(4,A3);

- catch

- end
```