

# Robô Manipulador Planar

Robótica Industrial 2020

Fábio Rodrigues de Andrade Santos

16/0151783

Engenharia Mecatrônica

santosfabio.ra@gmail.com

**Resumo**—O documento descreve os processos de montagem de um robô planar manipulador e os seus resultados em testes distintos. Com uma análise de resultados e apresentação teórica prévia para um melhor entendimento do projeto, o qual conterá códigos e simulações do próprio manipulador.

## I. OBJETIVO

Construção de um robô com 2 graus de liberdade para se passar em 4 etapas de testes distintas.

## II. INTRODUÇÃO

A necessidade humana de aprimorar a produção, tanto em questão de qualidade de produtos e em produtividade, fez com que se desenvolvesse tecnologias que marcaram pontos na história da humanidade, como a máquina a vapor em 1800 que estabeleceu a primeira revolução industrial. Agora a humanidade vem sendo marcada com o advento da robótica. Os robôs manipuladores vem sendo cada vez mais utilizados na manufatura de produtos, fábricas de carro são um grande exemplo de fabricações que utilizam braços robóticos.

A robótica é uma área a qual envolve diversas subáreas, em maior foco a engenharia mecânica, elétrica e da computação. O robô veio como uma ferramenta com o intuito de ser altamente eficiente e precisa, além de oferecer um meio de trabalhar com segurança em situações adversas, como o fundo do oceano. Para um melhor entendimento da robótica, é necessário uma definição de robô. Segundo a RIA, um robô é um manipulador reprogramável projetado para realizar uma variedade de tarefas com movimentos programados para tal realização.

O crescimento da utilização de robôs nas indústrias vem sendo dado pela inovação tecnológica, a qual permite preços cada vez mais acessíveis para tal projetos, mais também por uma crescente demanda de qualidade, eficácia e condições de trabalhos. O robô é um investimento realizado por empresas com um resultado significativo, sendo que o retorno financeiro deste investimento é estimado em um ano, de acordo com um estudo realizado pela *ABB Robotics*.

## III. PROJETO

### A. Especificações do Robô

O robô montado contém 2 graus de liberdade onde suas juntas são de rotação em torno do eixo Z, sendo assim sua área de atuação um plano.

- Distância entre os eixos: 100mm
- Motor: Micro Servomotor 9g (cada junta)

- Efetuador Terminal: Lápis
- Controlador: Arduino Uno

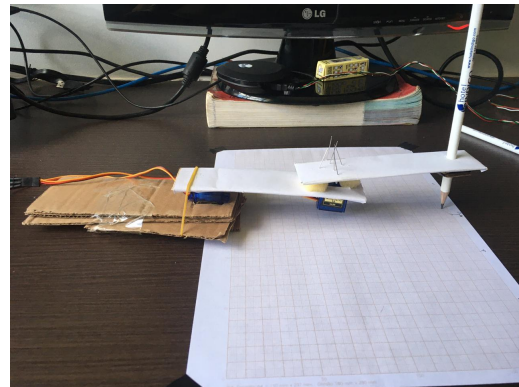


Figura 1. Manipulador Robótico

O manipulador apresentado foi montado com papelão e apresenta na região do seu ombro 2 espumas e 6 alfinetes para uma maior estabilidade do material. A base de papelão foi feita para se aplicar um peso que seguraria a movimentação do robô que fosse indesejada.

### B. Fases

O robô deve passar por 4 fases de testes, sendo elas:

- 1) Uma junta se move somente após a anterior tiver chegado a configuração final estabelecida
- 2) As duas juntas movem-se simultaneamente com a mesma velocidade, então a com maior variação angular demora mais tempo para completar seu trajeto
- 3) As duas juntas movem-se simultaneamente com velocidades distintas e tempos de percurso iguais
- 4) As juntas movem-se simultaneamente com velocidades distintas fazendo o efetuador terminal descrever uma trajetória linear até o ponto de parada

O funcionamento do servo motor é na escala de de  $0^\circ$  a  $180^\circ$ , então para contornar essa limitação e se pudesse atingir valores os quais um dos motores deveria girar em sentido horário, ângulos negativos, os motores foram iniciados em  $90^\circ$ . Com isso o intervalo  $[0^\circ \ 180^\circ]$  se equivale a  $[-90^\circ \ 90^\circ]$ .

### C. Modelagem

Para a modelagem do manipulador robótico foi usada a *Robotcis Toolbox*

Tabela I  
TABELA DENAVIT-HARTBERG

i	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$\theta_1$	0	100mm	0
2	$\theta_2$	0	100mm	0

Código Matlab:

- 1)  $dh = [0 \ 0 \ 100 \ 0; 0 \ 0 \ 100 \ 0];$
- 2)  $R = SerialLink(dh);$
- 3)  $R.name = 'Robo Planar';$
- 4)  $q = [0 \ 0];$
- 5)  $R.plot(q)$
- 6)  $R.teach$

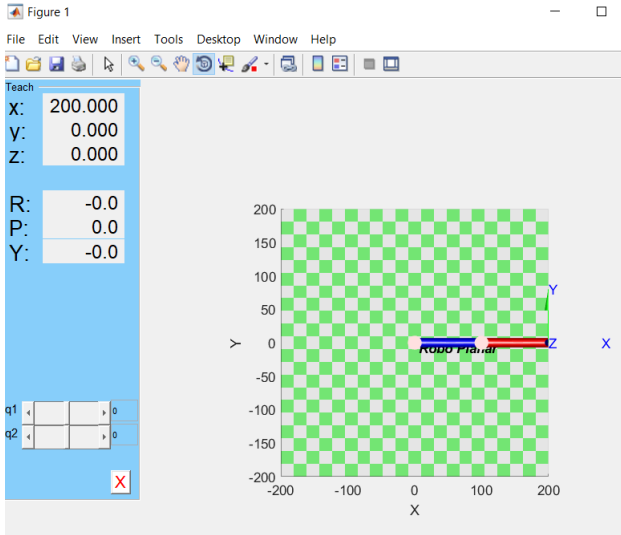


Figura 2. Modelagem do Robô Manipulador

#### D. Cálculos

- Cinemática Direta: A cinemática direta foi feita pelo método auxiliado pela tabela de Denavit-Harteberg (Tabela I)

$$A_0^2 = A_0^1 * A_2^1 \quad (1)$$

$$A_0^1 = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1 * \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1 * \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2^1 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 * \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 * \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

O resultado final será uma matriz 4x4, porém as coordenadas X e Y estão localizadas na última coluna onde a primeira linha é X e a segunda Y. Então:

$$X = a_1 * \cos(\theta_1) + a_2 * \cos(\theta_1) * \cos(\theta_2) - a_2 * \sin(\theta_1) * \sin(\theta_2) \quad (2)$$

$$Y = a_1 * \sin(\theta_1) + a_2 * \cos(\theta_1) * \sin(\theta_2) + a_2 * \sin(\theta_1) * \cos(\theta_2) \quad (3)$$

$$X = a_1 * \cos(\theta_1) + a_2 * \cos(\theta_1 + \theta_2)$$

$$Y = a_1 * \sin(\theta_1) + a_2 * \sin(\theta_1 + \theta_2)$$

- Cinemática Inversa: A cinemática inversa foi feita com o método geométrico, uma simples representação do robô foi construída para uma melhor visualização dos resultados

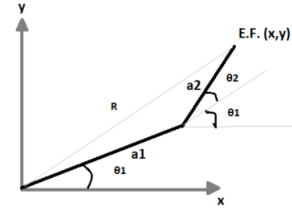


Figura 3. Cinemática Inversa

Tem-se que  $R^2 = x^2 + y^2$ , sendo que com o triângulo formado entre o eixo das abscissas e a junta que define o braço  $a_1$  e outro triângulo com essa mesma junta até o efetuador terminal, que segue a mesma lógica, mas leva-se em consideração a distância que este se encontra afastado da origem, é possível notar que:

$$x^2 + y^2 = a_1^2 + a_2^2 + 2a_1a_2\cos\theta_2$$

$$x^2 + y^2 = a_1^2 + a_2^2 + 2a_1a_2(\cos(\theta_1)\cos(\theta_1 + \theta_2) + \sin(\theta_1)\sin(\theta_1 + \theta_2))$$

Sustituindo  $a_1$  e  $a_2$  por 100, obtém-se o resultado final igual a:

$$\theta_2 = + - \arccos\left(\frac{x^2 + y^2 - 2(100^2)}{2 * 100^2}\right) \quad (4)$$

Com o rascunho a seguir é possível ver que:

$$tg(\alpha) = (a_2\sin(\theta_2))/(a_1 + a_2\cos(\theta_2))$$

$$tg(\alpha + \theta) = y/x$$

Resultando em:

$$\theta_1 = \arctg\left(\frac{y(100(1 + \cos(\theta_2))) - x * 100\sin(\theta_2)}{x(100(1 + \cos(\theta_2))) + y * 100\sin(\theta_2)}\right) \quad (5)$$

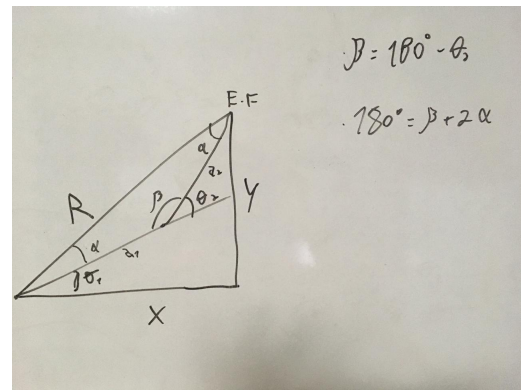


Figura 4. Rascunho Auxiliar

#### E. Programação do Robô

A programação do robô é feita em duas etapas. A primeira etapa é no Matlab, o qual definirá qual fase o robô irá reproduzir e com as coordenadas que o efetuador terminal

deve alcançar, fazer os cálculos cinemáticos para determinação dos ângulos que cada junta deverá mover. Após isso, o Arduino recebe essas variáveis por meio da porta serial e faz a movimentação das juntas baseado na fase decidida.

- Matlab

```

1 %Montagem do Robo e sua simulação
2 dh = [0 0 10 0; 0 0 10 0];
3 R = SerialLink(dh,'name','Robo Planar');
4 q= [0 0];
5 R.plot(q);
6 %Porta Serial
7 s= serial('COM4');
8 fopen(s);%lembrar de fechar
9 exit = 1;
10 while(exit)
11 %Interção Usuário
12 fase = input('Qual a fase do projeto: ');
13 x = input('coordenada x: ');
14 y = input('coordenada y: ');
15 if ((fase<=0 || fase >3) || (x>20 || x<0) || (y<-20 || y>20))
16     fase = 1;
17     x = 20;
18     y = 0;
19     disp('Entrada Inválida! Manipulador retornará para posição inicial')
20 end
21 fase = num2str(fase);
22 %cinematica inversa + conversão para graus
23 T = transl(x,y,0);
24 movimento= R.lkine(T,'q0',[0 0],'mask',[1 1 0 0 0 0]);
25 angulos = rad2deg(movimento);
26 theta1 = round(angulos(1,1),3)+90;
27 theta2 = round(angulos(1,2),3)+90;
28 t1 = num2str(theta1);
29 t2 = num2str(theta2);
30 %Interação Arduino
31 envio = ''+fase+' '+t1+' '+t2
32 fprintf(s,envio);
33 exit = input('Aperte 0 para sair e 1 para continuar: ');
34 end
35 %Fechando Serial
36 fclose(s);

```

Figura 5. Código Matlab

O código da figura 5 apresenta inicialmente uma modelagem semelhante a apresentada na seção III-C, excluindo somente a última linha, a qual permite o controle de posição na própria simulação. Em seguida faz-se a conexão com a porta serial para a comunicação com o Arduino Uno e finalmente entra em um *loop* para a interação com usuário.

Neste *loop* é pedido 3 entradas distintas, a fase que o robô deverá realizar e as coordenadas x e y, respectivamente. Após uma validação dessas entradas, que caso tenha algum valor inconsistente, o robô retornará a origem e será impresso na tela que houve erro. Com os valores no plano cartesiano definidos e validados, é feita o cinemática inversa para obtenção dos ângulos, que são convertidos de radianos em graus, para serem enviados ao processador. Logo que a operação finaliza, o Usuário pode acabar o programa apertando 0 ou reiniciar o processo. Finalizando o código, o fechamento da porta serial é realizado.

- Arduino

Essa etapa do projeto se refere ao controlador do robô. Inicialmente são incluídas as bibliotecas necessárias e declaradas as variáveis globais usadas nas seções *setup* e *loop*.

Na seção de *setup* os motores são estabelecidos nas entradas da placa e seus ângulos são postos em 90°.

A parte do código que ficará em *loop* durante o processo foi separada em, primeiramente a interação com o Matlab e em seguida entra em um *switch* de acordo com

a entrada da variável fase. Cada opção equivale ao código necessário para a determinada fase, apresentadas na seção III-B. O fluxograma a seguir apresenta a lógica das fases que foram implementadas e testadas.

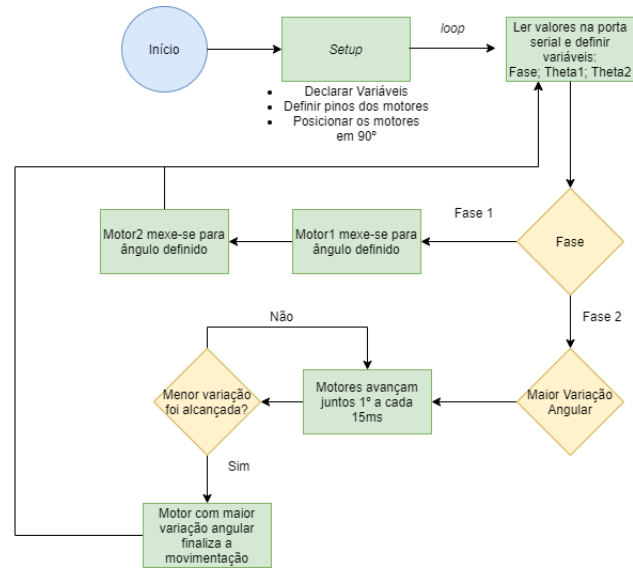


Figura 6. Lógica Arduino

## IV. TESTES E RESULTADOS

Os testes foram realizados escolhendo 4 pontos válidos aleatoriamente no plano cartesiano e anotando os erros nas direções X e Y.

### A. Fase I

Ponto(X,Y)	$\Delta X(mm)$	$\Delta Y(mm)$
$P_0(12,9)$	5	5
$P_1(8,-11)$	3	4
$P_2(5,13)$	7	6
$P_3(11,-11)$	1	1

Tabela II  
PONTOS E ERROS DA FASE I

```

Command Window
>> Trab
Qual a fase do projeto: 1
coordenada x: 12
coordenada y: 9

envio =

"1 85.46 172.819"

fx Aperte 0 para sair e 1 para continuar: |

```

Figura 7. Command Window  $P_0$



Figura 8. Ponto  $P_0$

### B. Fase II

Ponto(X,Y)	$\Delta X(mm)$	$\Delta Y(mm)$
$P_0(8,-11)$	3	6
$P_1(13,-13)$	0	4
$P_2(9,5)$	10	11
$P_3(15,9)$	4	5

Tabela III

PONTOS E ERROS DA FASE II

```

Aperte 0 para sair e 1 para continuar: 1
Qual a fase do projeto: 2
coordenada x: 8
coordenada y: -11

envio =

"2 83.178 -4.301"

```

Figura 9. Command Window  $P_0$

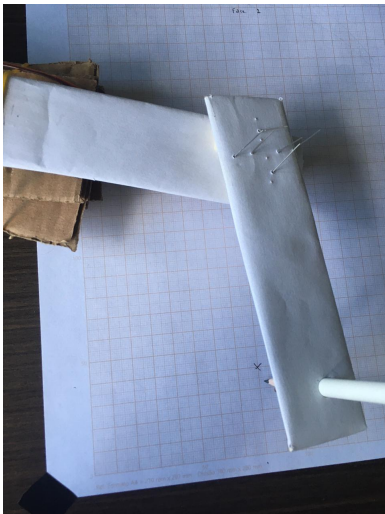


Figura 10. Ponto  $P_0$

## V. ANÁLISES DE RESULTADOS

Fase	Média $\Delta X$	Média $\Delta Y$	$\sigma X$	$\sigma Y$
I	4mm	4mm	2.58199	2.16025
II	4.25mm	6.5mm	4.19325	3.10913

Tabela IV  
MÉDIA DE ERRO

Pelas tabelas II, III e IV é possível ver que o robô segue as trajetórias desejadas com um resultado satisfatório. Pode-se notar que quanto mais próximo a origem, maior será o erro do manipulador, como os casos dos pontos  $P_2$  da fase I com o erro em X de 7mm e  $P_2$  da segunda fase que ambas coordenadas obtiveram os maiores erros dos testes.

## VI. CONCLUSÃO

O manipulador apresenta uma estabilidade não ideal, porém mesmo com essa falha, foi possível obter resultados satisfatórios. O objetivo das 4 fases não foi alcançado, porém as fases implementadas obtiveram bons resultados. Para a implementação das fases restantes não é necessário mexer na estrutura do robô, somente na programação do mesmo.

## REFERÊNCIAS

- [1] João Maurício Rosário, Princípios de Mecatrônica.,