



DOCUMENTO DE REFERÊNCIA

BRB-AS5



2020

RESUMO

Este documento visa trazer ao usuário informações necessárias para o melhor controle do braço robótico antropomórfico, abordando as questões da cinemática, programação e eletrônica do manipulador.

Palavras-chave: Robótica. Programação. Trajetória.



SUMÁRIO

1. INTRODUÇÃO.....	4
2. DESCRIÇÃO AVANÇADA.....	5
2.1. CINEMÁTICA.....	5
3. EXEMPLOS.....	7
3.1. CIRCUITO ELÉTRICO/ELETRÔNICO.....	8
3.2. PROGRAMAÇÃO.....	9



1. INTRODUÇÃO

Com o aumento da tecnologia, os robôs passaram a ocupar um espaço cada vez mais significativo na indústria. Dessa forma, os investimentos em pesquisas na área de robótica aumentaram exponencialmente. Estas pesquisas não se restringem somente no âmbito industrial, mas também para o desenvolvimento de robôs capazes de interagir com seres humanos ou até mesmo realizar tarefas cotidianas. Com isso, os sistemas robóticos deixaram de ser exclusivamente utilizados na indústria e passaram a adotar novas aplicações, o que permitiu o desenvolvimento de robôs antropomórficos. A BRBOTS pensou nessa situação e se desenvolveu através da criação e desenvolvimento de dispositivos robóticos de forma acessível.

Um robô antropomórfico recebe este nome, devido à semelhança da configuração de seus elos e juntas com a anatomia de um braço humano. É uma categoria de robô muito utilizado na indústria, largamente empregado para transposição de cargas, nos processos de pintura e soldagem. Dessa forma, cada configuração de robôs possui uma aplicação. Veja exemplos do catálogo de produtos da nossa empresa.



Figura 1: Manipuladores da BRBOTS



2. DESCRIÇÃO AVANÇADA

O manipulador que você adquiriu possui 5 graus de liberdade de cadeia aberta, com 5 juntas de rotação, possuindo efetuator de movimento paralelo, cujo punho possui 2 movimentos independentes: Arfagem (pitch), Rolamento (roll).

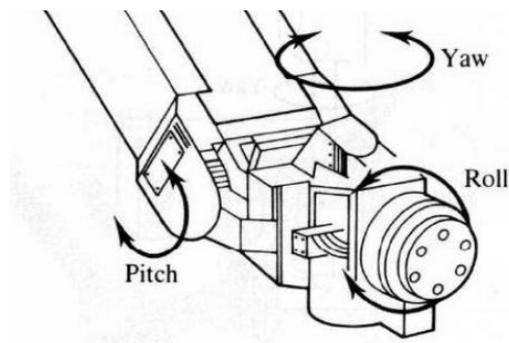
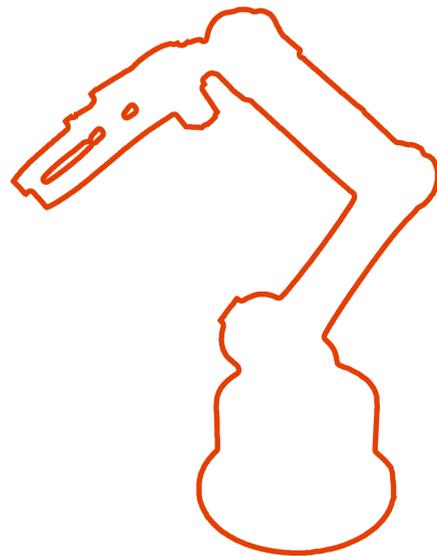


Figura 2: Representação dos 3 movimento de um punho robótico esférico genérico

A escolha do material utilizado para a impressão 3D da sua estrutura foi resultado de análises e testes minuciosos. O material que apresentou o melhor aproveitamento, com base nos desejos da equipe de engenheiros para o manipulador, foi o ABS, devido a suas propriedades mecânicas, como a resiliência (resistência) e a densidade.

Seus atuadores são servomotores MG996R e SG90. Eles foram escolhidos devido ao alto torque necessário para o movimento desejado e seus baixos custos.



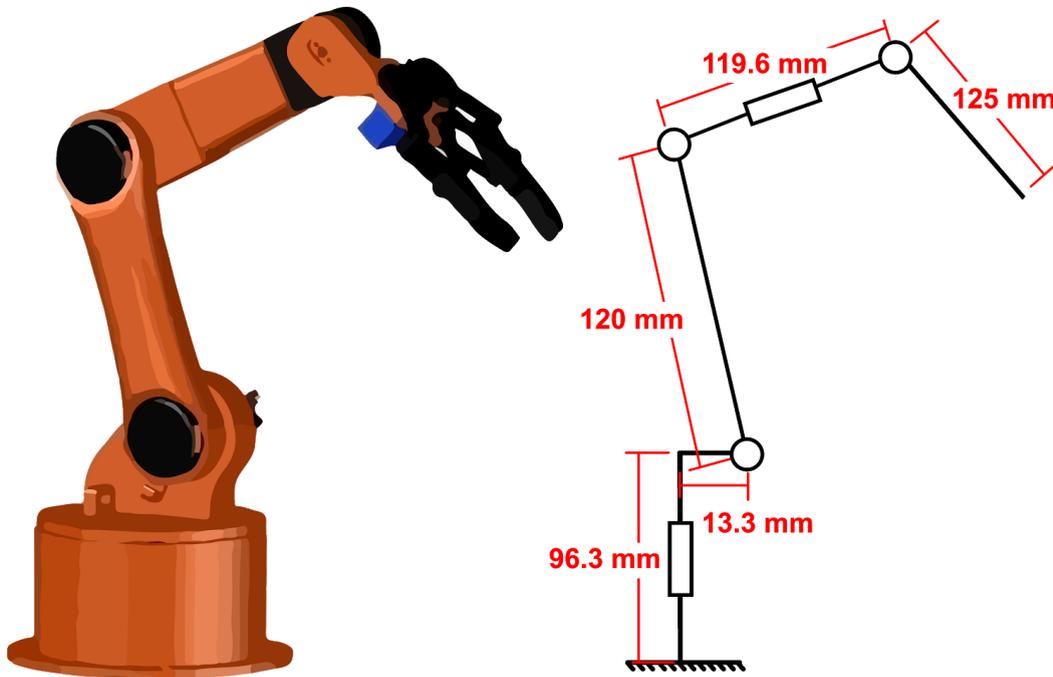
→ Itens inclusos:

- a. Manipulador robótico com 5 graus de liberdade;
- b. 6 servo motores
 - i. 3x MG996R;
 - ii. 3x SG90;

2.1. CINEMÁTICA

Para a cinemática, pode ser adotada a notação de Denavit-Hartenberg, que facilita os cálculos da posição do efetuador em relação à posição da base. Tal notação utiliza de 4 parâmetros para descrever a posição do efetuador de um manipulador robótico de acordo com os ângulos em cada uma de suas juntas.

As medidas do manipulador podem ser observadas na imagem a seguir.



Figuras 3 e 4: Geometria do manipulador e sua cadeia cinemática

Com as medidas definidas, deve-se encontrar o eixo de coordenadas em cada uma das juntas da cadeia cinemática, como na imagem a seguir.

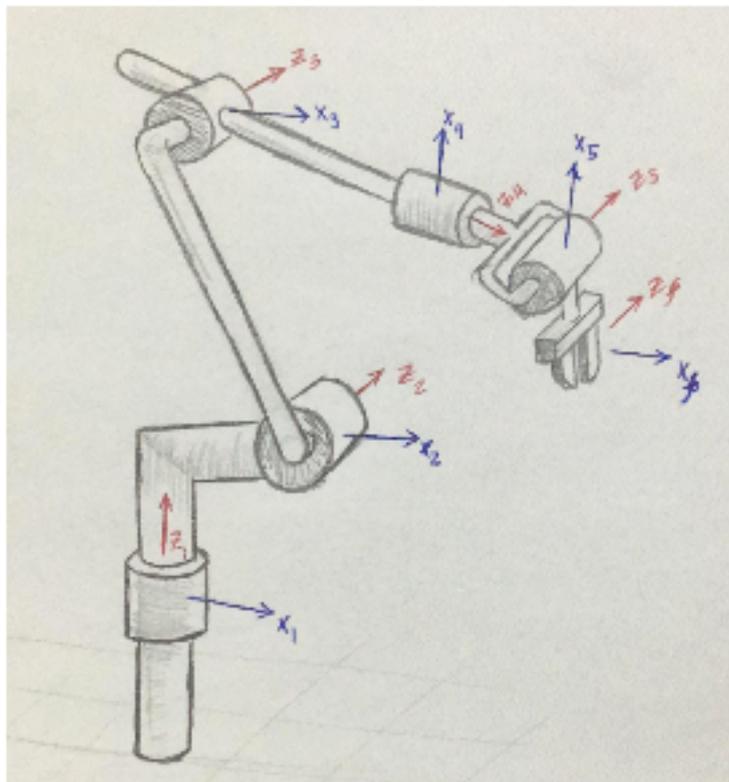


Figura 8: Eixo de coordenadas para os parâmetros de DH

Após a análise, é possível encontrar os parâmetros de Denavit-Hartenberg como:

θ (rad)	d (mm)	a (mm)	α (rad)
θ_1	96,3	13,3	$-\pi/2$ (-90°)
θ_2	0	120	0
θ_3	0	0	$-\pi/2$ (-90°)
θ_4	119,6	0	$\pi/2$ (90°)
θ_5	0	40	$-\pi/2$ (-90°)

Tabela 1: Parâmetros de DH

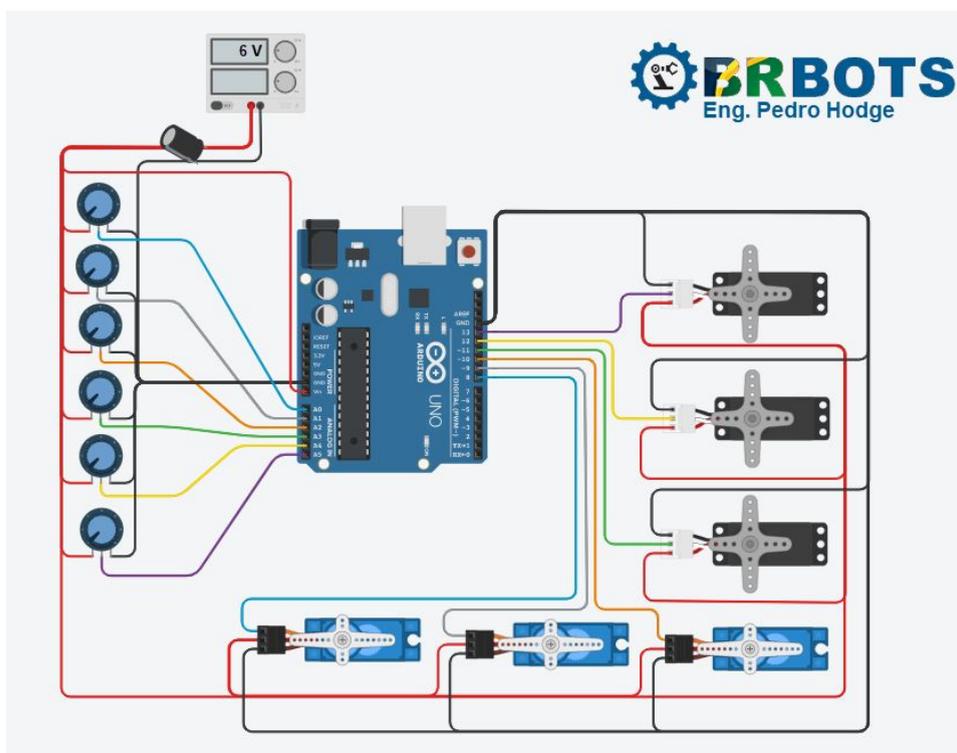
3. EXEMPLOS

Como muitos usuários não possuem conhecimento técnico avançado, esse documento proporciona, também, um simples exemplo de montagem de circuito e programação em Arduino para controle do manipulador adquirido.

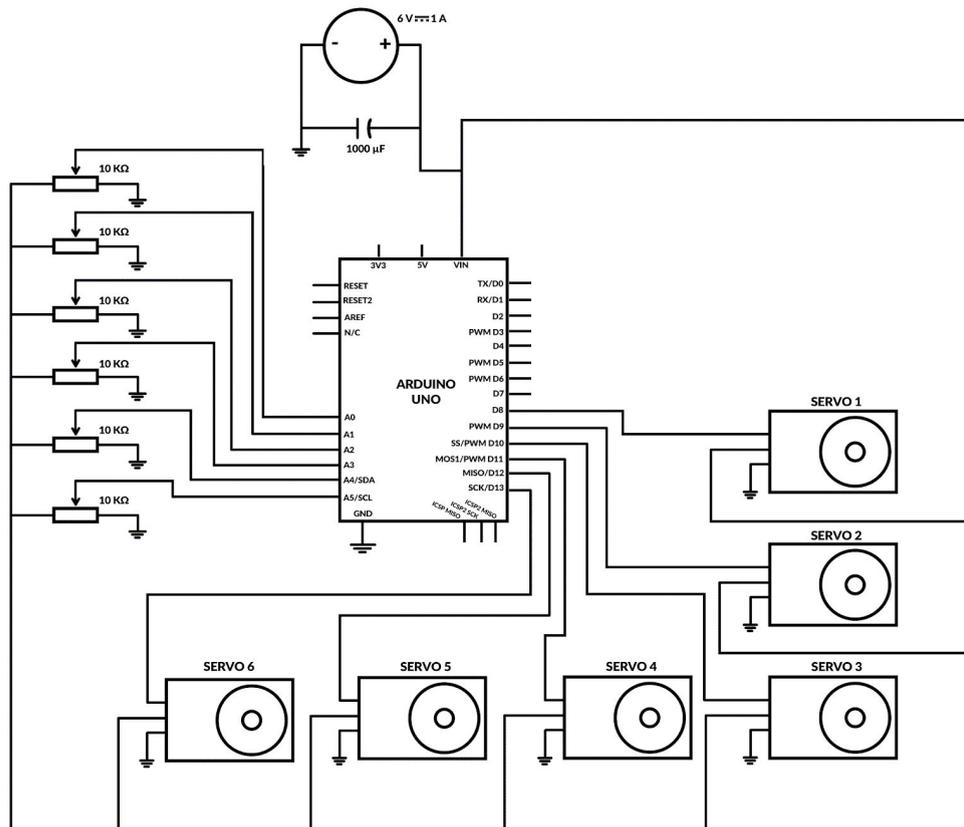
3.1. CIRCUITO ELÉTRICO/ELETRÔNICO

A montagem eletrônica utilizada neste exemplo é, basicamente, um circuito para controlar o ângulo em cada servo motor do robô através de potenciômetros lidos pelo Arduino através de suas portas analógicas.

A montagem do circuito foi feita utilizando o Arduino UNO, porém, pode ser feita em outros controladores da família Arduino.



Figuras 5: Circuito simulado para o controle do manipulador



Figuras 6: Circuito técnico para o controle do manipulador

Os servos possuem 3 fios em sua estrutura: O vermelho, para alimentação +5V; O preto, ou marrom, para o GND (terra); e o laranja, ou branco, será conectado ao pino digital correspondente ao motor no Arduino.

Os potenciômetros também têm 3 pinos de saída, sendo que os das extremidades são conectados ao GND e ao +5V e o pino central será conectado ao pino analógico correspondente no Arduino. O valor da resistência pode ser escolhida conforme a preferência, porém, para este exemplo, foi utilizado 10kΩ.

Além disso, é necessário conectar uma fonte externa de 6V e 1A, tanto para alimentar o Arduino quanto para alimentar os servos, já que o controlador não consegue prover carga suficiente.

3.2. PROGRAMAÇÃO

Para controlar os servos através dos potenciômetros, será utilizado utilizar a função `map`, que nesse caso, converte o valor lido na porta analógica do Arduino (entre 0 e 1023 bits), para um valor entre 0 e 180, onde o 0 equivale a 0° e o 180

equivale a 180° no servo motor. Para realizar o acionamento dos servos motores, foi utilizada uma biblioteca, já inclusa na IDE do Arduino, chamada “Servo.h”.

Conecte o Arduino ao computador, abra a IDE, confira a placa (que nesse exemplo é o Arduino UNO), a porta serial utilizada e, em seguida, carregue o código na placa.

Um código de exemplo se encontra a seguir.

```
1  #include <Servo.h>
2
3  // Cria os objetos do tipo Servo
4  Servo servo1, servo2, servo3, servo4, servo5, servo6;
5
6  void setup () {
7      //Anexa os servos aos respectivos pinos digitais do Arduino
8      servo1.attach (8);
9      servo2.attach (9);
10     servo3.attach (10);
11     servo4.attach (11);
12     servo5.attach (12);
13     servo6.attach (13);
14 }
15
16 void loop () {
17     //Lê os valores dos potenciômetros
18     int angulo1 = analogRead (0);
19     int angulo2 = analogRead (1);
20     int angulo3 = analogRead (2);
21     int angulo4 = analogRead (3);
22     int angulo5 = analogRead (4);
23     int angulo6 = analogRead (5);
24
25     //Mapeia os valores de 0 a 180 graus
26     angulo1 = map (angulo1, 0, 1023, 0, 180);
27     angulo2 = map (angulo2, 0, 1023, 0, 180);
28     angulo3 = map (angulo3, 0, 1023, 0, 180);
29     angulo4 = map (angulo4, 0, 1023, 0, 180);
30     angulo5 = map (angulo5, 0, 1023, 0, 180);
31     angulo6 = map (angulo6, 0, 1023, 0, 180);
32
33     //Passa o ângulo para os servos
34     servo1.write (angulo1);
35     servo2.write (angulo2);
36     servo3.write (angulo3);
37     servo4.write (angulo4);
38     servo5.write (angulo5);
```



```
39     servo6.write (angulo6);
40
41     //Delay de 15ms entre as leituras
42     delay (15);
43 }
```

Após carregar o código, gire os eixos dos potenciômetros para controlar a posição dos servo motores. As variáveis “angulo’n”, onde ‘n’ é o índice da variável (“angulo1”, por exemplo), recebem os valores resultantes da função “map()” e atribuem à função “servo.write()” o valor do ângulo para qual o servo deve se mover.

Agora, repare no trecho abaixo.

```
25     //Mapeia os valores de 0 a 180 graus
26     angulo1 = map (angulo1, 0, 1023, 0, 180);
27     angulo2 = map (angulo2, 0, 1023, 0, 180);
28     angulo3 = map (angulo3, 0, 1023, 0, 180);
29     angulo4 = map (angulo4, 0, 1023, 0, 180);
30     angulo5 = map (angulo5, 0, 1023, 0, 180);
31     angulo6 = map (angulo6, 0, 1023, 0, 180);
```

Os valores 0 e 180 na função “map()” podem ser diferentes para cada motor. Eles representam a faixa de movimento possível para o servo motor e variam de acordo com a calibragem feita na hora da montagem do manipulador.

