

Sobre o desenvolvimento do controle de uma máquina delta para aplicações industriais

Gabriel da Silva Teixeira
Jhonatan da Silva
Pedro Henrique Medeiros Godoi
Sarah Emmendorfer
Florianópolis, Santa Catarina

1. Introdução geral sobre manipuladores Delta

Existem quatro tipos de robôs industriais: Articulado, cartesiano, delta e SCARA (*Selective Compliance Assembly Robot Arm*). Algumas aplicações exigem alta velocidade, nesses casos faz-se do uso de manipuladores Delta.

Robôs delta são usados principalmente em aplicações onde é necessário pegar produtos em grupos e colocá-los em um container ou ajustá-los para um padrão de montagem.[1]. Eles são amplamente utilizados na indústria e são fabricados por grandes empresas como FANUC, ABB, Festo, Kawasaki, Omron etc.

Figura 1: Fanuc Delta



Fonte : FANUC

2. Proposta de projeto

Temos como objetivo principal realizar a construção da parte eletrônica e de controle do projeto para fins de demonstrações didáticas tanto quanto posterior utilização da máquina no projeto de módulo IX onde a mesma irá ser parte de um módulo de montagem industrial.

2.1. Requisitos e restrições

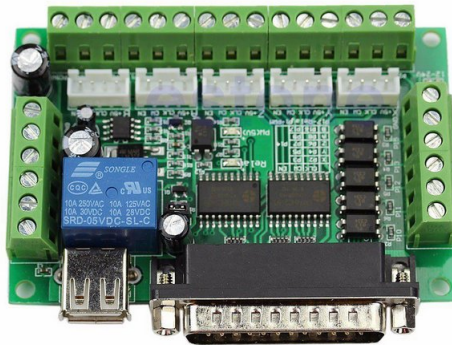
Alguns requisitos foram levantados devido a demanda da utilização da máquina como uma das partes do projeto integrador do módulo IX como

- Hardware funcional para diferentes níveis lógicos;
- Máquina deverá ser modular de maneira que permita interfaceamento por meio de barramentos com outros tipos de hardwares e softwares como LinuxCNC, FPGA, Raspberry PI, Arduino e outros;
- Permitir a modularização do manipulador, para que exista a possibilidade de ser testado diferentes tipos de manipuladores para diferentes tipos de aplicações;
- Interface gráfica

2.2. Especificações

- Permitir opção de controle da máquina via placa LPT-CNC-A, Figura 2

Figura 2: Placa LPT-CNC-A

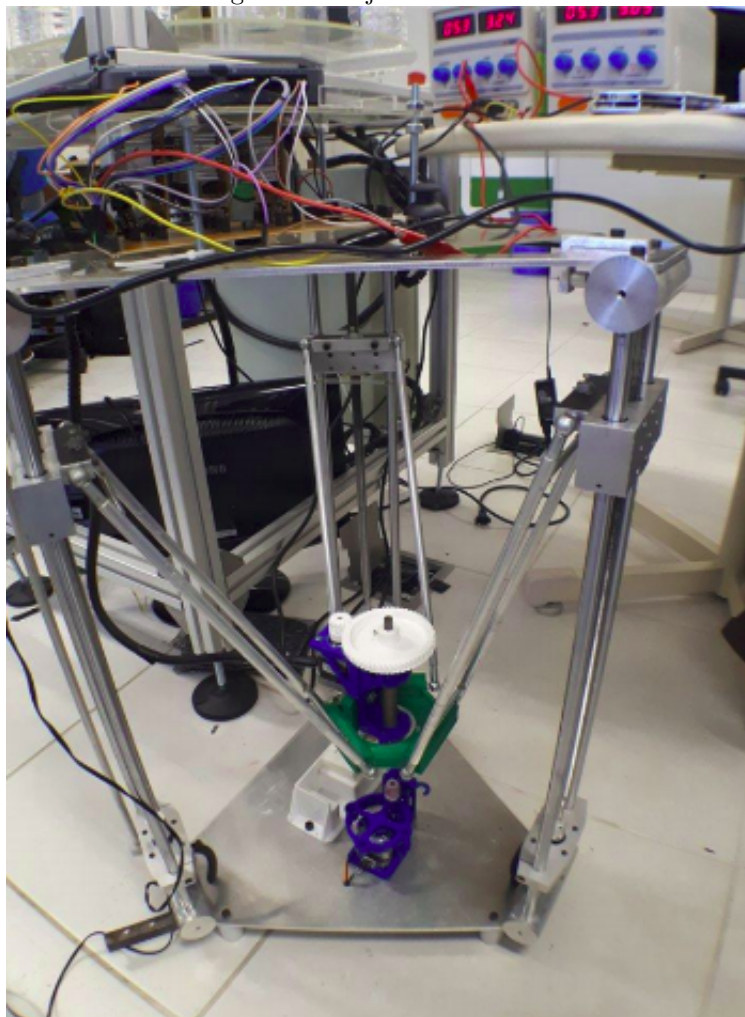


Fonte : Eletrogate

3. Mudanças na topologia da máquina

O projeto inicial apresentava a seguinte topologia, os fios dos motores de passo eram passados pelo lado da estrutura via dois tubos, depois eram colocados na placa projetada com 3 drivers feito pelos alunos que construíram a máquina.

Figura 3: Projeto inicial delta

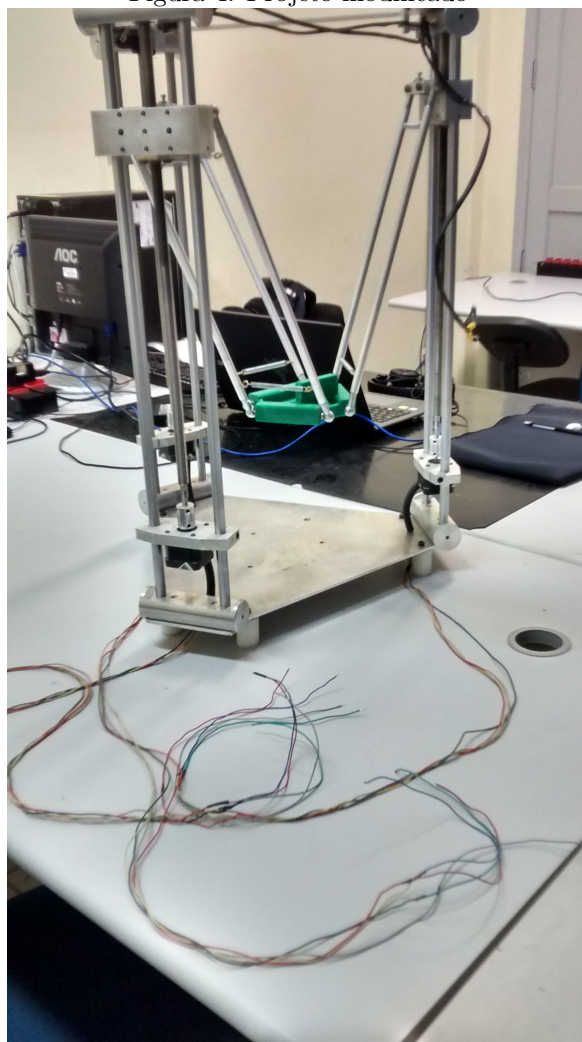


Fonte : Bruno Duarte

A topologia da máquina foi mudada a fim de contemplar os requisitos e

especificações desse projeto. A placa de controle da delta que ficava no topo da estrutura foi desmontada, os fios foram passados para a parte inferior.

Figura 4: Projeto modificado



Fonte : Autoria própria

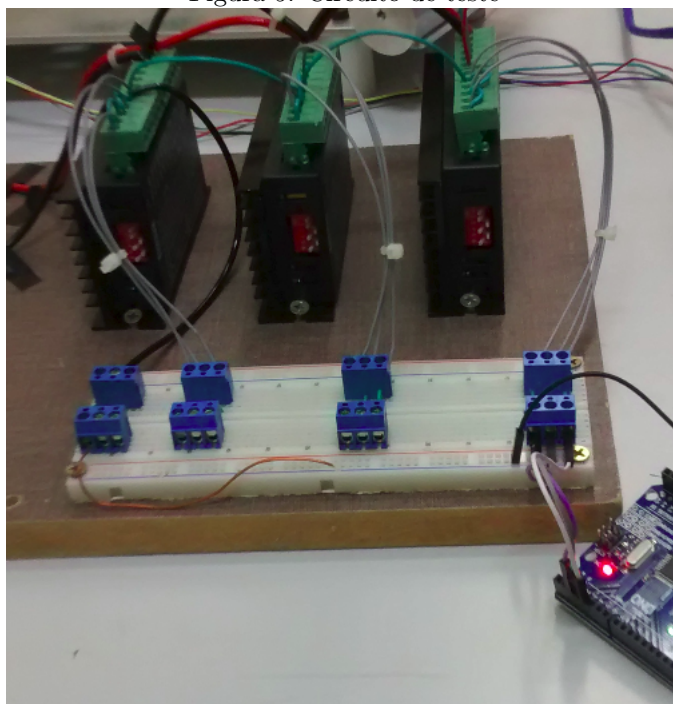
3.1. Primeiros testes

Para testes iniciais foi feita um estrutura simples com os drivers de maneira que poderíamos conectá-los tanto nos motores quanto no arduino.

3.2. *Testando todos os motores*

Para o teste com todos os motores fora feito uma estrutura um pouco mais robusta de maneira que ficasse mais simples a conexão e desconexão dos cabos.

Figura 5: Circuito de teste



Fonte : Autoria própria

3.3. Código de teste dos 3 motores

```
1  const int ena1 = 4;
   const int dir1 = 3;
3  const int pul1 = 2;

5  const int ena2 = 7;
   const int dir2 = 6;
7  const int pul2 = 5;

9  const int ena3 = 8;
   const int dir3 = 9;
11 const int pul3 = 10;

13 const int intervalo = 350;

15 boolean pulso = LOW;

17 void setup() {
   pinMode(ena1,OUTPUT);
19   pinMode(dir1,OUTPUT);
   pinMode(pul1,OUTPUT);
21   digitalWrite(ena1,LOW);
   digitalWrite(dir1,HIGH);
23   digitalWrite(pul1,HIGH);

25   pinMode(ena2,OUTPUT);
   pinMode(dir2,OUTPUT);
27   pinMode(pul2,OUTPUT);
   digitalWrite(ena2,LOW);
29   digitalWrite(dir2,LOW);
   digitalWrite(pul2,HIGH);

31   pinMode(ena3,OUTPUT);
   pinMode(dir3,OUTPUT);
33   pinMode(pul3,OUTPUT);
   digitalWrite(ena3,LOW);
35   digitalWrite(dir3,LOW);
   digitalWrite(pul3,HIGH);
37

39 }

41 void loop() {
   pulso = !pulso;
43   digitalWrite(pul1,pulso);
   digitalWrite(pul2,pulso);
```

```
45 | digitalWrite(pul3,pulso);  
    | delayMicroseconds(intervalo);  
47 | }
```

4. Referências

- [1] **Motion Control Robotics**,
Disponível em : <<https://motioncontrolsrobotics.com/delta-robots>>
Acesso em : 11/08/2018 09:19