Universidade Federal de Minas Gerais – UFMG
Introdução à Robótica
Relatório do Trabalho Prático 2

Alunos: Iuri Bueno Drumond de Andrade
Leandro Mercêdo Moreira Branco
Mateus Vilas Boas
Matheus Brasileiro Passos

Objetivo

- Maior Familiarização com a HandyBoard e com o ambiente de programção IC.
- Adquirir noções sobre o funcionamento de sensores e a melhor utilização destes com a HandyBoard.
- Noções basicas sobre processamento dos sinais adquiridos atraves dos sensores.

Montagem do robô

As unicas modificações no robô foram as intalações dos sensores e a adição de um pequeno compartimento para a realização da identificação dos blocos, cujo objetivo é manter as condições de luminosidade constanstes.

Dificuldades encontradas

- Lidar com os sensores

Os sensores se mostraram um pouco ineficientes quando a superficie do objeto a ser identificado nao era liso (exemplo:Blocos fornecidos, eram muito porosos o que dificulta a reflexão da luz proveniente do emissor). Tambem apresentaram resultados ruins quando havia muito reflexão na superfie (exemplo: plista sobre a qual foi montada a trajetoria com a linha preta quando esta estava sujeita a forte iluminação artificial).

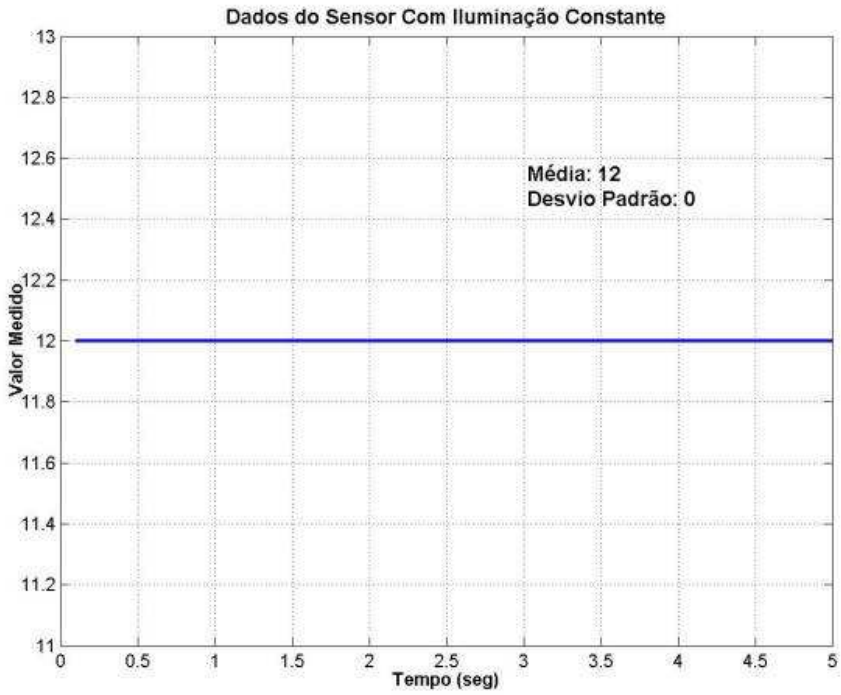
- Resolução dos problemas

A solução é tratar os sinais dos sensores por meio de algoritimos adequados a cada aplicação.

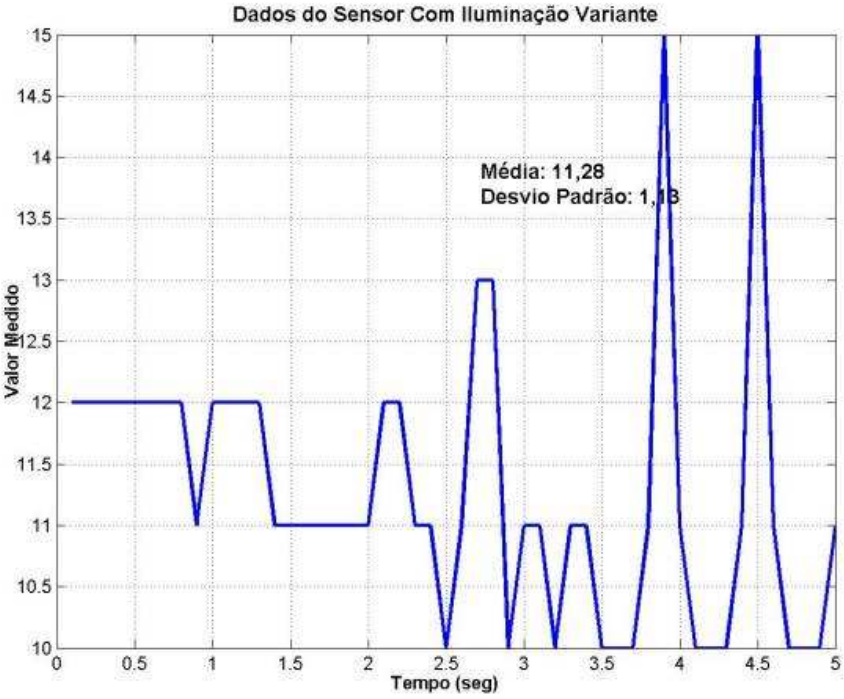
Resultados Obtidos dos testes

Apos a realização dos testes propostos obtivemos os seguintes resultados:

Iluminação constante, distância 3mm

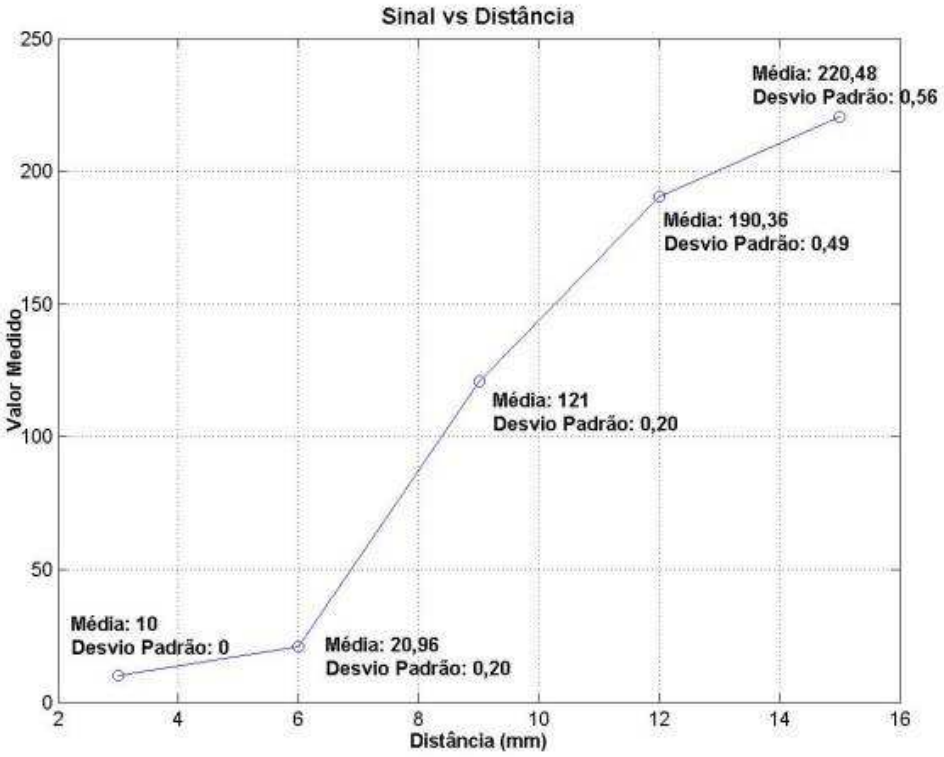


Sem Iluminação constante, distância 3mm



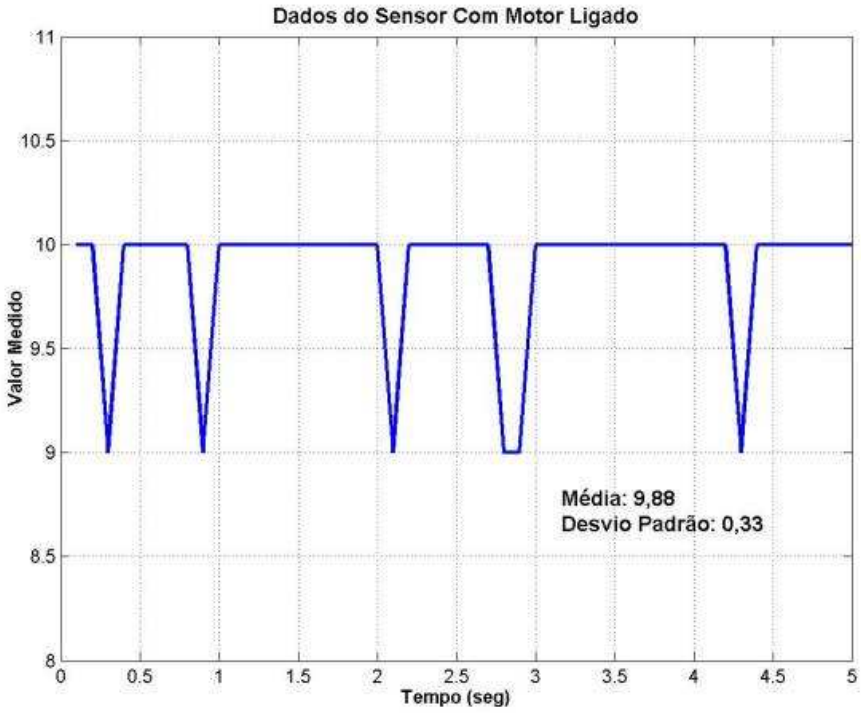
Podemos observar que a variação da luminosidade causou tambem uma flutuação do sinal em torno da média e um deslocamento para baixo da média.

Curva Sinal x Distância



O comportamento é não linear, mas para distâncias menores que 10mm e maiores que 3mm podemos fazer uma aproximação linear razuavel.

Efeito dos motores na leitura do sensor



Os motores provocam oscilação do sinal e queda no valor médio.

Validação

Para distâncias diferentes das anteriores obtivemos 4 medidas e 4 valores calculados apartir da linearização e os respectivos erros:

* Para Dist = 4,5mm

```
Medido = 11
Calculado = -7,48
Erro = 18,48
```

* Para Dist = 7,5mm

```
Medido = 92,98
Calculado = 69,94
Erro = 23,04
```

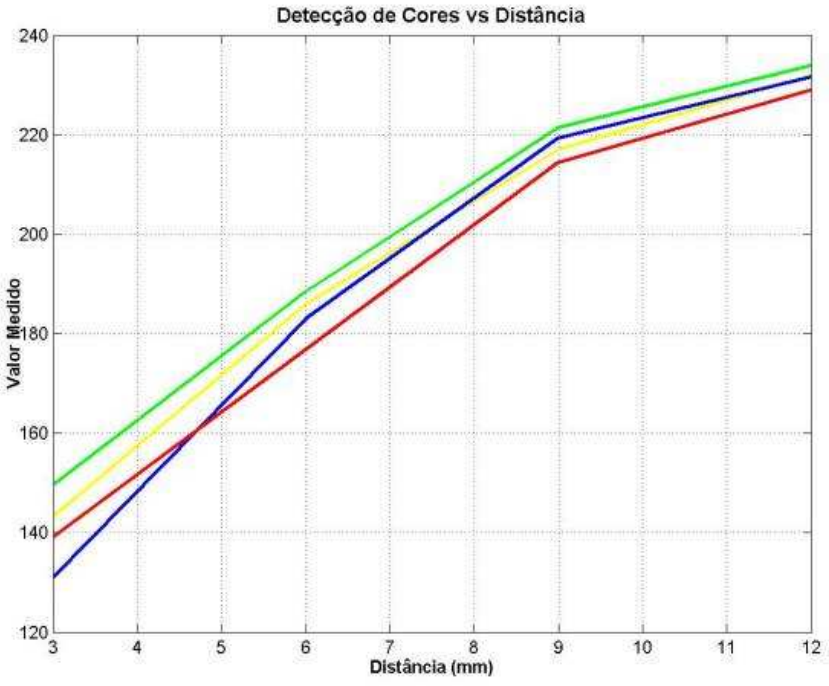
* Para Dist = 10,5mm

```
Medido = 171,84
Calculado = 162,65
Erro = 9,19
```

* Para Dist = 13,5mm

```
Medido = 212,92
Calculado = 206,49
Erro = 6,43
```

Influência da superfície na identificação dos blocos



Vemos pelo gráfico que a melhor distancia é 3mm, pois propicia a maior distancia entre os sinais. Além disso vimos que penas variações na distancia podem provocar troca entre as cores. O algoritimo que utilizamos é baseado na media do sinal obtido atraves do sensor.

Lista de Exercícios – Robótica

- Normalização:

1-
Para os valores medidos do sensor temos:

Máximo de Luz Refletida = 6
Mínimo de Luz Refletida = 252

```
/*normal.c*/
int normalize (int light)
{
int MAX_LIGHT = 6;
int MIN_LIGHT = 252;

int output = 100 - ( (light – MAX_LIGHT)*100 / (MIN_LIGHT – MAX_LIGHT) );

if (output < 0) output = 0;
if (output > 100) output = 100;

return output;
}
```

2 -

Utilizando a função normalize para o sensor fotoelétrico QRB1113 obtivemos os valores esperado, uma vez que foram normalizados. A faixa de valores que a função retorna é de zero a cem.

3 -

Quando o sensor estiver em contato com uma superfície mais refletora do que a utilizada na calibração (superfícies brancas) ou quando a superfície for menos refletora que a utilizada na calibração (superfícies pretas) o valor encontrado estará fora do range; ou seja, MAX_LITHG > valor medido ou MIN_LIGHT < valor medido.

- **Valores Históricos:**

1 -

Para um array de 50 posições e uma frequência de aquisição de dados de 7,8125Hz temos:

Tempo = 50/7,8125 e *Tempo = 6,4 segundos*

3 -

```
/* stuck.c*/

void STUCKROBOT()
{
int valores[50];
int i = 0;
float x;
while(1)
{
  If(i!=0)
  {
    x = analog(2);
    if( x != valores( i – 1 ) ) { i = 0; }
    valores( i ) = x;
    if( i == 49) { printf(“stucked”); }
  }
  else{ valores( i ) = analog(2); }
  i++;
  sleep(0.1);
}
```

O programa compara o valor lido com o último valor do buffer, caso seja igual armazena na posição corrente do buffer, caso seja diferente armazena na primeira posição do buffer. Se o buffer encher indica que todos os valores são iguais e que o robô está parado.