



# PMR 3408 Instrumentação

Experiência sobre

## ***Ball and Beam, Sensor de distância e PixyCam***

### **Objetivo**

O objetivo dessa experiência é implementar um controle de posição de uma esfera sobre uma viga, com a inclinação da viga sendo o atuador e verificando o efeito no sistema de dois possíveis sensores:

- 1) Sensor Infravermelho VL53L0X da ST Microelectronics;
- 2) PixyCam;

Nome _____	n°USP _____

## 1. Bancada experimental *Ball and Beam*

O sistema de equilíbrio de uma esfera em um prato é um excelente experimento didático para que um mecatrônico implemente um sistema de controle. Como o sistema é simples, o foco recai sobre o sensoriamento e a atuação.

A figura 1 exemplifica esse ponto, uma vez que foi desenvolvida como um trabalho semestral de uma matéria do MIT, que utiliza apenas um sensor de imagem para determinar a posição da esfera. Como foi utilizada a PixyCam, foram utilizadas tanto a biblioteca OpenCV (Open Computer Vision) em um Raspberry Pi quanto a saída do processamento integrado na PixyCam, através do protocolo de comunicação SPI.

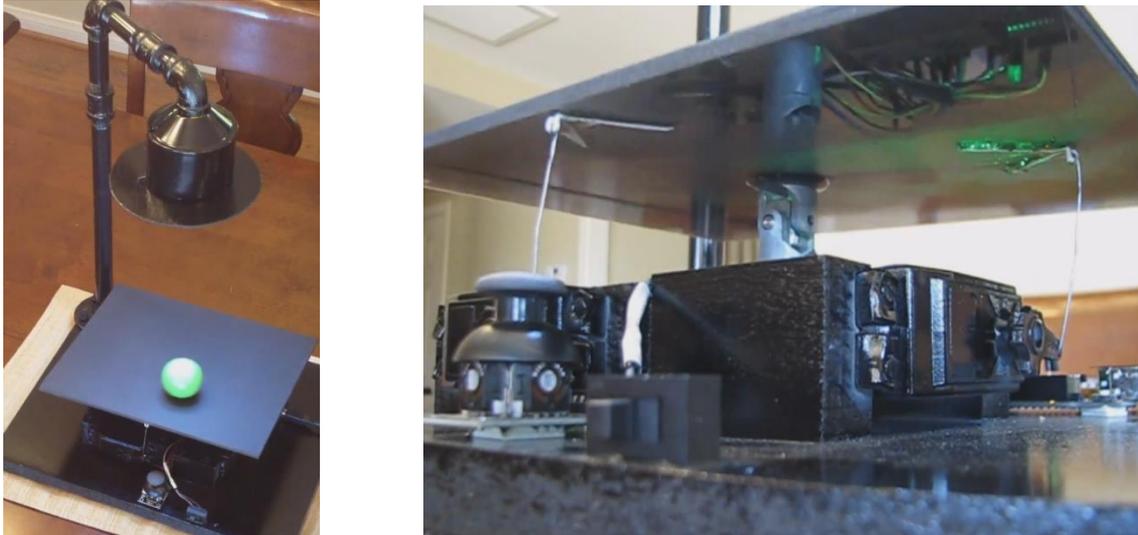


Figura 1 – Sistema de equilíbrio bola e prato, detalhe da atuação com dois servo motores.

Entretanto, como temos pouco tempo em nosso laboratório, iremos implementar o controle em um sistema *Ball and Beam*, ou seja, esfera e viga.

O aparato experimental encontra-se disponível em sua bancada com a viga, duas bolas oficiais de tênis de mesa de diferentes cores, um sensor de distância infravermelho e uma PixyCam. O foco da experiência será o sensoriamento.

Para facilitar o controle, foi escolhido um servomotor de alto torque, modelo PODI-6221MG, como atuador. Este tem como características:

- Engrenagens de alumínio;
- Torque de saída de aproximadamente 18kgf.cm;
- Velocidade de operação de 0,18 seg/60º.

Entretanto, para que seja possível atuar com este torque, o servomotor necessita de uma quantidade considerável de corrente. Sendo assim, usaremos uma fonte regulável. **Lembre-se de verificar se o terra da fonte está ligado tanto ao servo motor quanto ao Arduino.**

## 2. Sensor de distância

O sensor VL53L0X da ST Microelectronics é um para pequenas distâncias com protocolo de comunicação I<sup>2</sup>C. O sensor está representado na figura 2.

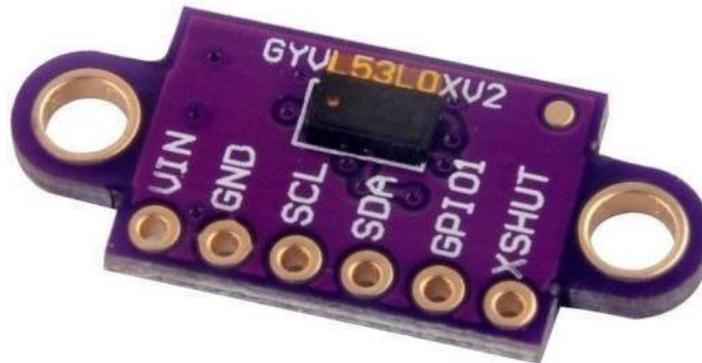


Figura 2 – Sensor VL53L0X.

O VL53L0 usa a tecnologia FlightSense da ST para medir com precisão quanto tempo leva para que os pulsos emitidos da luz laser infravermelha alcancem o objeto mais próximo e sejam refletidos de volta para um detector. Essa medida de tempo de voo (ToF) permite determinar com precisão a distância absoluta para um alvo sem que a refletância do objeto influencie muito a medição. O sensor pode reportar distâncias de até 2m com resolução de 1 mm, mas seu alcance efetivo e precisão (ruído) dependem muito das condições do ambiente e das características do alvo, como refletância e tamanho, bem como da configuração do sensor. A precisão do sensor é especificada para variar de  $\pm 3\%$  na melhor das hipóteses a mais de  $\pm 10\%$  em condições menos ideais. O sensor é capaz de operar em frequências bastante altas, sendo limitada, normalmente pela frequência da comunicação I<sup>2</sup>C.

Foi instalada na IDE do Arduino uma biblioteca para o sensor, fornecida pelo fabricante.

Sua primeira tarefa vale 2 pontos e consiste em gerar pares de distância real e da distância lida pelo sensor, com no mínimo 20 pontos, e fazer gráficos com distância real no eixo y e distância fornecida pelo sensor no eixo x.

Valores estimados pela curva do sensor fornecida pelo fabricante, disponível no código.

Por fim, **determine se há necessidade de correção dos dados fornecidos pelo sensor e, em caso positivo, qual seria essa equação. Os dados, gráficos e equações podem ser entregues no moodle em arquivo anexo.**

### 3. PixyCam

A PixyCam, desenvolvida pelo *CharmedLabs*, é uma câmera e CI com processamento de imagem integrado e saídas em formatos I<sup>2</sup>C, SPI, UART e USB. Com um sistema simples de calibração, ela é capaz adquirir imagens e fazer o rastreamento a 50Hz. A PixyCam está ilustrada na figura 3.

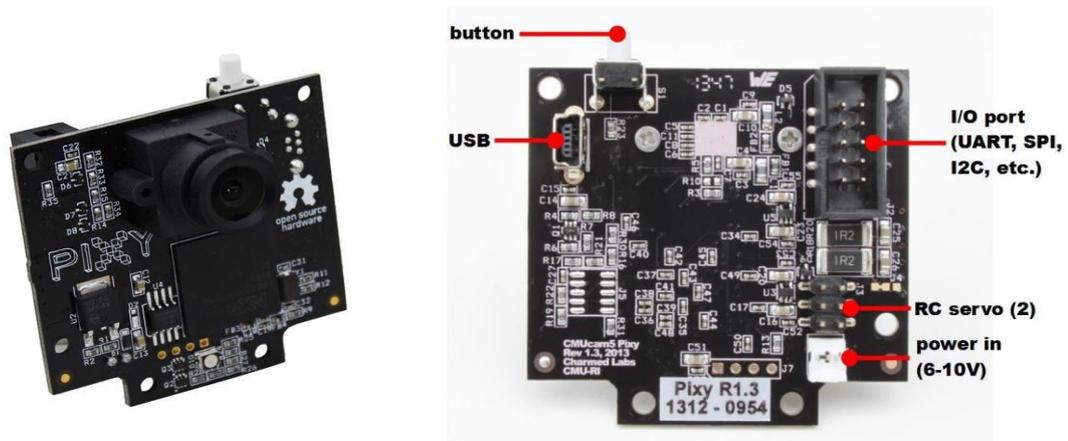


Figura 3 – PixyCam e parte traseira com conectores.

Para utilizar a câmera, a empresa fornece o software PixyMon.

Para calibrar a câmera, inicialmente deve-se regular o foco diretamente na câmera, girando o suporte de sua lente. Em seguida, deve-se limpar todas as assinaturas (actions -> clear all signatures) já existentes e escolher uma nova assinatura. As assinaturas servem para escolher uma das cores detectadas pela câmera, vários tons da mesma cor são detectados. Recomenda-se que se use apenas cores que se destaquem das demais. Coloque a bola laranja de tênis de mesa na canaleta do dispositivo e escolha uma região da imagem que se deseja detectar. A figura 4 demonstra a escolha de assinatura no PixyMon.

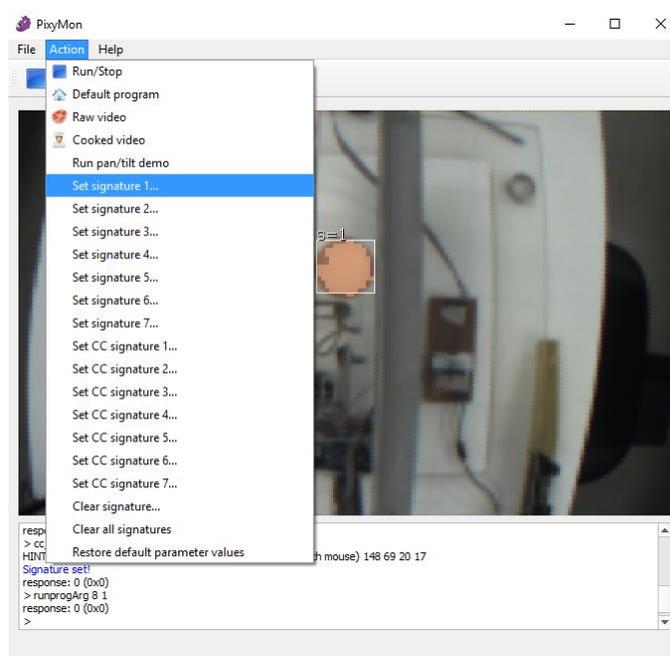


Figura 4 – PixyMon em uso.

Na tela de configurações, é possível escolher alguns parâmetros da câmera. A primeira é a faixa de tons de cores detectadas pela assinatura. Coloque outros objetos com cores semelhantes no alcance da

câmera e varie o parâmetro para notar a diferença. Para valores mais altos a iluminação tem menor interferência na detecção do objeto, mas tome cuidado para não detectar outros objetos nas proximidades. Teste também a variação de brilho da câmera.

A PixyCam é capaz de detectar até 1000 objetos simultaneamente. Em “Blocks”, certifique-se de que somente um objeto pode ser detectado por vez, apenas o que tiver maiores dimensões será detectado. A figura 5 demonstra uma falha de detecção.

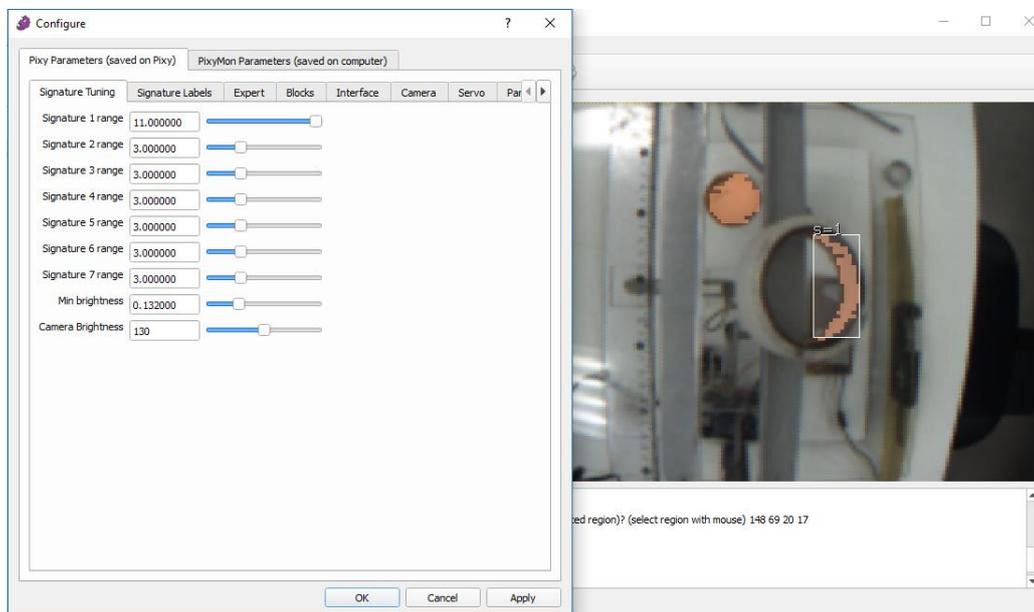


Figura 5 – PixyCam detectando a fita crepe ao invés da bola de tênis de mesa.

Nas configurações de interface, verifique se a opção Arduino ICSP SPI está selecionada. A PixyCam gera uma imagem completa e tem a capacidade de transferir ela para outros dispositivos como o computador. Entretanto, o arduino não suporta tais níveis de processamento. Escolha a opção “default program” (figura 6), essa opção transforma a imagem em informações de coordenadas e dimensões (em números de pixels) que são facilmente obtidas e interpretadas pelo o Arduino. Caso deseje voltar à câmera clique em “display cooked”.

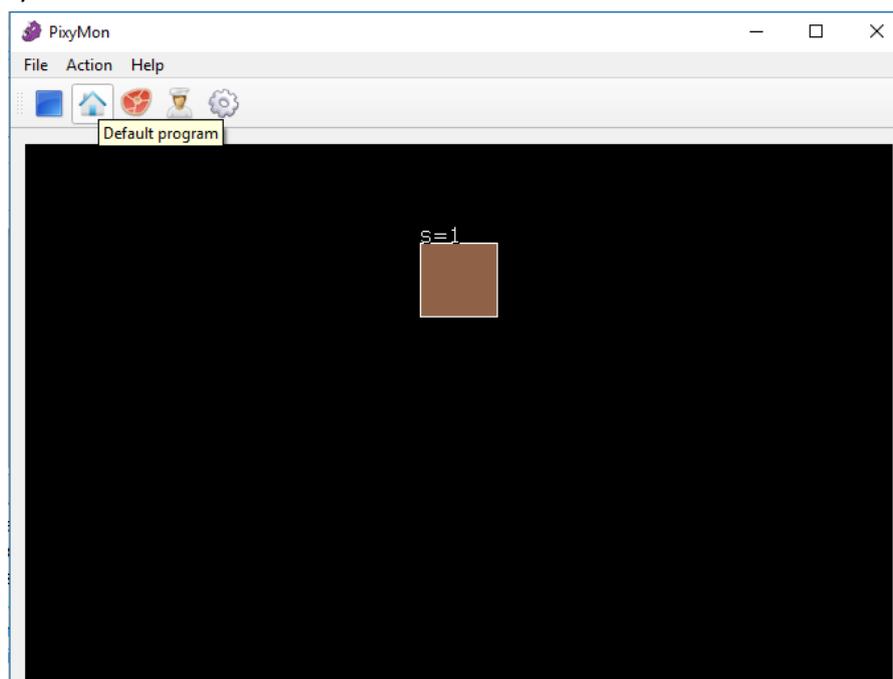


Figura 6 – PixyMon interpretando dados enviados para Arduino: posição e tamanho.

Na IDE do arduino, abra o exemplo de “hello world” da pixy e faça o upload para a placa. Verifique os dados através do Serial Monitor, e leia o programa para entender como os dados são recebidos.

**Exercício:** Calibre a câmera para detectar a bola colorida. Valendo 3 pontos, faça uma calibração entre a posição real e a posição em pixels na saída SPI da PixyCam, para pelo menos **duas inclinações diferentes**. Novamente, **escreva as duas tabelas e faça um gráfico (distância real no eixo y e posição em pixels no eixo x)**. Os dados, gráficos e equações podem ser entregues no moodle em arquivo anexo.

#### 4. Controle

Faça uma cópia do esqueleto.ino que está na área de trabalho para a sua pasta. Abra e leia o programa inteiro.

Com base nas calibrações dos sensores anteriores e na introdução do pré-relatório, implemente e calibre os seguintes controles:

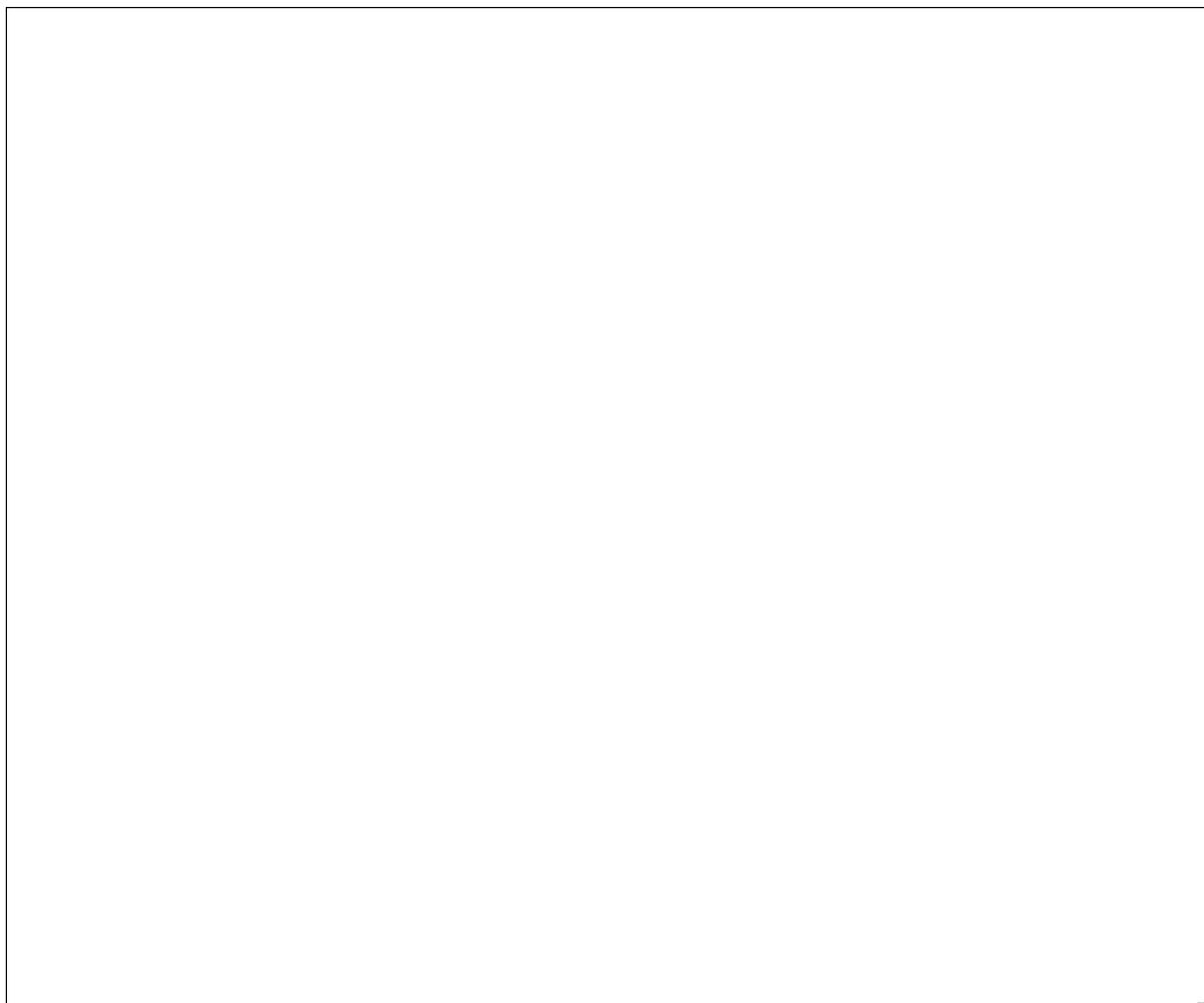
- Usando como sensor de posição o ToF, implemente um controle proporcional, um proporcional integral e um proporcional derivativo integral.
- Usando como sensor de posição a PixyCam, implemente um controle proporcional, um proporcional integral e um proporcional derivativo integral.
- Usando como sensor de posição (50% ToF + 50% PixyCam), implemente um controle proporcional, um proporcional integral e um proporcional derivativo integral.

Obs1: não se esqueça de saturar a saída. O motor tem um tempo de resposta baixo, portanto mover ângulos muito grandes pode causar instabilidades no sistema real (esse efeito não seria percebido na simulação, é obrigação do projetista prever a saturação). Utilize valores entre  $10^\circ$  e  $15^\circ$

Obs2: assim como deve ter ocorrido no pré-relatório, é muito difícil calibrar o integrador, o que pode causar instabilidades no sistema. Isso ocorre devido ao efeito de windup. O integrador soma indefinidamente e faz com que a soma das três parcelas sejam sempre o valor da saturação.

Para fazer o anti-windup basta manter o último valor do integrador imediatamente antes da saturação da saída.

**Cada implementação vale 0,5 ponto por sensor, sendo que vocês devem salvar na área de trabalho cada tipo de controle/sensor em diferentes pastas. Também devem ser entregues gráficos da posição real e da posição desejada pelo tempo para cada tipo de controle.**



**5. Controle por realimentação de estados**

Valendo 1 ponto, no script do MATLAB, implemente o anti-windup e simule novamente os 3 itens do exercício 5 e implemente o controlador por realimentação de estados no arduino. Em seguida, valendo mais 1 ponto, simule e implemente o controlador com amortecimento crítico.

**6. Finalização**

Por fim, valendo o último meio ponto, explique qual tipo de controle apresentou melhor comportamento e enumere os motivos.

Exemplifique os problemas nessa versão da construção (folgas e atritos) e deixe opiniões, comentários e sugestões para que o professor possa melhorar esta experiência para os próximos anos.