



Universidade de Brasília
Projeto de Extensão

Relatório 01:
Relatório do Sensor MPU6050

	Participantes:
Arthur Sena	Matricula: 22/1038767
Eduardo Rossi	Matricula: 22/1007125

Professor:
Walter Vidal Filho

14 de junho de 2024

Sumário

1	Introdução MPU6050	2
2	Princípio de funcionamento	2
2.1	Interface I2C	2
2.2	Giroscópio	3
2.2.1	Giroscópio vibratório de Coriolis com tecnologia capacitiva . . .	3
2.2.2	Efeito Força Coriolis	4
2.2.3	Interferência da aceleração linear	4
2.3	Acelerômetro	5
2.4	MPU6050 + Software	5
2.4.1	Configurações	5
3	Experimento realizado	9
3.1	Metodo	9
3.2	Problemas nas medidas	9
3.3	Dados	10
3.4	Interpretacao dos dados	10
4	Código	10

Abstract. Report covering the operating principles of the MPU-6050 board regarding obtaining angulation in real time.

Resumo. Relatório abordando os princípios de funcionamento da placa MPU-6050 referentes a obtenção da angulação em tempo real.

1. Introdução MPU6050

O sensor MPU6050, muito utilizado em diversos projetos mecatrônicos, conta com estratégias das mais variadas para a obtenção de dados referentes a velocidade de rotação e aceleração nos três eixos (6 DOF), além de temperatura interna em graus célcus. Nesse contexto, com tantos dados a serem passados ao microcontrolador, faz-se necessária utilização da interface de comunicação Inter-Integrate Circuit (I2C) ou Circuito Interintegrado, metodologia essa que possibilita o processamento dos dados citados em troca de uma latência mínima.

Também presente na placa, das estratégias minuciosamente pensadas para a coleta precisa de dados, os sensores Giroscópio MEMS e Acelerômetro MEMS se destacam nas engenhosidades de suas estruturas.

2. Princípio de funcionamento

A explicação do funcionamento do Sensor está dividido em três etapas principais: comunicação com o microcontrolador que se dá através da interface I2C, giroscópio e acelerômetro. Levando em consideração suas devidas importâncias e requisitos para o funcionamento ótimo do MPU6050.

2.1. Interface I2C

A comunicação entre o sensor em questão e o microcontrolador se dá por meio da interface I2C que nada mais é do que um protocolo de comunicação serial síncrona que permite a interconexão de dispositivos em um barramento comum. A comunicação com o sensor utiliza as linhas SDA (Serial Data Line) e SCL (Serial Clock Line), portas (A4) e (A5) do Arduino Uno, respectivamente.

Pense no I2C como uma forma de se comunicar e, o sensor MPU, como uma sala de aula com sete alunos agrupados em dois grupos de três e um grupo de um arranjados em 3,1,3; giroscópio, temperatura e aceleração respectivamente. Além disso, na sala também tem a presença de um monitor, um professor e assistentes. Os alunos estão fazendo uma leitura de texto, e o professor deseja questionar qual leitura um aluno está a ler. Para isso, ele utiliza a "metodologia I2C", onde fala a posição do grupo de alunos do aluno desejado ao monitor, bit a bit, e solicita o trecho lido pelo aluno desejado, daquele endereço, naquele momento. Nesse cenário, o monitor encosta no aluno, e este já sabe o que fazer. Com os dois de cabeça levantada, o aluno abaixa a cabeça para ler; em seguida, o monitor abaixa a cabeça para escutar. O aluno, levantando a cabeça, fala o bit e abaixa a cabeça em seguida; o monitor levanta a cabeça e fala ao professor o valor do bit e, logo após, abaixa a cabeça para escutar o aluno falando o próximo bit ao levantar a cabeça

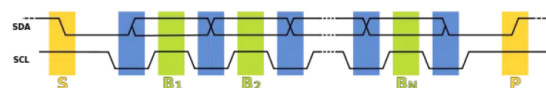


Figura 1. Comunicação entre MPU6050 e Arduino (via dupla).

novamente. Assim, após abaixá-la e o monitor levantá-la, respectivamente, este fala o bit recebido ao professor.

O processo se repete até que o monitor e o aluno estão os dois de cabeça levantada ao mesmo tempo, nesse momento o professor pode solicitar a leitura para outro aluno desde que o novo aluno requisitado esteja logo ao lado do aluno, se não, o monitor será chamado e a comunicação com o grupo será encerrada até que seja solicitada novamente.

Essa é a explicação aproximada de como é feita a comunicação entre a leitura do sensor e o microcontrolador. O professor representa o microcontrolador, o monitor representa a linha SCL e as vezes SDA, e o aluno representa a linha SDA (ver: Figura 1), além de o aluno também ser o leitor do valor captado pelo sensor, seja essa leitura a velocidade de rotação, a aceleração ou a temperatura interna. Além disso, a representação de apenas um monitor na sala, aliado ao levantar e abaixar de cabeças, é uma analogia ao fato de a comunicação via interface I2C ser serial e síncrona.

Finalmente, junto a todo mundo na sala, os assistentes desempenham papéis fundamentais na sofisticação e dinamismo do sensor. Para tanto, adjunto ao monitor para se comunicar, este pode solicitar ao assistente, prontamente comando do professor, diferentes funções como: mudança dos textos de leitura através da definição da configuração do giroscópio e do acelerômetro, sair/entrar no modo sono, definir taxa de amostragem, etc. Estas e outras funções serão abordadas mais adiante neste documento.

Portanto, a comunicação entre o mestre Arduino (professor) e o escravo placa MPU6050 (sala de aula) é efetuada mediante interface I2C. Ou seja, há o levantamento e abaixamento de "cabeças" ordenadamente, um único monitor na sala, endereçamento da placa e dos registradores, além de confirmação ACK após cada byte transmitido.

2.2. Giroscópio

Tendo entendido a comunicação entre dispositivos, da-se início agora aos princípios de funcionamento dos sensores, mais especificamente o sensor giroscópio MEMS (Microelectromechanical Systems) presente na placa em estudo.

Giroscópio MEMS funciona a partir da junção entre o efeito de Coriolis e a tecnologia capacitiva, junção essa que possibilita, após simplificação, a obtenção da velocidade angular da placa. Muitos de nós estamos familiarizados com o giroscópio tradicional – uma roda que gira rapidamente montada em um suporte que permite a rotação livre em qualquer direção. No entanto, o giroscópio vibratório de Coriolis, embora opere sob princípios similares, tem um design e aplicação distintamente diferentes.

2.2.1. Giroscópio vibratório de Coriolis com tecnologia capacitiva

No centro do Giroscópio MEMS, há uma pequena massa vibrante chamada de proof mass (massa de prova). Essa massa é forçada a vibrar em uma frequência específica

e em uma direção determinada. Esse processo de forçamento é realizado por atuadores que são feitos de materiais piezoelétricos. Esses atuadores têm a função de iniciar e manter a vibração da massa, aplicando uma tensão que faz com que ela se mova de forma sinusoidal, constantemente.

Além dos atuadores, para que todo o sistema funcione, são necessários os elementos de suspensão. Esses elementos são responsáveis por manter a massa vibrante alinhada além de permitir seu movimento estimulado pela força Coriolis em resposta à rotação. Eles são basicamente estruturas de silício finas que funcionam como molas, fornecendo a flexibilidade necessária.

Por último, inserida em toda essa estrutura, temos os sensores capacitivos responsáveis por medir a deflexão da massa vibrante estimulada pela força de Coriolis. Esses sensores consistem em placas de capacitor que, quando se movem, alteram a capacitância proporcionalmente à força sentida pela massa de prova. Dessa forma, a deflexão da massa afeta a distância entre essas placas, resultando em uma alteração na capacitância que é medida eletronicamente.

Esses componentes abordados formam o núcleo funcional do giroscópio presente na placa, permitindo a detecção precisa de rotação e movimento em diferentes contextos tecnológicos.

2.2.2. Efeito Força Coriolis

Na física por de trás da obtenção do ângulo pelo giroscópio citado anteriormente, há o efeito Coriolis. O efeito Coriolis se caracteriza por ser a principal componente na captação dos dados conferido pelo sensor em questão. Este efeito é causado quando há a presença de um referencial não-inercial vibracional que vibra de maneira periódica forçada pelos atuadores (eixo X na Figura 2) e, além disso, há presença de rotação no eixo perpendicular à vibração forçada (eixo Z na Figura 2). Assim, a partir da fórmula da força de Coriolis:

$$F_c = -2m(w * v). \quad (1)$$

Tal que m é a massa de prova, w a velocidade angular e v a velocidade linear da vibração forçada, tem-se a Força Coriolis perpendicular e proporcional a w e v . Como F_c é quantizado pela variação de capacitância, pode-se isolar w e obter:

$$w = F_c / (-2mv). \quad (2)$$

Dessa forma, obtém-se a velocidade angular w da placa.

2.2.3. Interferência da aceleração linear

Por outro lado, a aceleração linear pode interferir na coleta de dados, causando o que é chamado de "cross-axis sensitivity" ou sensibilidade cruzada. Isso acontece quando uma aceleração linear é interpretada pelo giroscópio como uma rotação, levando a leituras incorretas. Essas correções são imprescindíveis em situações com ambas forças atuantes

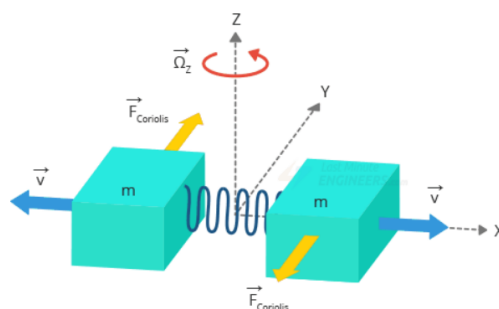


Figura 2. Força de Coriolis.

no sistema, e exige filtro(s) capaz(es) de compensar a interferência causada pela sensibilidade cruzada. Assim, para melhor compreensão desse fenômeno, aproveita-se o gancho e inicia-se a explicação do sensor acelerômetro MEMS de 3 eixos presentes na placa

2.3. Acelerômetro

O chip acelerômetro é um componente importantíssimo na robótica móvel, podendo ser utilizado para determinar a aceleração instantânea de um corpo nas componentes x, y e z e, por consequência, a sua velocidade e posição por meio da integração. O seu funcionamento utiliza como base sistemas massa mola, que são comprimidos ou esticados pela ação da força aplicada no chip, essas forças são medidas por meio de um cristal piezoelétrico e usadas para calcular a aceleração usando a Segunda Lei de Newton, que define a aceleração de um corpo sendo proporcional a força aplicada sobre ele. Contudo, no caso específico deste estudo, em que queremos obter o ângulo de inclinação da placa, a aceleração linear não foi utilizada diretamente, mas sim para obter o ângulo de inclinação usando o arcotangente entre as componentes da aceleração gravitacional. Essa forma é bastante confiável e contém poucos erros por conta da aceleração gravitacional local ser constante, porém não pode ser usada para calcular o ângulo no plano xy, uma vez que a aceleração estará sempre alinhada com o eixo z.

2.4. MPU6050 + Software

Apesar de uma boa estrutura, parece que quanto melhor é a metodologia de sensoriamento, mais baixo nível de linguagem o software exige.

2.4.1. Configurações

Com todas essas características e dinamismos, o sensor MPU6050 conta com configurações modificáveis para diferentes funcionalidades, essas configurações serão explicadas levando em conta como elas modificam a leitura da placa em dois contextos cada, assim como se segue:

1. Acordar/Dormir o MPU6050:

Essa funcionalidade permite colocar os sensores em modo soneca ou em modo coleta de dados. Com isso em mãos, é possibilitado a economia de energia em momentos em que sua funcionalidade de sensoriamento não está sendo requisitada.

- Contexto 1: Dispositivos Móveis
Configuração: 0x00 (Acordar o dispositivo).
Por que? Em smartphones e tablets, é crucial que o MPU6050 esteja sempre pronto para fornecer dados de movimento e orientação para funcionalidades como rotação automática da tela e estabilização de imagem.
- Contexto 2: Sensores de Segurança
Configuração: 0x01 (Mantém o dispositivo em modo de economia de energia até ser requisitado).
Por que? Em sistemas de segurança que só precisam do sensor em determinados momentos (como durante uma intrusão), manter o MPU6050 em modo de economia de energia ajuda a prolongar a vida útil da bateria.

2. Aumentar/diminuir a Taxa de Amostragem

O registrador SMPLRT_DIV ajusta a taxa de amostragem do MPU6050. A taxa de amostragem é a frequência com que o sensor coleta e atualiza os dados.

- Contexto 1: Realidade Virtual e Aumentada
Configuração: SMPLRT_DIV = 0x01 (Taxa de amostragem alta, 500 Hz).
Por que? Em aplicações de VR e AR, uma alta taxa de amostragem garante uma resposta rápida e precisa aos movimentos do usuário, proporcionando uma experiência imersiva e sem latência perceptível. Movimentos são capturados com maior frequência, o que é crucial para manter a sensação de presença e reduzir a vertigem.
- Contexto 2: Monitoramento de Vibrações em Máquinas
Configuração: SMPLRT_DIV = 0x09 (Taxa de amostragem baixa, 100 Hz).
Por que? Para monitorar vibrações em máquinas industriais, uma taxa de amostragem de 100 Hz é suficiente para detectar falhas e evitar desgaste prematuro. Uma taxa de amostragem mais baixa economiza energia e reduz a quantidade de dados processados, o que é importante para sistemas que operam continuamente.

3. Alarga/estreita a banda do Filtro de Passagem Baixa

O registrador CONFIG define a configuração do filtro de passagem baixa (DLPF). O DLPF filtra ruídos de alta frequência, permitindo apenas que frequências mais baixas (e menos ruidosas) passem, melhorando a qualidade dos dados sensoriais.

- Contexto 1: Estabilização de Imagem
Configuração: CONFIG = 0x01 (Filtro de passagem baixa com banda larga).
Por que? Em câmeras de smartphones, um filtro de passagem baixa menos agressivo (banda larga) permite que as leituras do giroscópio respondam

rapidamente aos movimentos. Isso é essencial para a estabilização de imagem e gravação de vídeo, pois minimiza a latência e mantém a suavidade dos vídeos mesmo em movimento.

- Contexto 2: Monitoramento de Saúde

Configuração: CONFIG = 0x06 (Filtro de passagem baixa com banda estreita).

Por que? Em dispositivos de monitoramento de saúde, como sensores de movimento usados em fisioterapia, um filtro de passagem baixa mais agressivo (banda estreita) elimina o ruído e suaviza os dados, garantindo medições precisas dos movimentos do paciente. Isso é crucial para a análise correta dos padrões de movimento e para monitorar a recuperação de pacientes.

4. Aumentar/diminuir o Intervalo do Giroscópio e sua sensibilidade

O registrador GYRO_CONFIG define a faixa de sensibilidade do giroscópio. Dependendo da configuração, o giroscópio pode medir diferentes intervalos de velocidade angular, de $\pm 250^\circ/\text{s}$ a $\pm 2000^\circ/\text{s}$

- Contexto 1: Jogos e Simulações

Configuração: GYRO_CONFIG = 0x18 ($\pm 2000^\circ/\text{s}$).

Por que? Em controladores de jogos e simuladores de voo, uma faixa de $\pm 2000^\circ/\text{s}$ permite detectar movimentos rápidos e abruptos, proporcionando uma experiência mais realista e responsiva. Jogadores e pilotos virtuais podem realizar manobras rápidas e precisas, aumentando a imersão e o realismo.

- Contexto 2: Dispositivos Médicos

Configuração: GYRO_CONFIG = 0x00 ($\pm 250^\circ/\text{s}$).

Por que? Em dispositivos médicos que monitoram movimentos sutis, como tremores em pacientes com Parkinson, uma faixa de $\pm 250^\circ/\text{s}$ é preferível para obter leituras mais precisas e detalhadas de pequenos movimentos. A sensibilidade aumentada permite detectar e quantificar tremores com maior precisão, ajudando no diagnóstico e tratamento.

5. Aumentar/diminuir o Intervalo do Acelerômetro e sua sensibilidade

O registrador ACCEL_CONFIG define a faixa de sensibilidade do acelerômetro. Dependendo da configuração, o acelerômetro pode medir diferentes intervalos de aceleração:

- Contexto 1: Wearables e Dispositivos de Fitness

Configuração: ACCEL_CONFIG = 0x00 ($\pm 2g$).

Por que? Em wearables, como smartwatches e rastreadores de fitness, uma faixa de $\pm 2g$ é ideal para detectar atividades diárias e movimentos suaves com alta precisão. Isso é essencial para monitorar passos, calorias

queimadas e padrões de sono. A alta sensibilidade permite capturar detalhes finos dos movimentos do usuário, melhorando a precisão das métricas de fitness.

- Contexto 2: Dispositivos de Segurança Automotiva
Configuração: ACCEL_CONFIG = 0x18 ($\pm 16g$).

Por que Em sistemas de segurança automotiva, como sensores de impacto em carros, uma faixa de $\pm 16g$ é necessária para detectar eventos de alta intensidade, como colisões. As forças g durante um acidente podem ser muito altas, e uma faixa de $\pm 16g$ garante que esses eventos sejam registrados com precisão, ativando sistemas de segurança passiva (como airbags) de forma oportuna.

6. Habilita/desabilita o FIFO e o DMP

O registrador FIFO_EN controla quais dados são armazenados no buffer FIFO. O FIFO permite o armazenamento temporário de dados, enquanto o DMP (Digital Motion Processor) processa os dados de movimento internamente.

- Contexto 1: Dispositivos de Medição de Desempenho Esportivo
Configuração: FIFO_EN = 0x78 (Acelerômetro, Giroscópio e DMP no FIFO).

Por que Em dispositivos de medição de desempenho esportivo, habilitar o FIFO permite a coleta contínua de dados de movimento, que podem ser analisados posteriormente para avaliar o desempenho do atleta. Isso é útil para detectar padrões de movimento, identificar áreas de melhoria e fornecer feedback detalhado aos atletas.

- Contexto 2: Sistemas de Navegação Inercial
Configuração: FIFO_EN = 0x00 (Desabilitar FIFO).

Por que? Em sistemas de navegação inercial, como aqueles usados em submarinos ou aeronaves, que requerem processamento de dados em tempo real, desabilitar o FIFO pode ser preferível para minimizar a latência. Isso garante que os dados de movimento sejam processados imediatamente, permitindo cálculos de posição e orientação precisos e atualizados.

7. Habilita/desabilita Interrupções

O registrador INT_ENABLE controla as interrupções geradas pelo MPU6050. Quando uma interrupção é habilitada, o sensor notifica o microcontrolador assim que novos dados estão prontos, facilitando a leitura e processamento eficiente dos dados.

- Contexto 1: Controle de Movimento em Robôs
Configuração: INT_ENABLE = 0x01 (Interrupção de dados prontos).

Por que? Permite ao microcontrolador responder rapidamente a novos dados de movimento, ajustando o controle do robô em tempo real, essencial

para navegação precisa e resposta rápida a mudanças no ambiente.

- Contexto 2: Assistentes Pessoais Inteligentes
Configuração: `INT_ENABLE = 0x00` (Desabilitar interrupções).
Por que Em dispositivos onde o processamento de dados pode ser feito periodicamente, desabilitar interrupções simplifica o design do software e reduz a complexidade do gerenciamento de interrupções, adequado para tarefas não críticas de tempo.

3. Experimento realizado

3.1. Metodo

Para se analisar os erros de cada eixo de rotação, foram feitos 3 experimentos utilizando uma base angulada.

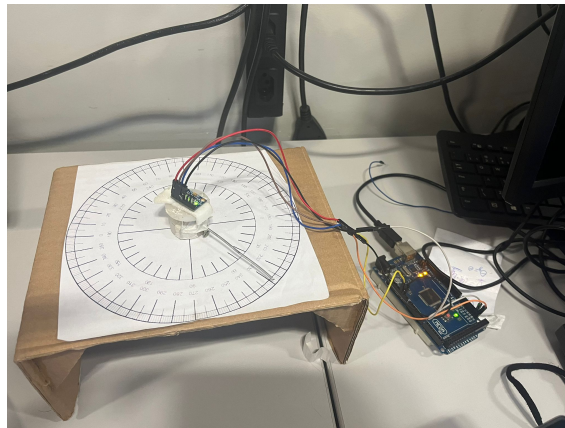


Figura 3. Base angulada utilizada para os testes

Nesse sentido, posicionou-se o sensor de forma que o eixo de rotação da base estivesse colinear ao eixo estudado e mediu-se os ângulos de 15 em 15 graus usando os dados do sensor e comparou-os com os ângulos marcados, fazendo quatro medidas afim de estipular um erro experimental.

3.2. Problemas nas medidas

O primeiro problema enfrentado foi a aceleração feita ao girar o medidor, uma vez que o código integra as velocidades, gerando erros caso a variação fosse muito brusca. Para esse problema não pode se obter uma solução muito eficiente, por conta de como os testes foram feitos, somente garantindo que o sensor fosse girado na velocidade mais constante possível.

Houve também um problema no uso do acelerômetro ao medir a angulação, pois uma vez que o sensor foi posicionado alinhado ao eixo da gravidade para fazer os testes, o vetor da aceleração gravitacional não formava ângulos com nenhuma das componentes do sensor. Esse defeito afetava os resultados, uma vez que o peso utilizado para calcular os ângulos eram 98 por cento do valor dado pelo giroscópio e 2 por cento do valor do acelerômetro, logo, os ângulos tendiam com o tempo a retornar o valor zero pois não havia a contribuição do acelerômetro. A solução encontrada foi retirar a medida da aceleração para o cálculo, o que não afetaria os casos em que a inclinação ocorresse em apenas um eixo.

3.3. Dados

Z(Real)	Z1	Z2	Z3	Z4	media	desvio padrao(Z)
15	15.09	15.08	15.03	14.63	14.96	0.2
30	30.05	30.65	29.98	29.09	29.94	0.56
45	44.86	45.68	44.28	43.83	44.66	0.7
60	60.35	60.3	59.06	58.42	59.53	0.83
75	75.92	75.62	74.30	74.00	74.96	0.83
90	91.20	90.60	89.70	88.76	90.06	0.93

Y(Real)	Y1	Y2	Y3	Y4	media	desvio padrao(Y)
15	12.9	13.1	15.3	15.1	14.1	1.11
30	26.3	26.8	27.3	29.6	27.5	1.27
45	40.4	41.1	42.3	44.0	41.95	1.37
60	55.8	55.2	56.9	58.1	56.5	1.12
75	69.9	70.6	71.8	72.6	71.22	1.05
90	84.7	82.8	85.3	87.1	84.97	1.55

X(Real)	X1	X2	X3	X4	media	desvio padrao(X)
15	14.5	14.6	14.3	15.2	14.65	0.34
30	28.2	29.1	29.7	30.3	29.32	0.78
45	43.3	43.3	43.6	44.7	43.72	0.58
60	59.5	58.5	59.0	60.8	59.45	0.86
75	73.1	72.1	74.1	75.8	73.77	1.37
90	90.6	85.8	90.0	89.8	89.05	1.9

3.4. Interpretacao dos dados

Os dados obtidos permitem entender que o sensor possui diferentes erros para cada eixo medido. Essa relação se dá pela própria modelagem do MPU que pode possuir defeitos nos eixos de cada direção.

Percebe-se também que em todos os eixos os erros tendem a aumentar com o tempo, isso é esperado, uma vez que os erros advindos do processo de integração tendem a se acumular.

4. Código

```
/* Get tilt angles on X and Y, and rotation angle on Z
 * Angles are given in degrees
 *
 * License: MIT
 */
```

```
#include "Wire.h"
#include <MPU6050_light.h>
```

```
MPU6050 mpu( Wire );
```

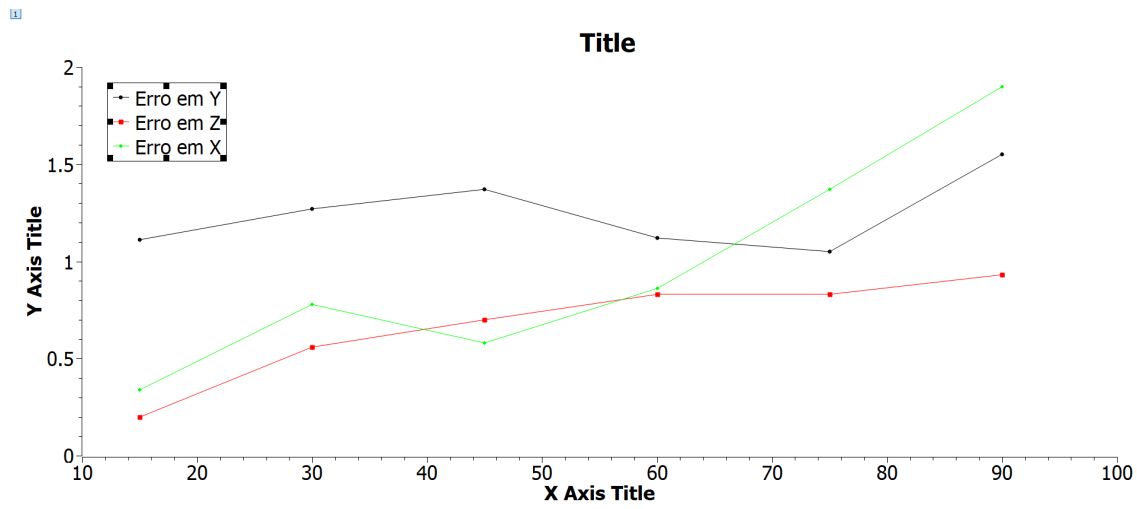


Figura 4. Erros em X, Y e Z por angulos medidos

```

unsigned long timer = 0;

void setup() {
  Serial.begin(9600);
  Wire.begin();

  byte status = mpu.begin();
  Serial.print(F("MPU6050 status: "));
  Serial.println(status);
  while(status!=0){ } // stop everything if could not connect to MPU6050

  Serial.println(F("Calculating offsets , do not move MPU6050"));
  delay(1000);
  // mpu.upsideDownMounting = true; // uncomment this line if the MPU6050
  mpu.calcOffsets(); // gyro and accelero
  Serial.println("Done!\n");
  mpu.setFilterGyroCoef(1);
}

void loop() {
  mpu.update();

  if((millis()-timer)>10){ // print data every 10ms
    Serial.print("X : ");
    Serial.print(mpu.getAngleX());
    Serial.print("\tY : ");
    Serial.print(mpu.getAngleY());
    Serial.print("\tZ : ");
    Serial.println(mpu.getAngleZ());
    timer = millis();
  }
}

```

```
}  
}
```

Referências

[1] Interface MPU6050 Accelerometer and Gyroscope Sensor with Arduino. Disponível em: <https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/>. Acesso em: 06 jun. 2024.

[2] Documentation MPU6050light.h library. Disponível em: https://github.com/rfetick/MPU6050_light/blob/master/documentation_MPU6050_light.pdf. Acesso em: 02 jun. 2024.

<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>