

Calibração Estática do Sensor

Instrumentação Eletrônica para Engenharia

Felipe Costa Gomes, Davi Dantas Lopes

Engenharia Eletrônica

Universidade de Brasília - Faculdade do Gama

Gama, Distrito Federal

E-mails: 190012757@aluno.unb.br , 170031934@aluno.unb.br

Resumo—O propósito deste trabalho consiste em descrever o processo de calibração estática do sensor acelerômetro. Utilizando um mecanismo disponibilizado, o sensor foi posicionado em seis posições diferentes, permitindo a coleta de um mínimo de 100 leituras em cada eixo. Com base nas leituras coletadas, foram calculados os valores médios para cada eixo, representados por $\bar{A}_{+1g,i}$ e $\bar{A}_{-1g,i}$. A partir desses valores, um sistema de equações foi construído para estimar os parâmetros *Ganho* e A_{offset} em cada eixo. Esses parâmetros foram utilizados posteriormente para compensar as medições realizadas, garantindo maior precisão e confiabilidade. A calibração estática do sensor acelerômetro é essencial para minimizar erros sistemáticos e obter medições mais precisas em projetos que fazem uso desse sensor.

palavras-chave: Calibração estática. Acelerômetro. Ganho e Aoffset. Erros sistemáticos.

I. INTRODUÇÃO

O MPU-6050 [1] é um sensor amplamente utilizado que combina um giroscópio de três eixos, um acelerômetro de três eixos e um processador de movimento digital em um único pacote compacto. Com suas capacidades de rastreamento de seis eixos, proporciona seis graus de liberdade para medições de movimento. Com a possibilidade de conexão de uma bússola externa de três eixos, é possível obter uma saída de fusão de movimento completa com nove eixos. Além disso, o MPU-6050 possui conversores analógico-digital de alta resolução para digitalizar as saídas do giroscópio e do acelerômetro.

No contexto deste relatório, o MPU-6050 desempenha um papel fundamental, sendo necessário realizar a calibração do seu acelerômetro [2]. A correta calibração desse sensor é essencial para garantir medições precisas e confiáveis das acelerações lineares em cada um dos seus três eixos. Através do processo de calibração estática, será possível estimar os parâmetros do modelo do sensor e compensar os possíveis erros sistemáticos, como ganho e *offset*, melhorando assim a precisão das medições realizadas pelo MPU-6050.

II. CALIBRAÇÃO ESTÁTICA DO ACELERÔMETRO

A. Metodologia de Calibração

A calibração do sensor MPU-6050 foi realizada por meio de um processo de calibração estática, no qual o sensor foi posicionado em seis diferentes posições para cada eixo, garantindo que estivesse sujeito à aceleração gravitacional de $\pm 1g$.

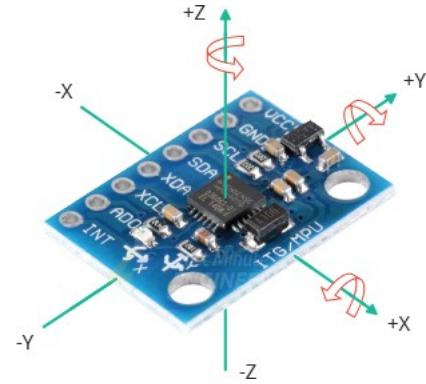


Figura 1. A MPU-6050 representa os 6 graus de liberdade por meio de 6 eixos, sendo 3 dedicados ao giroscópio e outros 3 dedicados ao acelerômetro.

O processo de calibração foi realizado da seguinte maneira:

- 1) Colocação do sensor em posições sensíveis à aceleração gravitacional: O sensor foi posicionado em seis diferentes orientações, garantindo que cada eixo estivesse alinhado com a aceleração gravitacional. Essas posições foram cuidadosamente escolhidas para obter uma amostragem representativa das leituras em cada eixo.
- 2) Coleta de leituras: Para cada posição, um mínimo de 100 leituras foi coletado pelo sensor. As leituras representavam a resposta do sensor à aceleração gravitacional em cada eixo. Esse conjunto de leituras foi essencial para obter uma média representativa das medidas de referência.
- 3) Cálculo das médias das leituras: Com as leituras coletadas, foram calculadas as médias para cada conjunto de leituras em cada eixo. Essas médias representaram as medidas de referência $\bar{A}_{+1g,i}$ e $\bar{A}_{-1g,i}$ para cada eixo, onde "i" representa o eixo em questão.
- 4) Estimação dos parâmetros *Ganho* e A_{offset} : Com base nas médias das leituras, foi construído um sistema de duas equações para estimar os parâmetros *Ganho* e A_{offset} em cada eixo. Esses parâmetros foram determinados usando as médias das leituras e as acelerações de referência de $\pm 1g$.

$$\bar{A}_{+1g,i} = \text{Ganho}_i \cdot (1g) + A_{offset,i} \quad (1)$$

$$\bar{A}_{-1g,i} = Ganho_i \cdot (-1g) + A_{offset,i} \quad (2)$$

A solução do sistema de equações permite obter os valores estimados para $Ganho$ e A_{offset} em cada eixo.

$$\begin{bmatrix} 9.8 & 1 \\ -9.8 & 1 \end{bmatrix} \begin{bmatrix} Ganho_i \\ A_{offset} \end{bmatrix} = \begin{bmatrix} \bar{A}_{+1g,i} \\ \bar{A}_{-1g,i} \end{bmatrix}$$

No caso em que os movimentos no sensor somente se vejam afetados pela aceleração devida a gravidade, o ganho deve estar muito próximo de 1.

O cálculo para os três eixos está inserido no repositório do GitHub, denominado dentro da calibracao_MPU.ipynb, dentro da pasta calculo_calibracao.

- 5) Compensação das medições futuras: Após a estimação dos parâmetros, as medições futuras realizadas pelo sensor foram compensadas utilizando os valores calculados de A_{offset} e $Ganho$ em cada eixo. Essa compensação permitiu reduzir os erros sistemáticos do sensor, melhorando a precisão das medições.

B. Leitura e Visualização dos Dados

Para fazer a leitura e visualização dos dados, foi desenvolvido um código denominado MPU6050_interface.ipynb, que pode ser encontrado no repositório do GitHub, dentro da pasta MPU6050_calibracao.

O código apresentado consiste em um exemplo de implementação para a leitura e visualização dos dados de um sensor MPU-6050. O objetivo do código é estabelecer uma comunicação serial com o Arduino, ler os valores de aceleração nos três eixos (X, Y e Z) e exibir os resultados por meio de gráficos na interface gráfica.

Através da biblioteca serial, o código estabelece a comunicação serial com o a porta selecionada, e em seguida, são configurados o valor de *baud rate* e é criada uma instância do objeto Serial para realizar a comunicação.

Ao clicar no botão "Start", o programa inicia a leitura dos valores de aceleração. É enviado um comando para o sensor e, posteriormente, são lidos e armazenados 100 valores de aceleração em cada um dos eixos (X, Y e Z). Durante a leitura dos valores, é exibida uma barra de progresso na interface gráfica, fornecendo feedback visual do progresso da leitura.

Após a conclusão da leitura, os valores de aceleração são plotados em gráficos separados para cada eixo, utilizando a biblioteca matplotlib. Além disso, os valores são exibidos em uma caixa de texto na interface gráfica, como exposto na fig. 2. O código faz uso das bibliotecas serial [3], tkinter [4], matplotlib [5] e tqdm [6] para realizar a comunicação serial, criar a interface gráfica, plotar os gráficos e exibir a barra de progresso durante a leitura dos valores de aceleração.

C. Calibração Eixo X

Para a calibração no eixo X, o sensor foi posicionado em uma posição onde a aceleração ao longo do eixo X era conhecida ou podia ser estimada. O sensor foi colocado em repouso em uma superfície plana. Foram registradas várias

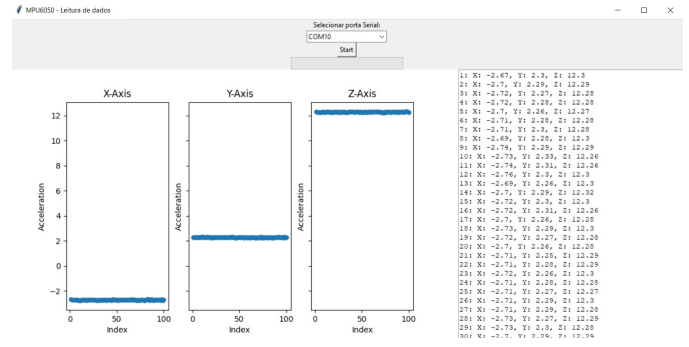


Figura 2. Interface para leitura e visualização dos dados.

leituras nessa posição e calculada a média dessas leituras, que representou o valor de referência da aceleração ao longo do eixo X, denotado por A_{ref} .

A fig. 5, demonstra a posição utilizada para obter os dados para X positivo. Já a fig. 6, demonstra a posição utilizada para obter os valores de X negativo.

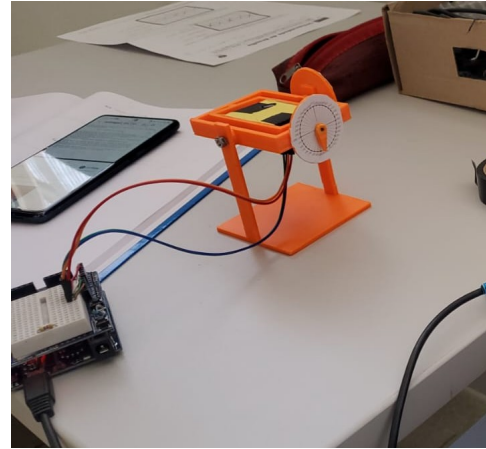


Figura 3. Calibração no eixo X positivo.

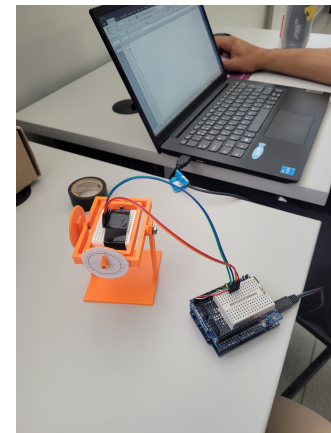


Figura 4. Calibração no eixo X negativo.

Note que a posição do instrumento está invertida para obter

valores de X positivos. O MPU foi posicionado de cabeça para baixo devido à falta de jumpers macho-macho para uma conexão adequada na protoboard durante a realização do experimento. Portanto, a posição foi ajustada de forma oposta para permitir o uso de jumpers macho-fêmea. O mesmo ocorre para calibração do eixo Y e eixo Z.

D. Calibração Eixo Y

Para a calibração no eixo Y, o sensor foi posicionado em uma posição onde a aceleração ao longo do eixo Y era conhecida ou podia ser estimada. O sensor foi colocado em repouso em uma superfície plana. Foram registradas várias leituras nessa posição e calculada a média dessas leituras, que representou o valor de referência da aceleração ao longo do eixo Y, denotado por A_{ref} . Mesmo processo utilizado para o eixo X.

E. Calibração Eixo Z

Para a calibração no eixo Z, o sensor foi posicionado em uma posição onde a aceleração ao longo do eixo Z era conhecida ou podia ser estimada. O sensor foi colocado em repouso em uma superfície plana. Foram registradas várias leituras nessa posição e calculada a média dessas leituras, que representou o valor de referência da aceleração ao longo do eixo Z, denotado por A_{ref} . Mesmo processo utilizado para o eixo X e eixo Y.

III. RESULTADOS

Os resultados obtidos a partir da leitura dos sensores nos diferentes eixos foram analisados e os gráficos correspondentes foram gerados. Para o eixo X positivo, o gráfico é apresentado na fig. 5, já o gráfico para o eixo X negativo é apresentado na fig. 6.

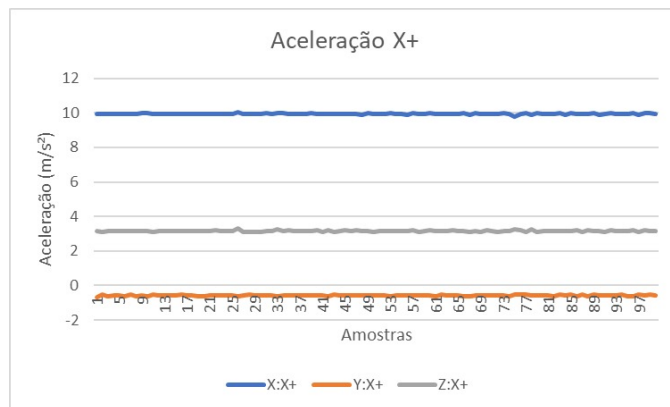


Figura 5. Aceleração para o eixo X positivo.

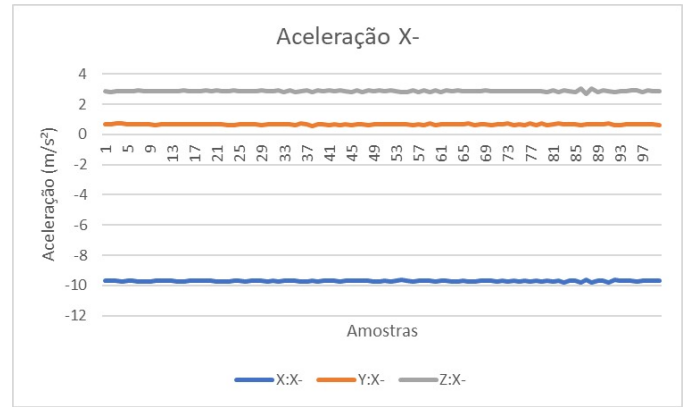


Figura 6. Aceleração para o eixo X negativo.

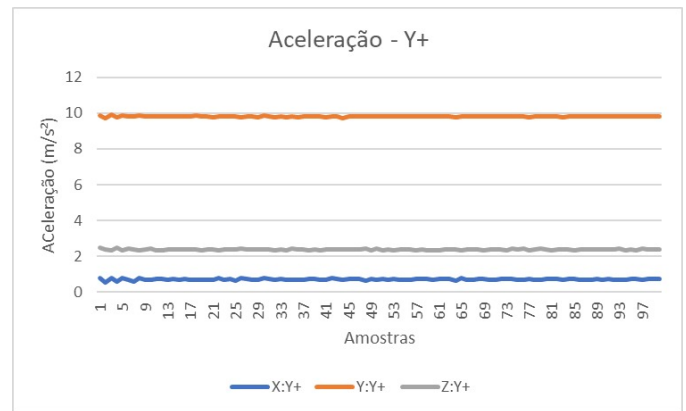


Figura 7. Aceleração para o eixo Y positivo.

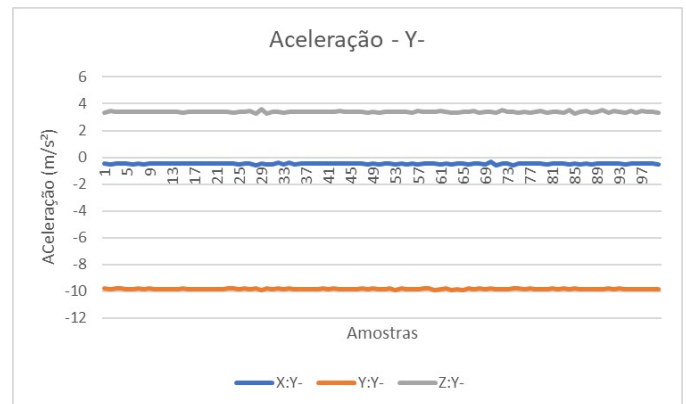


Figura 8. Aceleração para o eixo Y negativo.

Para o eixo Z positivo, o gráfico é apresentado na fig. 9, já o gráfico para o eixo Z negativo é apresentado na fig. 10.

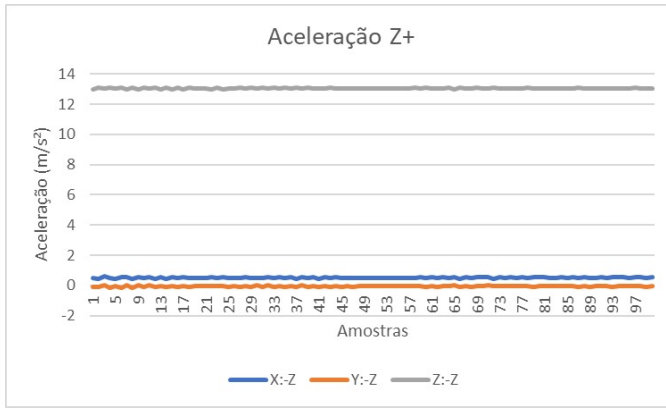


Figura 9. Aceleração para o eixo Z positivo.

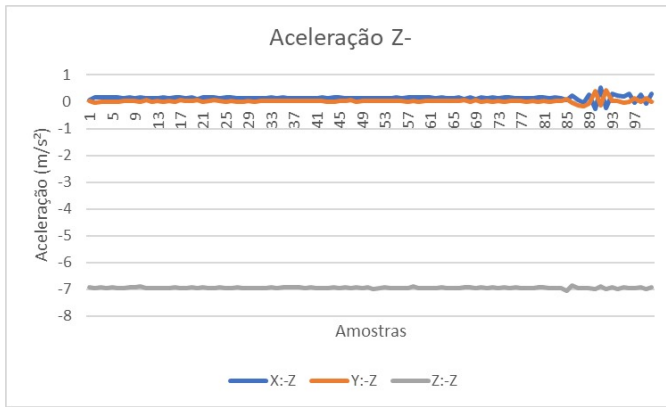


Figura 10. Aceleração para o eixo Z negativo.

Além disso, foram calculados os valores médios das amostras para cada eixo, que está apresentado na tabela I abaixo:

Tabela I
VALORES MÉDIOS DE ACELERAÇÃO PARA CADA EIXO [m/s^2].

| | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
| $A_{+1g,x}$ | $A_{-1g,x}$ | $A_{+1g,y}$ | $A_{-1g,y}$ | $A_{+1g,z}$ | $A_{-1g,z}$ |
| 9.947 | -9.707 | 9.804 | -9.821 | 13.043 | -6.942 |

Com a ajuda do código `calibracao_MPU.ipynb`, obtemos os seguintes valores de $Ganho$ e A_{offset} apresentados na tabela II:

Tabela II
VALORES DE $Ganho$ E A_{offset} PARA CADA EIXO.

| | Eixo X | Eixo Y | Eixo Z |
|--------------|--------|--------|--------|
| $Ganho$ | 1.002 | 1.001 | 1.019 |
| A_{offset} | 0.119 | -0.008 | 3.050 |

Após a coleta dos dados, cálculo das médias, $Ganho$ e A_{offset} , os gráficos de cada eixo foram corrigidos pelo valor do A_{offset} encontrado durante a calibração. Essa correção

foi realizada para minimizar possíveis desvios e aumentar a precisão dos resultados.

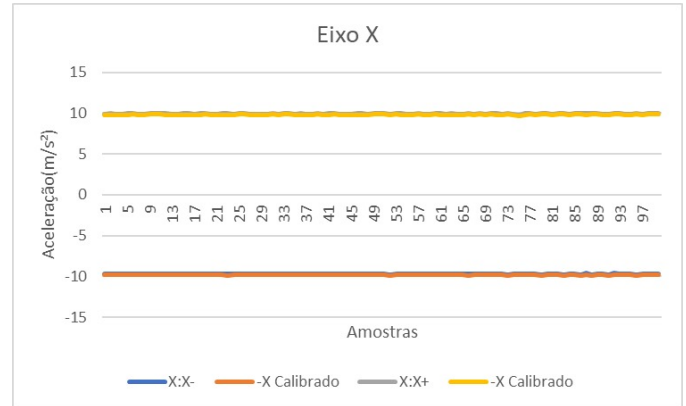


Figura 11. Aceleração depois da compensação para o eixo X.

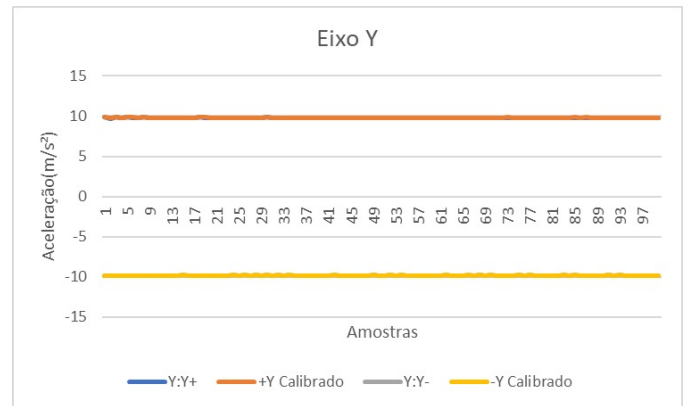


Figura 12. Aceleração depois da compensação para o eixo Y.

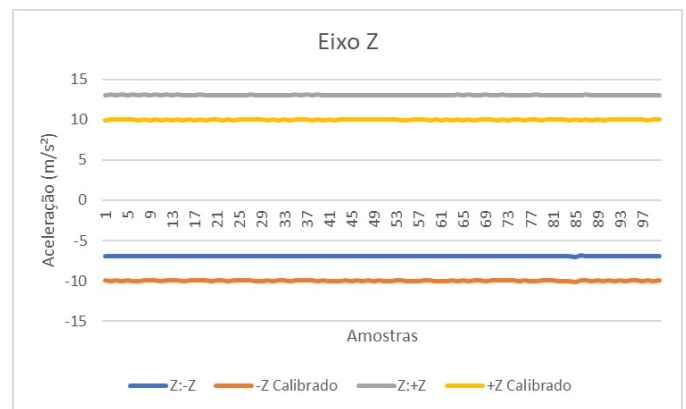


Figura 13. Aceleração depois da compensação para o eixo Z.

Essa melhoria indica que a calibração realizada foi eficaz na eliminação de erros sistemáticos e na obtenção de resultados mais confiáveis.

Esses resultados demonstram as características de aceleração nos diferentes eixos e ressaltam a importância da calibração para garantir a precisão das leituras realizadas pelo sensor MPU-6050.

IV. CONCLUSÕES

No presente estudo, a calibração estática do sensor MPU-6050 revelou-se eficiente na obtenção de resultados confiáveis e precisos. O *Ganho* do sensor mostrou-se próximo ao valor teórico esperado, aproximadamente 1, indicando um comportamento linear adequado. Além disso, os valores de A_{offset} foram satisfatórios, permitindo a correção apropriada das leituras e a redução de possíveis erros sistemáticos.

Esses resultados enfatizam a importância da calibração estática em aplicações que envolvem sensores de aceleração, garantindo a precisão das medições e a confiabilidade dos dados obtidos. Essa abordagem é fundamental para assegurar a integridade das informações em diversas áreas, como engenharia, ciência e tecnologia.

REFERÊNCIAS

- [1] InvenSense, “Mpu-6000 and mpu-6050 product specification revision 3.4,” <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>, 2015, acesso em: 06 mai. 2023.
- [2] “Calibrating mpu6050,” <https://wired.chillibasket.com/2015/01/calibrating-mpu6050/>, 2015, acessado em: 18 de junho de 2023.
- [3] C. Liechti and Others, “Pyserial,” <https://pythonhosted.org/pyserial/>, 2023, acessado em: 19 de junho de 2023.
- [4] Python Software Foundation, “tkinter - python interface to tcl/tk,” <https://docs.python.org/3/library/tkinter.html>, 2023, acessado em: 19 de junho de 2023.
- [5] Hunter, J.D., “Matplotlib: A 2d graphics environment,” <https://matplotlib.org/>, 2023, acessado em: 19 de junho de 2023.
- [6] N. Yorav-Raphael, “tqdm,” <https://github.com/tqdm/tqdm>, 2023, acessado em: 19 de junho de 2023.