UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO DE JOINVILLE

CURSO DE ENGENHARIA MECATRÔNICA

Relatório Técnico: Sistema de Leitura de Orientação com MPU6050 para Submarino Autônomo

Luan Misael dos Santos

Joinville

2025

Conteúdo

1 Introdução

Este relatório descreve a implementação de um sistema baseado no sensor inercial MPU6050 para um submarino autônomo, utilizando o microcontrolador Raspberry Pi Pico no ambiente Arduino. O MPU6050, que combina acelerômetro e giroscópio, fornece dados de aceleração (m/s^2) e velocidade angular (rad/s) nos eixos X, Y e Z, essenciais para controle de orientação e estabilização. O código foi migrado do Arduino Uno para o Pico, com otimizações para reduzir a poluição no Monitor Serial e estabilizar leituras por meio de filtros (limiar e offsets de calibração). Todos os testes foram realizados em bancada, sem validação em condições subaquáticas reais, o que limita a aplicabilidade dos resultados em cenários operacionais.

2 Hardware Utilizado

- **Sensor MPU6050**: Módulo inercial com acelerômetro e giroscópio de 3 eixos, configurado para ± 2g (acelerômetro) e $\pm 250^{o}$/s (giroscópio).
- **Microcontrolador**: Raspberry Pi Pico, responsável por leitura, processamento e comunicação serial via USB.
- **Comunicação**: Protocolo I2C (pinos GP4/SDA, GP5/SCL) para interface com o sensor.
- **Interface Serial**: USB para monitoramento e depuração no Monitor Serial (57600 baud).

3 Lógica do Sistema

3.1 Inicialização

A inicialização utiliza a biblioteca `Wire.h` para configurar a comunicação I2C a 100 kHz. O sensor é configurado com:

- Faixa do acelerômetro: ± 2g (registrador 0x1C = 0x00).
- Faixa do giroscópio: $\pm 250^{o}$/s (registrador 0x1B = 0x00).
- Modo sleep desativado (registrador 0x6B = 0x00).

O registrador WHO_AM_I (0x75) é verificado (valor esperado: 0x68) para confirmar a conexão.

3.2 Calibração

A calibração, realizada com o sensor parado, coleta 3000 leituras (20 ms de intervalo) para calcular offsets do acelerômetro (AccX/Y/Z, em g) e giroscópio (GyroX/Y/Z, em ^{o}/s). Offsets são aplicados para zerar AccX/Y e GyroX/Y/Z, e ajustar AccZ para ≈ 9.80665 m/s^2 (gravidade).

3.3 Leitura e Processamento de Dados

O sistema lê continuamente:

- **Aceleração** (eixos X, Y, Z, em m/s^2): Convertida de dados brutos (16384 LSB/g) e ajustada com offsets.
- **Velocidade angular** (eixos X, Y, Z, em rad/s): Convertida de dados brutos (131 LSB/^{o}/s) e ajustada com offsets.

Um filtro de limiar zera valores de AccX/Y abaixo de 0.49 m/s^2 (0.05g) e GyroX/Y/Z abaixo de 0.0087 rad/s (0.5^o/s), eliminando ruído. A norma da aceleração ($\sqrt{\text{AccX}^2 + \text{AccY}^2 + \text{AccZ}^2}$) é calculada como métrica auxiliar.

3.4 Saída de Dados

Os dados são exibidos no Monitor Serial a cada 50 ms (20 Hz), incluindo:

- Aceleração (AccX, AccY, AccZ, em m/s^2, 3 casas decimais).
- Velocidade angular (GyroX, GyroY, GyroZ, em rad/s, 3 casas decimais).
- Norma da aceleração (m/s^2, 2 casas decimais).

Exemplo de saída (sensor parado):

```
AccX (m/sš): 0.000
AccY (m/sš): 0.000
AccZ (m/sš): 9.807
GyroX (rad/s): 0.000
GyroY (rad/s): 0.000
GyroZ (rad/s): 0.000
Norma (m/sš): 9.81
```

4 Estrutura do Código

O projeto está organizado em dois arquivos:

4.1 mpu6050.h

Contém funções para:

- Inicialização do sensor (`mpu_begin`).
- Leitura de dados brutos (`readAcel`, `readGiro`).
- Calibração (`mpu_calibrate`).

- Processamento e acesso aos dados (`mpu_loop`, `getAccX`, etc.).

4.2 giroscopio.ino

Controla o fluxo principal:

- Executa a calibração inicial.
- Realiza leituras contínuas a 20 Hz.
- Exibe resultados no Monitor Serial.

5 Resultados Obtidos

- **Estabilidade**: As leituras de AccX/Y e GyroX/Y/Z são zeradas quando o sensor está parado, e AccZ é estabilizado em ≈ 9.80665 m/s^2, graças aos offsets e filtro de limiar.
- **Norma**: A norma da aceleração (≈ 9.81 m/s^2 quando parado) confirma a consistência dos dados.
- **Redução de ruído**: O filtro de limiar elimina flutuações pequenas, e a saída no Monitor Serial foi otimizada para maior clareza.

6 Limitações

- **Testes em bancada**: Os experimentos foram limitados a ambiente controlado, sem validação em condições subaquáticas (e.g., vibrações, pressão, umidade).
- **Filtro simplificado**: A remoção do filtro de Kalman reduz a complexidade, mas limita a fusão de dados para ângulos de orientação.

7 Melhorias Futuras

- **Reintrodução do filtro de Kalman**: Para calcular ângulos de roll e pitch com maior precisão.
- **Testes subaquáticos**: Validar o sistema em condições reais, considerando interferências ambientais.
- **Controle PID**: Implementar um controlador PID para estabilização automática do submarino.
- **Armazenamento**: Adicionar um cartão SD para registrar leituras.

8 Conclusão

O sistema com MPU6050 e Raspberry Pi Pico demonstrou eficácia na leitura de aceleração e velocidade angular em bancada, com dados estabilizados por offsets e filtro de limiar. A norma da aceleração (≈ 9.81 m/s^2) e a ausência de ruído nas leituras confirmam a robustez do sistema. Contudo, a falta de testes subaquáticos e de cálculo de ângulos de orientação destaca a necessidade de melhorias, como validação em condições reais e integração com controladores. Com as otimizações propostas, o sistema tem potencial para atender às demandas de navegação autônoma em aplicações subaquáticas.