*※보고서 제목 양식 예 : 이름\_20240403*

|  |
| --- |
| **2024 ALTIS SW 개인 활동 보고서** |

|  |  |
| --- | --- |
| **활동 개요** | 가속도 자이로 값으로 각도 추정하는 방법(필터,적분,중력가속도) |
| **일시** | 2024년 04월 12일 21시 |
| **작성자** | 김승창 |
| **활동 내용** | 가속도 자이로 값으로 각도를 추정하는 방법에는 여러 가지가 있습니다.  1. 상보 필터 (Complementary Filter):  가속도계와 자이로스코프를 사용하여 각도를 추정하는 방법 중 하나입니다. 가속도계는 저주파수 영역에서 정확하고, 고주파수 영역에서 노이즈가 많습니다. 자이로스코프는 단기간의 정확도가 높지만 장기간의 측정에는 drift가 발생할 수 있습니다. 상보 필터는 이 두 가지 센서를 조합하여 각도 추정에 사용됩니다. 가속도계로는 저주파수 성분을 추정하고, 자이로스코프로는 고주파수 성분을 보상하여 최종 각도를 추정합니다.  2. 칼만 필터 (Kalman Filter):  칼만 필터는 시스템의 상태를 추정하는 데 사용되는 재귀적인 추정 알고리즘입니다. 가속도계와 자이로스코프의 측정 값을 사용하여 시스템의 상태(여기서는 각도)를 추정합니다. 칼만 필터는 선형 시스템 및 가우시안 잡음에 대해 가장 효과적입니다.  3. 보상 적분 (Integral Compensation):  자이로스코프를 사용하여 각 속도를 측정하고 이를 적분하여 각 위치(즉, 각도)를 추정하는 방법입니다. 그러나 이 방법은 자이로스코프의 drift에 영향을 받을 수 있으므로 장기간 사용할 경우 보정이 필요합니다.  4. 중력 가속도 활용 (Utilizing Gravity Acceleration):  가속도계를 사용하여 중력 가속도 성분을 추출하고, 이를 기준으로 각도를 추정하는 방법입니다. 이 방법은 기본적으로 센서가 고정되어 있는 상태에서만 사용 가능하며, 자이로스코프의 drift에 영향을 받지 않습니다.  상보필터 사용 이유:  가속도센서는 진동에 약하기 때문에 짧은 시간의 변화에는 약하지만, 긴 시간에 대해서는 정확한 각도 값을 줄 수 있습니다. 반면, 자이로센서는 진동에 강하기 때문에 짧은 시간에 대해선 정확한 각도를 얻어낼 수 있지만 오차가 쌓이기 때문에 긴 시간에 대해서는 취약합니다. 이러한 문제를 해결할 수 있는 게 상보필터입니다.  #include<Wire.h>  const int MPU\_ADDR = 0x68;  int16\_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;  double angleAcX, angleAcY, angleAcZ;  double angleGyX, angleGyY, angleGyZ;  double angleFiX, angleFiY, angleFiZ;//필터를 통해 얻어진 각도를 저장하는 변수를 설정  const double RADIAN\_TO\_DEGREE = 180 / 3.14159;  const double DEGREE\_PER\_SECOND = 32767 / 250;  const double ALPHA = 1 / (1 + 0.04);  unsigned long now = 0;  unsigned long past = 0;  double dt = 0;  double baseAcX, baseAcY, baseAcZ;  double baseGyX, baseGyY, baseGyZ;  void setup() {  initSensor();  Serial.begin(115200);  calibrateSensor();  past = millis();  }  void loop() {  getData();  getDT();  angleAcY = atan(-AcX / sqrt(pow(AcY, 2) + pow(AcZ, 2)));  angleAcY \*= RADIAN\_TO\_DEGREE;  angleGyY = ((GyY - baseGyY) / DEGREE\_PER\_SECOND) \* dt;//자이로센서로부터 얻은 각도를 적분  double angleTmp = angleFiY + angleGyY;  angleFiY = ALPHA \* angleTmp + (1.0 - ALPHA) \* angleAcY;//상보필터    Serial.println(angleFiY);  }  void initSensor() {  Wire.begin();  Wire.beginTransmission(MPU\_ADDR);  Wire.write(0x6B);  Wire.write(0);  Wire.endTransmission(true);  }  void getData() {  Wire.beginTransmission(MPU\_ADDR);  Wire.write(0x3B);  Wire.endTransmission(false);  Wire.requestFrom(MPU\_ADDR, 14, true);  AcX = Wire.read() << 8 | Wire.read();  AcY = Wire.read() << 8 | Wire.read();  AcZ = Wire.read() << 8 | Wire.read();  Tmp = Wire.read() << 8 | Wire.read();  GyX = Wire.read() << 8 | Wire.read();  GyY = Wire.read() << 8 | Wire.read();  GyZ = Wire.read() << 8 | Wire.read();  }  void getDT() {  now = millis();  dt = (now - past) / 1000.0;  past = now;  }  void calibrateSensor() {  double sumAcX = 0, sumAcY = 0, sumAcZ = 0;  double sumGyX = 0, sumGyY = 0, sumGyZ = 0;  getData();  for (int i = 0; i < 10 ; i++) {  getData();  sumAcX += AcX; sumAcY += AcY; sumAcZ += AcZ;  sumGyX += GyX; sumGyY += GyY; sumGyZ += GyZ;  delay(100);  }  baseAcX = sumAcX / 10;  baseAcY = sumAcY / 10;  baseAcZ = sumAcZ / 10;  baseGyX = sumGyX / 10;  baseGyY = sumGyY / 10;  baseGyZ = sumGyZ / 10;  } |
| **활동 사진** | mpu6050 예제 코드는 잘 돌아갔으나 다른 코드를 돌릴때 보드가 안찾아지는 문제가 생겨 다른 코드를 돌려보지 못했습니다. |
| **활동 결과** | 가속도 자이로 값으로 각도를 추정하는 방법에는 여러 가지가 있습니다. 가속도센서는 진동에 약하기 때문에 짧은 시간의 변화에는 약하지만, 긴 시간에 대해서는 정확한 각도 값을 줄 수 있습니다. 반면, 자이로센서는 진동에 강하기 때문에 짧은 시간에 대해선 정확한 각도를 얻어낼 수 있지만 오차가 쌓이기 때문에 긴 시간에 대해서는 취약합니다. 이러한 문제를 상보필터를 통해 해결 할 수 있습니다. |
| **계획** | 보드 연결 관련 문제를 해결해오겠습니다. |

2024년 04월 13일

작성자 : 김승창 (인)