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

|  |  |
| --- | --- |
| **활동 개요** | 가속도 자이로 값으로 각도 추정하는 방법  (필터,적분,중력가속도) |
| **일시** | 2024년 04월 12일 21시 |
| **작성자** | 박상하 |
| **활동 내용** | 가속도 자이로 값으로 각도를 추정하는 방법  1. 상보 필터  1. 상보 필터 (Complementary Filter):  - 가속도계와 자이로스코프를 사용하는 방법.  2. 칼만 필터 (Kalman Filter):  - 시스템의 상태를 추정하는 데 사용하는 방법.  - 가속도계와 자이로스코프의 측정 값을 사용하여 시스템의 상태인 각도를 사용하여 추정.  3. 보상 적분 (Integral Compensation):  - 자이로스코프를 사용하는 방법.  - 각 속도 측정 후, 적분하여 위치(즉, 각도) 추정.  칼만 필터 사용 이유:  시스템의 상태를 추정할 때, 지금까지의 누적된 결과들을 반영하는 방법이 정확성을 높이고, 다른 요인에 의해서 변동 가능성이 매우 적기 때문에 칼만 필터를 선택하게 되었다. 그리고 제일 중요한 실시간으로 데이터를 처리하는 부분에서 가장 큰 영향을 미쳤다고 생각한다. |
| **활동 사진** | 사용 코드:  #include <Wire.h>  #define mpu\_add 0x68  //mpu6050 address  class kalman {    public :      double getkalman(double acc, double gyro, double dt) {        //project the state ahead        angle += dt \* (gyro - bias) ;        //Project the error covariance ahead        P[0][0] += dt \* (dt \* P[1][1] - P[0][1] - P[1][0] + Q\_angle) ;        P[0][1] -= dt \* P[1][1] ;        P[1][0] -= dt \* P[1][1] ;        P[1][1] += Q\_gyro \* dt ;        //Compute the Kalman gain        double S = P[0][0] + R\_measure ;        K[0] = P[0][0] / S ;        K[1] = P[1][0] / S ;        //Update estimate with measurement z        double y = acc - angle ;        angle += K[0] \* y ;        bias += K[1] \* y ;        //Update the error covariance        double P\_temp[2] = {P[0][0], P[0][1]} ;        P[0][0] -= K[0] \* P\_temp[0] ;        P[0][1] -= K[0] \* P\_temp[1] ;        P[1][0] -= K[1] \* P\_temp[0] ;        P[1][1] -= K[1] \* P\_temp[1] ;        return angle ;      } ;      void init(double angle, double gyro, double measure) {        Q\_angle = angle ;        Q\_gyro = gyro ;        R\_measure = measure ;        angle = 0 ;        bias = 0 ;        P[0][0] = 0 ;        P[0][1] = 0 ;        P[1][0] = 0 ;        P[1][1] = 0 ;      } ;      double getvar(int num) {        switch (num) {          case 0 :            return Q\_angle ;            break ;          case 1 :            return Q\_gyro ;            break ;          case 2 :            return R\_measure ;            break ;        }      } ;    private :      double Q\_angle, Q\_gyro, R\_measure ;      double angle, bias ;      double P[2][2], K[2] ;  } ;  kalman kal ;  long ac\_x, ac\_y, ac\_z, gy\_x, gy\_y, gy\_z ;  double deg, dgy\_y ;  double dt ;  uint32\_t pasttime ;  void setup() {    // put your setup code here, to run once:    Serial.begin(9600) ;    Wire.begin() ;    Wire.beginTransmission(mpu\_add) ;    Wire.write(0x6B) ;    Wire.write(0) ;    Wire.endTransmission(true) ;    kal.init(0.001, 0.003, 0.03) ;  //init kalman filter    Serial.println() ;    Serial.print("parameter") ;    Serial.print("\t") ;    Serial.print(kal.getvar(0), 4) ;    Serial.print("\t") ;    Serial.print(kal.getvar(1), 4) ;    Serial.print("\t") ;    Serial.println(kal.getvar(2), 4) ;  }  void loop() {    // put your main code here, to run repeatedly:    Wire.beginTransmission(mpu\_add) ; //get acc data    Wire.write(0x3B) ;    Wire.endTransmission(false) ;    Wire.requestFrom(mpu\_add, 6, true) ;    ac\_x = Wire.read() << 8 | Wire.read() ;    ac\_y = Wire.read() << 8 | Wire.read() ;    ac\_z = Wire.read() << 8 | Wire.read() ;    Wire.beginTransmission(mpu\_add) ; //get gyro data    Wire.write(0x43) ;    Wire.endTransmission(false) ;    Wire.requestFrom(mpu\_add, 6, true) ;    gy\_x = Wire.read() << 8 | Wire.read() ;    gy\_y = Wire.read() << 8 | Wire.read() ;    gy\_z = Wire.read() << 8 | Wire.read() ;    deg = atan2(ac\_x, ac\_z) \* 180 / PI ;  //acc data to degree data    dgy\_y = gy\_y / 131. ;  //gyro output to    dt = (double)(micros() - pasttime) / 1000000;    pasttime = micros();  //convert output to understandable data    double val = kal.getkalman(deg, dgy\_y, dt) ;  //get kalman data    Serial.print("kalman degree") ;    Serial.print("\t") ;    Serial.println(val) ;  }  처음 실행했을 때에는 잘 출력되다가 마지막에 결과를 출력하는 과정에서 오류가 발생하여 따로 결과화면을 촬영하지 못했습니다. |
| **활동 결과** | 칼만 필터를 사용해 mpu6050으로 직접 실행해 볼 수 있었다. 처음에는 실시간으로 값을 잘 받아오는 것을 확인할 수 있었다. 비록 직접 연결을 해서 사용하였기 때문에 로켓에서 사용하기는 어렵다는 판단을 하였다. |
| **계획** | 해당 코드를 실행할 때 왜 마지막에 다시 실행하면 결과화면에서 오류가 발생했는지 파악하여 해결해 보도록 하겠습니다. |

2024년 04월 12일

작성자 : 박상하 (인)