MỤC LỤC

[CHƯƠNG 1: PHẦN CỨNG 2](#_Toc176869433)

[1.1: SƠ ĐỒ ĐẤU DÂY 2](#_Toc176869434)

[CHƯƠNG 2: PHẦN MỀM 3](#_Toc176869435)

[2.1: MPU6050 3](#_Toc176869436)

[2.1.1: Tổng quát về MPU6050 3](#_Toc176869437)

[2.1.2: Xử lý dữ liệu từ MPU6050 3](#_Toc176869438)

[2.2: HMC5883L 6](#_Toc176869439)

[2.2.1: Tổng quát về HMC5883L 6](#_Toc176869440)

[2.2.2: Xử lý dữ liệu từ HMC5883L 6](#_Toc176869441)

[2.3: BỘ LỌC KALMAN 10](#_Toc176869442)

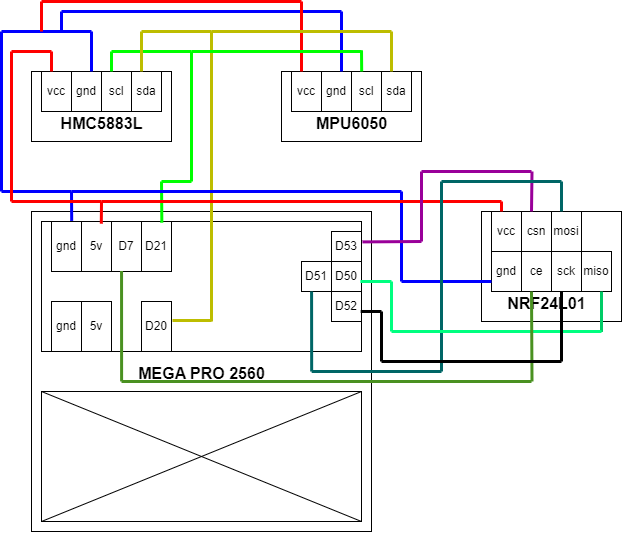
[2.3.1: Prediction: 11](#_Toc176869443)

[2.3.2: Kalman Gain: 12](#_Toc176869444)

[2.3.3: Updation: 13](#_Toc176869445)

# PHẦN CỨNG

## SƠ ĐỒ ĐẤU DÂY



Hình 1: Sơ đồ đấu dây

# PHẦN MỀM

## MPU6050

### Tổng quát về MPU6050

MPU6050 là một cảm biến kết hợp bao gồm gia tốc kế 3 trục (3-axis accelerometer) và con quay hồi chuyển 3 trục (3-axis gyroscope). Cảm biến này được sử dụng để đo lường các chuyển động liên quan đến gia tốc và góc quay (roll, pitch, yaw).

###### Các đặc điểm chính của MPU6050:

* Đo gia tốc dọc theo ba trục (X, Y, Z). Dải đo có thể điều chỉnh từ ±2g, ±4g, ±8g đến ±16g (g là gia tốc trọng trường, khoảng 9.81 m/s²).
* Đo tốc độ góc quanh ba trục (X, Y, Z). Dải đo có thể điều chỉnh từ ±250, ±500, ±1000 đến ±2000 độ/giây.
* Có thể giao tiếp với vi điều khiển hoặc các thiết bị khác qua giao thức I2C hoặc SPI.

### Xử lý dữ liệu từ MPU6050

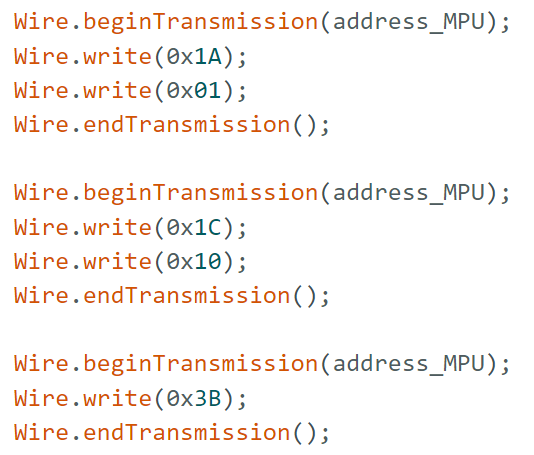
#### Đọc dữ liệu từ MPU6050

Sử dụng hàm Wire trong Arduino để truy cập vào địa chỉ I2C của cảm biến MPU6050 (0x68) để thực hiện giao tiếp với cảm biến.

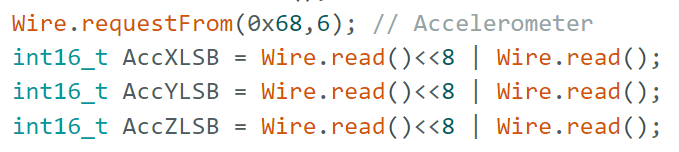
Cấu hình thanh ghi 0x1A với 0x01 để bật bộ lọc Low Pass Filter, bộ lọc sẽ có băng thông 184Hz cho Accelerometer và 188Hz cho Gyroscope. Độ trễ của bộ lọc là 2ms.

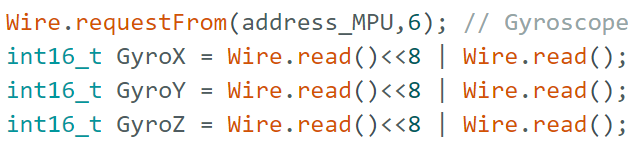
Cấu hình thanh ghi 0x1C với 0x10 để chọn full scale ở mức 8g.

Bắt đầu đọc dữ liệu của accelerometer từ thanh ghi 0x3B và đọc dữ liệu của gyroscope từ thanh ghi 0x43.



Giá trị đọc về của mỗi trục là 16 bit. Thực hiện đọc tín hiệu từ 3 trục X, Y, Z của accelerometer và 3 trục X, Y, Z của gyroscope bằng cách cộng 8 bit MSB với LSB của mỗi trục.

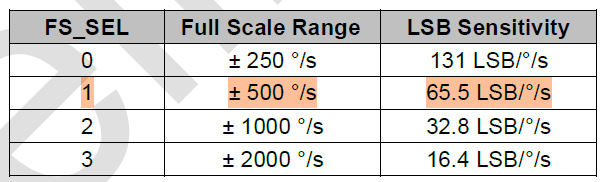




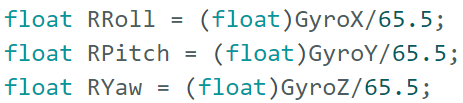
#### Calibration tín hiệu đọc từ MPU6050

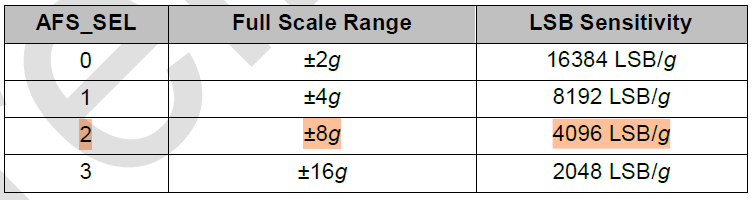
###### Các bước hiệu chỉnh tín hiệu:

Quy đổi các dữ liệu thô của Gyroscope và Accelerometer với độ nhạy đã chọn lần lượt là 65.5 và 4096.

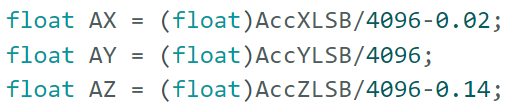


Hình 2: Gyroscope sensitivity

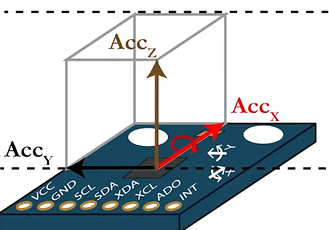




Hình 3: Accelerometer sensitivity

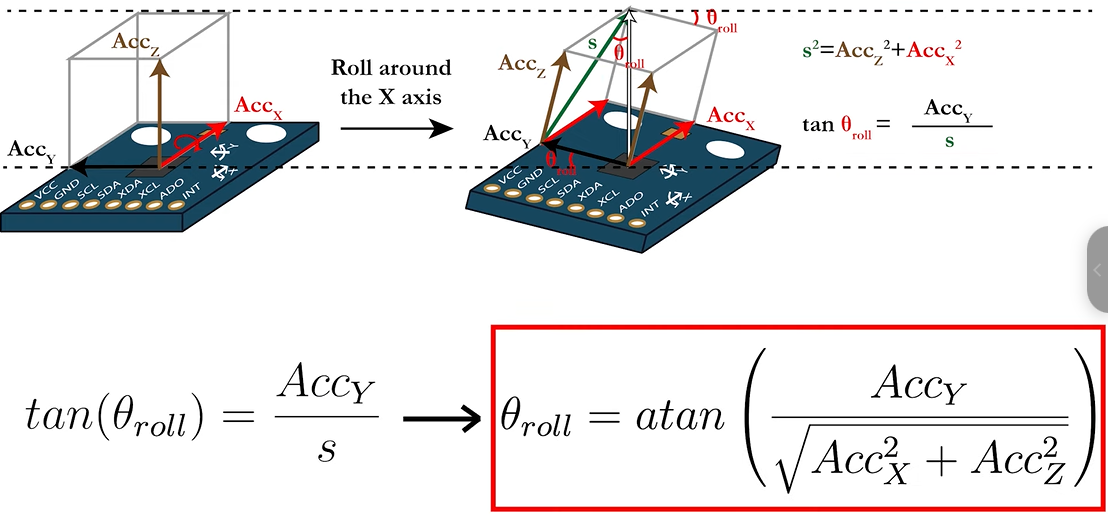


Tìm phương trình chuyển đổi MPU6050 bằng cách đặt các trục X, Y, Z như sau:



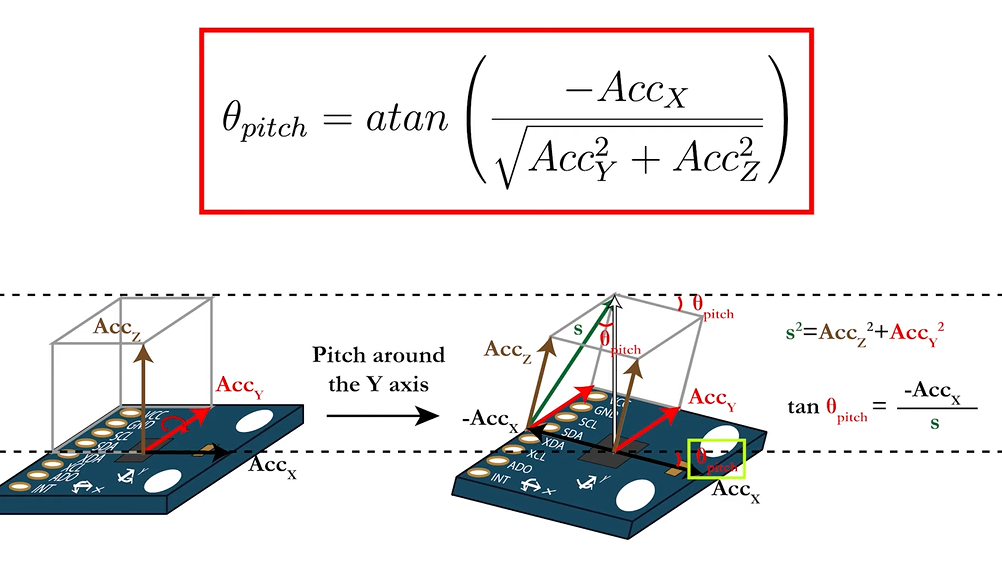
Hình 4: Hình minh họa đặt trục{accelerometer, #1}

Xoay cảm biến theo trục AccX để tính góc xoay roll của cảm biến.



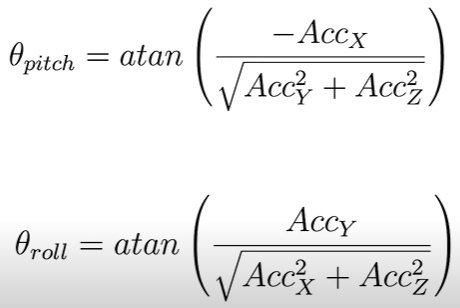
Hình 5: Tính toán và hình minh họa

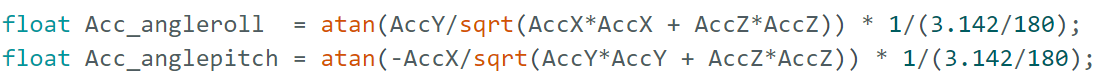
Tương tự với góc xoay pitch.



Hình 6: Tính toán và hình minh họa

Tính góc roll và góc pitch theo phương trình tìm được như sau:





Từ phương trình này, ta có được góc xoay theo trục X và Y của MPU6050.

###### Kết quả hiệu chỉnh

## HMC5883L

### Tổng quát về HMC5883L

HMC5883L là một cảm biến từ trường (la bàn số) 3 trục (three-axis magnetometer) được sử dụng để đo cường độ và hướng của từ trường xung quanh nó.

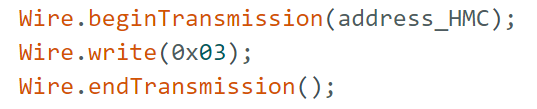
###### Các đặc điểm chính của HMC5883L:

* Đo từ trường theo ba trục không gian (X, Y, Z).
* Sử dụng giao thức giao tiếp I2C
* Có độ nhạy cao và khả năng đo chính xác từ trường. Từ 8 gauss đến mili-gauss.
* Có độ nhạy thấp theo trục chéo

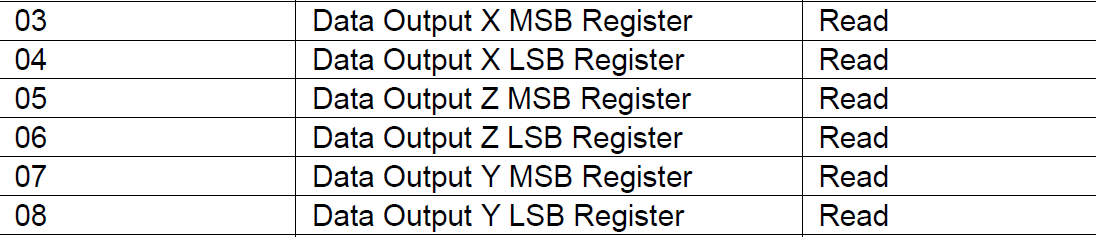
### Xử lý dữ liệu từ HMC5883L

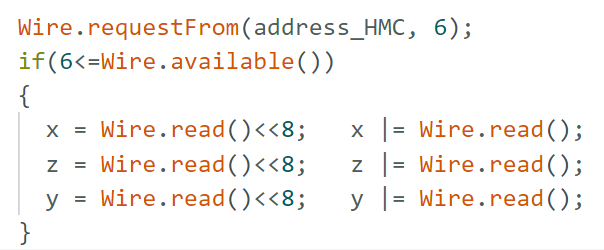
#### Đọc dữ liệu từ HMC5883L

Sử dụng hàm Wire trong Arduino để truy cập vào địa chỉ I2C của cảm biến HMC5883L (0x1E) để thực hiện giao tiếp với cảm biến.



Giá trị đọc về của mỗi trục là 16 bit. Thực hiện đọc tín hiệu từ 3 trục X, Y, Z bằng cách cộng 8 bit MSB với LSB của mỗi trục.

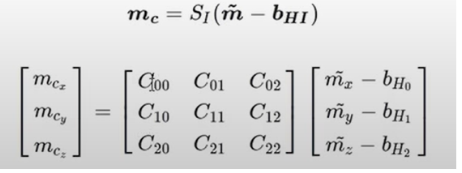




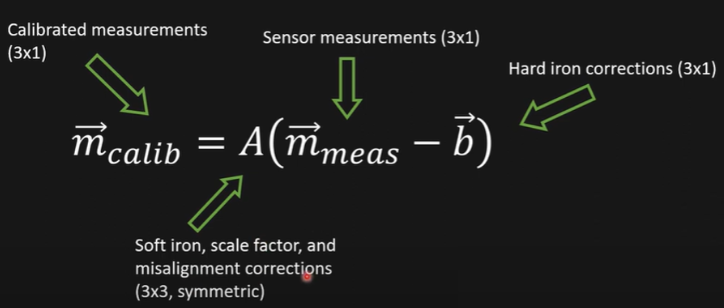
#### Hiệu chỉnh (Calibration) tín hiệu đọc từ HMC5883L

Đối với cảm biến HMC5883L, có hai loại nhiễu chính là Hard Iron Distortions (Biến dạng sắt cứng) gây ra bởi các từ trường cố định, liên tục tồn tại xung quanh cảm biến (thường có ở nam châm kim loại, các dây dẫn mang dòng điện cao,v.v), và Soft Iron Distortions (Biến dạng sắt mềm) gây ra bởi các vật liệu paramagnetic (vật liệu có tính từ kém, không giữ từ trường lâu dài) hoặc vật liệu từ tính khác làm thay đổi hình dạng từ trường địa phương (thường có ở các vật liệu như sắt, thép, v.v)

Hai loại nhiễu trên sẽ làm cho tín hiệu đọc về của hmc5883L bị méo dạng, do đó, tín hiệu đọc về cần được hiệu chỉnh. Phương trình cho việc hiểu chỉnh như sau:

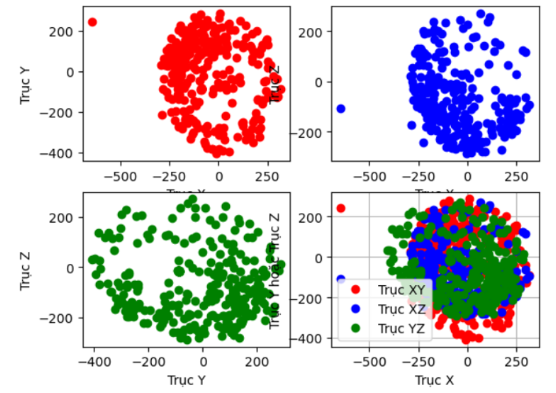


**Chú thích:**



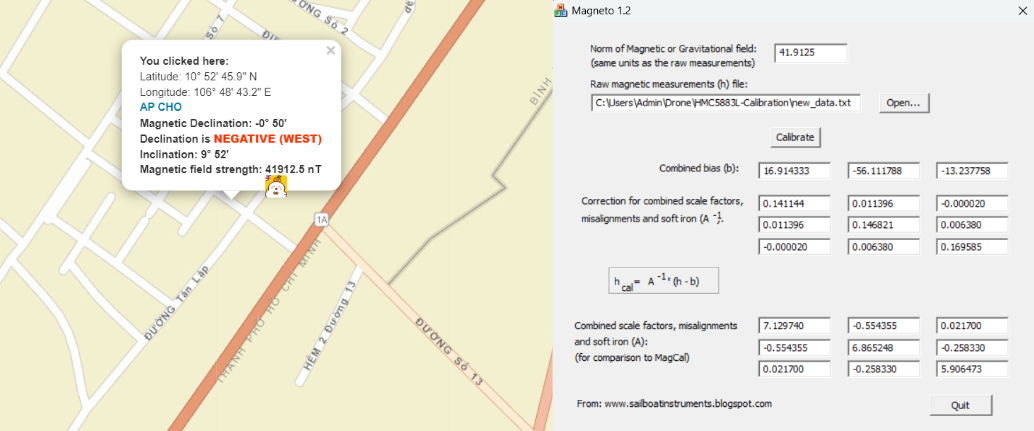
###### Các bước hiệu chỉnh tín hiệu:

* Lấy dữ liệu thô



Hình 7: Dữ liệu thô

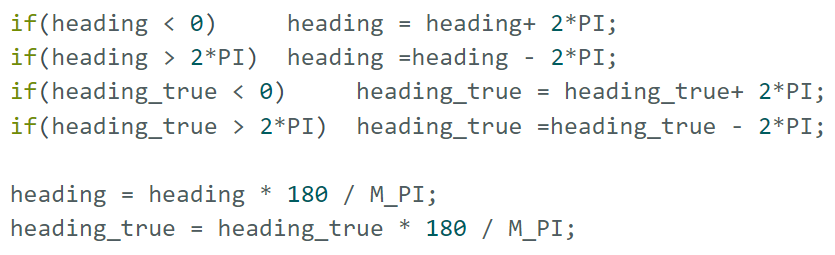
* Sử dụng phần mềm để tìm ma trận A, b và góc từ thiên (góc giữa hướng Bắc từ trường và hướng Bắc thực tế): <https://www.magnetic-declination.com/> và phần mềm magneto



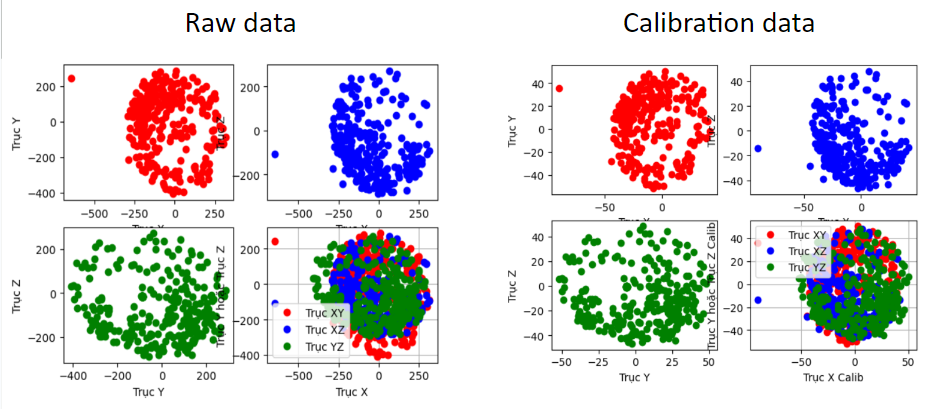
* Nhập các ma trận tìm được vào Arduino IDE và sử dụng phương trình Calib để tính toán.



* Thêm các điều kiện để chỉ lấy góc từ 0 đến 360 độ.



###### Kết quả hiệu chỉnh

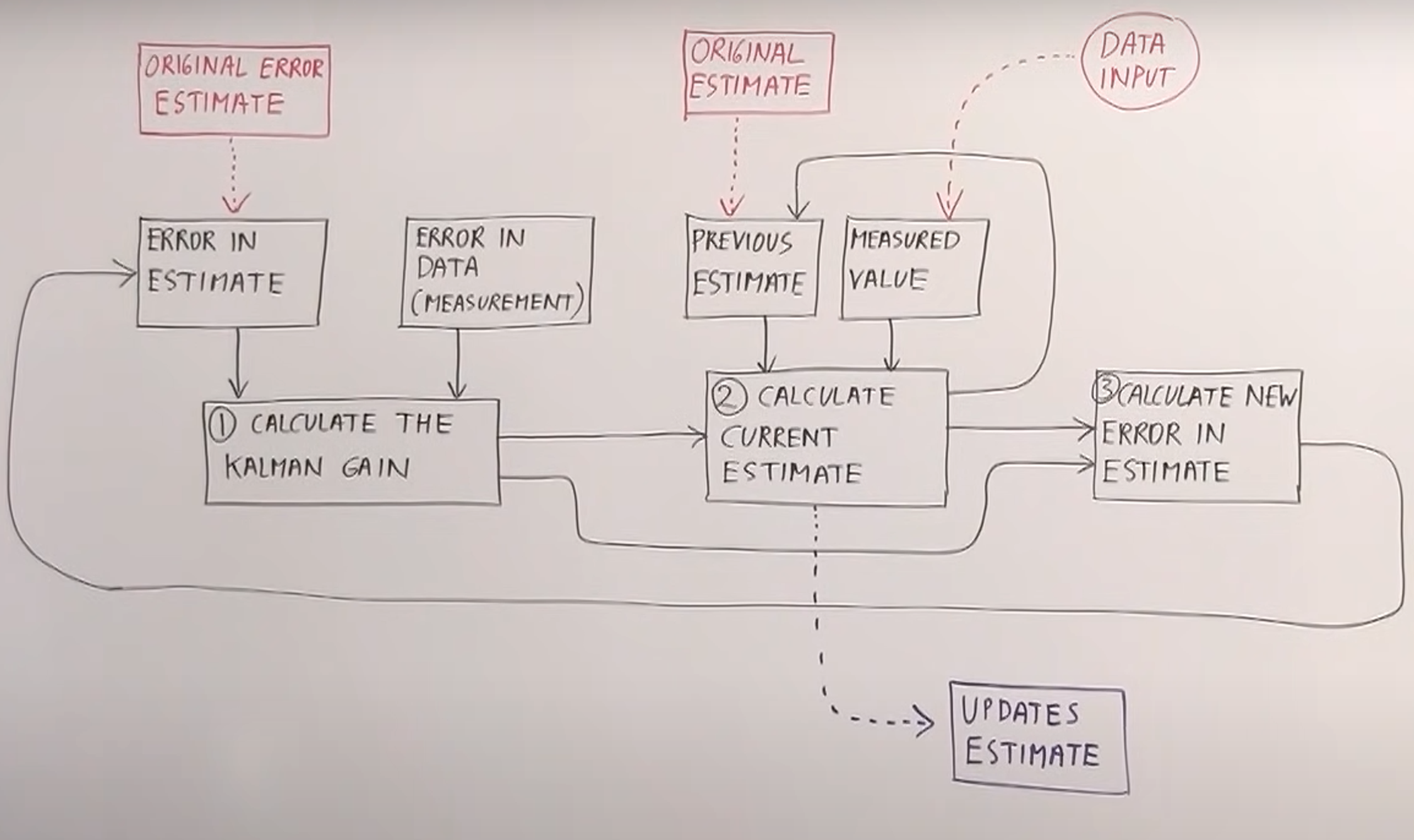


Hình 8: Kết quả hiệu chỉnh

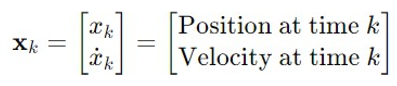


Hình 9: So sánh kết quả đo của cảm biến với điện thoại

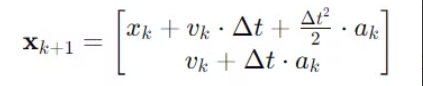
## BỘ LỌC KALMAN



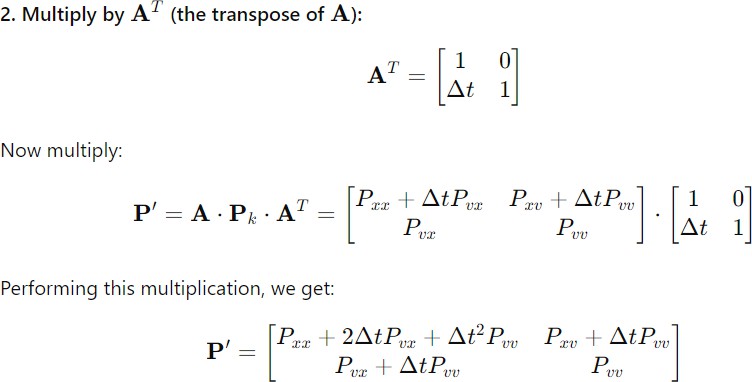
### Prediction:

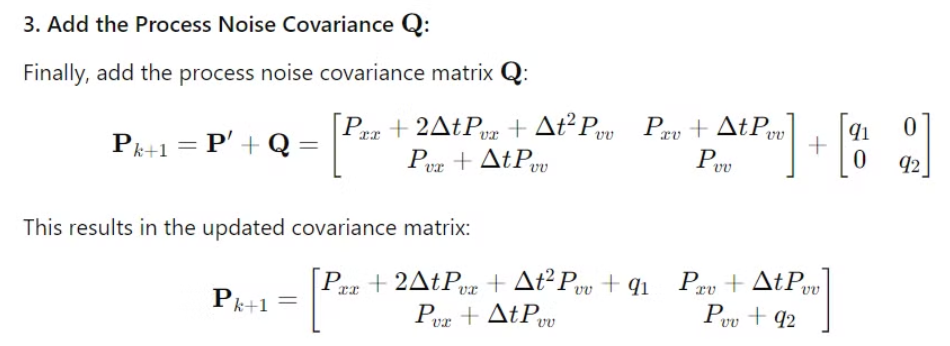
  



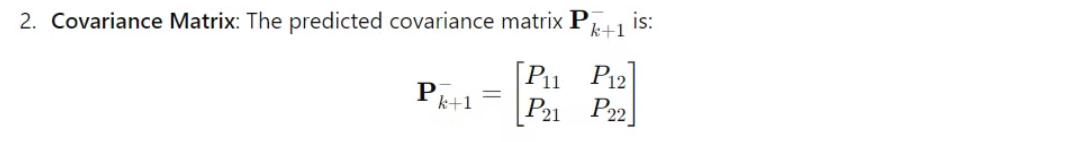


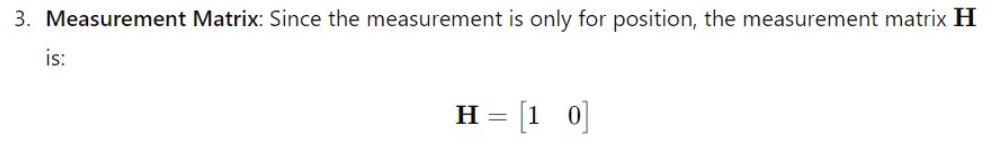






### Kalman Gain:

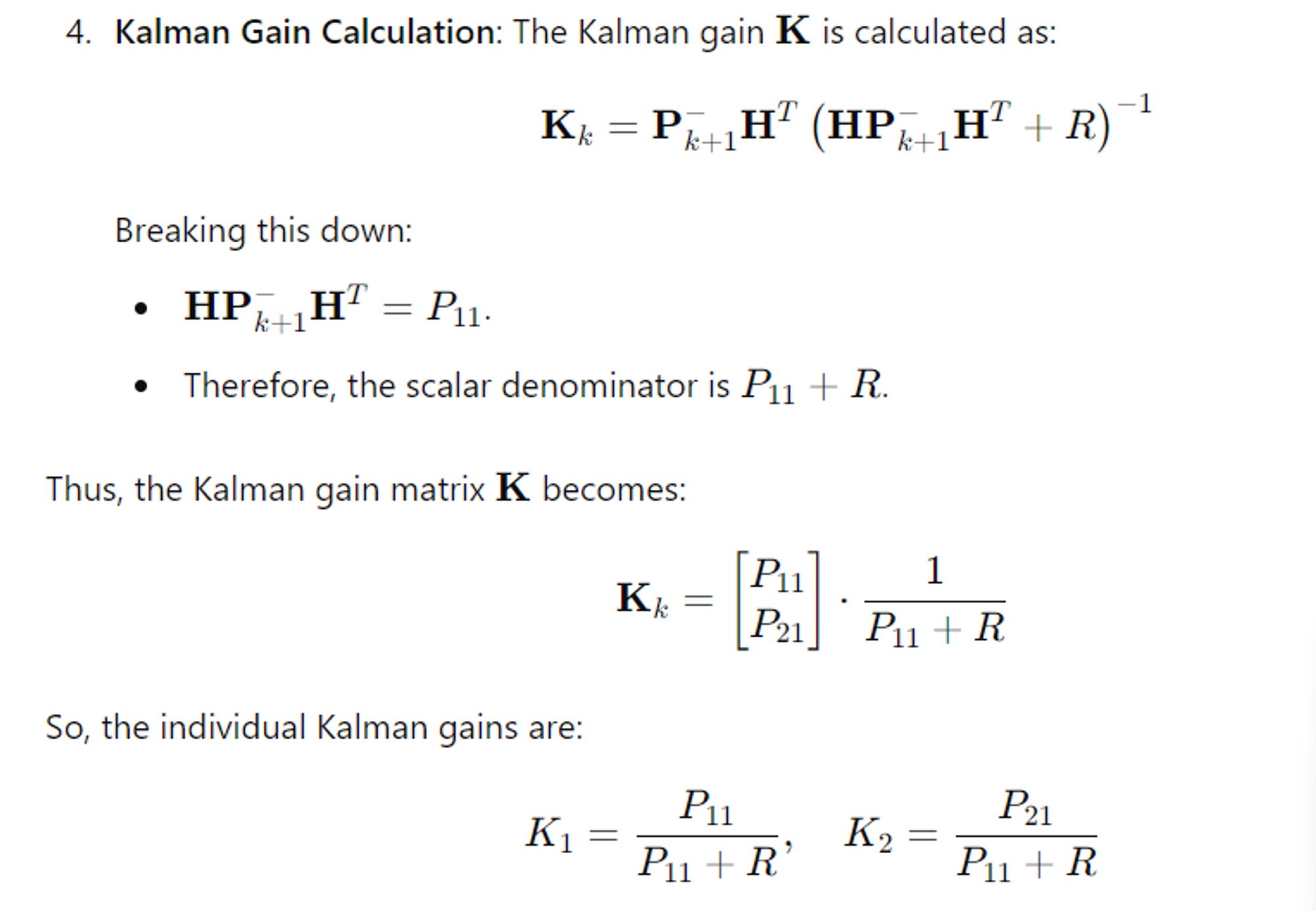
****

****

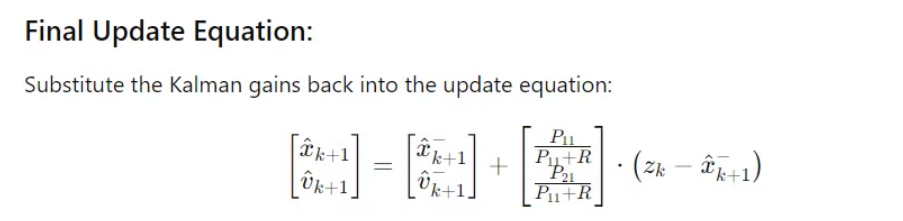
Nếu ma trận H là [1 0], thì phép đo **zk** chủ yếu dự vào dữ liệu vị trí (position) đo được và bỏ qua dữ liệu vận tốc đo được.

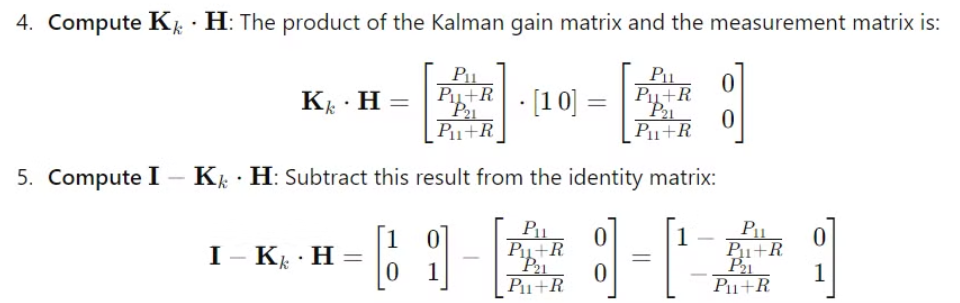
**NOTE**:

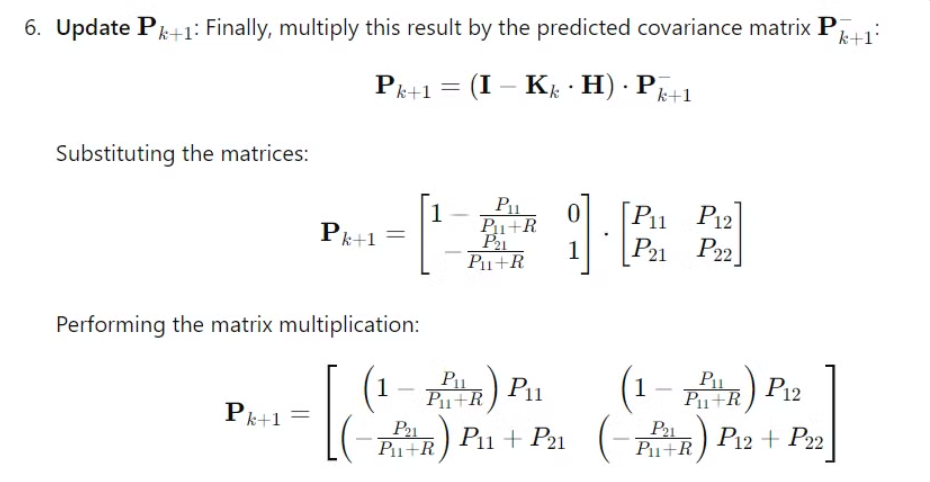
* Nếu ma trận R → 0 thì K → 1, từ đó khâu cập nhật của bộ lọc sẽ chủ yếu dựa vào dữ liệu thực đo được.
* Nếu ma trận R càng lớn thì K → 0, từ đó khâu cập nhật của bộ lọc sẽ chủ yếu dựa vào giá trị dự đoán của bộ lọc.
* Ma trận Q được dùng để thêm vào giá trị sai số (nhiễu của quá trình dự đoán) để tránh ma trận phương sai của giá trị dự đoán bằng 0 (nếu P = 0, dữ liệu từ dữ liệu thực hầu như bị bỏ qua, dẫn đến bộ lọc không còn hiệu quả).

****

### Updation:

****

****

****