

Bài viết hay mình lấy từ trang đam mê điện tử, vì trước mình gặp trường hợp này nhưng không biết gỡ như thế nào. Khi ta sử dụng cảm biến, giá trị trả về từ chúng luôn thay đổi quanh vị trí cân bằng dù là rất nhỏ, và bạn biết nguyên nhân của hiện tượng này là do nhiễu, bạn luôn muốn loại bỏ nhiễu nhưng việc đó dường như ngoài tầm với của bạn. Đừng lo, mình đã có giải pháp, Bạn chỉ cần đọc bài viết này!

## 1. Nhiễu

Trong đo lường, do các yếu tố cả chủ quan lẫn khách quan mà kết quả đo đạc luôn chỉ được coi là tương đối so với giá trị thực cần đo. Độ chênh lệch giữa giá trị đo được với giá trị thực được gọi là Sai số.

Sai số gây ra bởi nhiễu, chúng được phân làm 2 loại chính: sai số hệ thống hoặc sai số ngẫu nhiên.

Giả sử ta tiến hành đo giá trị điện áp của một cục pin năng lượng mặt trời, kết quả thu được trên đồ thị như sau :

Giả sử giá trị điện áp thực của pin mặt trời là  $u_0 = 1.9545 \text{ vol}$ .

Do ảnh hưởng bởi nhiễu yếu tố trong quá trình đo đạc, sai số xuất hiện là điều rất khó tránh khỏi, kết quả là ta chỉ có thể đo giá trị điện áp bằng cách lấy tương đối kết quả trung bình các phép đo .

Tức  $u_0 = u \pm \Delta e = 1.95 \pm 1\%$ , (với  $\Delta e$  là sai số)

## 2. Loại bỏ nhiễu bằng thuật toán lọc Kalman

Phương pháp này được đề xuất năm 1960 bởi nhà khoa học có tên Kalman.

Để chứng minh hiệu quả của phương pháp này mình sẽ có một phép thử mô phỏng như sau.

```
float u0 = 100.0; // giá trị thực (không đổi)
float e; // nhiễu
float u; // giá trị đo được (có thêm nhiễu)
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  randomSeed(millis());
  e = (float)random(-100, 100);
  u = u0 + e;
  Serial.println(u);
}
```

```
float u0 = 100.0; // giá trị thực (không đổi)
float e; // nhiễu
float u; // giá trị đo được (có thêm nhiễu)
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  randomSeed(millis());
  e = (float)random(-100, 100);
  u = u0 + e;
  Serial.println(u);
}
```

Nạp code cho arduino rồi sau đó mở cổng Serial plotter để xem dưới dạng đồ thị:

Kết quả hiển thị trên Serial plotter:

Xem lại code bên trên ta thấy có vài điều như sau:

Gọi  $u_0=100.0$  là giá trị thực tế của vật thể, cũng là giá trị mà ta mong muốn thu được, vì  $u_0$  là hằng số, (nếu như không có nhiễu). Lý tưởng thì trên đồ thị ta sẽ thu được một đường thẳng song song với trục thời gian  $t$ .

Thường thì nhiễu chỉ dao động trong khoảng  $e=\pm 10\%$  giá trị thực đã được coi là rất ổn rồi (noise).

Để tăng độ khó, mình đã cố ý cho  $e=\pm 100\%$   $u_0$  bằng hàm Random khiến cho giá trị đo bị nhiễu hoàn toàn và gần như rất khó để thu thập lần tính toán sau này.

Sử dụng bộ lọc Kalman

Như đã thống nhất, trong thực tế  $u_0$  là giá trị mình không biết, việc sử dụng bộ lọc sẽ phải giúp ta loại bỏ các nhiễu, khi đó giá trị đo được phải gần đường  $u_0=100$  hơn .

Vì đây là mô phư nên giá trị  $u_0$  cần được cho trước (chỉ mình và bạn biết) để có thể kiểm chứng tính đúng đắn của kết quả trước và sau khi lọc. (bằng cách trộn  $u_0$  với nhiễu rồi cho arduino lọc)

So với code bên trên ,phần code này chỉ cần thêm một dòng lệnh duy nhất:

Gọi  $u\_kalman$  là giá trị đo đã qua bộ lọc Kalman:

```
u_kalman=bo_loc.updateEstimate(u);
```

Code:

```
#include
SimpleKalmanFilter bo_loc(2, 2, 0.001);

float u0 = 100.0; // giá trị thực (không đổi)
float e; // nhiễu
float u; // giá trị đo được (có thêm nhiễu)
float u_kalman; // giá được lọc nhiễu
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  randomSeed(millis());
  e = (float)random(-100, 100);
  u = u0 + e;
  Serial.print(u);
  Serial.print(",");
  u_kalman = bo_loc.updateEstimate(u);
  Serial.print(u_kalman);
  Serial.println();
}
```

```
#include
SimpleKalmanFilter bo_loc(2, 2, 0.001);

float u0 = 100.0; // giá trị thực (không đổi)
float e; // nhiễu
float u; // giá trị đo được (có thêm nhiễu)
float u_kalman; // giá được lọc nhiễu
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  randomSeed(millis());
  e = (float)random(-100, 100);
  u = u0 + e;
  Serial.print(u);
  Serial.print(",");
  u_kalman = bo_loc.updateEstimate(u);
  Serial.print(u_kalman);
  Serial.println();
}
```

Và đây là kết quả khi sử dụng thêm bộ lọc:

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

Đường màu xanh: u.

Đường màu vàng: u\_kalman.

Dừng lại một chút để quan sát đồ thị, hẳn bạn cũng đồng ý với mình thuật toán lọc Kalman tỏ ra rất hiệu quả, có những lúc nhiễu dồn ra biên cực đại ( $\pm 100\%u_0$ ). nhưng giá trị vẫn khá sát đường  $u_0$ .

Ghép tầng các bộ lọc ta thu được kết quả chính xác hơn, tất nhiên nó sẽ đáp ứng trễ hơn 1 tầng

```
#include
```

```
SimpleKalmanFilter bo_loc(2, 2, 0.001);
```

```
float u0 = 100.0; // giá trị thực (không đổi)
```

```
float e; // nhiễu
```

```
float u; // giá trị đo được (có thêm nhiễu)
```

```
float u_kalman; // giá được lọc nhiễu
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
randomSeed(millis());
```

```
e = (float)random(-100, 100);
```

```
u = u0 + e;
```

```
Serial.print(u);
```

```
Serial.print(",");
```

```
u_kalman = bo_loc.updateEstimate(u); // tầng 1
```

```
u_kalman = bo_loc.updateEstimate(u_kalman); // tầng 2
```

```
u_kalman = bo_loc.updateEstimate(u_kalman); // tầng 3
```

```
u_kalman = bo_loc.updateEstimate(u_kalman); // tầng 4
```

```
Serial.print(u_kalman);
```

```
Serial.println();
```

```
}
```

```
#include
```

```
SimpleKalmanFilter bo_loc(2, 2, 0.001);
```

```
float u0 = 100.0; // giá trị thực (không đổi)
```

```
float e; // nhiễu
```

```
float u; // giá trị đo được (có thêm nhiễu)
```

```
float u_kalman; // giá được lọc nhiễu
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
}
```

```
void loop()
{
  randomSeed(millis());
  e = (float)random(-100, 100);
  u = u0 + e;
  Serial.print(u);
  Serial.print(",");
  u_kalman = bo_loc.updateEstimate(u); // tầng 1
  u_kalman = bo_loc.updateEstimate(u_kalman); // tầng 2
  u_kalman = bo_loc.updateEstimate(u_kalman); // tầng 3
  u_kalman = bo_loc.updateEstimate(u_kalman); // tầng 4
  Serial.print(u_kalman);
  Serial.println();
}
Kết quả cho 4 tầng lọc
```

### 3. Ma thuật hay trí tuệ nhân tạo

Đây là thành quả nghiên cứu có từ 56 năm trước, chủ nhân của thuật toán là ông Rudolf (Rudy) E. Kálmán

Bộ lọc này được sử dụng rộng rãi trong các máy tính kỹ thuật số của các hệ thống điều khiển, Hệ thống định vị, Hệ thống điện tử, RADA, vệ tinh dẫn đường để lọc nhiễu.

Bộ lọc hoạt động ổn định đến mức, nó đã được sử dụng trong chương trình Apollo, tàu con thoi của NASA, tàu ngầm Hải quân và xe tự hành không gian không người lái và vũ khí, tên lửa hành trình.

Với thành công đó, ông cũng đã thấy rất nhiều giải thưởng lớn khác nhau.

<http://vi.wikipedia.org/wiki/Rudolf...>

### 4. Thuật toán Kalman trong C

Đây là thuật toán đơn giản mình tìm được, hãy vào địa chỉ bên dưới để hiểu được ý nghĩa của các tham số K (Kalman Gain) ,P,Q,R....

<http://malcolmmielle.wordpress.com/...>

```
C++
/*
 * Need to tweak value of sensor and process noise
 * arguments :
 * process noise covariance
 * measurement noise covariance
 * estimation error covariance */
```

```
class Kalman_Filter_Distance {
protected:
double _q; //process noise covariance
```

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

```
double _q_init;
double _r; //measurement noise covariance
double _r_init;
double _x; //value
double _p; //estimation error covariance
double _p_init;
double _k; //kalman gain

public:
/*
* Need to tweak value of sensor and process noise
* arguments :
* process noise covariance
* measurement noise covariance
* estimation error covariance */
Kalman_Filter_Distance(double q, double r, double p)
: _q(q)
, _q_init(q)
, _r(r)
, _r_init(0)
, _x(0)
, _p(p)
, _p_init(p)
, _k(_p / (_p + _r)){};
virtual void init(double x) { _x = x; }
virtual void setProcessNoiseCovariance(double i)
{
    _q = i;
    _q_init = i;
}
virtual void setMeasurementNoiseCovariance(double i)
{
    _r = i;
    _r_init = i;
}
virtual void setEstimationErrorCovariance(double i)
{
    _p = i;
    _p_init = i;
}
virtual double kalmanUpdate(double measurement);
virtual void reset()
{
    _q = _q_init;
    _r = _r_init;
    _p = _p_init;
```

```
};  
};
```

```
double Kalman_Filter_Distance::kalmanUpdate(double measurement)  
{  
    //prediction update  
    //omit _x = _x  
    _p = _p + _q;  
  
    //measurement update  
    _k = _p / (_p + _r);  
    _x = _x + _k * (measurement - _x);  
    _p = (1 - _k) * _p;  
  
    return _x;  
}
```

```
/*  
 * Need to tweak value of sensor and process noise  
 * arguments :  
 * process noise covariance  
 * measurement noise covariance  
 * estimation error covariance */  
  
class Kalman_Filter_Distance {  
protected:  
    double _q; //process noise covariance  
    double _q_init;  
    double _r; //measurement noise covariance  
    double _r_init;  
    double _x; //value  
    double _p; //estimation error covariance  
    double _p_init;  
    double _k; //kalman gain  
  
public:  
    /*  
    * Need to tweak value of sensor and process noise  
    * arguments :  
    * process noise covariance  
    * measurement noise covariance  
    * estimation error covariance */  
    Kalman_Filter_Distance(double q, double r, double p)  
    : _q(q)  
    , _q_init(q)
```

```
, _r(r)
, _r_init(0)
, _x(0)
, _p(p)
, _p_init(p)
, _k(_p / (_p + _r)){};
virtual void init(double x) { _x = x; }
virtual void setProcessNoiseCovariance(double i)
{
    _q = i;
    _q_init = i;
}
virtual void setMeasurementNoiseCovariance(double i)
{
    _r = i;
    _r_init = i;
}
virtual void setEstimationErrorCovariance(double i)
{
    _p = i;
    _p_init = i;
}
virtual double kalmanUpdate(double measurement);
virtual void reset()
{
    _q = _q_init;
    _r = _r_init;
    _p = _p_init;
};
};

double Kalman_Filter_Distance::kalmanUpdate(double measurement)
{
    //prediction update
    //omit _x = _x
    _p = _p + _q;

    //measurement update
    _k = _p / (_p + _r);
    _x = _x + _k * (measurement - _x);
    _p = (1 - _k) * _p;
```

TRUNG TÂM SỬA CHỮA ĐIỆN TỬ QUẢNG BÌNH

**MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình**



GIÁ RẺ

NHANH CHÓNG

LINH KIẾN CHÍNH HÃNG


 SANYO ELEC MSUNG  
 Panasonic TOSHIBA BISHI

## TRUNG TÂM SỬA CHỮA ĐIỆN TỬ XÔ NGUYỄN

- Dịch vụ sửa chữa điện tử tại nhà
- Cung cấp linh kiện điện tử
- Tư vấn lắp đặt nhà thông minh

Đc: Quảng Thuận, tx Ba Đồn,  
 tỉnh Quảng Bình - 0901.679.359

```
return _x;
}
```

Nguồn: <http://dammedientu.vn/bo-loc-kalman-giai-phap-chong-nhieu-khi-su-dung-cam-bien/>

### Các bài viết tương tự:

- [biến trở và Inverter - a chị em xin cho hỏi biến trở và Inverter hoạt động như thế nào a Inverter làm tăng giảm động cơ \(động cơ vd như máy bơm động cơ điện\) còn biến trở có thể tăng giảm động cơ như Inverter hok](#)
- [cân giúp đỡ âm ly 8 sò 2 ngày vẫn chưa tìm ra bệnh\\_áp đối xứng +-17vol qua 2 ỏn áp 7912 7812 cấp cho rơ le mạch music master mic,,+-52 cho công suất - ban đầu hỏng công suất chết câu chì,,thay thế và kiểm tra các điện áp chân b công suất =nhau 52 vol,các tầng khuếch đại thúc, đệm, trở tụ tốt,\(bo nguồn ,ổn áp và công suất đi liền\),,,tháo đường 52 vol thì rơ le lại đóng cấp vào lại ko đóng ,bỏ 1 câu chì 1 về lại đóng\(về đã bị nổ câu chì lúc đầu\),,,kiểm tra ko thấy bị sao? 2 trở cân bằng về rơ le bảo vệ loa em đo 1 đường về 52vol còn 1 đường vài mili vol,,ko hiểu là sao lại chênh lệch thế,,](#)
- [đầu dvd tàu calindax - chạy đĩa hát dc 1 bài tự động reset load lại đĩa hiện no disc lúc mất hiển thị led lại load lại chạy,hát dc 1 bài lại bị đã vệ sinh vì các đường truyền vẫn báo đèn nguồn xanh keyy đĩa](#)
- [em mới mua ve chai 4 vĩ crt 5 vĩ bếp từ 1 nguồn ATX mới và được em vang sài gòn chạy \(BL 0306 ,,50k\) về sửa,,test thử karaoke - chỉnh vẫn chưa chuẩn lắm,,](#)
- [máy giặt sharp 75EV - máy không bấm được nút nguồn . mình đã thay thạch anh 16M , nhưng vẫn không được còn máy con tụ chưa thay được vì nó là tụ dán mình không biết trị số nó là bao nhiêu để thay .](#)
- [Mấy hôm nay làm có 2 hiện tượng thấy lạ như ma ám.hj. 1là tivr tq, nên đo lè nỏ đường hôj, đo đường kR =10v. Tháo vĩ đèn ra đo cũng 10v. Sau đó rút con 4282 trên đg kr ra đo có 150v trên kr, sau đó lắp lại máy đã chạy bình thườg ko pjt bị j lun hehe. 2. Máy trung quốc chj? Bị lỏng mạch nhug khj đo H thấy 22v. Nhug vẫn chạy pjh thuog lạ thât. - .](#)

7. [Nokia 6300 - Tự nhấn phím số 0 làm liệt các phím trừ các phím số từ 1 đến 9, \\* và #.](#)
8. [Sam sung cs 21z45ml - Khởi động nguồn cho chạy , rít cao áp , nóng sò ngang . E đã kt các tụ và diot xung quanh sò , cũng đã thay thử cao áp và sò , nhưng vẫn vậy .](#)
9. [tivi BTV. mất model - bị cao áp đánh vào R\(220k\) đường ABL, đang sáng thì được 15s thì tối dần và bây giờ đang bị tối màn như giảm độ sáng của mà hình, đã thay cao áp và R\(220k\) mà màn hình vẫn tối...](#)
10. [Tivi LG model 21FU6LR - Chạy ic màn hình STV 9326, nửa màn hình dưới bình thường, trên giữa màn hình có vệt sáng hơn và hình bị gấp, phía trên thì hình bị dẫn, kiểm tra nguồn 26v đủ, đường ra chân số 5 cao 22v, thay ic màn hình và các tụ hóa nhưng vẫn chưa ra bệnh](#)
11. [Tủ đông DARLING 210lít - Mỗi lần xả tủ là bị nghẹt k làm lạnh đc,mình xa gas rút chân không nạp gas lại thì chạy đc,nhưng khi tủ bám tuyết nhiều xả đá xong thì block vẫn chạy nhưng k làm lạnh đc,kt đông hô gas thì báo dưới 0psi,mong ae chỉ cách trị pan này](#)
12. [Tủ quây SANAKY - chiếc tủ quây SANAKY cắm điện chạy bình thường nhưng ở phía dưới đáy tủ không lạnh , không làm đá đc ,ở trên mặt kính tủ bị đóng tuyết gân tấm kính](#)