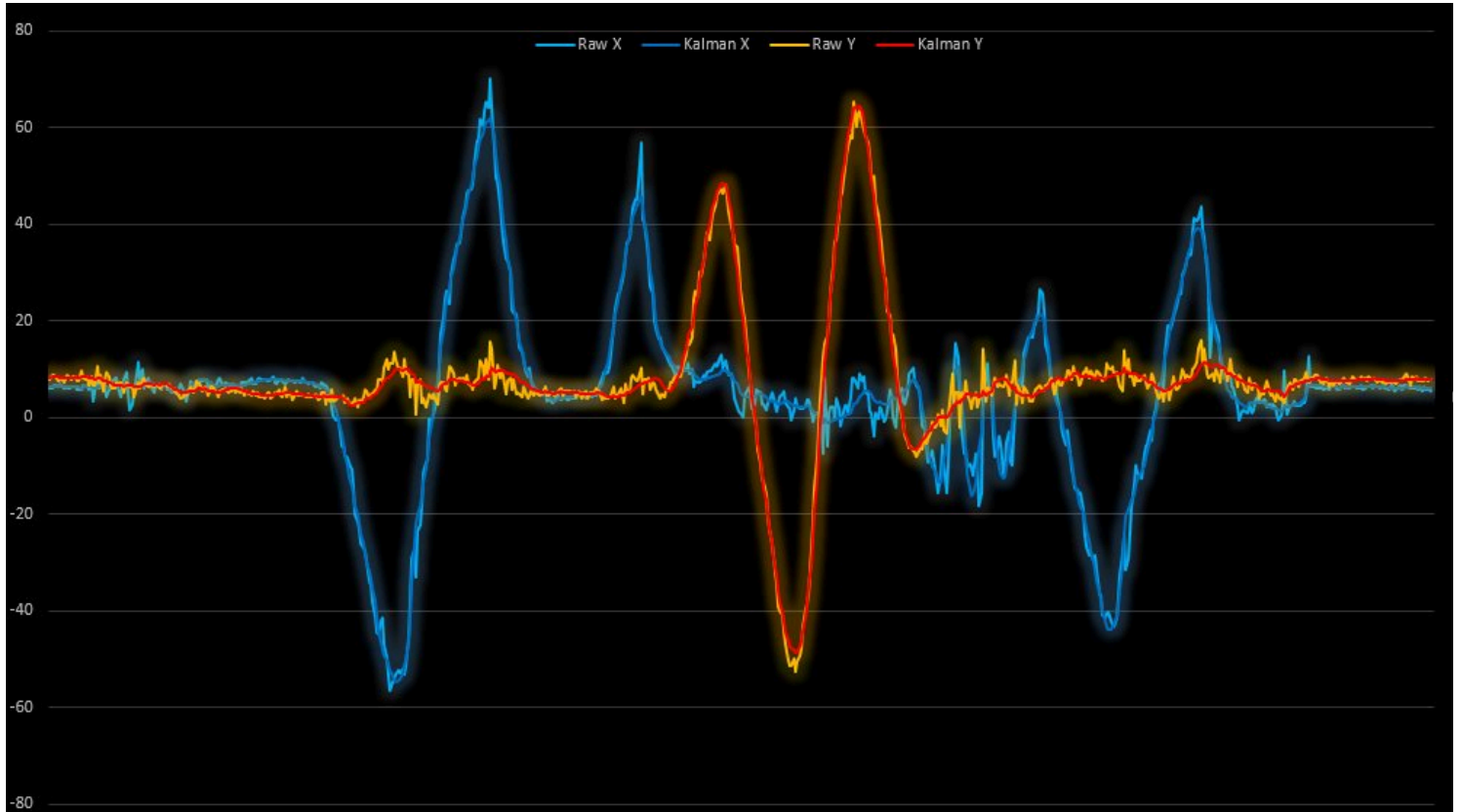


ozzmaker.com



BerryIMU

## Guide to interfacing a Gyro and Accelerometer with a Raspberry Pi – Kalman Filter

| 16 Comments

In this guide we will go over some very basics on the use of a Kalman filter for sensor fusion. There is some very complex math involved which is well over my head, however we do have some working code and very good reference sites.

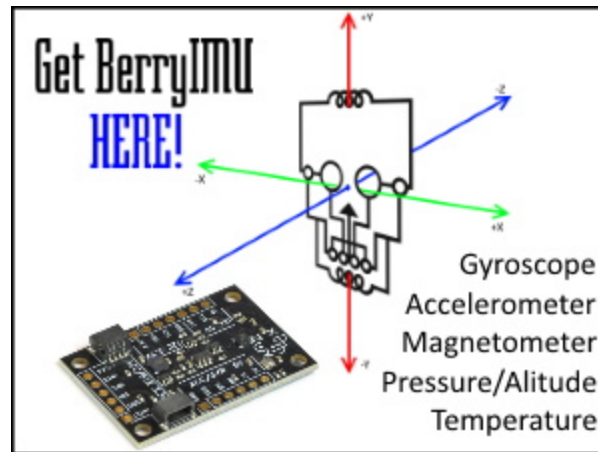
A prerequisite for this guide is to have a gyro and accelerometer from an IMU already up and running on your Raspberry Pi. A guide to interfacing an IMU with a Raspberry Pi can be found [here](#).

Git repository [here](#)

The code can be pulled down to your Raspberry Pi with;

```
pi@raspberrypi ~ $ git clone https://github.com/ozzmaker/BerryIMU.git
```

The code for this guide can be found under the [gyro\\_accelerometer\\_tutorial03\\_kalman\\_filter](#) directory.



## Kalman Filter

The Kalman filter, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The filter is named for Rudolf (Rudy) E. Kálmán, one of the primary developers of its theory.

The Kalman filter has numerous applications in technology. A common application is for guidance, navigation and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometric.

*FYI: A Kalman filter was used to assist with navigation in the Apollo 11 moon landing.*

If we had to explain it in one sentence what a Kalman filter is: “It’s a method of predicting the future state of a system based on the previous ones.”

## References

If you want to understand the ins and outs of the Kalman filter, then TKJ Electronics [post](#) is a must read. I think I must have read through it 15 times.

These three other links are also worth a read;

<http://bilgin.esme.org/BitsBytes/KalmanFilterforDummies.aspx>

<http://tom.pycke.be/mav/71/kalman-filtering-of-imu-data>

<http://credentiality2.blogspot.com.au/2010/08/simple-kalman-filter-example.html>

## The Code

Once you have a working Kalman filter function, it is very easy to slot into existing code.

Below is our Kalman filter for the Y axis, I have another one for X axis;

```

1  float kalmanFilterY(float accAngle, float gyroRate)
2  {
3      float y, S;
4      float K_0, K_1;
5      KFangleY += DT * (gyroRate - y_bias);
6      YP_00 += - DT * (YP_10 + YP_01) + Q_angle * DT;
7      YP_01 += - DT * YP_11;
8      YP_10 += - DT * YP_11;
9      YP_11 += + Q_gyro * DT;
10     y = accAngle - KFangleY;
11     S = YP_00 + R_angle;
12     K_0 = YP_00 / S;
13     K_1 = YP_10 / S;
14     KFangleY += K_0 * y;
15     y_bias += K_1 * y;
16     YP_00 -= K_0 * YP_00;
17     YP_01 -= K_0 * YP_01;
18     YP_10 -= K_1 * YP_00;
19     YP_11 -= K_1 * YP_01;
20     return KFangleY;
21 }
```

We will also need to declare our global variables

```

1  float Q_angle = 0.01;
2  float Q_gyro = 0.0003;
3  float R_angle = 0.01;
4  float x_bias = 0;
5  float y_bias = 0;
```

```
6 | float XP_00 = 0, XP_01 = 0, XP_10 = 0, XP_11 = 0;  
7 | float YP_00 = 0, YP_01 = 0, YP_10 = 0, YP_11 = 0;  
8 | float KFAngleX = 0.0;  
9 | float KFAngleY = 0.0;
```

You then call the Kalman filter functions with the current accelerometer angles and the current gyro rotation rate

```
1 | float kalmanX = kalmanFilterX(AccXangle, rate_gyr_x);  
2 | float kalmanY = kalmanFilterY(AccYangle, rate_gyr_y);
```

The code in the repository will print out both the complimentary and Kalman filter values.

If you have any recommendations or improvements, please let us know.

[wp\_ad\_camp\_3]

## Other Guides and Tutorials

- [Guide to interfacing a Gyro and Accelerometer with a Raspberry Pi](#)
- [Guide to interfacing a Gyro and Accelerometer with a Raspberry Pi – Kalman Filter](#)
- [Create a Digital Compass with the Raspberry Pi – Part 1 – “The Basics”](#)
- [Create a Digital Compass with the Raspberry Pi – Part 2 – “Tilt Compensation”](#)
- [Create a Digital Compass with the Raspberry Pi – Part 3 – “Calibration”](#)
- [Create a Digital Compass with the Raspberry Pi – Part 4- “Smartphone Replica”](#)
- [Converting values from an Accelerometer to Gs – “ Proper Acceleration”](#)
- [Using the BerryIMUv3 on a Raspberry Pi Pico](#)
- [Double tap detection with BerryIMUv3](#)
- [How to Create an Inclinator using a Raspberry Pi and an IMU](#)
- [Raspberry Pi Digital Spirit Level](#)



## 16 THOUGHTS ON “GUIDE TO INTERFACING A GYRO AND ACCELEROMETER WITH A RASPBERRY PI – KALMAN FILTER”

**skyguy**

JANUARY 1, 2016 AT 6:50 AM

hi,

your IMU work great! Could you write the same kalman function for python?

---

**CARLOS GOMEZ**

MAY 15, 2016 AT 12:15 AM

Hi Mark...

Please, can you explain to me why when i run the BerryIMU.py the GYRY and GRYZ values not stop to increase.

I am PILOT, and try to create a Raspberry Pi-EFIS is like “Dynon D2” —>

<https://www.youtube.com/watch?v=vj7mjKI2GHk>

Thanks you and I appreciate your help.

CARLOS F. GOMEZ

From: Bogota, Colombia

---

★ **Mark Williams**

MAY 15, 2016 AT 10:18 PM

That is normal. It is from the noise which all gyros have. This is why the angles from an accelerometer and gyro need to be fused.

*Gyros – A gyro measures the rate of rotation, which has to be tracked over time to calculate the current angle. This tracking causes the gyro to drift. However, gyros are good at measuring quick sharp movements.*

More info here;<https://ozzmaker.com/berryimu/>

---

**CARLOS**

MAY 23, 2016 AT 6:01 AM

Hi Mark

In your code is not calculated YAW value. please can you show us how we can to calculate a YAW value?

Thanks

CARLOS F. GOMEZ

---

**soheil**

SEPTEMBER 15, 2016 AT 4:10 PM

hello mr williams , i have a basic question , its kind of you if help me, as we know, we can have angle from gyro but with drift and we have angle from accelerometer but its not usefull because static and dinamic acc are combined. in overal i want to know that can we have angle by combining gyro and acc that was very fast and removed dinamic acc from that? i tried this code and also complimentary code but in both of them is static acc.

---

**Yasir**

NOVEMBER 15, 2016 AT 5:35 AM

I was also wondering about the yaw?

Also is there a way to actually compute the Zangle or rotation?

---

★ **Mark Williams**

NOVEMBER 15, 2016 AT 5:02 PM

For yaw, you can use the heading from the compass.

You can fuse the compass heading and gyro Z values. Just like you would fuse the accel X and gyro X values

---

**N**

MARCH 16, 2017 AT 12:22 AM

Hi Mark,

My heading seems to vary quite a bit so I was just fusing it with the gyroZ value but my understanding of kalman filters is pretty crap, so I was wondering if you could tell me how these values were calculated (and would they change to fuse the heading)?:

$Q_{\text{angle}} = 0.02$

Q\_gyro = 0.0015

R\_angle = 0.005

Thank you =)

---

**Francesco**

NOVEMBER 18, 2017 AT 10:12 PM

To my understanding, those values are just tuning parameters you need to select . As a matter of fact, in Kalman filtering theory they have a precise meaning and are related to the statistical properties of the noise you are dealing with and may also be time-varying! However, for this type of applications it is quite unlikely to be able to get closed-form expressions for R and Q so the best you can do is to consider those as parameters. Hopefully that replies to your question.

---

**Michael**

JUNE 11, 2018 AT 1:31 AM

Hello Mr. Williams,

How would we merge this filter code with a Angular Acceleration tracking code? I am using a Raspberry Pi 3 with a SenseHAT (IMU), my objective is to track angular acceleration on a steering wheel bearing to track spikes in the wheels movements but ignore regular turns or movements.

Thanks,  
MC

---

**john**

APRIL 4, 2019 AT 12:01 AM

Hey mark or anyone in the comments who have worked on this, Can you guys tell me how to calculate AccZangle? What formula should i use? Plus how to add AccZangle in the rotation value "If Else" condition after that in the code.

Thanks

---

**John marshall**

APRIL 4, 2019 AT 1:04 AM

Hi mark,

If i want to introduce KalmanFilterZ do i need to extend the matrix aswell like for example  
float XP\_00 = 0, XP\_01 = 0, XP\_10 = 0, XP\_11 = 0;

```
float YP_00 = 0, YP_01 = 0, YP_10 = 0, YP_11 = 0;
```

do i need to extend the current XP\_00 with XP\_000? because if i am introducing float ZP, do i need to add extra digit or can follow the same as with two digits ZP\_00 = 0 ?

---

**Peter**

APRIL 30, 2019 AT 5:45 AM

hi,

How fast i can read from your gyroscope and accelerometer module?

I have MPU6050 and every information take 5 milisecond.

I need faster.

---

**★ Mark Williams**

APRIL 30, 2019 AT 10:54 AM

The gyroscope and accelerometer on the BerryIMUv2 can do 952 times a second, just above 1 millisecond.

You need to make sure your MCU can read this fast. The arduino can.

---

**Alfan**

JUNE 5, 2019 AT 9:39 PM

hi William, how do we know the AccZangle and the kalman output for Z axis?

---

**Giuseppe Porciani**

AUGUST 29, 2022 AT 3:01 AM

Hi,

I would be interested in your BerryGPS-IMU board. I know almost nothing of programming. I would need gps position accuracy of about 1m. I do not know if that is possible using EKF for IMU-GPS data. If that is the case I would need a guide to use EKF for the IMU-GPS data of the above board.

Thank you in advance for any help.

---

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)