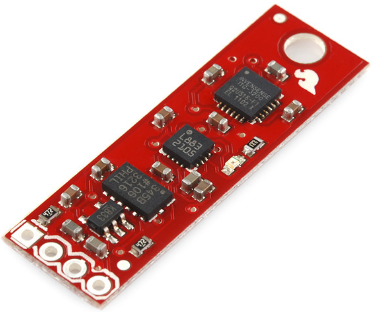
ARDUINO, QUADRIROTOR

THE QUADCOPTER : HOW TO COMPUTE THE PITCH, ROLL AND YAW

SEPTEMBER 19, 2012 | ALEX | 25 COMMENTS

After having introduced here the basics of an aircraft orientation and how to control it, this article is about actually computing the orientation of the quadcopter in space with sensors and with the Arduino.



The stick IMU from sparkfun: ADXL345 accelerometer, ITG3200 gyroscope, HMC5883L compass.

Which sensors to use?

3 axis accelerometer + compass for yaw

To get easily the orientation of a non-moving object (pitch and roll), a 3-axis accelerometer (how does it work?) can be used. For a static object, it gives the value of the gravity field on 3 axes, therefore its direction. And since it always points to the center of the earth, we can therefore know how the accelerometer is inclined with the help of man's best friend, trigonometry.

This method has been used in a lot of smartphones and gives pretty accurate results, if you are not moving. Indeed, if you start translating the phone in space, you are creating a force on it, therefore you are changing its acceleration. The assumption we made previously to compute the orientation is not valid anymore, so the calculated orientation won't be accurate.

Another problem of accelerometers is their sensitivity to vibrations.(it is actually not a "problem" per say, it is just a consequence of the forces created by vibrations, which are essentially shocks). If you want a stable quadcopter, you absolutely need a smooth angle! I think it is by far the most important thing about a quadcopter. If you're stuck with your PID tuning and that all the settings you try are not efficient, it probably means that your angles are not smooth/ accurate enough.

As you can see, using only an accelerometer is not a valid option. You can of course try to filter the signal and reduce the amount of vibrations received by the sensor(using some foam or anything to free the accelerometer from the motors vibrations can really change a lot the output of the sensor) but it will never be precise/ fast enough to satisfy the needs of this application.

Gyroscope

The most common used sensor in quadcopter control boards is the gyroscope sensor. It gives the angular rate around the 3 axes of space in deg/s, so, as for the accelerometer, some simple maths are needed to compute the actual angle by integration. But using only a gyroscope raises problems:

- The first problem raised is caused by the nature of the sensor. It just gives an angular rate, not an absolute measure. So if you start up the quadcopter on a crooked floor, the initials pitch/ roll angles shouldn't be 0, but they will be in your program since the gyroscope will just output null angular rates!
- All the common MEMS gyroscopes used with Arduinos have a drift. It means that even if you stay steady and don't move, the sensor will output values different than zero. The drift can be pretty big for some sensors (it can go up 2 deg/s on the z axis of our L3G4200d !), therefore ruining the accuracy of the measurement when you integrate the values ! But it's not as bad as it could sound, most of the gyro drift can be subtracted from the measurement since it's a constant value (given a certain temperature).

So, both accelerometers and gyros are bad? The answer is no!

There is actually many different ways of getting the most accurate orientation from a combination of sensors and they all have their good and bad sides. The simplest approach we've taken so far for our quadcopter, and which turns out doing pretty good, is a complementary filter.

The idea of the complementary filter is the following : the filtered accelerometer value of the angle is not subject to drift, so we use it to "correct" the value given by the gyroscope(more precise and less subject to vibrations noises). How do we do that? We combine the two values like so:

$$Angle_{accurate} = (GyroPercentage)*Angle_{Gyro} + (1 - GyroPercentage)*Angle_{Accel}$$

GyroPercentage is just a floating value between 0 and 1. It is typically ranged from 0.9 to almost 1, depending on how much you can trust your gyroscope and accelerometer.

The angle given by this very simple method is actually pretty accurate and isn't too much time consuming for the Arduino. So if you don't want to go too deep into the maths, I would suggest you to use this method, it is very satisfactory and easy to understand 😉

On the yaw axis, the same complementary filter is used with a compass (which gives the direction of the earth's magnetic field), giving us the 3D orientation of the quadcopter like so:

$$Yaw_{accurate} = (GyroPercentage)*Yaw_{Gyro} + (1 - GyroPercentage)*Yaw_{Compass}$$

Last, but not least, **filter your signals!!** It is fundamental to reduce the impact of the frame vibrations on the angle computation, if your computed angles are not accurate, your quadcopter will never fly. It is really the most important thing to do. And filtering the signals means both in software and in the conception of the frame. Use strong materials (such as carbon fiber), try to protect your IMU(Inertial Measurement Unit = Accelerometer + gyroscope + compass) from the vibrations. On the software side, if your sensors have a built in low pass filter (like on the ITG3200 and the L3G4200D), activate them by writing in the good registers. If you want to program your own software low pass filter for you sensors, a simple method is to smooth the values given by the sensor like it is done in this pseudo C code:

```
/* This code shows an easy way to smooth readings from a sensor subject to
 high frequency noise.
It uses a low pass filter on a circular buffer.
This circular buffer always contains the last BUFFER_SIZE-1 readings from
the sensor.
The new reading is then added to this buffer, from which wecompute the
mean value by simply dividing the sum of the readings in the buffer by the
number of readings in the buffer.
*/

int indexBuffer;
float circularBuffer[BUFFER_SIZE];
float sensorDataCircularSum;
int BUFFER_SIZE; // Number of samples you want to filter on.
float filteredOutput;
float sensorRawData; // typically the value you read from your sensor
 //in your loop() function

void smoothSensorReadings(){
 // We remove the oldest value from the buffer
 sensorDataCircularSum= sensorDataCircularSum - circularBuffer[indexBuffer];
  // The new input from the sensor is placed in the buffer
 circularBuffer[indexBuffer]=sensorRawData;
// It is also added to the total sum of the last  BUFFER_SIZE readings
// This method avoids to sum all the elements every time this function is ca
 sensorDataCircularSum+=sensorRawData;
// We increment the cursor
 indexBuffer++;

 if (indexBuffer>=BUFFER_SIZE) indexBuffer=0;// We test if we arrived to the
//of the buffer, in which case we start again from index 0
 filteredOutput =(SensorDataCircularSum/BUFFER_SIZE); // The output is the t
//value of the circular buffer.
}
```

What is basically done here is just an average value of the *BUFFER_SIZE* last inputs from the sensor. This is a simple way to get rid of high frequency noise and can also be used to smooth PWM inputs from a RC control, for example 🙂

If you made it until here, I hope you are now able to implement by yourself your own implementation of orientation computing. It's actually a big area of research and much more complex solutions are still developed (I invite you to look at Sebastian Madgwick fusion filter algorithm 🙂).

• ACCELEROMETER • COMPUTE • GYROSCOPE • IMU • PITCH • QUADCOPTER • ROLL • YAW

25 THOUGHTS ON “THE QUADCOPTER : HOW TO COMPUTE THE PITCH, ROLL AND YAW”

Moustafa

JANUARY 28, 2013 AT 9:20 AM

Hey! AWESOME JOB!
You may be the BEST blog online explaining how to combine accelerometer data and gyro data simply to obtain an accurate angle.
I have some questions though, can you elaborate on how to actually GET the GyroAngle and AccelAngle mathematically?

Another thing, after i get these readings, are they reliable enough to be fed into a PID controller? I'm using only unfiltered, raw acceleromter data with a PID controller based on the Arduino PID library on an ArduPilot 2.5 Mega and I'm failing MISERABLY, the thing just keeps on oscillating and never returning to its setpoint which i set to 0's (so that x and y values are read to zeros)

Any help is appreciated!
Thanks for all the help!

★ Alex

JANUARY 28, 2013 AT 6:03 PM

Hi Moustafa !
Thanks for your comment, it's nice to see that some people like our articles 🙂
If you are only using an accelerometer to get the angles, I'd rather stop you now : there is NO way it can work 😀 As I said in the article, an accelerometer can only give you the orientation for a static object (which is not really the point of a quadcopter), so you need a gyroscope. You also should really insist on filtering your sensors data, especially the data from your accelerometer. There are so many vibrations on a quadcopter that instant readings can get highly inaccurate and generate the oscillations you are talking about!

About actually explaining the angle computing, it's right here 😀 I will update the article to redirect towards this one in case people want more maths 😉

Nlanaconda

MARCH 29, 2013 AT 2:22 AM

Hi Guys,

Wonderfull blog, you giving me some hope again.
I'm trying to do the same thing: building a quad copter.

I thought I had it all sorted out, angle calculation, calibration, noise reduction, etc.
But as I was playing I did see (in my inexperienced eyes) something weird.
The pitch (or the roll, 1 of them) didn't have a drgree scale from 360 (-180 to 180) but 180 (-90 to 90).
I was braking my head over it, thinking there are 360 degrees on an axis, I only have 180, I must been doing something wrong.
So I started searching the Internet finding solutions I couldn't understand: quaternions, DCM calculations, you name it.
I was giving up. My quad was never going to fly.

But now I found your blog and I did read all articles and you make it look (relatively) so simple again.
The math here is simple enough for me to understand and to build further on.

But the question I can't find an answer to:
1. Do you also have one axis from -90 to 90 degrees?
2. WHYYYYY is it this way? There are 360 degrees on an axis why can I miss 180 of them?
3. Can I go on and buy the moters and stuff if I have these steady 360 deg X, 180 deg Y and 360 deg Z readings?

(or am I completely wrong and have you guys sneaky been using these weird quaternions or complex filter thingies, and didn't tell about?)

Ben

MARCH 29, 2013 AT 11:46 AM

Hi there,

What you can try is this : first read your raw accelerometer and gyroscope values.

From your accelerometer the raw data is something like proportional to the angle from horizontal. you can find the coef by rotating your drone from right vertical to left vertical. Doing this, remark that you made your drone accomplish a 180° rotation, not a 360° one.

For your gyroscope it's a little more touchy since you have no direct way to find the coef but reading the datasheet that give you the correspondance between the raw data and the actual ° (or rad) per second (or min) . Then by multiplying this value by the time for one loop you find your angle (you integrate the value actually).

I'm not sure if I answered your question but you can ask more (and maybe give here the incriminated part of your code so we can take a look !)

Last thing : don't give up ! It really worth the headeaches 🙂

Wilbert

MARCH 29, 2013 AT 6:08 PM

Thanks for your reply,

Well, I already have the degrees from the Accelero and the Gyroscope with a Complementary Filter.

I calculate the angles this way:

For the Accelerometer (just raw to degrees):

```
X = atan2(rawX, sqrt(rawY*rawY + rawZ*rawZ)) * 180 / PI;  //This spits o
Y = atan2(rawY,rawZ) * 180 / PI;  //This spits out values between -180 to
```

So if the plane makes an acrobatic roll of 360 degrees the sensors will pick this up and wont miss a degree of it.

Imagine the plane makes a looping, I will get the same readings for a pitch of 30 and a pitch of 130.
Instead of giving me a value of 130 it gives me a value of 30 again because the scale is 180 degrees instead of 360.

I've put my best photoshop skills in action and made an image to explain:
pitch180degreesscale.png
In black you see the 360 degrees.
The red arrows point at the degree readings I get.
The red half of the circle is just a mirror of the green.

Ok, in reality the copter wont fly upside down and loopings are not my goals. But I break my head over it, WHYY 😛
Do you guys also have a -90 to 90 scale for the pitch?

I made this movie http://www.youtube.com/watch?v=jmT5RPGf6R4 (Yaw is disabled "0", so dont look at that).
Im ok when Im not crossing the 90 or -90 degrees barrier. But look what happens when I go upside down.

Wilbert

MARCH 30, 2013 AT 12:02 PM

(Crap I used my nick and real name, in the same guy 😉)

★ Alex

MARCH 30, 2013 AT 12:13 PM

Hi Wilbert!
I see your problem, and if you don't want to get into complex maths I suggest you test the signs of your accelerometer's raw data to know if the quadcopter is upside down or not. Then you can correct the angle

in your code by changing the output value in degrees. It's the quick and dirty fix. Our quadcopter was not supposed to make loops, so the angles could go crazy if it turned around 😀
If you want to get serious about angle computing, Eulers angles have some limitations that make them inappropriate for such a use. In complex VR applications, quaternions and rotation matrices are always used. It's much less intuitive, but it works better 😀 I recommend you take a look at Sebastian Madgwick fusion filter algorithm . He provides a C implementation of his algorithm in the PDF he wrote. It's so far the best algorithm for orientation computing(even better than Kalman filters, another famous algorithm). This algorithm is the one used in the FreeIMU library for Arduino, You should take a look at it! It's not that hard to implement. To understand it is another thing, though 😀

Cheers 🙂

Wilbert

MARCH 30, 2013 AT 2:43 PM

Thanks Alex,

Im not intending to make my quad do loops and other acrobatic stuff. I only want to get it airborn. More or less the same as your quad. 🙂

I thought I did something fundamentally wrong because I get these readings.
By reading your reply I conclude that it's not a problem. I wont fly upside down or do crazy things 🙂

Maybe later I will look at algorythms and stuff 🙂

Thanks and thanks again, I go to the next step 🙂 !

Jannek

APRIL 27, 2013 AT 1:45 AM

Hi, very good explanation.

One question though:

Instead of integrating the compass reading in the complementary filter for the yaw angle, can't i just use the yaw from acc like you did for the other angles ?

THX!

★ Alex

APRIL 30, 2013 AT 11:58 AM

Hi Jannek,
The thing is that you can't get the yaw from the accelerometer only. Let's say the accelerometer is on a flat

surface on a table, not moving. The readings on the X,Y,Z axis of the acc should be 0,0,-1 (or 1 depending on where the Z axis is heading). Now, if you turn your acc on this flat surface (so around the Z axis), the readings will still be exactly the same. So you can't really count on it for measuring the yaw. If you don't have a compass, it actually doesn't matter if you don't need to know if your quadcopter is heading north, south etc... Indeed, the yaw angle is pretty much useless when it comes to regular piloting of the drone. You want to control the yaw rate with your yaw stick, not its absolute value. And the gyroscope gives you directly this rate! 🙂

Hope I made myself clear 😀

Chetan Kumbhar

JULY 9, 2013 AT 9:24 AM

hey plzzz help me ....i have got the readings from the Gyroscope and Accelerometer. I have displayed it on my LCD for gyroscope i get all the 3 -axis zero value wen stable ,bt as i move the board the values changes n setles down to zero whatever may be the orientation.

Accelerometer : on LCD i am getting values as: 1g,2g or (-1g),(-2g)...dats it bec i hav converted the values..using sensitivity parameter given in the datasheet......

should i convert them or no........plzzz suggest me....or jst use the data which i am getting frm the sensor that is in HEX values format.......
plzzzz do help me...plzzzzz....

★ Alex

JULY 9, 2013 AT 11:34 AM

Hi,
What do you want to do? I am not going to write the code you should write but I can help you if you tell what your goal is 🙂

---

chetan kumbhar

AUGUST 27, 2013 AT 7:02 AM

hey i want to fuse the data of accelerometer and gyro in order to remove the effect of gravity......
plzzz help me asap....

Bruno F. M. Callegaro

JULY 21, 2013 AT 12:12 AM

Hi, i'm having some problems with high frequency noise even after passing them on the filters. I've enabled the built-in filters in the sensors and after that, added a first order low pass filter on both sensors ,and merged them both with the complementary filter and even with all these stuff i sill get very noise readings when my motors reach some velocity. I don't know more what i have to do, someone had the same problem!?

By the way, thank you for the great post you have made , helped me a lot on my building, and sry for my bad grammar.

★ Alex

JULY 21, 2013 AT 11:29 AM

Hi Bruno, and thanks for your comment ! 🙂
What percentage of the acc are you using for your complementary filter? Even when it's filtered, it can get very noisy when the motors are turning. The point of the acc is really just to keep the gyro from drifting, so a percentage as low as 0.003% should do the job, while keeping the noise away 😀

Andy

JULY 21, 2013 AT 1:34 AM

It's a rate gyro. It tells you how fast you are turning not the absolute direction. You have to integrate to get the angle.

Jan R

SEPTEMBER 1, 2013 AT 11:20 PM

Hi,

I am glad to read your sites about quadro 🙂 !!

My question is: How can I get Angle (accurate) ? from your pattern - If I use Gyropercentage = 1, so Angle from Accelerometer will by a zero! and if I use 0.9 so the most significance angle will be from Gyro... Is that correct?

★ Alex

SEPTEMBER 2, 2013 AT 11:24 PM

Yes it is! 🙂
The accelerometer is only used to correct the drift of the Gyro, which is way less sensitive to the frame vibrations 🙂

Jan Ropek

SEPTEMBER 3, 2013 AT 8:44 PM

Thanks for reply!

I suppose that this video is your job: https://www.youtube.com/watch?v=ejml0YumyDg and is in the video used this method retrieve angle?

And how much BUFFER_SIZE did you use for Gyro and how much for Accelerometer (my frequency of reading new data form Gyro and Accelerometer is 100Hz)?

And Can your Quadro run as autonomous mode?

★ **Alex**

SEPTEMBER 3, 2013 AT 10:40 PM

Yes this video was shot with our own flight controller, so everything you can find here was applied !

I honestly don't remember which buffer size was used, and I don't have the code here, but keep in mmind that the more you flatten your signals, the slower it will react to a movement. So you have to find a sweet spot by experimenting I guess 😀

Our quad with our custom flight controller could not fly autonomously, it was fairly simple. We now use an Ardupilot which can do pretty much anything, including autonomous flight 🙂

Jan Ropek

SEPTEMBER 4, 2013 AT 10:52 PM

I am little confused how can i get Angle form Gyro: I have Gyro with 250 DPS So I integrate tha value of Gyro but the result it's not a deegre.

```
Angle+=Gyro_Y*dt; //Where Gyro_Y are filtered data form Y axis GYRO, and dt is periode to calculating, and Angle is result in Deegres,
```

but it doesn't work.. where is problem?

Jan Ropek

SEPTEMBER 7, 2013 AT 8:59 PM

I have to multiple by 250 and divide by 65536/2 🙂

Harshith

OCTOBER 5, 2013 AT 5:41 PM

Hi Alex,
This is really a good place to start with..Congrats..
I'm an amateur in quadrotor topic,

One question(one of the many!!)..
Should we control the horizontal movement of the quadrotor with just yaw,pitch and roll??
i.e..
if quadrotor should move left by 5m,
then roll a little with some throttle(i.e. 5m left with 1m up) ,later reduce the throttle(1m down) to compensate

for the altitude??
or
is it possible with other combination of motor control(say motor1 and motor2 combined,)??

Thanks for reply,

Ali

NOVEMBER 21, 2013 AT 7:30 PM

Great Work! This blog was really helpful...
i am having a really tough time regarding my project on yaw,tilt, pitch sensor.. am wondering if you could help!
Am really running out of time 🙁

Harrison

DECEMBER 21, 2013 AT 6:03 PM

Hey guys, really, really great work.

It is so freaking impressive to see your drones flying so smoothly and I admire the fact that you built all this after deciding to persevere through your initial designs.

My friend and I have been trying to get our quad to fly with an Arduino Mega (ITG-3200, LSM303DLHC for accel/mag) but the angles that we're computing are atrocious. They're beautiful when the quad is still, but once the motors start vibrating the chassis our accelerometer goes bonkers and messes up the pitch and roll values.

We're just trying to get it to hover for now, regardless of yaw. How many points did you need to average for the accelerometer? Did incorporating the magnetometer readings help a lot? I'll look into this magnetometer filter, thank you so much for sharing your work, this blog is me and my partner's last ray of hope!!!