

Questo sito fa uso di cookies.

Cliccando un qualunque link accetti al loro utilizzo.

OK

No

Varesano.net[Home](#)[About](#)[Topics](#)[Projects](#)[Socialize](#)[Contact](#)[Home](#) » [Blogs](#) » [fabio's blog](#)

Simple gravity compensation for 9 DOM IMUs

Submitted by fabio on Wed, 2011-07-06 15:12.

An accelerometer is subject to dynamic (or external) and static (gravity) accelerations (see [my thesis](#), chapter 5.1). This means that whenever you are interested in measuring dynamic accelerations rather than gravity you have to compensate for gravity.

I implemented a very simple approach for doing so, assuming that you have a 9 degrees of measurement inertial measurement unit as my [FreeIMU](#), from which you are able to compute its orientation quaternion representing the orientation of the Earth frame with respect to the Sensor frame and that you have the readings coming from the accelerometer.

This is the code, implemented in Python:

```
# compensate the accelerometer readings from gravity.
# @param q the quaternion representing the orientation of a 9DOM MARG sensor array
# @param acc the readings coming from an accelerometer expressed in g
#
# @return a 3d vector representing dynamic acceleration expressed in g
def gravity_compensate(q, acc):
    g = [0.0, 0.0, 0.0]

    # get expected direction of gravity
    g[0] = 2 * (q[1] * q[3] - q[0] * q[2])
    g[1] = 2 * (q[0] * q[1] + q[2] * q[3])
    g[2] = q[0] * q[0] - q[1] * q[1] - q[2] * q[2] + q[3] * q[3]

    # compensate accelerometer readings with the expected direction of gravity
    return [acc[0] - g[0], acc[1] - g[1], acc[2] - g[2]]
```

The idea behind this is pretty simple. With the quaternion we can compute the expected direction of gravity and then subtract that from the accelerometer readings. The result is the dynamic acceleration.

I'm sure that there could be a better solution but this is just the result of a 10 minutes coding and seems working pretty good.

[Login](#) to post comments 68458 reads

Posted in: [FreeIMU](#)

g[0] is the only one with an accurate reading

Submitted by Anonymous (not verified) on Sat, 2014-07-12 02:23.

I'm working with a 9-axis device. I'm comparing the raw accelerometer readings to see if the algorithm can at least produce the same gravity readings as the accelerometer (when it's not moving). X is the only one that matches but with the sign reversed. The Z axis responds but only when gravity gets diverted to the x axis, not the y (when it's pitched about the x axis).

[Login](#) to post comments

A Chicken Egg Issue - Kalmann and Gravity Componenets

Stickers



Popular content

All time:

- [Installing Flash Player Plugin on Firefox without having Administrator Access or Permissions](#)
- [FreeIMU: an Open Hardware Framework for Orientation and Motion Sensing](#)
- [Disable related videos on a embedded Youtube player](#)
- [h264 videos on Flash: cool example](#)
- [Serial RS232 connections in Python](#)
- [Serial communication with Arduino and Processing: simple examples and an Arduino based gamepad interfacing with Processing](#)
- [My first 6 DOF IMU Sensors Fusion Implementation: ADXL345, ITG3200, Arduino and Processing](#)
- [Multiple forums for Organic Groups: my patch committed](#)
- [An homebrew DIY breakout board for the ITG3200 gyroscope: tested with Arduino and Processing](#)

Submitted by Joy Bhattacharjee (not verified) on Thu, 2014-05-22 09:09.

In my case I have a 6DOF arduino setup with ADXL345Accelerometer and ITG-3200 rate Gyro.

We are getting the quaternions in local frame after using a Kalmann Filter (which uses Acceleromter data to correct gyro data).

My point is, in this case can we again derive gx, gy, gz components from q0, q1, q2, q3 and again subtract them from accelerometer data? Seems like a chicken-egg situation to me.

[Login](#) to post comments

How do you obtain the quaternion

Submitted by Naveen (not verified) on Sun, 2014-03-09 20:32.

Hi,

May I know how do you obtain the quaternion which represents the orientation of a 9dof sensor at any one point in time?

Thanks

[Login](#) to post comments

need clarification for Gravity calc

Submitted by Evan (not verified) on Fri, 2014-01-24 15:30.

Hi Fabio;

the gravity vector can be calculated from quaternions in the way you presented. Could you give me material you used for the calculation?

or the math behind this way.

thank

[Login](#) to post comments

Gravity vector calculation

Submitted by Flipordie (not verified) on Fri, 2014-01-24 09:03.

Good day sir,

I appreciate your work. However, I can't seem to get the math right to check whether the gravity vector can be calculated from quaternions in the way you presented. Could you give me some pointers and material you used for the calculation?

Br,

Flip

[Login](#) to post comments

Compensating Gravity

Submitted by Ankur (not verified) on Thu, 2013-11-07 00:54.

Hi,

i saw your post about the gravity compensation. i am also doing the same thing in my project. but, in a little different way.

i dont have the quartino values. i just have the raw g's (m/s^2) and raw degree/sec

- [Femtoduino: an ultrasmall \(20.7x15.2 mm\) libre Arduino compatible board](#)

Recent comments

- [prom read address 0](#)
5 years 11 weeks ago
- [Sorry if this sounds stupid,](#)
5 years 12 weeks ago
- [Starting with MS5611](#)
5 years 12 weeks ago
- [problems in calibration mag data](#)
5 years 13 weeks ago
- [Code to test IMU \(The cube form\).](#)
5 years 16 weeks ago
- [error while compiling](#)
5 years 18 weeks ago
- [Re: Hi I have read that in](#)
5 years 19 weeks ago
- [wonderful!](#)
5 years 20 weeks ago
- [You may want to test if the](#)
5 years 21 weeks ago
- [Hello, Doing this in the](#)
5 years 21 weeks ago

Advertising

(IMU6050). i want to calculate linear acceleration in all the three directions and dont know how to do it. the process you explained includes the quaternions and the rotation matrices but, i don't have any knowledge about them. could you please suggest me some idea that i can implement.

thank you

[Login](#) to post comments

Are you sure it works?

Submitted by Roberto (not verified) on Wed, 2013-07-24 20:43.

Hi Fabio,

I was looking on internet for some gravity compensation method and I came across this website and your algorithm.

If my understanding is correct what you do is computing the gravity components by using the angles previously given by the IMU and then you simply subtract the gravity that you just obtained to the accelerometer reading (which includes both gravity and dynamic acceleration). Is this correct?

If I got it right, you are using the angles that are already affected by a non inertial acceleration (the dynamic one) sensed by the accelerometer (because as you know in the IMU the accelerometer corrects the gyro's angles) thus the angles themselves are wrong. Therefore I think that what you eventually get is not correct.

Please let me know if I misunderstood something.

[Login](#) to post comments

how does this deal with stuff like centripetal acceleration

Submitted by Hari (not verified) on Thu, 2012-11-22 09:17.

Suppose you have the IMU unit on the end of a rope, being swung around in a vertical circle. The accelerometer would be sensing the centripetal acceleration as dynamic acceleration right ? Wouldn't that confuse this algorithm when used for computing vertical position (complementary filter with baro pressure sensor).

[Login](#) to post comments

Yeah, it will sense the

Submitted by fabio on Thu, 2012-11-22 12:01.

Yeah, it will sense the centripetal acceleration.

In theory, given a perfect filter and a perfect sensor, it wouldn't effect the altitude estimation.. but in reality, tests are needed and probably the algorithm parameters as well as the sensor configurations should be adapted to this use case.

NOTE: We are trying to move to the newly launched [FreeIMU community website](#) to answer such kind of questions. Please post there your future questions. Thanks!

[Login](#) to post comments

Fabio a quick question on the

Submitted by Anonymous (not verified) on Tue, 2012-10-16 23:57.

Fabio a quick question on the units:

Are you assuming that q_0 , q_1 , q_2 and q_3 form a unit quaternion? And the units for g_z g_y g_z are in 'g'?

-S

[Login](#) to post comments

Yes. Yes. :-)

Submitted by fabio on Wed, 2012-10-17 09:50.

Yes. Yes. :-)

[Login](#) to post comments

Can it be used for 6DOF IMU?

Submitted by Primoz (not verified) on Thu, 2012-09-27 09:36.

I would also like to remove gravity component from my accelerometer readings - however, I have only 6DOF IMU (no magnetometer). Is it still possible to use this code? I always get wrong results (but this may be because of wrong implementation on my side OR the fact that this code can't be used for 6DOF IMU - which is correct?).

[Login](#) to post comments

Hi Fabio, can you please

Submitted by chris (not verified) on Sat, 2012-08-11 10:01.

Hi Fabio,
can you please explain me your calculation? What are you doing in the first step of your calculation and the second step? Moreover what are the requirements for the quaternion. Do you assume that you know the absolute orientation based on a start position? Or do you just take the relative orientation change from the last update of the AHRS algorithm?

(I am the guy from mathematics, if you are wondering)

Best regards

Chris

[Login](#) to post comments

Given a vector of dimension 1

Submitted by fabio on Sun, 2012-08-12 17:32.

Given a vector of dimension 1 normal to the Earth plane pointing the sky expressed in the Earth frame representing the gravity (1g) and given a quaternion coming out from the sensor fusion algorithm representing the orientation of the board with respect to the Earth, the first part of the code express the vector into the sensor frame.

Then as your accelerometers readings expressed in g in the sensor frame are effected by gravity too, you can simply subtract the gravity vector expressed in the sensor frame computed in the first step and you obtain gravity free accelerometer readings.

Hope this helps, if you know a better way or if it is not clear, let me know.

[Login](#) to post comments

Hi Fabio, thanks for the fast

Submitted by chris (not verified) on Sun, 2012-08-12 22:10.

Hi Fabio,
thanks for the fast answer.

Ok I know understand what you have done, but the problem is that it is only working well if you know your device orientation at the beginning, because the AHRS algorithm is only giving relative orientation changes.
Or do I have overlooked something?

What do you think about my approach on stackoverflow?

Best regards
chris

[Login](#) to post comments

No, the orientation from the

Submitted by fabio on Sun, 2012-08-12 22:42.

No, the orientation from the AHRS is absolute. I personally used this code in a project so it should work. what stackoverflow are you talking about?

[Login](#) to post comments

Yes the orientation is

Submitted by chris (not verified) on Mon, 2012-08-13 07:25.

Yes the orientation is absolute but only based on the initial conditions, as far as I understand and tried on implementing myself.

I am talking about this thread, where you have answered to my questions:
<http://stackoverflow.com/questions/10801951/remove-gravity-from-raw-acce...>

Do you have shared the sources of mentioned project? I would like to have a look.

Thanks Chris

[Login](#) to post comments

Not entirely true..

Submitted by fabio on Mon, 2012-08-13 09:25.

Yes the orientation is absolute but only based on the initial conditions, as far as I understand and tried on implementing myself.

That would be true for a 6DOF board (gyro + acc) but for a 9 DOF (gyro + acc + magn) it is not. Despite of the board initial conditions, the algorithm always get to the correct absolute orientation.

That source code is part of a bigger project which I still didn't published so I still can't share such code. I will do it in September or October.

[Login](#) to post comments

gravity compensation

Submitted by Markus (not verified) on Mon, 2011-11-28 15:47.

This could be also a good idea for vertical position estimation:

- Compensate gravity from vertical acc.
 - Double integrate vertical acc.
 - Use complementary filter (high-pass) for double integrated acc and lowpass for baro.
- > Drift from double integral does not affect to result and baro can be filtered at much lower freq.

[Login](#) to post comments

Yep! That's exactly how it's

Submitted by fabio on Mon, 2011-11-28 16:09.

Yep! That's exactly how it's done in the MultiWii software for controlling quadcopters! See the code available from <http://code.google.com/p/multiwii/>

[Login](#) to post comments

Copyright © 2007-2013 Fabio Varesano - [Some rights reserved](#).

[Privacy Policy](#)