International Journal of Scientific & Engineering Research, Volume 7, Issue 10, October-2016 140
ISSN 2229-5518

Method of Transforming Coordinates of a Vehicle-mounted Accelerometer: Example Application of Calculating Lean Angle of a Motorcycle

Andrei Nicolae

Abstract—When mounting an accelerometer onto a vehicle, it may be difficult or inconvenient to align the sensor's axes with the desired directions of x, y, z. One way to overcome this issue is to mount the sensor in an arbitrary direction and then mathematically transform the acceleration readings from the sensor coordinate system to the vehicle's coordinate system. Described is a simple method to determine the transformation constants (components of the rotation matrix) and a sample code for an Arduino microcontroller. Once computed, these 9 constants can be used to transform any other vector quantity between the sensor and vehicle reference frames, for example, gyro and magnetometer readings. A practical example is given where an accelerometer/gyro module is used on a motorcycle to calculate forward acceleration and lean angle during a turn.

Index Terms—accelerometer, coordinate transform, gyroscope, lean angle, motorcycle, rotation matrix, tilt sensor.

1 Introduction

MEMSbased accelerometer/gyro sensors have become overwhelmingly popular in the last few years due to low cost and increased availability of electronic platforms such as Arduino/Raspberry Pi microcontrollers and smartphones [1], [2], [3]. These sensors are often constructed to give an output in three orthogonal directions x,y,z[4]. When mounting such devices onto a vehicle, it may not be practical to align the sensor's axes to the vehicle's axes within the desired tolerance due to the lack of physical references. For example, attaching an accelerometer to a curved motorcycle frame while keeping all axes aligned properly is a tedious task and prone to human error. Therefore, it is much preferred that the sensor can be mounted in any direction and its readings transformed to the bike's reference frame.

All necessary constants for deriving the transformation equations can be obtained by collecting accelerometer data with the vehicle tilted about the forward axis (such as placing a motorcycle on a kickstand or tilting a four-wheeled vehicle on its side wheels) and another set of readings while the vehicle is upright on a flat surface. An example is then shown how this technique is used to build a device capable of measuring a motorcycle's forward/braking acceleration and also determine its true lean angle during turns. This has previously been done using optical vision systems [5], [6] or sensor networks along with digital processing (for example, Kalman filtering) [7], [8]. However, the method presented utilizes only one chip (accelerometer/gyro) and is much easier to implement on a microcontroller.

2 Transformation Equations

Suppose that an accelerometer is mounted on a vehicle as shown in figure 1. The readings from the accelerometer will be given as components to the $(\hat{\mathbf{x}}_s, \hat{\mathbf{y}}_s, \hat{\mathbf{z}}_s)$basis. However, it is desired to obtain the acceleration vector expressed in the vehicle's reference frame which has basis$(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$(shown blue in figure 1).

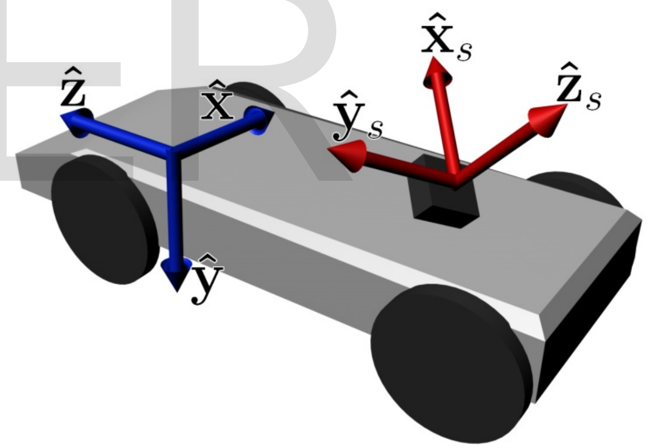

Figure 1.A sensor mounted on a vehicle in an arbitrary orientation with coordinate system shown in red. The desired coordinate system is shown in blue.

This can be done by applying a mathematical transformation to the accelerometer readings. The necessary equations are derived as follows [9].

Suppose that the vehicle is stationary on a flat, horizontal surface. In this case, the accelerometer will return readings denoted by$\left(a_{xF}, a_{yF}, a_{zF}\right)$. These are components to the gravity vector, expressed from the sensor's reference frame. Since $\hat{\mathbf{y}}$is parallel to gravity, this vector is simply $g\hat{\mathbf{y}}$.

$$\boldsymbol{v_1} = a_{xF}\hat{\mathbf{x}}_s + a_{yF}\hat{\mathbf{y}}_s + a_{zF}\hat{\mathbf{z}}_s = g\hat{\mathbf{y}} \tag{1}$$

Now suppose that the vehicleisstationary but tilted about the $\hat{\mathbf{z}}$axis by an angle α as shown in figure 2.

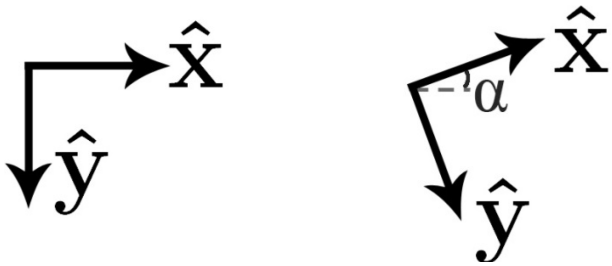

Figure 2. Tilting the vehicle about the ẑaxis by an angle α.

In the tilted case, the accelerometer will return readings denoted by (a_{xT}, a_{yT}, a_{zT}).These are still components of the gravity vector, which can be expressed in terms of the$\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$vectorsby $g\cos(\alpha)\hat{\boldsymbol{y}} - g\sin(\alpha)\,\hat{\boldsymbol{x}}$.

$$\begin{aligned} \boldsymbol{v_2} &= a_{xT}\hat{\mathbf{x}}_s + a_{yT}\hat{\mathbf{y}}_s + a_{zT}\hat{\mathbf{z}}_s \\ &= g\cos(\alpha)\hat{\mathbf{y}} - g\sin(\alpha)\hat{\mathbf{x}} \end{aligned} \tag{2}$$

Taking the cross product of thesetwovectorsgives

$$\boldsymbol{v_1} \times \boldsymbol{v_2} = c_x\hat{\mathbf{x}}_s + c_y\hat{\mathbf{y}}_s + c_z\hat{\mathbf{z}}_s = g^2\sin(\alpha)\,\hat{\mathbf{z}} \tag{3}$$

where

$$\begin{aligned} c_x &= a_{yF}\,a_{zT} - a_{yT}\,a_{zF} \\ c_y &= a_{xT}\,a_{zF} - a_{xF}\,a_{zT} \\ c_z &= a_{xF}\,a_{yT} - a_{xT}\,a_{yF} \end{aligned} \tag{4}$$

Now the unit vectors$\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$canbesolved for :

$$\begin{aligned} \hat{\mathbf{y}} &= (a_{xT}\hat{\mathbf{x}}_s + a_{yT}\hat{\mathbf{y}}_s + a_{zT}\hat{\mathbf{z}}_s)/g \\ &= Y_x\hat{\mathbf{x}}_s + Y_y\hat{\mathbf{y}}_s + Y_z\hat{\mathbf{z}}_s \end{aligned} \tag{5}$$

$$\hat{\mathbf{z}} = \frac{c_x\hat{\mathbf{x}}_s + c_y\hat{\mathbf{y}}_s + c_z\hat{\mathbf{z}}_s}{\sqrt{c_x^2 + c_y^2 + c_z^2}} = Z_x\hat{\mathbf{x}}_s + Z_y\hat{\mathbf{y}}_s + Z_z\hat{\mathbf{z}}_s \tag{6}$$

$$\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{z}} = X_x\hat{\mathbf{x}}_s + X_y\hat{\mathbf{y}}_s + X_z\hat{\mathbf{z}}_s \tag{7}$$

with

$$\begin{aligned} X_x &= Y_yZ_z - Y_zZ_y \\ X_y &= Y_zZ_x - Y_xZ_z \\ X_z &= Y_xZ_y - Y_yZ_x \end{aligned} \tag{8}$$

The relation between basis vectorsistherefore

$$\begin{aligned} \hat{\mathbf{x}} &= X_x\hat{\mathbf{x}}_s + X_y\hat{\mathbf{y}}_s + X_z\hat{\mathbf{z}}_s \\ \hat{\mathbf{y}} &= Y_x\hat{\mathbf{x}}_s + Y_y\hat{\mathbf{y}}_s + Y_z\hat{\mathbf{z}}_s \\ \hat{\mathbf{z}} &= Z_x\hat{\mathbf{x}}_s + Z_y\hat{\mathbf{y}}_s + Z_z\hat{\mathbf{z}}_s \end{aligned} \tag{9}$$

Whenmeasuring an arbitraryacceleration, the components are given in the accelerometer'sreference frame. However, thissamevector has a representation in the vehicle'sframe.

$$\boldsymbol{a} = a_x\hat{\mathbf{x}}_s + a_y\hat{\mathbf{y}}_s + a_z\hat{\mathbf{z}}_s = A_x\hat{\mathbf{x}} + A_y\hat{\mathbf{y}} + A_z\hat{\mathbf{z}} \tag{10}$$

Substituting our transformation equations to change everything to the sensor frame basis gives

$$\begin{aligned} a_x\hat{\mathbf{x}}_s + a_y\hat{\mathbf{y}}_s + a_z\hat{\mathbf{z}}_s = {} & A_x(X_x\hat{\mathbf{x}}_s + X_y\hat{\mathbf{y}}_s + X_z\hat{\mathbf{z}}_s) \\ & + A_y(Y_x\hat{\mathbf{x}}_s + Y_y\hat{\mathbf{y}}_s + Y_z\hat{\mathbf{z}}_s) \\ & + A_z(Z_x\hat{\mathbf{x}}_s + Z_y\hat{\mathbf{y}}_s + Z_z\hat{\mathbf{z}}_s) \end{aligned} \tag{11}$$

Equatingvectorsyields the linear system

$$\begin{aligned} a_x &= A_xX_x + A_yY_x + A_zZ_x \\ a_y &= A_xX_y + A_yY_y + A_zZ_y \\ a_z &= A_xX_z + A_yY_z + A_zZ_z \end{aligned} \tag{12}$$

whichcanbewritten in matrix form

$$\begin{bmatrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix}\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = M\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{13}$$

The components of acceleration in the vehicle's frame canbefound by inverting M. However, sincethis transformation is pure rotation, M is orthogonal whichmeans $M^{-1} = M^T$. Therefore

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = M^T\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} X_x & X_y & X_z \\ Y_x & Y_y & Y_z \\ Z_x & Z_y & Z_z \end{bmatrix}\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{14}$$

The components of acceleration in the vehicle's frame canbecomputed by transforming the readingsfrom the accelerometerusing the followingequations

$$\begin{aligned} A_x &= X_xa_x + X_ya_y + X_za_z \\ A_y &= Y_xa_x + Y_ya_y + Y_za_z \\ A_z &= Z_xa_x + Z_ya_y + Z_za_z \end{aligned} \tag{15}$$

3 Implementation in C

To implement the above method, readings from the accelerometer must be collected with the vehicle in the flat and tilted positions. Normally, data should be collected over several seconds and averaged to avoid noise. These readings will be used to compute the 9 components of matrix M which is done once and stored in memory. The code below is one possible example of how to do so on an Arduino microcontroller.

```
#define G 9.81
float X[3], Y[3], Z[3]; // initialize arrays

void setup() {
```

ISSN 2229-5518

```
  // enter values given by accelerometer with vehicle flat
  floataF[] = {axF, ayF, azF};
  // enter values given by accelerometer with vehicle tilted
  floataT[] = {axT, ayT, azT};

    // computes cross product v1 x v2
  float c[3];
  c[0] = aF[1] * aT[2] - aT[1] * aF[2];
  c[1] = aT[0] * aF[2] - aF[0] * aT[2];
  c[2] = aF[0] * aT[1] - aT[0] * aF[2];

  for (int k = 0; k < 3; k++) {
      Y[k]  =aT[k] / G;
      Z[k]   =   c[k]/sqrt(c[0]*c[0]   +   c[1]*c[1]   + c[2]*c[2]);
    }

    // compute Xx, Xy, Xz by the cross product y x z
  X[0] = Y[1] * Z[2] - Z[1] * Y[2];
  X[1] = Z[0] * Y[2] - Y[0] * Z[2];
  X[2] = Y[0] * Z[1] - Z[0] * Y[2];
  }
  void loop() {
    // prints acceleration in x, y, z
  Serial.println(A('x'));
  Serial.println(A('y'));
  Serial.println(A('z'));
  }
  // assume ax, ay, az are values given by accelerometer
  float A(int  coordinate) {
  if (coordinate == 'x') {
  return ax * X[0] + ay * X[1] + az * X[2];
    } else if (coordinate == 'y') {
  return ax * Y[0] + ay * Y[1] + az * Y[2];
    } else {
  return ax * Z[0] + ay * Z[1] + az * Z[2];
    }
  }
```

4 Calibrating the Accelerometer

Capacitance based accelerometers often have some manufacturing variability [10], meaning that it will not give a reading of 0 when the sensor is still - this is called offset error. Also, it may be desired to express acceleration in units such as m/s². To do this, the sensor must be calibrated as follows:

1. Align the accelerometer's +x axis to point straight up (use a flat surface).
2. Obtain readings from the sensor and average them. Call this number xMax.
3. Flip the sensor such that the +x axis is pointing straight down. Obtain sensor readings again, average them, and call this number xMin.
4. Compute the calibration constants using the following formula:

$$\text{xOffset} = \frac{-g(\text{xMax} + \text{xMin})}{\text{xMax} - \text{xMin}} \tag{16}$$

$$\text{xOffset} = \frac{2g}{\text{xMax} - \text{xMin}} \tag{17}$$

This is to be performed for the y and z axes in the same manner. The calibrated acceleration (with same units as g) is

$$\text{axCalibrated} = \text{xSlope} * \text{ax} + \text{xOffset} \tag{18}$$

$$\text{ayCalibrated} = \text{ySlope} * \text{ay} + \text{yOffset} \tag{19}$$

$$\text{azCalibrated} = \text{zSlope} * \text{az} + \text{zOffset} \tag{20}$$

This calibration correction must be applied before acquiring data for calculating the components of M.

5 Example: Computing the Lean Angle of a Motorcycle

Vehicles such as bicycles, motorcycles, and to a lesser extent, cars, lean when driven in curved paths. A popular electronics project is to build a device that displays the lean angle (also known as roll) of a motorcycle. One method to compute this angle is using an accelerometer/gyro, assuming that both wheels contact a horizontal surface. If the motorcycle is stationary, the lean angle can be computed by

$$\theta = \cos^{-1}(g/A_y) \tag{21}$$

However, the goal is to compute the lean angle regardless of centripetal acceleration and rider position, therefore (21) cannot be used. An equation for the true angle (the angle between the motorcycle and the vertical, regardless of speed or rider position) can be derived as follows. Suppose that the motorcycle is leaned clockwise by an angle $\theta$from the vertical. Then,

$$A_y\cos(\theta) + A_x\sin(\theta) = g \tag{22}$$

$$A_x\cos(\theta) + A_y\sin(\theta) = s \tag{23}$$

$$A_x^2 + A_y^2 = g^2 + s^2 \tag{24}$$

wheres is the centripetal acceleration. Solving forθ,

$$\theta = \sin^{-1}\left(\frac{gA_x - sA_y}{A_x^2 + A_y^2}\right) \tag{25}$$

and

$$s = \text{sign}(s)\sqrt{A_x^2 + A_y^2 - g^2} \tag{26}$$

Computing the sign of s is therefore required. This can be done by looking at a gyro, whose readings of angular rate must also be transformed to the vehicle's reference frame. If the angular rate about the y-axis is positive, the motorcycle is making a right turn and therefore sign(s) is positive. Other-

ISSN 2229-5518

wise, sign(s) is negative. Some commercial MEMS sensors come with 3-axis acceleromenter / 3-axis gyro on the same chip, making implementation trivial.

6 Experimental Verification

An electronic device was built to measure the lean angle of a motorcycle in real time using the techniques described here. The sensor is tethered to the central processing module and may be adhered to the motorcycle frame in an arbitrary position. The device then collects data in two motorcycle positions (one on the kick-stand, and one upright) and computes the rotation matrix components.

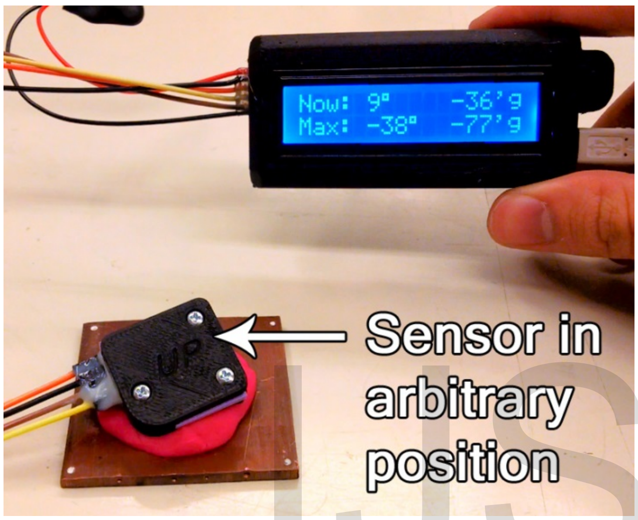

Figure 3. Prior to applying the coordinate transform, the LCD displays angle readings that don't match the desired reference frame.

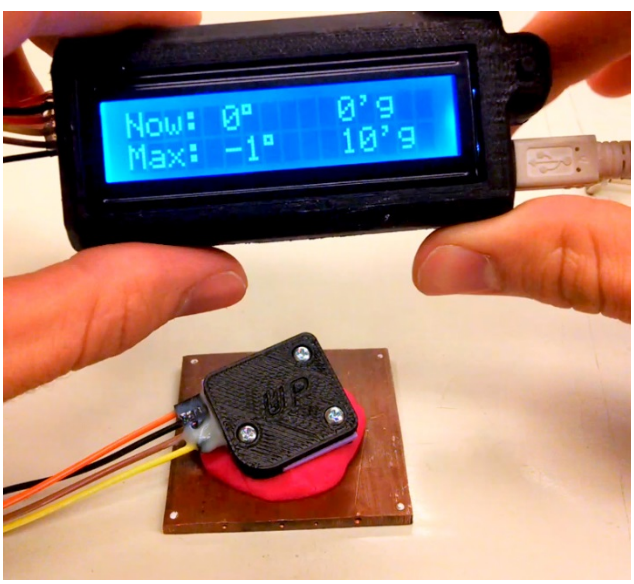

Figure 4. Display when the correct coordinate rotation is applied. The LCD shows 0^{O} tilt and 0 g acceleration.

Figures 3 and 4 show the algorithm working to correct for the arbitrary position of the accelerometer/gyro sensor. This technique is highly effective as shown here, and easy to use as seen by the equations and C implementation provided. Moving the sensor to a new location and orientation only requires the re-collection of sensor data in the two positions, a task that is completed in about 10 seconds.

Although not shown in this paper, the device was tested on a Yamaha R6 motorcycle with great success. The displayed angle was displayed correctly regardless of forward/braking acceleration or side acceleration (riding in a turn as opposed leaning the motorcycle while still). Even after months of use, the sensor system showed great stability and there was no need to perform another calibration or calculate the rotation coefficients.

References

[1] Robert Bogue. Recent developments in mems sensors: a review of applications,markets and technologies. *Sensor Review*, 33(4):300–304, 2013.

[2] M. Gunasekaran and R. Potluri. Low-cost undergraduate control systems experimentsusing microcontroller-based control of a dc motor. *IEEE Transactions on Education*,55(4):508–516, Nov 2012.

[3] D. K. Shaeffer. Mems inertial sensors: A tutorial overview. *IEEE CommunicationsMagazine*, 51(4):100–109, April 2013.

[4] W. T. Ang, S. Y. Khoo, P. K. Khosla, and C. N. Riviere. Physical model of a memsaccelerometer for low-g motion tracking applications. In *Robotics and Automation,2004. Proceedings. ICRA '04.* 2004 IEEE International Conference on, volume 2, pages 1345–1351 Vol.2, April 2004.

[5] X. Liu, Z. Cao, Chao Zhou, K. Ai, and M. Tan. A universal vision-based roll angleestimation method for mobile robots. In 2014 *9th IEEE Conference on IndustrialElectronics and Applications*, pages 468–472, June 2014.

[6] Michele Norgia, Ivo Boniolo, Mara Tanelli, Sergio M Savaresi, and Cesare Svelto.Optical sensors for real-time measurement of motorcycle tilt angle. *IEEE Transactionson Instrumentation and Measurement*, 58(5):1640–1649, 2009.

[7] I. Boniolo, M. Tanelli, and S. M. Savaresi. Roll angle estimation in two-wheeledvehicles. In *2008 IEEE International Conference on Control Applications*, pages31–36, Sept 2008.

[8] I. Boniolo, S. M. Savaresi, and M. Tanelli. Lean angle estimation in two-wheeledvehicles with a reduced sensor configuration. In 2012 *IEEE International Symposiumon Circuits and Systems*, pages 2573–2576, May 2012.

[9] Peter Chadwick. *Continuum mechanics: concise theory and problems*. Courier Corporation,2012.

[10] N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings ofthe IEEE*, 86(8):1640–1659, Aug 1998.