DT0140
Design tip

Tilt computation using accelerometer data for inclinometer applications

Andrea Vitali

Main components	
IIS2ICLX	High-accuracy, high-resolution, low-power, 2-axis digital inclinometer with embedded Machine Learning Core
IIS3DHHC	High-resolution, high-stability 3-axis digital accelerometer
ISM330DHCX	iNEMO inertial module with embedded Machine Learning Core: always-on 3D accelerometer and 3D gyroscope with digital output for industrial applications

Purpose and benefits

This design tip explains how to compute a tilt angle from accelerometer data. Benefits are:

- Better understanding of the computation and performance of the MotionTL and MotionTL2 libraries
- Guidelines on choosing 3-axis, 2-axis, and 1-axis accelerometers
- Guidelines on improving the accuracy

Introduction

An accelerometer measures acceleration, which means that in static condition it measures the gravity acceleration. The gravity vector is aligned along the vertical and points up.

When the accelerometer is in the horizontal plane and its Z-axis is pointing up, it will report $Acc_X = 0g$, $Acc_Y = 0g$, $Acc_Z = +1g$. The sign is positive because gravity is pulling down which causes the same effect as accelerating up.

When the accelerometer is tilted in any direction, the projection of the gravity vector along the Z-axis is reduced, while the projection along the X and Y axes is increased in the positive or negative direction. The amount of change is mathematically converted to the tilt angle (see next paragraph). This is how the tilt can be computed from accelerometer data.

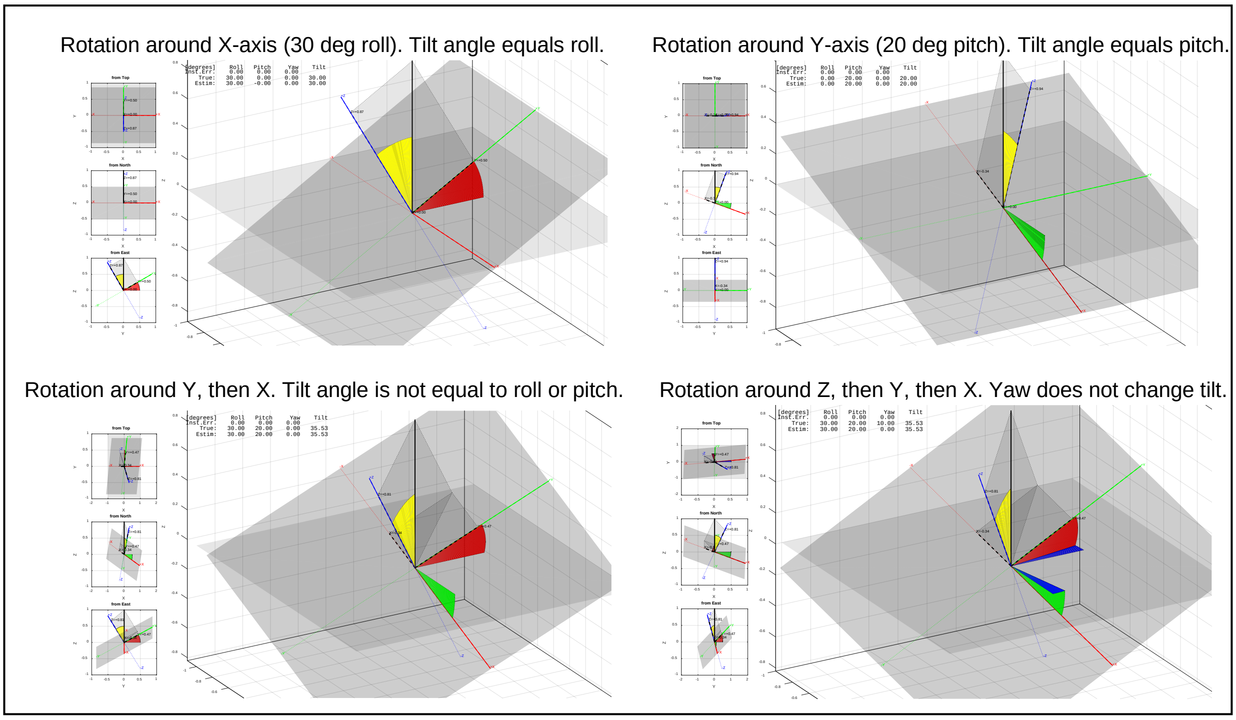
In Figure 1, one can see the effect of rotating the accelerometer around its X or Y axis. In general, the rotation can happen around any axis, but the final position can always be obtained as a sequence of two rotations: first a rotation around the Y-axis (also known as *pitch*, which causes the X and Z axes to move) and then a rotation around the X-axis (also known as *roll*, which causes the Y and Z axes to move).

Note that any rotation around the Z-axis (also known as *yaw*, which causes the X and Y axes to move) will not change accelerometer output. One needs an additional sensor, a magnetometer, to compute the yaw angle (see Design Tip DT0058).

The tilt angle does not correspond to the roll or to the pitch angles, but it is a combination of both. More precisely, the tilt is the angle between the vertical gravity vector and the vertical axis of the sensor (Z-axis if the sensor is in the horizontal plane). This is the same as the angle between the horizontal plane and the plane of the sensor (plane of the X-Y axes).

For other definitions of tilt and the corresponding formulas, see the last paragraph.

Figure 1. Accelerometer rotations: tilt angle is highlighted in yellow, roll in red, pitch in green, yaw in blue. Tilt depends only on roll and pitch. Yaw does not change it.



3-axis accelerometers, any rotation

The accelerometer is usually installed in the X-Y horizontal plane. There are two typical reference frames:

- **ENU**: X-axis East, Y-axis North, **Z-axis up**. At zero-degree tilt angle, the accelerometer output is $Acc_X = 0g$, $Acc_Y = 0g$, **$Acc_Z = +1g$**.
- **NED**: X-axis North, Y-axis East, **Z-axis down**. At zero-degree tilt angle, the accelerometer output is $Acc_X = 0g$, $Acc_Y = 0g$, **$Acc_Z = -1g$**

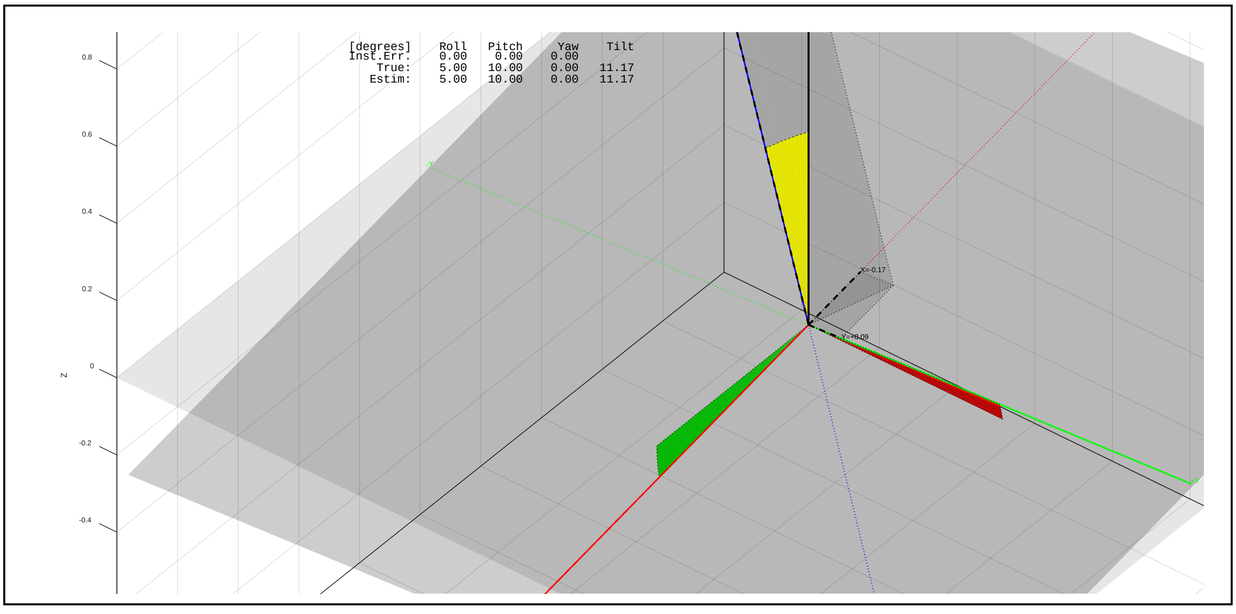
In the following formulas, the tilt is the angle between the vertical gravity vector and the Z-axis. See Figure 2 where it is highlighted in yellow.

By the definition of cosine, the projection of the gravity vector along the Z-axis is:
$Acc_Z = 1g \times \cos(\alpha)$ where "α" is the tilt angle.

By the definition of sine, the projection of the gravity vector on the plane perpendicular to the Z-axis is:
$Acc_{XY} = 1g \times \sin(\alpha)$, where $Acc_{XY} = +\mathrm{sqrt}(Acc_X^2 + Acc_Y^2)$, always positive.

By the definition of tangent: $\tan(\alpha) = \sin(\alpha) / \cos(\alpha) = Acc_{XY} / Acc_Z$.

Figure 2. Projection of the gravity vector along Z-axis and on the plane X-Y plane perpendicular to Z-axis. The tilt is the angle highlighted in yellow between the vertical gravity vector and the Z-axis.



Therefore, the tilt can be computed using one of the following formulas

- $\alpha = \arccos(Acc_Z / 1g)$, arccos output range is 0 to 180 degrees. From the point of view of noise and measurement errors, this formula is less accurate when $\mathrm{abs}(Acc_Z)$ is close to $1g$ ($\alpha < 30$ or $\alpha > 150$ degrees), see Figure 3. Note that arccos argument must be in the range [-1, +1] and clipping may be necessary.

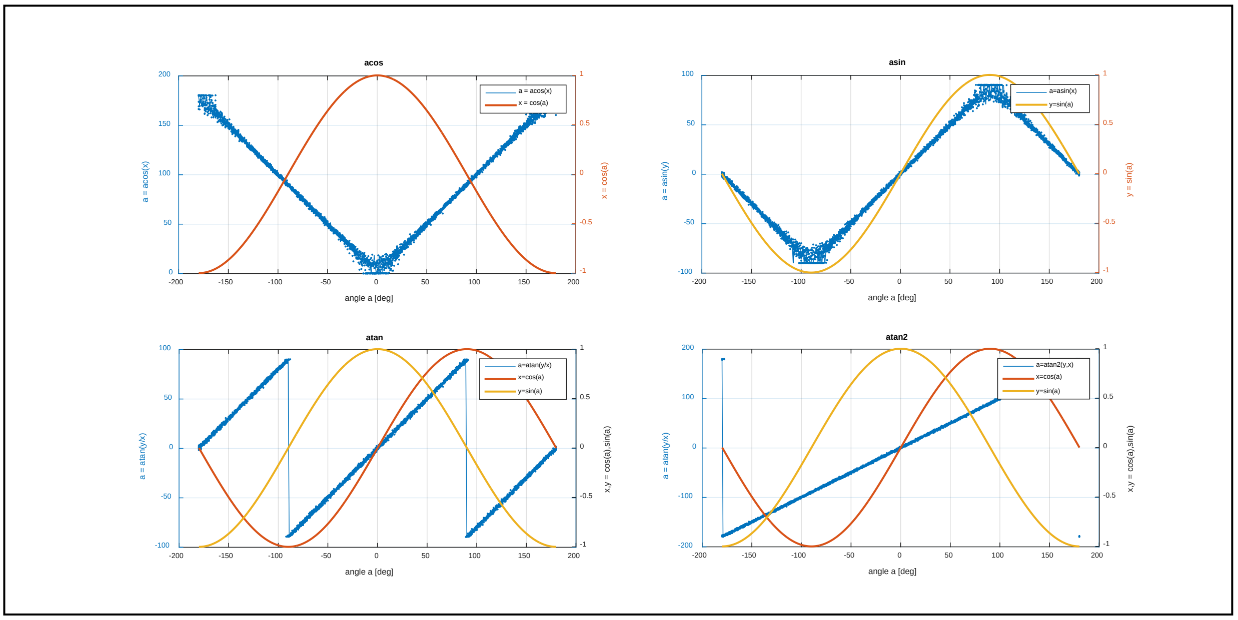
- $\alpha = \arcsin(Acc_{XY} / 1g)$, arcsin output range is -90 to +90 degrees, but here the argument Acc_{XY} is always positive, therefore the actual output range is 0 to +90 degrees. This formula is less accurate when Acc_{XY} is close to $1g$ (large tilt: $\alpha > 70$ degrees), see Figure 3. Note that arcsin argument must be in the range [-1, +1] and clipping may be necessary.

- $\alpha = \arctan(Acc_{XY} / Acc_Z)$, arctan output range is -90 to +90 degrees. The span is 180 degrees but there is a discontinuity at ±90 degrees (see Figure 3 and Figure 4), and for this reason, this formula should be avoided.

- $\alpha = \mathrm{arctan2}(Acc_{XY}, Acc_Z)$, arctan2 output range is -180 to +180 degrees, but Acc_{XY} is always positive, therefore the actual output range is 0 to 180 degrees. **There is**

no discontinuity, and this formula is always the most accurate, see Figure 3 and Figure 4. For these reasons, this formula should be selected. From the point of view of noise and measurement errors, the worst case is when tilt is halfway (α = 45 degrees).

Figure 3. Effect of noise for inverse trigonometric functions: arccos and arcsin have a larger sensitivity when their argument is close to ±1, conversely, arctan and arctan2 have uniform and lower sensitivity.



If the data is normalized by dividing by the norm n = sqrt($Acc_X^2 + Acc_Y^2 + Acc_Z^2$), the performance of the arccos and arcsin formula becomes close to the performance of the arctan2 formula, and clipping is no longer needed because their argument is guaranteed to be in the [-1, +1] range.

Figure 4 illustrates the output of the tilt formula. The accelerometer can be set in the ENU reference frame (Z-axis up) or NED reference frame (Z-axis down). When the accelerometer is tilted, the output is the number indicated by the corresponding arrow.

2-axis accelerometers, any rotation

In static conditions the squared norm of the measured acceleration ($Acc_X^2 + Acc_Y^2 + Acc_Z^2$) equals 1*g*. Therefore, the missing axis can be computed by the difference and square root, and all the formulas presented above for a 3-axis accelerometer can be used.

When the 2-axis accelerometer is installed in the horizontal plane, it measures Acc_X and Acc_Y and one can compute $Acc_Z = \pm sqrt(1g^2 - Acc_X^2 - Acc_Y^2)$. The Acc_Z is assumed positive but it could also be negative. The sign of Acc_Z is lost. See Figure 4, where “+z” indicates that Acc_Z is always positive. The output range for the tilt is restricted: it goes from 0 to 90 degrees.

For this reason, a 2-axis accelerometer should not be installed in the X-Y horizontal plane, unless the restricted range for the tilt angle is acceptable for the inclinometer application.

Note that the argument of the square root must be positive, and clipping may be necessary. From the point of view of noise and measurement errors, whatever formula is selected, the performance is always close to the performance of the arcsin formula, that is, the result is less accurate when the tilt is large (α > 70 degrees). Whatever formula is selected, the output tilt will be restricted to the range [0, 90] degrees.

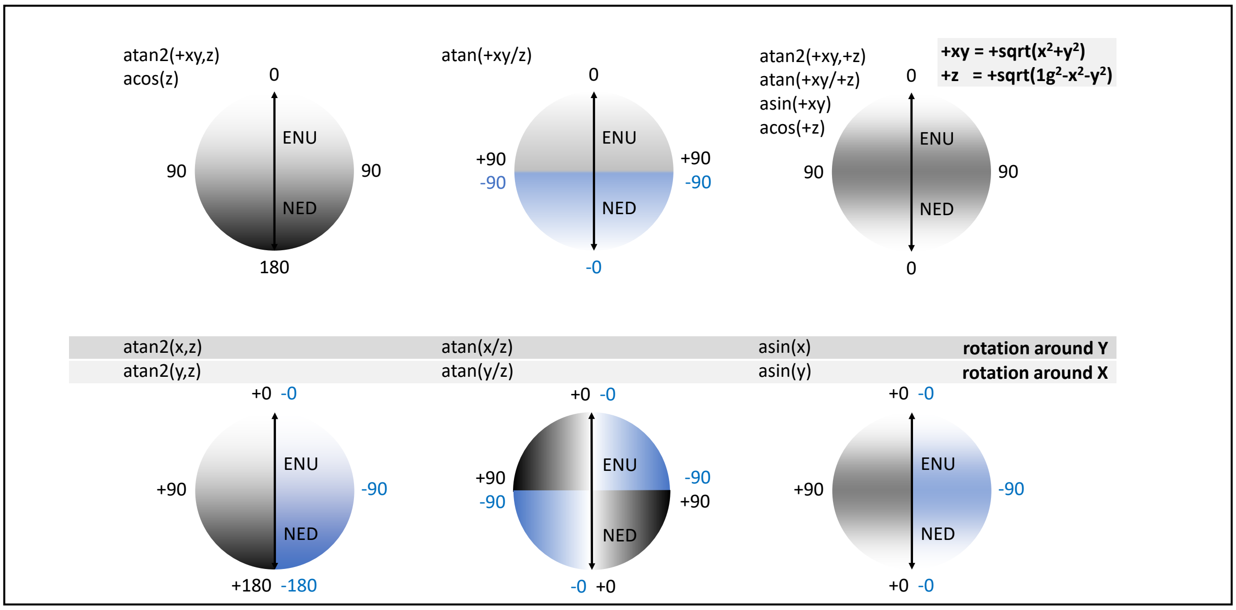
When the 2-axis accelerometer is installed in the vertical plane

- in the X-Z vertical plane, it measures Acc_X and Acc_Z and one can compute $Acc_Y = \pm\text{sqrt}(1g^2 - Acc_X^2 - Acc_Z^2)$. The sign of Acc_Y is also lost, but it does not matter because it is squared when computing Acc_{XY} and the result is always positive. There is no impact on the output range for the tilt. It goes from 0 to 180 degree as for a 3-axis accelerometer when using the arccos or arctan2 formulas.
- in the Y-Z vertical plane, it measures Acc_Y and Acc_Z and one can compute Acc_X. The performance is the same as when it is installed in the X-Z vertical plane (see previous point).

A 2-axis accelerometer should be installed in the X-Z or Y-Z vertical plane to get the full tilt range (0 to 180 degrees), see Figure 4.

Note that the argument of the square root must be positive, and clipping may be necessary. From the point of view of noise and measurement errors, whatever formula is selected, the performance is always close to the performance of the arccos formula, that is, the result is less accurate when the tilt is small (α < 30 or α > 150 degrees). Use the arctan2 or the arccos formula to get the full range for the output tilt [0, 180] degrees.

Figure 4. Output range for tilt. Top row: rotation around any axis. Bottom row: rotation restricted to be around the axis which is not measured (X or Y axis). '+' highlights the fact that the corresponding variable is always positive.



1-axis accelerometers, any rotation

If the accelerometer is measuring only Acc_X or only Acc_Y, there is not enough information to compute Acc_{XY} = sqrt(Acc_X^2+ Acc_Y^2). The α = arcsin(Acc_{XY} / 1*g*) formula cannot be used.

For this reason, the 1-axis accelerometer must be installed with its axis aligned along the vertical axis. The only formula that can be used is α = arccos(Acc_Z / 1*g*). The output tilt goes from 0 to 180 degrees as for a 3-axis accelerometer. As noted above, this formula is less accurate when abs(Acc_Z) is close to 1*g* (α < 30 or α > 150 degrees).

Rotation restricted to be around X-axis (or Y-axis)

If the accelerometer is installed in the X-Z plane and the **rotation happens to be around the Y-axis**:

- Acc_Z = 1g × cos(α), Acc_X = 1g × sin(α), **Acc_Y = 0 always**. This is the reason why a 3-axis accelerometer is not needed, and the reason why a 2-axis accelerometer is not installed in the horizontal plane: Acc_Y would not give any useful information.
- α = arccos(Acc_Z / 1*g*), arccos output range is 0 to 180 degrees. From the point of view of noise and measurement errors, this formula is less accurate when abs(Acc_Z) is close to 1*g* (α <30 or α > 150 degrees). Note that arccos argument must be in the range [-1, +1] and clipping may be necessary.
- α = arcsin(Acc_X / 1*g*), arcsin output range is -90 to +90 degrees. This formula is more accurate when abs(Acc_X) is close to 1*g* (abs(α) > 70 degrees). Note that arcsin argument must be in the range [-1, +1] and clipping may be necessary.
- α = arctan(Acc_X / Acc_Z), arctan output range is -90 to +90. The span is not a full circle and there is a discontinuity at ±90 (see Figure 3 and Figure 4). For these reasons, this formula should be avoided.
- α = arctan2(Acc_X, Acc_Z), arctan2 output range is -180 to +180 degrees. **Note that the output angle does span a full 360 degrees circle. Also, this is always the most accurate formula in all cases. For these reasons, a 2-axis accelerometer and this formula should be selected.** From the point of view of noise and measurement errors, the worst case is when tilt is halfway (α = 45 or 135 degrees).

Note that when 2 axes are available, and remembering that Acc_X is always 0, data can be normalized by dividing by the norm n = sqrt(Acc_X^2 + Acc_Z^2). In this case the performance of the arccos and arcsin formula becomes close to the performance of the arctan2 formula, and clipping is no longer needed because their argument is guaranteed to be in the [-1, +1] range. However, the use of the atan2 formula is recommended, and with atan2 the normalization does not change the output, in fact only the ratio of its argument matters.

Similar reasoning applies when the accelerometer is installed in the Y-Z axis plane and the rotation happens to be around the X-axis: **Acc_X = 0 always.** The best formula is α = arctan2(Acc_Y, Acc_Z), arctan2 output range is -180 to +180 degrees.

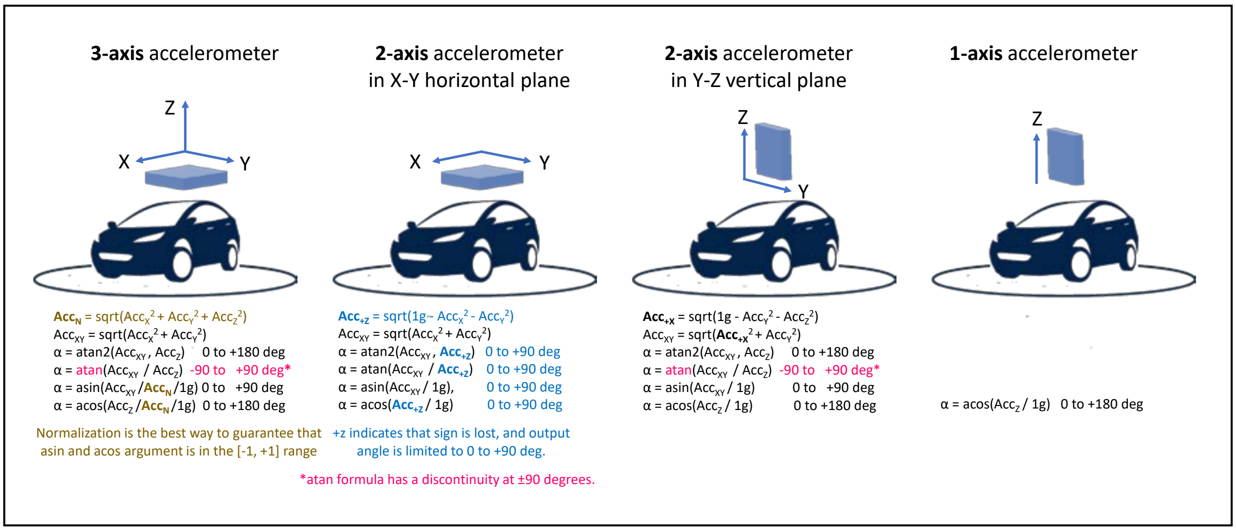
Tilt applications and guidelines

Here is a summary of the previous paragraphs. When rotation is not constrained (see also Figure 5):

- 3-axis accelerometers offer the best flexibility: full tilt range (0 to 180 degrees) and best accuracy by using the arctan2 formula; however, note that the out-of-plane Z-axis may be affected by larger errors. Worst case for accuracy is when tilt is halfway (α = 45 or 135 degrees).
- 2-axis accelerometers offer the best accuracy because there is no out-of-plane axis; if installed in the horizontal however tilt range is restricted (0 to 90 degrees) plane; if installed in the vertical plane full tilt range (0 to 180 degrees) can be achieved, but accuracy is somewhat reduced when tilt is small (α < 30 degrees).
- 1-axis accelerometers offer the lowest power consumption and full tilt range (0 to 180 degrees), but accuracy is reduced when tilt is small (α < 30 degrees).

As an example: a 2-axis accelerometer installed in the horizontal plane is an optimal choice for automotive inclinometer applications because tilt is limited, well below 70 degrees. The steepest road in the USA is Canton Avenue in Pittsburg, 37% (20.3 degrees); in San Francisco it is Filbert Street 31.5% (17.5 degrees); the maximum slope for driveways in the USA is 25% (14 degrees) and usually is <15% (8.5 degrees); local roads are usually <12-15% (6.8-8.5 degrees); maximum slope for federal roads in the USA is <7% (4 degrees).

Figure 5. Summary of accelerometer type and installation options, any rotation

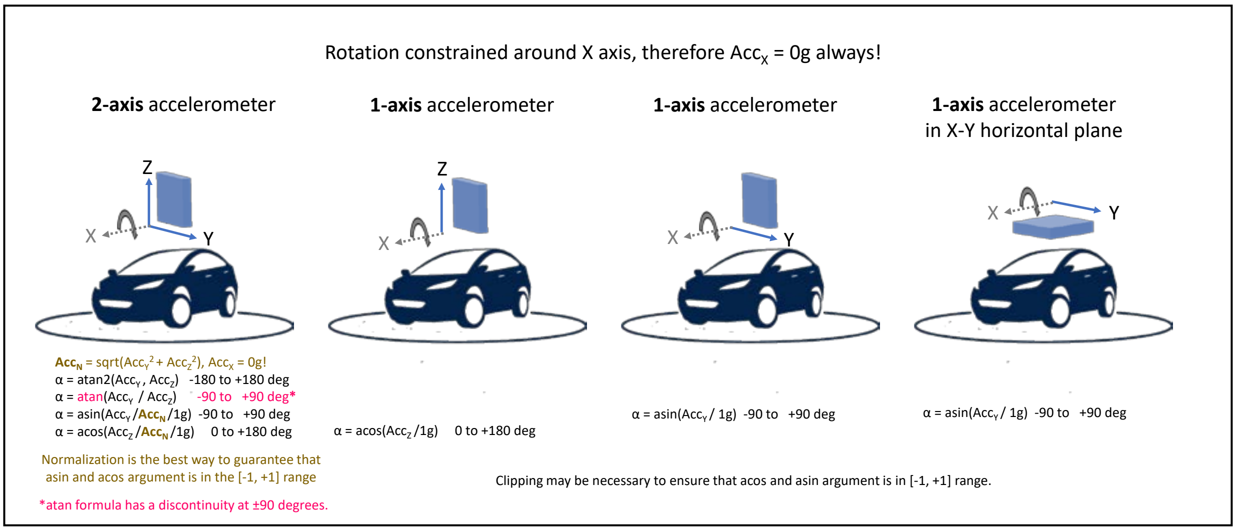


When rotation is constrained to be around the X or Y-axis (see also Figure 6):

- 3-axis accelerometers are not needed because the rotation axis will always give an output close to 0.

- 2-axis accelerometers offer the best accuracy by using the arctan2 formula.
- 1-axis accelerometers offer the lowest power consumption, but accuracy is reduced: if installed in the horizontal plane, accuracy is reduced when tilt is large (abs(α) > 70 degrees); if installed in the vertical plane it is reduced when tilt is small (α < 30 or α > 150 degrees).

Figure 6. Summary of accelerometer type and installation options, rotation constrained around X-axis



How to improve accuracy

Noise can be averaged out if it has zero mean. The higher the frequency, the easier it is to remove the noise. By averaging N samples, the RMS noise is reduced by a factor of sqrt(N).

If the sensor noise density ND is constant across its bandwidth BW, the RMS noise can be computed as RMS = ND * sqrt(BW * c). The bandwidth is at most equal to the output data rate divided by 2; it can be less if internal low-pass filtering is enabled. The constant c is a correction term to consider the fact that the internal filter that defines the bandwidth is not a boxcar filter, c is usually 1.6-1.7:

- The averaging is simple. If N is a power of two: just sum N samples and then shift right to remove log2(N) least significant bits. For an output which is continuously updated, the moving average is also very inexpensive: just use a circular FIFO buffer and maintain the accumulator by subtracting the oldest sample in the buffer and adding the newest sample. In this way, complexity does not grow with N (big-O notation: complexity is O(1) instead of O(N)). See reference C code below.
- The only disadvantage is that this filtering operation adds a latency proportional to N. N must be small or the output data rate must be high to reduce the overall latency.

As an example: if the RMS noise is 3m*g*, by averaging over 8 samples one can bring it down to 3m*g* / sqrt(8) = 1.06m*g*. If output data rate is 100Hz, one sample is taken every 10ms, averaging over 8 samples introduces a latency of 8 * 10ms = 80ms. If the output data rate is increased to 1kHz, one sample is taken every 1ms, and the latency is reduced to 8 * 1ms = 8ms.

Note that for time alignment purposes: the group delay of a moving average is equal to N/2 samples. The output of the moving average is referenced to t - (N/2) × T_S where t is the time when the output can be computed because all N samples are available, and T_S is the sample interval (inverse of sampling frequency: T_S = 1/ F_S).

Reference C code for the moving average is copied below for convenience. Note that when using the arctan or arctan2 formulas, the scale of the accelerometer data does not matter (because only the ratio matters). In this case division by N or shift to right by log2(N) can be skipped.

```
// Moving Average on N samples, note that outputs overwrite inputs!
#define N 8 // LOG2MALEN is log2(8) = 3
function AccMovAvg(int *AccX, int *AccY, int *AccZ) {
  static int MAidx = 0;
  static int MAaccX = 0, MAaccY = 0, MAaccZ = 0;
  static int MAbufX[N] = { 0 }, MAbufY[N] = { 0 }, MAbufZ[N] = { 0 };

  MAaccX -= MAbufX[MAidx]; MAaccX += AccX; MAbufX[MAidx] = AccX;
  MAaccY -= MAbufY[MAidx]; MAaccY += AccY; MAbufY[MAidx] = AccY;
  MAaccZ -= MAbufZ[MAidx]; MAaccZ += AccZ; MAbufZ[MAidx] = AccZ;

  MAidx = (MAidx+1) % N;
  // or: MAidx = (MAidx+1) & (N-1); // if N is 2^n
  // or: ++MAidx; if (MAidx==N) MAidx = 0; // if modulo is expensive

  *AccX = MAaccX / N; // or: *AccX = MAaccX >> LOG2MALEN; // if N is 2^n
  *AccY = MAaccY / N; // or: *AccY = MAaccY >> LOG2MALEN;
  *AccZ = MAaccZ / N; // or: *AccZ = MAaccZ >> LOG2MALEN;

  return;
}
```

Measurement errors can be compensated by calibration. The measurement can be affected by several errors. The linear sensor model is the following:

$$\begin{bmatrix} Acc_X \\ Acc_Y \\ Acc_Y \end{bmatrix} = \begin{bmatrix} S_X & C_{YX} & C_{ZX} \\ C_{XY} & S_Y & C_{ZY} \\ C_{XZ} & C_{YZ} & S_Z \end{bmatrix} \begin{bmatrix} True_X \\ True_Y \\ True_Z \end{bmatrix} + \begin{bmatrix} O_X \\ O_Y \\ O_Z \end{bmatrix}$$

Which corresponds to the following three equations:

$$Acc_X = S_X\ True_X + C_{YX}\ True_Y + C_{ZX}\ True_Z + O_X$$

$$Acc_Y = C_{XY}\ True_X + S_Y\ True_Y + C_{ZY}\ True_Z + O_Y$$

$$Acc_Z = C_{XZ}\ True_X + C_{YZ}\ True_Y + S_Z\ True_Z + O_Z$$

Where [$True_X$, $True_Y$, $True_Z$] is the true acceleration vector that stimulates the sensor; O coefficients are the offsets; S coefficients are the sensitivities; C coefficients are the cross-axis sensitivities. In an ideal scenario, offsets are 0, sensitivities are 1, cross-sensitivities are 0.

Note that all coefficients have a dependence on temperature. As an example: for IIS2ICLX the offset can be within ±8m*g* at room temperature (+25°C) but can increase up to ±14m*g* when the sensor is hot (+105°C).

The dependency on frequency is not relevant for inclinometer applications, because one is interested in the DC components. However high-frequency AC components can have a significant effect (e.g. vibration rectification error).

Offset and sensitivities are affected by mechanical stress induced by the package as well as external factors, and therefore they change when the sensor is soldered on its host board. They also change with temperature and time (aging). In this respect, ceramic packages, such as the one used for IIS2ICLX, have a much better behavior with respect to plastic packages.

Cross-axis sensitivities are affected by non-orthogonality and rotation. Non-orthogonality is usually negligible in MEMS sensors. Thanks to the accurate geometry of the silicon mechanical structure, all sensor axes are orthogonal, that is 90 degrees with respect to each other. Circuit cross-talk can however give a small contribution.

On the contrary, unwanted rotation can be significant. This is due to how the sensor is soldered on the board, how the board is enclosed in its housing, and how the final device is installed in the field. All these unwanted rotations end up in the so-called installation or alignment error. By compensating cross-axis sensitivities, the installation or alignment error is also compensated (but not the other way around: if there were some non-orthogonality, it could not be compensated by any de-rotation).

Calibration can help compensate for offset, sensitivity, and cross-axis errors:

- 1-tumble calibration removes the offset error (see Design tip DT0105, accurate orientation of the accelerometer is required: roll and pitch must be 0 degree to ensure the Z-axis is aligned with the vertical gravity vector in its positive direction)
- 3-tumble calibration removes the offset error and compensates for non-unity sensitivity for all axes (see Design tip DT0105, accurate orientation of the accelerometer is required to ensure each axis in turn is aligned with its vertical gravity vector in the positive direction).
- 6-tumble calibration removes the offset error, compensates for non-unity sensitivity, and removes any cross-axis sensitivity error (see Design tip DT0053, accurate orientation of the accelerometer is required to ensure each axis in turn is aligned with the vertical gravity vector in its positive and negative direction).
- If tumble positions cannot be set accurately, sphere/ellipsoid fitting can also be used (see Design tip DT0059). The accelerometer must be kept stationary and not moving when collecting each data point. The accuracy of the calibration is improved if the data points are spread evenly on the sphere/ellipsoid surface.
    - Sphere fitting removes the offset error.
    - Ellipsoid fitting removes the offset and sensitivity error.
    - Rotated ellipsoid fitting removes the offset, sensitivity, and cross-sensitivity error (but is reliable only if sensitivities are different from each other before the compensation).

- Calibration should be performed at different temperatures and compensation parameters should be interpolated/extrapolated accordingly. When possible, the coldest and hottest working temperature points should be used.

Installation or alignment error compensation, also known as output zeroing, must always be done (see Design tip DT0076). Note that this error is automatically compensated if 6-tumble calibration or rotated ellipsoid fitting is performed after installation.

The **MotionFE** fine alignment library is also available to automatically compute the de-rotation matrix and match the output of the on-board accelerometer to an external reference accelerometer (the library can also be used to match the output of the on-board gyroscope to an external reference gyroscope).

Gyroscope to improve stability and enable dynamic tilt tracking. If the accelerometer sensor is subject to any acceleration, the measured acceleration does not correspond anymore to the components of the gravity vector projected on the sensor axes. The additional acceleration will alter the measurement and compromise the tilt computation.

- The dynamic condition can often be detected by checking the norm n = sqrt(Acc_X^2+ Acc_Y^2+ Acc_Z^2): if it is far from 1*g*, one can flag the tilt computation output as wrong. Note that there can be cases where the vector sum of the gravity vector and the additional acceleration does have 1*g* norm. When this happens, the wrong output will not be flagged.

- When the dynamic condition is detected, the output angle can be updated by exploiting the gyroscope data. The gyroscope is not affected by linear acceleration. The gyroscope output is the angular velocity and by integration one can compute the update for the tilt angle (see Design tip DT0060 for an example on how this is done for Euler angles).

- When the gyroscope output is close to 0, the tilt angle can be assumed to remain the same. This fact can be used in adaptive filters such as the complementary filter or the Kalman filter: tilt computed from the accelerometer data is mixed with tilt updated from gyroscope data to keep the gyroscope integration error under control and stabilize the final tilt estimate.

- Note that the gyroscope output is affected by a spurious offset that must be estimated and compensated.

The MotionDI Dynamic Inclinometer library implements the solution described above. To achieve the best accuracy, samples from the accelerometer and gyroscope should be taken at the same time. This is guaranteed by design on the Inertial Measurement Unit ISM330DHCX.

See also the following documents:

- AN5551, application note on precise and accurate tilt sensing in industrial applications, in particular the paragraph dedicated to error and calibration.

- UM2182, the user manual of **MotionAC** accelerometer calibration library in the X-Cube-MEMS1 expansion package. This calibration compensates for offset and sensitivity errors. Measurement data can be taken by 6-tumble calibration (to have each of the three-axis pointing up, then down) or continuously.

- UM2774, the user manual of **MotionAC2** 2-axis accelerometer calibration library in the X-Cube-MEMS1 expansion package. This library compensates for offset and sensitivity errors by performing ellipse fitting on measurements data taken by 4-tumble calibration (rotation in Y-Z or X-Z vertical plane, to have the first, then second sensor axis pointing up, then down).

- UM2724, the user manual of **MotionDI** dynamic inclinometer library in the X-Cube-MEMS1 expansion package. This library includes the accelerometer calibration (MotionAC) and the gyroscope calibration (MotionGC). The adaptive Kalman filter is used for data fusion.

- TN0018, technical note, Surface mounting guidelines for MEMS sensors in an LGA package: PCB design guidelines, stencil design and solder paste application, process considerations, solder heat resistance and environmental specifications.

Effect of errors

The cumulative effect of offset, sensitivity, and cross-axis error can be computed using a Monte-Carlo simulation. The roll and pitch angles are swept. For each possible orientation, N error samples are taken and applied to the ideal measurement (N=1000 for the table below, N=100 for the plots in the figures). The tilt is computed using the specified formula, then the maximum and average absolute tilt error is recorded.

Two distributions are used for the error samples:

- Uniform distribution. This is useful to verify the worst-case performance by looking at the maximum tilt error. Note that when the number N of error samples is large, one should also check the maximum tilt error using the Gaussian distribution which is not bounded.

- Gaussian distribution (also known as Normal) with standard deviation chosen so that 99.6% of samples are in the specified error interval. Note that the actual error interval is not bounded: 0.4% of samples will be out of the specified interval. This distribution is useful to verify the typical error performance by looking at the average tilt error.

Figure 7 shows the performance of the industrial 3-axis accelerometer IIS3DHHC vs. the high-precision 2-axis inclinometer IIS2ICLX installed in the X-Y horizontal plane. Range of tilt is limited to ±45 degrees (remember that when $\alpha < 70$ degrees the performance of the 2-axis accelerometer installed in the horizontal plane is optimal). N=100 error samples for each possible orientation.

The table below summarizes the expected performance in the same conditions as Figure 7. N=1000 error samples for each possible orientation. Each cell contains the range for the tilt error, the average over all possible orientations is also shown in parenthesis.

Accelerometer and Tilt formula	Offset, sensitivity, and cross-axis max error	Mean error with Gaussian distribution	Maximum error with Gaussian distribution	Mean error with Uniform Distribution	Maximum error with Uniform distribution
IIS3DHHC 3-axis Atan2(Acc_{XY}, Acc_{Z})	60*mg*, 8.35%, 1%	0.84 – 1.44 deg (avg 1.17 deg)	3.22 – 8.23 deg (avg 5.06 deg)	1.48 – 3.64 deg (avg 2.06 deg)	4.49 – 10.14 deg (avg 7.64 deg)
II2ICLX 2-axis Atan2(Acc_{XY}, **Acc_{+Z}**)	14*mg*, 2.96%, 1%	0.20 – 0.78 deg (avg 0.40 deg)	0.78 – 4.44 deg (avg 1.70 deg)	0.36 – 1.35 deg (avg 0.70 deg)	0.88 – 5.48 deg (avg 2.35 deg)

Figure 7. Performance with maximum offset and sensitivity error including maximum temperature drift. Roll and pitch in [-45, +45] degree range, arctan2 formula.

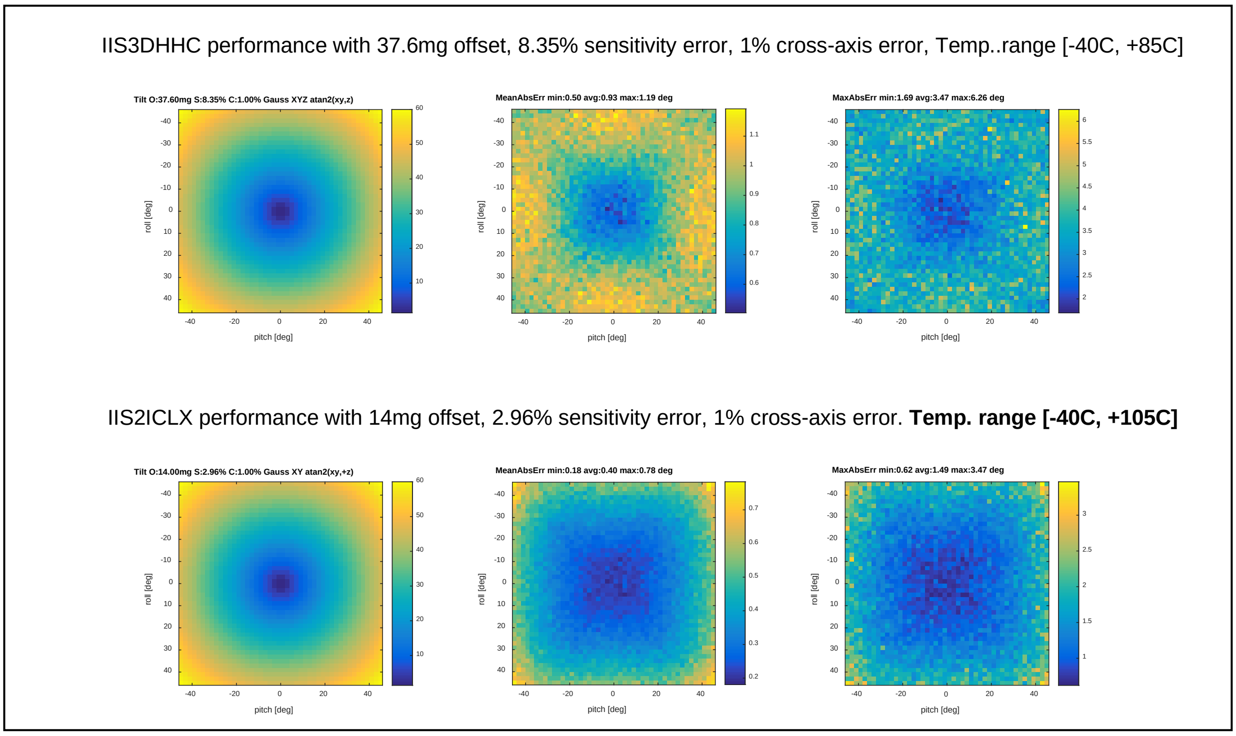


Figure 8 shows the performance of a generic high-performance accelerometer: 3-axis, vs. 2-axis installed in the X-Y horizontal plane, vs. 2-axis installed in the X-Z vertical plane, vs. 1-axis. Range of tilt is limited to 90 degrees (remember that tilt output is limited to 90 degrees for a 2-axis accelerometer installed in the horizontal plane).

Figure 9 same as Figure 8 but the full range for tilt is tested, up to 180 degrees. Note how the output from a 2-axis accelerometer installed in the X-Y horizontal plane is limited to 90 degrees.

Figure 8. Performance of a generic 3, 2, and 1-axis accelerometer with 10m*g* offset, 2% sensitivity error, and 1% cross-axis sensitivity error. Arctan2 formula is used in all cases, except for 1-axis accelerometer which can only use arccos. Roll and pitch in [-90, +90] degree range.

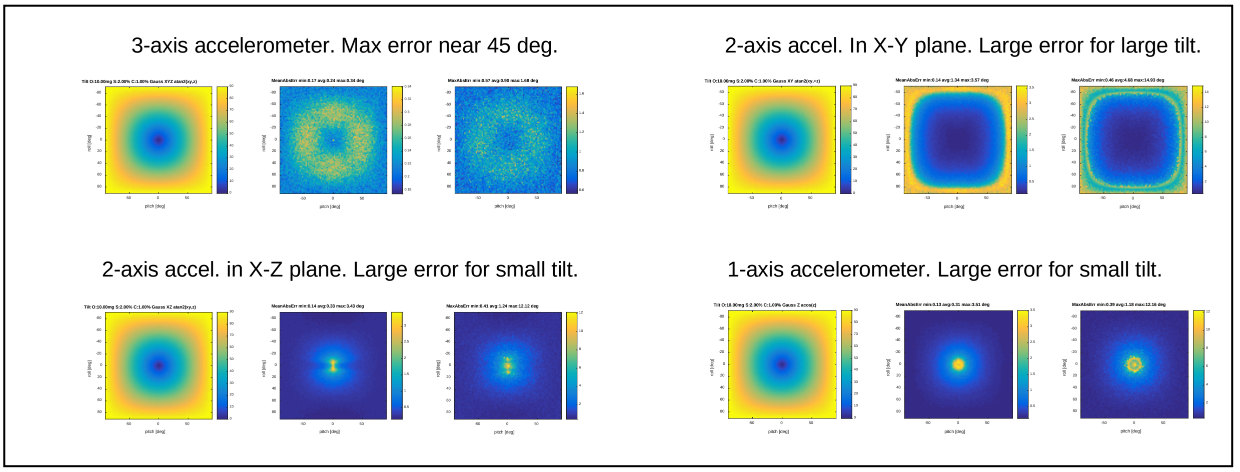
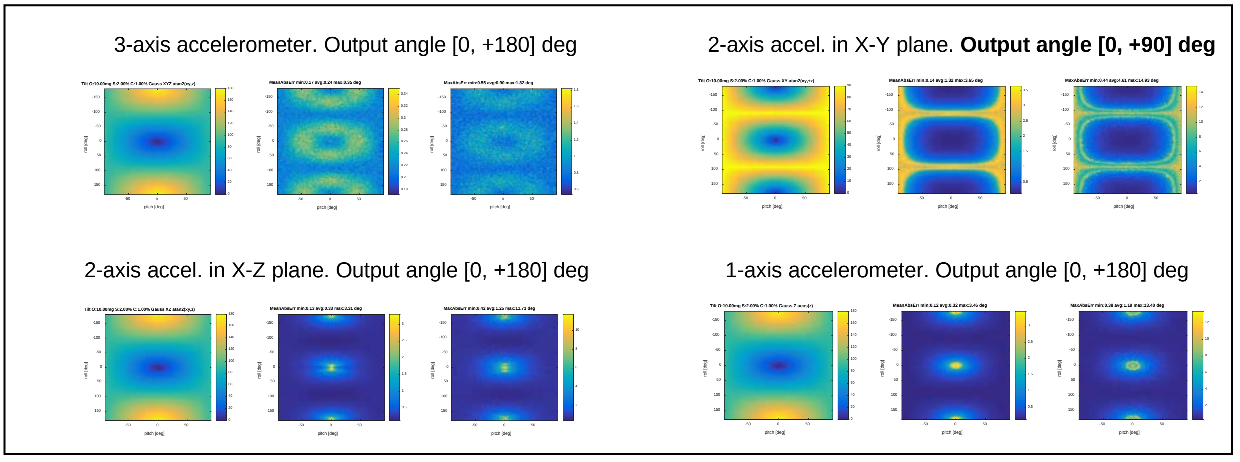


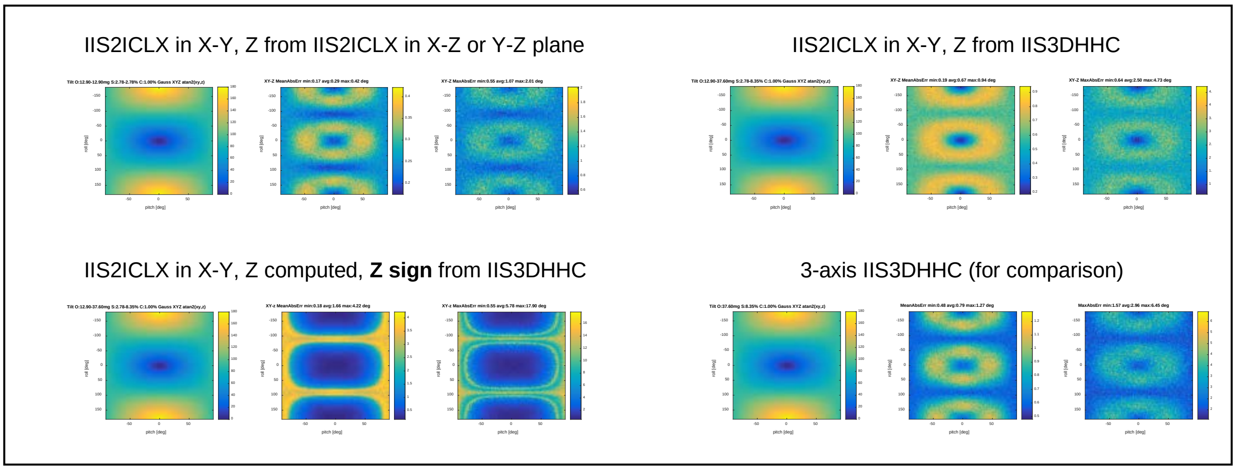
Figure 9. Same as Figure 8 but roll in [-180, +180] degree range, and pitch in [-90, +90] degree range.



Note that the worst case for a 3-axis accelerometer is when tilt angle is halfway (α near 45 degrees), for a 2-axis accelerometer in the X-Y horizontal plane is when tilt angle is large (α above 70 degrees), for a 2-axis accelerometer in the vertical plane and for a 1-axis accelerometer is when tilt angle is small (α below 30 degrees).

Figure 10 shows the performance for a 2-axis accelerometer in the X-Y plane complemented by a secondary accelerometer which provides Z, or the sign of Z when Z is computed from X and Y, in order to support the full tilt range from 0 to 180 degrees. From the point of view of the accuracy, it is better to use the Z-axis from the secondary accelerometer, rather than computing it and setting the correct sign.

Figure 10.Performance of IIS2ICLX in the X-Y plane, complemented by another IIS2ICLX or IIS3DHHC, vs the performance of IIS3DHHC alone. Worst case offset and sensitivity error computed for the common narrower temperature range (-40°C to +85°C).



Other definitions of tilt angle

In the previous paragraphs the tilt was defined as the angle "α" between the vertical gravity vector and the Z-axis, which is the same as the angle between the horizontal plane and the plane of the X-Y axes. It is the same because the horizontal plane is perpendicular to the vertical gravity vector, and the Z-axis is perpendicular to the X-Y axes. The cosine of the angle is the ratio between the gravity vector and its adjacent projection on the Z-axis, the sine of the angle is the ratio between the gravity vector and its perpendicular projection on the plane of the X-Y axes.

- Tilt for any rotation
  - $Acc_Z = 1g \times \cos(\alpha)$ and $Acc_{XY} = 1g \times \sin(\alpha)$ where $Acc_{XY} = \mathrm{sqrt}(Acc_X^2 + Acc_Y^2)$.
  - $\alpha = \arccos(Acc_Z / 1g)$, $\alpha = \arcsin(Acc_{XY} / 1g)$
  - $\alpha = \arctan(Acc_{XY} / Acc_Z)$, $\alpha = \mathrm{arctan2}(Acc_{XY}, Acc_Z)$
- Tilt when rotation is restricted around the Y-axis (moving X and Z axes)
  - $Acc_Z = 1g \times \cos(\alpha)$, $Acc_X = 1g \times \sin(\alpha)$, $Acc_Y = 0$ always (3-axis accelerometer not needed)
  - $\alpha = \arccos(Acc_Z / 1g)$, $\alpha = \arcsin(Acc_X / 1g)$
  - $\alpha = \arctan(Acc_X / Acc_Z)$, $\alpha = \mathrm{arctan2}(Acc_X, Acc_Z)$
- Tilt when rotation is restricted around the X-axis (moving Y and Z axes)
  - $Acc_Z = 1g \times \cos(\alpha)$, $Acc_Y = 1g \times \sin(\alpha)$, $Acc_X = 0$ always (3-axis accelerometer not needed)
  - $\alpha = \arccos(Acc_Z / 1g)$, $\alpha = \arcsin(Acc_Y / 1g)$
  - $\alpha = \arctan(Acc_Y / Acc_Z)$, $\alpha = \mathrm{arctan2}(Acc_Y, Acc_Z)$

The tilt can also be defined as the angle "β" between the horizontal plane and the X-axis, which is the same as the angle between the vertical gravity vector and the plane of the Y-Z axes. It is the same because the vertical gravity vector is perpendicular to the horizontal plane, and the Y-Z axes are perpendicular to X. The cosine of the angle is the ratio between the gravity vector and its adjacent projection on the plane of the Y-Z axes, the sine of the angle is the ratio between the gravity vector and its perpendicular projection on the X-axis.

- Tilt for any rotation
  - $Acc_X = 1g \times \sin(\beta)$ and $Acc_{YZ} = 1g \times \cos(\beta)$ where $Acc_{YZ} = \mathrm{sqrt}(Acc_Y^2 + Acc_Z^2)$
  - $\beta = \arcsin(Acc_X / 1g)$, $\beta = \arccos(Acc_{YZ} / 1g)$
  - $\beta = \arctan(Acc_X / Acc_{YZ})$, $\beta = \mathrm{arctan2}(Acc_X, Acc_{YZ})$
- Tilt when rotation is restricted around the Y-axis or X-axis: same as for angle “α” because $Acc_Z = 1g \times \cos(\beta)$, the other axis measured is $1g \times \sin(\beta)$, and the third axis (the rotation axis) is always 0.

Finally, the tilt can be defined as the angle “γ” between the horizontal plane and the Y-axis, which is the same as the angle between the vertical gravity vector and the plane of the X-Z axes. It is the same because the vertical gravity vector is perpendicular to the horizontal plane, and the X-Z axes are perpendicular to Y. The cosine of the angle is the ratio between the gravity vector and its adjacent projection on the plane of the X-Z axes, the sine of the angle is the ratio between the gravity vector and its perpendicular projection on the Y-axis.

- Tilt for any rotation
  - $Acc_Y = 1g \times \sin(\gamma)$ and $Acc_{XZ} = 1g \times \cos(\gamma)$ where $Acc_{XZ} = \mathrm{sqrt}(Acc_X^2 + Acc_Z^2)$
  - $\gamma = \arcsin(Acc_Y / 1g)$, $\gamma = \arccos(Acc_{XZ} / 1g)$
  - $\gamma = \arctan(Acc_Y / Acc_{XZ})$, $\gamma = \mathrm{arctan2}(Acc_Y, Acc_{XZ})$
- Tilt when rotation is restricted around the Y-axis or X-axis: same as for angle “α” because $Acc_Z = 1g \times \cos(c)$, the other axis measured is $1g \times \sin(b)$, and the third axis (the rotation axis) is always 0.

The last two definitions for tilt (angles “β” and “γ”) are adopted in AN5146, application note for IIS3DHHC 3-axis accelerometer, in the paragraph that discusses its application as high-precision inclinometer, where the corresponding arctan formulas are presented.

As a reference, the matching definitions in UM2277, the user manual of MotionTL tilt measurement software library in X-Cube-MEMS1 expansion package, are as follows:

- Angle “α” between the gravity vector and the Z-axis, or angle between the plane of the X-Y axes and the horizontal plane, is referenced as **phi** = $\arctan(Acc_{XY}, Acc_Z)$ and **gravity inclination** gi = $\arccos(Acc_Z / 1g)$ when rotation can happen around any axis.

- Angle "β" between the gravity vector and the plane of the Y-Z axes, or angle between the X-axis and the horizontal plane, is referenced as **theta** = arctan(Acc_X / Acc_{YZ}) when rotation can happen around any axis.

- Angle "β" is also referenced as **roll** = arcsin(Acc_X / 1g) when rotation is constrained around the Y-axis. The better formula roll = arctan2(Acc_X, Acc_Z) can also be used. Note that this roll is different with respect to roll defined as the rotation around the X-axis in the introduction of this design tip. Also note that angle α and β are equal because rotation is constrained, and for this reason the formulas are the same.

- Angle "γ" between the gravity vector and the plane of the X-Z axes, or angle between the Y-axis and the horizontal plane, is referenced as **psi** = arctan(Acc_Y / Acc_{XZ}) when rotation can happen around any axis.

- Angle "γ" is also referenced as **pitch** = arctan2(Acc_Y, Acc_Z) when rotation is constrained around the X-axis. The other formulas can also be used (see above). Note that this pitch is different with respect to pitch defined as rotation around the Y-axis in the introduction of this design tip. Also note that angle α and γ are equal because rotation is constrained, and for this reason the formulas are the same.

As a reference, the matching definitions in UM2775, the user manual of MotionTL2 2-axis tilt measurement software library in X-Cube-MEMS1 expansion package, are as follows:

- Single plane mode describes a 2-axis accelerometer in the vertical X-Z or Y-Z plane. First sensor axis (Acc_X) is horizontal, second sensor axis (Acc_Y) points up.
    - Vertical plane X-Z with rotation constrained around the Y-axis: the computed angle corresponds to "α" which is equal to "β". Use the formulas presented here but replace Acc_Z with Acc_Y because it is the axis pointing up.
    - Vertical plane Y-Z with rotation constrained around the X-axis: the computed angle corresponds to "α" which is equal to "γ". Use the formulas presented here but replace Acc_Z with Acc_Y and replace Acc_Y with Acc_X because it is the horizontal axis.

- Dual plane mode describes a 2-axis accelerometer in the horizontal X-Y plane.
    - Angle "α" between the plane of the X-Y axes and the horizontal plane is referenced as **phi**.
    - Angle "β" between the X-axis and the horizontal plane is referenced as **theta**.
    - Angle "γ" between the Y-axis and the horizontal plane is referenced as **psi**.

Support material

Documentation
User manual, UM2182, Getting started with MotionAC accelerometer calibration library in X-CUBE-MEMS1 expansion for STM32Cube
User manual, UM2277, Getting started with MotionTL tilt measurement library in X-CUBE-MEMS1 expansion for STM32Cube
User manual, UM2724, Getting started with MotionDI dynamic inclinometer library in X-CUBE-MEMS1 expansion for STM32Cube
User manual, UM2774, Getting started with MotionAC 2-axis accelerometer calibration library in X-CUBE-MEMS1 expansion for STM32Cube
User manual, UM2775, Getting started with MotionTL2 2-axis tilt measurement library in X-CUBE-MEMS1 expansion for STM32Cube
Application note, AN5146, IIS3DHHC: high-resolution, high-stability 3-axis digital accelerometer
Application note, AN5551, Precise and accurate tilt sensing in industrial applications
Technical note, TN0018, Surface mounting guidelines for MEMS sensors in an LGA package
Design tip, DT0053, 6-point tumble sensor calibration
Design tip, DT0058, Computing tilt measurement and tilt-compensated e-compass
Design tip, DT0105, 1-point or 3-point tumble sensor calibration
Design tip, DT0076, Compensating for accelerometer installation error: zeroing pitch and roll for a reference orientation

Revision history

Date	Version	Changes
20-Nov-2020	1	Initial release