

# Strapdown integration for MEMS inertial sensors

Henk Luinge  
infertia.com

August 20, 2025

## Abstract

Applications using MEMS inertial sensors tend to use the movement signals at a much lower rate than the dynamics the movement, especially for movements that include impact or vibrations. This report describes a method for strapdown integration that preserves the integrated change in orientation and velocity the method by calculating it at a high rate and making it available to the user at a much lower rate. Rather than communicating gyroscope and accelerometer signals to the user application using a per-channel aliasing filter, it transmits a change in orientation and a change in acceleration. These incremental pose updates are computed on the IMU at a high internal frequency before being transmitted to the host. Reference code is provided.

## 1 Introduction

We consider applications in which a 3D inertial measurement unit (IMU) is used to track its change in movement. Were not only interested in the momentary sensor signals (angular velocity and accelerometer signals), but also in the integrated quantities: change in orientation, change in velocity and change in position. Typically these quantities are computed on a micro controller using a finite difference integration process also known as strap down integration, see e.g. Woodman [2].

Small errors in sensor data as well as sampling artefacts will cause integration errors, leading to an increasing uncertainty in computed quantities. One specific error cause is loss of integrated values due to a too low strap down computation rate. In 3D movement, the change in orientation and velocity does not follow a simple linear integration, but a differential equation [1]. This means that a simple low-pass averaging filter is not sufficient to reconstruct the movement.

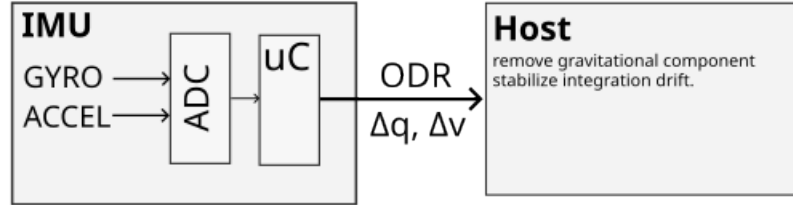
In several applications in which movement is recorded using an IMU, the rate in which the user gets the data (ODR - output data rate) is lower than the dynamics as well as the internal sampling rate.

Two examples:

- A wireless, foot-mounted IMU will record signals with a bandwidth of several hundreds of Hz. When using zigbee or bluetooth, the update rate will often be lower than 100Hz due to constraints in wireless transmission.
- A MEMS IMU mounted on a vibrating mechanical construction can be subjected to vibrations in the kHz range. Because vibratory rotations can be coupled with translations, the strap down integration has to be solved at a rate higher than the Nyquist frequency. If the data is used as part of a controller, the controlled will generally operate on a lower sampling rate.

In the above examples, if the strap down integration is done on the output data rate (ODR), the non commutativity problems can be a substantial part of the integration error, in particular when there is impact or vibrations.

With the amazing stability and linearity of modern IMU's (Bosch, TDK, ST MEMS gyroscopes and accelerometers), dealing with the strap down integration matters to reduce the integration drift and get more reliable movement sensing.



**Figure 1:** Inertial sensors are sampled at a sufficiently high rate at the IMU side and the change in acceleration and velocity is computed. The IMU rate could be 1kHz depending on movement. The computed increments are then used by a host or user at a much lower ODR data rate. The host would take care of removing gravitational acceleration to compute the velocity and removing longer term integration drift.

A solution to the above problems is to compute the strap down integration over a short interval and send the change in orientation and change in velocity to a host, see figure 1. For the foot-mounted application with 10 Hz ODR, this could mean that an internal sampling rate of 1 kHz could be used. Then every 100 samples will be combined to a change in velocity and change in acceleration using strap down integration. These results will then be sent to the host with 10 Hz. Although the user will still have the signals at 10 Hz, there is no loss due to a low-rate strap down integration.

The document describes the implementation of the strap down integration that is divided up in a fast update section and a reduced rate section. The effect of sensing errors on the integration is not treated, nor a derivation of 3D kinematics to a finite difference equation. For reconstructing the velocity we assume a local tangent plane with fixed, known gravitational acceleration.

This report accompanies a jupyter notebook with the implementation. Emphasis is placed on creating an intuition between the equations describing the kinematics and the implementation. The content of the report is not new, it is intended to serve as a guide for implementation. This way of providing change in orientation/velocity rather than angular velocity/acceleration is already used by e.g. iMAR GmbH or Movella.

First, the general, well known strap down integration is introduced, as is proven for inertial sensors on a fixed local tangent plane. Then the equation is chopped up in a part that can be computed on the IMU part and a part that can run on the host.

## 2 IMU sampling

We assume a calibrated inertial measurement unit (IMU) that samples angular velocity and acceleration, each expressed as a vector along an coordinate system (S) attached to the IMU.

The signals are sampled at times  $t_i$  with  $i = 1..ns$ ,  $ns$  represents the number of samples used in the analysis.

Conventions:

- Time is measured at times  $t_i$ . The subscripts  $t_i$ ,  $t$  and  $i$  are used interchangeably.
- The first index (zero) is for the starting position, also in the python reference code. The starting position is assumed to be known. In practise, the first accel/gyro samples (index 0) are used to create an initial orientation estimate and it's uncertainty. Then the second sample (index 1) gives the first integration interval.

## 3 Strap down integration for ideal sensors

Define an ideal gyroscope and accelerometer, not corrupted by noise or calibration errors:

$$\begin{aligned} \mathbf{y}_{gyr,t} &=^S \boldsymbol{\omega}_t \\ \mathbf{y}_{acc,t} &=^S \mathbf{a}_t -^S \mathbf{g}_t \end{aligned} \tag{1}$$

with  $\omega$  the angular velocity,  $a$  the acceleration of the IMU and  $g$  the gravitational component of acceleration.

We assume that the angular velocity  $\boldsymbol{\omega}_t$  is constant during the sampling interval  $t - T$  to and including  $t$ . Then the orientation change between two sampling times  $t_{i-1}$  and  $t_i$  is:

$$\delta \mathbf{q}_i = \begin{bmatrix} \cos(|\boldsymbol{\omega}_i|/2) \\ \sin(\boldsymbol{\omega}_i/2) \end{bmatrix} = \exp(\boldsymbol{\omega}_i/2) = \exp(\mathbf{y}_{gyr,i}/2) \tag{2}$$

The subscript  $i$  is used to indicate the time  $t_i$  and the left superscript specifies the coordinate frame the vector is expressed. The orientation change up to sample  $j$  at time  $t_j$  is obtained by concatenating the orientation differences:

$${}^{LS}\mathbf{q}_j = {}^{LS}\mathbf{q}_0 \cdot \prod_{i=1}^j \delta \mathbf{q}_i \quad (3)$$

or in recursive form, omitting the starting orientation:

$${}^{LS}\mathbf{q}_i = {}^{LS}\mathbf{q}_{i-1} \cdot \delta \mathbf{q}_i \quad (4)$$

The multiplication dot between two quaternions is the quaternion multiplication, implemented in the python/kinematics algebra module, see also e.g. [lisyarus](#). Due to the non-commutativity, equation (3) must be evaluated in order of increasing time.

L represents a non-moving inertial coordinate system. It can be the local tangent plane, or lab coordinate system, or it can be the coordinate system of the device at start of recording.

To compute the velocity, the acceleration is integrated, assuming the acceleration was constant over the period  $t_{i-1}..t_i$ .

$$\begin{aligned} {}^L\mathbf{v}_i &= {}^L\mathbf{v}_{i-1} + T \cdot {}^L\mathbf{a}_i \\ &= {}^L\mathbf{v}_{i-1} + T \cdot {}^L\mathbf{y}_{acc,i} + T \cdot {}^L\mathbf{g} \\ &= {}^L\mathbf{v}_{i-1} + T \cdot {}^{LS}\mathbf{R}_i \cdot {}^S\mathbf{y}_{acc,i} + T \cdot {}^L\mathbf{g} \end{aligned} \quad (5)$$

or in direct form:

$${}^L\mathbf{v}_j = {}^L\mathbf{v}_0 + T \sum_{i=1}^j {}^{LS}\mathbf{R} \cdot \mathbf{y}_{acc,i} + n_s \cdot T \cdot {}^L\mathbf{g} \quad (6)$$

In case the local L frame is defined with Z-axis upwards the gravity is described as:

$${}^L\mathbf{g} = [0 \quad 0 \quad -g_c]^T \quad (7)$$

with  $g_c$  the local gravitational constant, e.g.  $9.81m/s^2$  in the Netherlands.

## 4 Sectioning the integral

Define the following **time instants**:

- High-frequency sampling times  $t_i$  with  $i \in 0..n_t$ . This would typically be the rate of the ADC converter, at a rate of hundreds to perhaps thousands of Hz. L would represent the current time that the pose needs to be calculated.
- the time instants of the output data rate. These are given by  $D_j$  with  $j \in 0..n_j$ . j starts at zero and grows with the duration of the measurement. M would be the pose at current time:  $t_{nt} = D_{nj}$

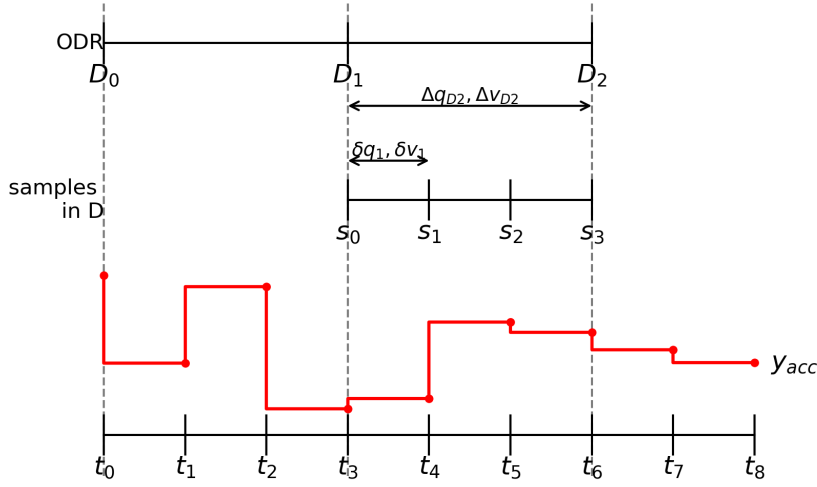
- High frequency sampling times within the interval  $s_k$  with  $k \in 0..n_k$ . The index  $k$  resets with every new ODR interval.

Define the following **intervals**:

- The change in velocity or change in orientation over a short interval from  $s_{k-1}$  to  $s_k$  is given by  $\delta_k$ .
- The change in velocity or change in orientation over an ODR interval from  $D_{j-1}$  to  $D_j$  is given by  $\Delta_j$ .

We assume that when a signal such as accelerometer signal  $y_{acc}$  is measured at some time  $t_i$ , it was constant between  $t_{i-1}$  and  $t_i$ .

A graphical summary of the above is given in figure 2.



**Figure 2:** Definition of timelines and intervals. We seek to capture integrated quantities from each ODR interval ( $D_0, D_1, \dots$ ). The samples within the ODR interval are written as  $s_i$ .

Following from 3, the orientation at some time  $t = t_L$  is chopped up in two multiplication sequences:

$${}^{LS_t} \mathbf{q} = \prod_{j=1}^{n_j} \Delta \mathbf{q}_j = \prod_{j=1}^{n_j} \prod_{k=1}^{n_k} \delta \mathbf{q}_k \quad (8)$$

Similarly following 3, the change in velocity at the end of ODR interval  $j$ :

$$\begin{aligned} {}^L \mathbf{v}_j &= {}^L \mathbf{v}_0 + T \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} {}^{LS} \mathbf{R} \cdot \mathbf{y}_{acc,i} + n_j \cdot n_s \cdot T \cdot {}^L \mathbf{g} \\ &= {}^L \mathbf{v}_0 + T \sum_{j=1}^{n_j} \Delta v_t + n_j \cdot n_s \cdot T \cdot {}^L \mathbf{g} \end{aligned} \quad (9)$$

In recursive format, again omitting the starting velocity:

$${}^L\mathbf{v}_t = {}^L\mathbf{v}_{t-1} + \Delta\mathbf{v}_t + n_s \cdot T \cdot {}^L\mathbf{g} \quad (10)$$

#### 4.1 On-imu calculations

Using the quantities that can be computed on the IMU, the change in orientation in and interval  $j$ , using samples  $i=1..n_s$  within that interval:

$$\begin{aligned} \Delta\mathbf{q}_j &= \prod_{i=1}^{n_s} \exp(T\mathbf{y}_{gyr,i}/2) \\ \Delta v_j &= T \sum_{i=1}^{n_s} {}^L\mathbf{R}_i \cdot \mathbf{y}_{acc,i} \end{aligned} \quad (11)$$

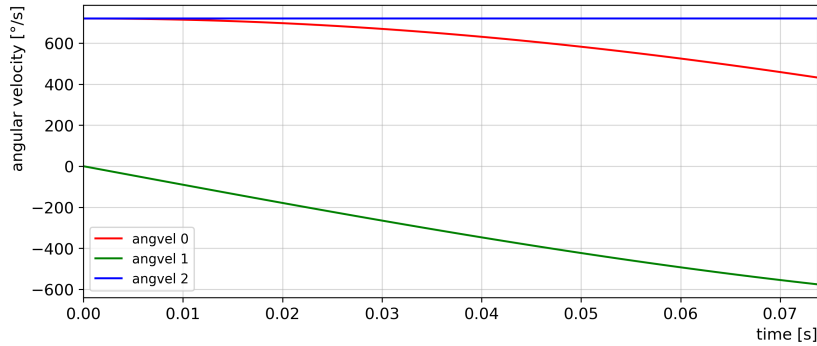
### 5 Benefit

#### 5.1 Coning motion

The performance increase depends on the movement: more 1 dimensional rotations give less noncommutativity errors than movements in which the direction of rotation is changing. As a test case we use coning motion, sometimes called helical motion. The orientation change is defined by

$$\boldsymbol{\omega}(t) = \omega \begin{bmatrix} \sin\theta \cdot \cos(\omega t) \\ -\sin\theta \cdot \sin(\omega t) \\ (1 - \cos\theta) \end{bmatrix} \quad (12)$$

The coning motion is interesting because the angular velocity is constantly changing direction and therefore challenging. For the illustration, we use  $\theta = \pi/2$  to maximize the coning effect and  $\omega = 2[\text{Hz}] \cdot 2\pi$ , corresponding to e.g. a brisk walk.



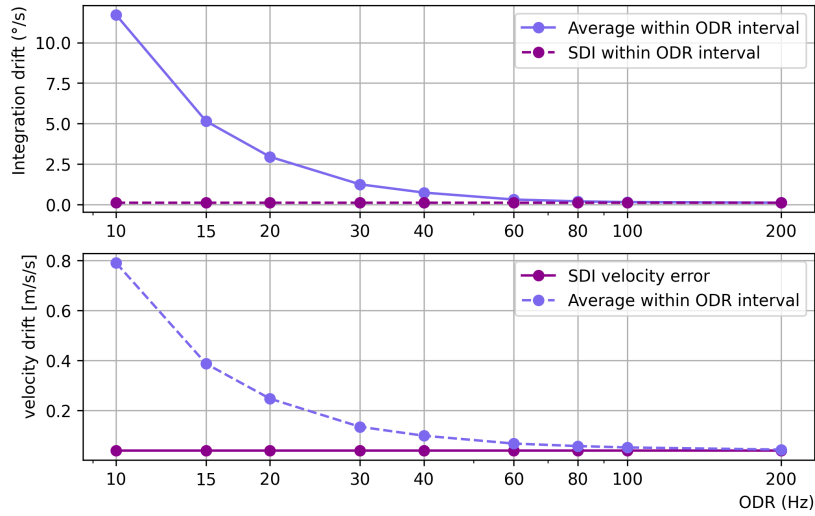
**Figure 3:** Coning angular velocity with duration 4[s],  $\theta = \pi/2$ ,  $\omega = 2 \cdot 2\pi$ , corresponding to 2Hz.

## 5.2 Effect of ODR

The effect of on-IMU strapdown integration is illustrated in figure 4. Two strapdown integration mechanizations are compared:

- on-IMU strap down integration of 800 Hz, then output to the host on the frequency specified by ODR (dashed line). The host then concatenates the change in velocity and change in acceleration as described above.
- A mechanization in which the average gyroscope/accelerometer signal during an ODR sampling period is transmitted to the host. The host then properly integrates the angular velocity and acceleration assuming a constant angular velocity/acceleration during the sampling interval. This case is representative of an anti aliasing filter on each of the sensor signals.

Particularly for update rates below 40Hz, the coning effect can become too large to be useful. With on-IMU strapdown integration, the strap down integration errors are determined by the internal sampling frequency which can in general be chosen sufficiently high - in this case 800Hz.



**Figure 4:** Effect of on-IMU strap down integration is largest for low communication rates.

A 20Hz update rate could be sufficient for visualization purposes or perhaps a motor control application. Still due to sampling alone, the integration drift can be significant. If then inevitable sensors errors such as calibration errors and noise are present, the drift will quickly be a limitation to accurate pose tracking.

## References

- [1] John Bortz. “A New Mathematical Formulation for Strapdown Inertial Navigation”. In: (1970).
- [2] Oliver Woodman. “An introduction to inertial navigation”. In: Aug. 2007.