imucal - A Python library to calibrate 6 DOF IMUs

Arne Küderle[1][¶], Nils Roth[1], Robert Richer[1], and Bjoern M. Eskofier[1]

1 Machine Learning and Data Analytics Lab (MaD Lab), Department Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) ¶ Corresponding author

Summary

Inertial measurement units (IMUs) have wide application areas from human movement analysis to commercial drone navigation. However, to use modern micro-electromechanical systems (MEMS)-based IMUs a calibration is required to transform the raw output of the sensor into physically meaningful units. To obtain such calibrations one needs to perform sets of predefined motions with the sensor unit and then apply a calibration algorithm to obtain the required transformation and correction factors for this unit. The `imucal` library implements the calibration algorithm described by Ferraris et al. (F. Ferraris et al., 1994; Franco Ferraris et al., 1995) and provides functionality to calculate calibration parameters and apply them to new measurements. As typically multiple calibrations are recorded per sensor over time, `imucal` further provides a set of opinionated tools to save, organize, and retrieve recorded calibrations. This helps to make sure that always the best possible calibration is applied for each recording even when dealing with multiple sensors and dozens of measurements.

Statement of Need

When working with MEMS-based IMUs, calibrations are required to correct sensor errors like bias, scaling, or non-orthogonality of the included gyroscope and accelerometer as well as to transform the raw sensor output into physical units. While out-of-the-box factory calibrations and self-calibration procedures have become better over the past years, high precision applications still benefit from manual calibrations temporally close to the measurement itself. This is because the parameters of the sensor can change because of the soldering process, change in humidity, or temperature. Also, it could simply change over time as the silicon ages. Various algorithms and protocols exist to tackle this issue. To calibrate the accelerometer, most of them require the sensor unit to be placed in multiple well-defined orientations relative to gravity. To calibrate the gyroscope, the sensor is required to be rotated either with known angular rate or by a known degree. From the data recorded in the different phases of the calibrations, the correction and transformation parameters for a specific sensor can be calculated.
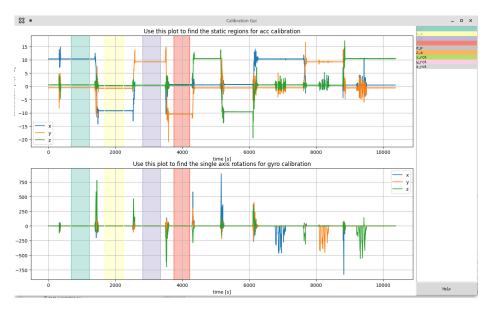
While multiple of these procedures have been published in literature, for example (F. Ferraris et al., 1994; Kozlov et al., 2014; Qureshi & Golnaraghi, 2017; Scapellato et al., 2005; Skog & Händel, 2006; Tedaldi et al., 2014; Zhang et al., 2009), no high-quality code implementations are available for most of them. Existing implementations that can be found on the internet are usually "one-off" scripts that would require adaptation and tinkering to make them usable for custom use cases. Further, many practical aspects of calibrating IMUs, like which information needs to be stored to make sure that a calibration can be correctly applied to a new recording, are usually not discussed in research papers or are not easily available online.

Hence, well-maintained reference implementations of algorithms, clear guidelines, and informal guides are needed to make the procedure of creating and using calibrations easier.

Provided Functionality

With `imucal` and its documentation, we address all the above needs and hope to even further expand on that in the future based on community feedback. The library provides a sensor-agnostic object-oriented implementation of the calibration algorithm by Ferraris et al. Franco Ferraris et al. ([1995](#)) and functionality to apply it to new data. Further, we provide a simple GUI interface to annotate recorded calibration sessions ([Figure 1](#)).

Figure 1: Screenshot of the GUI to annotate recorded Ferraris sessions. Each region corresponds to one of the required static positions or rotations. The annotation is performed using the mouse with support for keyboard shortcuts to speed up some interactions.

When working with sensors and multiple calibrations, storing and managing them can become complicated. Therefore, `imucal` also implements a set of opinionated helpers to store the calibrations and required metadata as *.json* files and functions to retrieve them based on sensor ID, date, type of calibration, or custom metadata.

While `imucal` itself only implements a single calibration algorithm so far, all tools in the library have been designed with the idea of having multiple algorithms in mind. Therefore, the provided structure and base classes should provide a solid basis to implement further algorithms, either as part of the library itself or as part of custom software packages.

To ensure that all the provided tools are usable, the documentation contains full guides on how to practically perform the calibration and then use `imucal` to process the recorded sessions. As much as possible, these guides include informal tips to avoid common pitfalls.

The `imucal` library has been used extensively in the background of all movement research at the Machine Learning and Data Analytics Lab (MaD Lab) to calibrate our over 100 custom and commercial IMU sensors. Therefore, we hope this library can bring similar value to research groups working on IMU-related topics.

Availability

The software is available as a pip installable package (`pip install imucal`) and via [GitHub](#). Documentation can be found on [Read the Docs](#).

Acknowledgments

Ferraris, F., Gorini, I., Grimaldi, U., & Parvis, M. (1994). Calibration of three-axial rate gyros without angular velocity standards. *Sensors and Actuators A: Physical*, *42*(1-3), 446–449. https://doi.org/10.1016/0924-4247(94)80031-6

Ferraris, Franco, Grimaldi, U., & Parvis, M. (1995). Procedure for effortless in-field calibration of three-axial rate gyro and accelerometers. *Sensors and Materials*, *7*(5), 311–330. https://myukk.org/SM2017/article.php?ss=10210

Kozlov, A., Sazonov, I., & Vavilova, N. (2014). IMU calibration on a low grade turntable: Embedded estimation of the instrument displacement from the axis of rotation. *2014 International Symposium on Inertial Sensors and Systems (ISISS)*, 1–4. https://doi.org/10.1109/ISISS.2014.6782525

Qureshi, U., & Golnaraghi, F. (2017). An Algorithm for the In-Field Calibration of a MEMS IMU. *IEEE Sensors Journal*, *17*(22), 7479–7486. https://doi.org/10.1109/JSEN.2017.2751572

Scapellato, S., Cavallo, F., Martelloni, C., & Sabatini, A. M. (2005). In-use calibration of body-mounted gyroscopes for applications in gait analysis. *Sensors and Actuators A: Physical*, *123–124*, 418–422. https://doi.org/10.1016/j.sna.2005.03.052

Skog, I., & Händel, P. (2006). Calibration of a MEMS inertial measurement unit. *In Proc. XVII IMEKO WORLD CONGRESS, (Rio de Janeiro.*

Tedaldi, D., Pretto, A., & Menegatti, E. (2014). A robust and easy to implement method for IMU calibration without external equipments. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 3042–3049. https://doi.org/10.1109/ICRA.2014.6907297

Zhang, H., Wu, Y., Wu, W., Wu, M., & Hu, X. (2009). Improved multi-position calibration for inertial measurement units. *Measurement Science and Technology*, *21*(1), 015107. https://doi.org/10.1088/0957-0233/21/1/015107