Self-Calibration of Accelerometer Arrays

P. Schopp, H. Graf, W. Burgard, *Fellow, IEEE*, and Y. Manoli, *Senior Member, IEEE*

***Abstract*—A gyroscope-free inertial measurement unit (GF-IMU) employs solely accelerometers to capture the motion of a body in the form of its linear and angular acceleration as well as its angular velocity. For that, multiple transducers are fixed at distinct locations of the body that together form an accelerometer array. To accurately estimate the motion, the poses of the sensors, i.e., their positions and orientations, must be known precisely. Unfortunately, these parameters are typically hard to assess. Current state-of-the-art calibration methods are able to reconstruct the geometrical sensor configuration based on a set of motion data and corresponding acceleration measurements. However, to impose a reference motion on the sensor array and to capture that motion with the necessary accuracy requires sophisticated laboratory equipment. In this work, we present a method to estimate the transducer poses using only their own measurements without depending on reference motion data. It is based on an iterative graph-optimization that considers both the sensor poses and the motion as target variables. Initially, this results in infinitely many solutions. We reduce the solutions to only one global optimum by explicitly modeling the used triple-axis accelerometers as sensor triads and furthermore taking the temporal dependence of the acceleration samples into account. We compare our method to the conventional calibration using reference data in terms of its estimation accuracy. Furthermore, we analyze the convergence properties of our method by evaluating its tolerance to initial pose deviations. For both, we use synthetic and experimental data recorded on a 3-D rotation table.**

***Index Terms*—Gyroscope-free inertial measurement unit, Accelerometer array, Self-calibration, Reference-free, Calibration**

I. Introduction

A conventional inertial measurement unit (IMU) consists of three accelerometers and three gyroscopes to capture the motion of the body in the form of its linear acceleration and its angular velocity [1], [2]. In contrast to this, a gyroscope-free inertial measurement unit (GF-IMU) comprises only accelerometers to determine the motion. The approach exploits the fact that the acceleration field of a body becomes inhomogeneous if an angular motion is present. By taking samples of the acceleration field with multiple transducers at distinct locations of the body the linear as well as the angular motion can be reconstructed. The positions of the sensors must remain constant relative to each other. Therefore, a GF-IMU is also referred to as an accelerometer array.

There are various reasons for employing accelerometer arrays as a GF-IMU. E.g., the approach allows to measure the angular acceleration directly, without the differentiation of the angular velocity. Thus, the noise-amplification problem caused by the differentiation can be avoided [3]. In [4], this advantage was used to construct a surgical tool that detects the tremor of its user. Furthermore, accelerometer arrays can be used to implement a GF-IMU that features a lower power consumption compared to a conventional IMU. To measure the angular velocity micromechanical gyroscopes sense the Coriolis force that acts on an oscillating proof mass. Because of the required mechanical excitation, they are considered as active devices [5], [6]. In contrast to this, accelerometers can be constructed as passive devices [7], [8]. As a result, the power consumption of a gyroscope is $\sim$20 times higher compared to an accelerometer. Thus, a GF-IMU is a competitive alternative to a conventional IMU especially if there is the demand for a low power consumption. Accelerometer arrays were successfully employed in various applications, e.g., in human motion analysis [9], automotive navigation [10], or robotics [11].

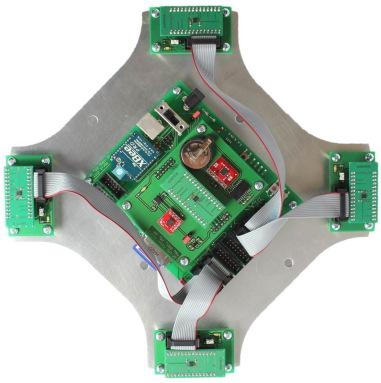
To infer the motion, the positions and the orientations of the accelerometers, i.e., their poses, must be known precisely. Even small errors within the assumed poses can cause large deviations in the estimation of the motion as the measured accelerations are interpreted falsely [12]. In certain application scenarios the sensor poses are available in the form of a construction plan. Still, the poses are only known up to a certain precision due to unavoidable tolerances of manufacturing process and an imperfect mounting of the sensors. However, the transducer poses can be recovered by calibration. This is valuable especially when building a prototype because the accelerometers can be attached on the body without defining exact mountings beforehand. State of the art calibration methods impose a known reference motion on the accelerometer array while recording the transducer measurements. A numerical optimization estimates the transducer poses based on the values of the reference motion and the acceleration measurements. The main challenge of these methods is to obtain values for the reference motion. Either, there is a mechanical manipulator that imposes the motion on the accelerometer array very accurately such that the predefined motion can serve as a reference or an additional measurement system records the imposed motion. E.g., we use a 3-D rotation table to impose a defined motion on our prototype (Fig. 1). However, such kind of reference systems are expensive laboratory facilities. Furthermore, they may not even be applicable if the geometric size of the accelerometer array becomes large. From an economic point of view, a calibration run for each produced unit using a reference measurement system equals additional production costs. In this context, we consider the necessity to acquire the imposed motion as the most significant drawback of all currently known approaches to a calibration of accelerometer

P. Schopp and H. Graf are with the Fritz Huettinger Chair of Microelectronics, Department of Microsystems Engineering – IMTEK, University of Freiburg, Freiburg, Germany (e-mail: patrick.schopp@imtek.uni-freiburg.de)

W. Burgard is with the Laboratory for Autonomous Intelligent Systems, Department of Computer Science, University of Freiburg, Germany

Y. Manoli is with the Fritz Huettinger Chair of Microelectronics, Department of Microsystems Engineering – IMTEK, University of Freiburg, Germany, and also with Hahn-Schickard, Wilhelm-Schickard-Straße 10, 78052 Villingen-Schwenningen, Germany

(a) Rotation table.



(b) GF-IMU prototype.

Fig. 1. Experimental setup. The 3-D rotation table (a) allows to execute a predefined motion. At the same time it allows to record this motion with high accuracy by means of its integrated instrumentation. The custom GF-IMU prototype (b) contains 5 accelerometer triads in total of which 4 are placed on satellite boards.

arrays. To overcome this shortcoming, this work presents a self-calibration for accelerometer arrays. The method estimates the poses of the transducers using only their own measurements without the need for any external reference.

This paper is organized as follows. In the next section we recapitulate the fundamentals of the GF-IMU approach. Using the derived sensor equations we summarize the approaches to a conventional calibration with reference data in Section III. In Section IV, we present our approach to a self-calibration. We analyze its estimation accuracy and precision in Section V and discuss its convergence properties in Section VI. In Section VII, we conclude this paper by comparing the properties of the derived self-calibration to those of a calibration with reference data before we give an outlook to our future research in Section VIII.

II. GF-IMU Fundamentals

There is a large variety of approaches for inferring the motion from the measurements of an accelerometer array. However, they are all based on the same mechanical equation that allows to compute the acceleration measurement of the used transducers given the motion. In the following, we state this fundamental equation and recapitulate the control system formulation for arbitrary accelerometer arrays, which we derived in our previous work [13].

An accelerometer array can be regarded as a control system by defining the motion of the body as the internal state of the system and the accelerometer measurements as its outputs. The discrete-time state-space formulation of a general nonlinear control system Σ can be given by

$$\Sigma : \begin{aligned} \boldsymbol{x}_{i+1} &= \boldsymbol{f}\left(\boldsymbol{x}_i,\, \boldsymbol{u}_i\right) + \boldsymbol{w}_i \\ \boldsymbol{z}_i &= \boldsymbol{h}\left(\boldsymbol{x}_i\right) + \boldsymbol{v}_i \end{aligned} \tag{1}$$

with $\boldsymbol{x} \in \mathbb{R}^n$ being the state of the system, $\boldsymbol{z} \in \mathbb{R}^m$ its outputs, and vector $\boldsymbol{u} \in \mathbb{R}^l$ a known control input. The process model $\boldsymbol{f}$ propagates the state $\boldsymbol{x}$ from time-step i to the next time-step $i+1$. The output $\boldsymbol{z}$ is given by the observation model $\boldsymbol{h}$ and depends only on the state $\boldsymbol{x}$. The random vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ are error terms representing the uncertainty of the models. Both are assumed to be drawn from normal distributions with zero-mean and the covariances ${}^{\mathrm{w}}\Sigma$ and ${}^{\mathrm{v}}\Sigma$. In the following, we derive all parts of this description for the GF-IMU control system by means of the equation that calculates the acceleration ${}^{\mathrm{r}}\boldsymbol{a}$ at position $\boldsymbol{r}$ with respect to the body frame.

Assuming that the body is rigid, i.e., $\boldsymbol{r}$ is constant over time, ${}^{\mathrm{r}}\boldsymbol{a}$ can be computed as

$${}^{\mathrm{r}}\boldsymbol{a} = {}^{\mathrm{l}}\boldsymbol{a} + \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r})\,, \tag{2}$$

where the motion of the body is described by its linear acceleration ${}^{\mathrm{l}}\boldsymbol{a}$, its angular velocity $\boldsymbol{\omega}$ and its angular acceleration $\dot{\boldsymbol{\omega}}$. The linear acceleration ${}^{\mathrm{l}}\boldsymbol{a}$ covers all accelerations that are homogeneous throughout the body. E.g., the acceleration caused by gravity is also part of ${}^{\mathrm{l}}\boldsymbol{a}$.

As required by (1), the state $\boldsymbol{x}$ must include all variables that cause a change of the outputs $\boldsymbol{z}$ as it is the only input to the observation model $\boldsymbol{h}$. As $\boldsymbol{r}$ is constant, we define the state of the system as

$$\boldsymbol{x} \equiv {}^{\mathrm{b}}\boldsymbol{x} = \begin{bmatrix} {}^{\mathrm{l}}\boldsymbol{a}^T & \boldsymbol{\omega}^T & \dot{\boldsymbol{\omega}}^T \end{bmatrix}^T, \tag{3}$$

where the superscript b indicates that ${}^{\mathrm{b}}\boldsymbol{x}$ is the motion of the body.

The sensors sample the acceleration ${}^{\mathrm{r}}\boldsymbol{a}$ at their position with respect to the sensor frame. Thus, to compute the scalar acceleration measurement ${}^{\mathrm{sm}}a$ of a transducer, ${}^{\mathrm{r}}\boldsymbol{a}$ is first rotated into the sensor frame to the acceleration ${}^{\mathrm{s}}\boldsymbol{a}$ before a sensor model maps ${}^{\mathrm{s}}\boldsymbol{a}$ to the scalar transducer measurement ${}^{\mathrm{sm}}a$. As an efficient, singularity-free representation of the sensor orientation we use a quaternion $\boldsymbol{q}$ to describe the rotation from the body to the sensor frame. We encapsulate the steps to rotate a vector $\boldsymbol{v}$ by a unit-quaternion $\boldsymbol{q}$ by means of the rotation operator $R_{\boldsymbol{q}}(\boldsymbol{v})$. The required mathematical tools for a quaternion rotation can be found in [14]. As modern accelerometers are designed for a linear sensor behavior, it is common to use a linear mapping between ${}^{\mathrm{s}}\boldsymbol{a}$ and ${}^{\mathrm{sm}}a$, which can be given by

$$\begin{aligned} {}^{\mathrm{sm}}a &= \boldsymbol{s} \cdot {}^{\mathrm{s}}\boldsymbol{a} + {}^{\mathrm{o}}a \\ &= \boldsymbol{s} \cdot R_{\boldsymbol{q}}({}^{\mathrm{r}}\boldsymbol{a}) + {}^{\mathrm{o}}a, \end{aligned} \tag{4}$$

where the scalar multiplication $\boldsymbol{s} \cdot {}^{\mathrm{s}}\boldsymbol{a}$ maps ${}^{\mathrm{s}}\boldsymbol{a}$ onto the sensitivity axis $\boldsymbol{s}$ of the accelerometer relative to the sensor frame. The

signal offset ${}^{\mathrm{o}}a$ accounts for the constant measurement error of the accelerometer. The parameters of (4) can be divided into the ones that describe the placement of the transducer and the ones that originate from the sensor characteristics. The placement of the sensor is defined by its position $\boldsymbol{r}$ and its orientation given by $\boldsymbol{q}$. The combination of both is also referred to as the pose $P = \{\boldsymbol{q}, \boldsymbol{r}\}$ of the transducer. The parameters of the linear sensor model are the sensitivity axis $\boldsymbol{s}$ and the signal offset ${}^{\mathrm{o}}a$ of the transducer. We will denote their combination as $C = \{\boldsymbol{s}, {}^{\mathrm{o}}a\}$ and refer to them as the sensor parameters. Here, we want to mention that $\boldsymbol{q}$ and $\boldsymbol{s}$ are redundant parameters in (4). We can account for every change in the orientation with a corresponding rotation of the sensitive axis. It is equally possible to describe the sensitivity axis $\boldsymbol{s}$ in the body frame thereby omitting a separate parameter for the orientation of the sensor. However, we separate the parameters to have a clear distinction between the geometry dependent pose P and the transducer dependent parameters C.

To construct the observation model $\boldsymbol{h}$ of the accelerometer array we introduce the notation ${}^{\mathrm{sm}}a_j = {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}, P_j, C_j)$ to indicate that the function ${}^{\mathrm{s}}h$ implements the calculations of (4) to compute the measured acceleration ${}^{\mathrm{sm}}a_j$ of the jth single-axis transducer. The function depends on ${}^{\mathrm{b}}\boldsymbol{x}$ and is parameterized by the pose P_j and the constants C_j of the jth sensor. For an accelerometer array that employs m single-axis sensors, observation model $\boldsymbol{h}$ is a combination of m sensor equations, which leads to

$$\boldsymbol{h}\left(\boldsymbol{x}\right) \equiv {}^{\mathrm{sa}}\boldsymbol{h}({}^{\mathrm{b}}\boldsymbol{x}_i, P_{1:m}, C_{1:m}) = \begin{bmatrix} {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}_i, P_1, C_1) \\ \vdots \\ {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}_i, P_m, C_m) \end{bmatrix} \tag{5}$$

where the superscript sa stands for sensor array. The observation $\boldsymbol{z}$ holds the measured accelerations and is defined as

$$\boldsymbol{z} \equiv {}^{\mathrm{sm}}\boldsymbol{a} = \begin{bmatrix} {}^{\mathrm{sm}}a_1 & \dots & {}^{\mathrm{sm}}a_m \end{bmatrix}^T \tag{6}$$

The error term $\boldsymbol{v}_i$ accounts for the noise of the sensors. Hence, we define its covariance as ${}^{\mathrm{v}}\Sigma \equiv \mathrm{diag}([{}^{\mathrm{s}}\sigma_1^2 \ \dots \ {}^{\mathrm{s}}\sigma_m^2])$, where ${}^{\mathrm{s}}\sigma_{1:m}^2$ are the variances of each individual transducer.

As the process model $\boldsymbol{f}$ gives the state at the next time-step based on the current state it describes how the state variables affect each other. The inputs $\boldsymbol{u}$ allow to describe the mechanics of a known external excitation. For the GF-IMU we define $\boldsymbol{f}$ as

$$\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i) \equiv {}^{\mathrm{sa}}\boldsymbol{f}({}^{\mathrm{b}}\boldsymbol{x}_i) = \begin{bmatrix} {}^{\mathrm{l}}\boldsymbol{a}_i \\ \boldsymbol{\omega}_i + \dot{\boldsymbol{\omega}}_i \Delta t_i \\ \dot{\boldsymbol{\omega}}_i \end{bmatrix} \tag{7}$$

where Δt_i is the time between i and $i+1$. In contrast to the general formulation, the process model ${}^{\mathrm{sa}}\boldsymbol{f}$ does not include a control input $\boldsymbol{u}$ because we do not know the stimulation of the motion. Instead, the external excitation is solely modeled by the error term $\boldsymbol{w}_i$. The process model implements the assumption that the accelerations ${}^{\mathrm{l}}\boldsymbol{a}$ and $\dot{\boldsymbol{\omega}}$ remain constant between two time-steps. This allows us to integrate the angular acceleration as $\dot{\boldsymbol{\omega}}_i \Delta t_i$ to compute the angular velocity at $i+1$. However, the assumption of constant acceleration will not hold true once there is an external excitation of the motion. Thus, the error term $\boldsymbol{w}$ accounts for ${}^{\mathrm{b}}\boldsymbol{a}$ and $\dot{\boldsymbol{\omega}}$ not being constant over time and for the error of the integration term $\dot{\boldsymbol{\omega}}_i \Delta t_i$ caused by a non-constant $\dot{\boldsymbol{\omega}}$. We define the covariance ${}^{\mathrm{w}}\Sigma$ corresponding to $\boldsymbol{w}$ as

$${}^{\mathrm{w}}\Sigma_i \equiv \Delta t_i^2 \begin{bmatrix} \sigma_a^2 \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_{\mathrm{int}}^2 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \sigma_{\dot{\omega}}^2 \boldsymbol{I} \end{bmatrix}, \tag{8}$$

where $\boldsymbol{I}$ denotes an identity matrix and $\boldsymbol{0}$ a block of zeros of size 3×3. The variances σ_a^2 and $\sigma_{\dot{\omega}}^2$ model the change of the accelerations ${}^{\mathrm{l}}\boldsymbol{a}$ and $\dot{\boldsymbol{\omega}}$, while σ_{int}^2 describes the errors of the integration term $\dot{\boldsymbol{\omega}}_i \Delta t_i$. As we do not want to make an assumption about how the true course of $\dot{\boldsymbol{\omega}}$ deviates from $\dot{\boldsymbol{\omega}}_i$, we model the error of constant angular acceleration and the integration error as independent of each other. Therefore, we set the respective correlation terms in ${}^{\mathrm{w}}\Sigma$ to zero. Furthermore, we assume that the described errors grow linearly with Δt for reasonable sampling frequencies. Following the law of error propagation this introduces the term Δt^2 to the calculation of ${}^{\mathrm{w}}\Sigma$. This formulation has the advantage that once we set the values for variances σ_a^2, $\sigma_{\dot{\omega}}^2$, and σ_{int}^2 we can recompute ${}^{\mathrm{w}}\Sigma$ for varying sampling frequencies.

The system description given by the formulas (3), (5), and (7) is valid for a general accelerometer array with m sensors and an arbitrary placement of the transducers. However, not all configurations result in a proper GF-IMU. The array must suffice some requirements regarding the number of sensors and their arrangement. In [13], we showed that the state of the system described by (3), (5), and (7) is locally weakly observable with at least three accelerometer triads, i.e., three groups of three orthogonally mounted single-axis sensors, commercially available as triple-axis sensors. Thus, $m = 9$ or more acceleration sensitive axis are necessary to directly capture the entire motion, which ensures a drift-free estimation of the angular velocity. Furthermore, the positions of the triads must span a surface. If they reside on a line the rotation around that axis cannot be detected.

III. Conventional Calibration

The calibration of an accelerometer used in a conventional IMU estimates the sensor parameters C but does not consider the pose P of the transducer. Its position is mostly defined to be the center of the body and its orientation to be aligned with the frame of the body. Compared to this, present calibration methods for accelerometer arrays infer both transducer poses P as well as the parameters of the sensor model C for each employed sensor. Based on a set of reference motion data ${}^{\mathrm{b}}\boldsymbol{x}_{1:n}^{\mathrm{R}}$ and corresponding acceleration measurements ${}^{\mathrm{sm}}a_{1:n}$ the methods set-up a least-squares optimization with the parameter sets P and C as target variables. The optimization has the form

$$\underset{P,\,C}{\arg\min} \sum_{i=1}^{n} \left({}^{\mathrm{sm}}a_i - {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}_i^{\mathrm{R}},\, P,\, C)\right)^2 \tag{9}$$

where n is the number of samples. In (9), the parameters P and C are optimized such that the sensor model ${}^{\mathrm{s}}h$ best explains the measurements ${}^{\mathrm{sm}}a_{1:n}$. The optimization is performed individually for each one of the m sensors. Throughout this

paper, we refer to methods based on reference data of the motion as conventional calibration.

Although the published approaches to a conventional calibration for accelerometer arrays can be summarized by the formulation given in (9), the actual implementations differ in their methodology. As the orientation and the sensitive axis of the transducer are redundant parameters most implementations describe the sensitive axis in the body frame, which allows to drop the orientation from the equations. As an alternative, we present a model for sensor triads in Appendix A, which aligns the sensitive axes with the sensor frame and thus removes the redundancy of both parameters. Furthermore, some methods split the calibration into two parts [15]–[18]. They estimate the orientation, the sensitive axis, and the offset in a static part without any rotation. Here, the transducer array is placed into several known poses. As the rotational state variables $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are zero in (2) the reference acceleration can be calculated based on the current orientation and the gravity constant. Subsequently, the position of the accelerometer is estimated in a dynamic part at a defined rotation. In contrast to this, the entire parameter set can also be estimated in only one single step [12], [19]. By rearranging (4), the regression problem of (9) can be formulated as a simple matrix equation, which can be solved efficiently using a singular value decomposition.

The formulation of (9) clearly shows that the present calibration methods for accelerometer arrays depend on the exact knowledge of the reference motion ${}^{\mathrm{b}}\boldsymbol{x}^{\mathrm{R}}_{1:n}$. Even if available, errors in ${}^{\mathrm{b}}\boldsymbol{x}^{\mathrm{R}}_{1:n}$ lead to errors in the estimated sensor poses P and parameters C. As discussed in the introduction, this is a severe burden for the calibration of accelerometer arrays, because it requires a highly accurate reference measurement system. We address this problem by presenting a self-calibration for accelerometer arrays. Thus, there is no need to know ${}^{\mathrm{b}}\boldsymbol{x}^{\mathrm{R}}_{1:n}$ to perform the calibration.

In fact, there are already interesting approaches to a reference-free calibration of the sensor parameters C of single accelerometer triads. Without any additional acceleration driven by motion, the Euclidean length of the measurement of the accelerometer triad must be the earth gravitation: 1 g. These assumptions are used to formulate an error function, which is minimized in an optimization process. Its target values are the parameters C of the linear sensor model. The only input to the optimization are measurements taken at multiple, arbitrary orientations within the gravitational field. The solutions proposed in literature differ in the applied sensor model and also in the applied optimization method. Methods that have been successfully applied include the Newton-Raphson method [20], maximum likelihood estimation [21], or Levenberg-Marquardt minimization [22], [23]. Recently, we presented an iterative solution in [24] where we employed an UKF to recursively optimize the sensor parameters of an accelerometer. Furthermore, there are a few approaches towards a reference-free calibration of the accelerometer poses P. E.g., Nilsson *et. al.* present a method to align an array of multiple IMUs based on the gravity vector [25]. However, the method does not consider the positions of the accelerometers. Kozlov *et. al.* are concerned with the simultaneous estimation of the position displacement of an IMU on a single axis rotation table together with the calibration parameters of a conventional IMU [26]. However, the method is tailored for a rotation around a single axis and hence cannot recover a tree-dimensional sensor position.

In summary, there are already solutions for a reference-free calibration of C and the orientation of the transducers. However, none of the existing reference-free calibration methods include the three-dimensional positions of the sensors. We advance the state of the art here by presenting calibration method for the entire geometrical setup of the accelerometer array. Thus, the sensor poses P consisting of the orientation and the position are estimated without requiring reference data of the imposed motion.

IV. Graph Optimization

A conventional calibration estimates the parameters of a proposed model based on known system states and corresponding observations. In contrast to this, self-calibration inherently contains a circular dependency because both the model parameters and the system states are unknown, i.e., the system states are required to estimate the model parameters and in turn, the model parameters are necessary to compute the system states. Hence, both must be estimated jointly. Only having an estimate of both allows to compute expected acceleration measurements using (4) and thus to determine the sensor poses according to the sensor model. However, if the optimization only considers the sensor equations as in (9) either the parameters or the assumed state can always be adjusted to account for a certain measurement. This results in infinitely many solutions.

We address this problem by introducing prior knowledge of the system to the optimization in order to impose constraints on the possible solutions such that the optimization converges to the correct geometrical setup and the correct motion. Concretely, we derive constraints from the following facts: Firstly, the accelerometers are organized as triads as we only employ triple-axis accelerometers. We assume that the sensor triads are calibrated, thus, the parameters C are known for each of the three sensitive axes. Secondly, the time-intervals between the acceleration measurements are known because we use a defined sampling rate.

A. Nonlinear Least-Squares Optimization

A general least squares optimization can be represented by a factor graph in an intuitive and effective way as the structure of the graph highlights the relations between the variables [27]–[29]. A factor graph is a bipartite graph, i.e., there exist exactly two types of nodes and there can only be a link from a node of the first type to a node of the second type. Specific to a least squares optimization, nodes of the first type embody the unknown variables $\boldsymbol{x}$, which are the target of the optimization. Nodes of the second type represent constraints between the variables that originate from observations $\boldsymbol{z}$. Each one refers to a certain error function $\boldsymbol{e}$ being of the form

$$\boldsymbol{e}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij}) = \boldsymbol{z}_{ij} - \boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{10}$$

where $\boldsymbol{h}$ computes the expected observation. Furthermore, each constraint node corresponds to an information matrix Σ_{ij}^{-1}

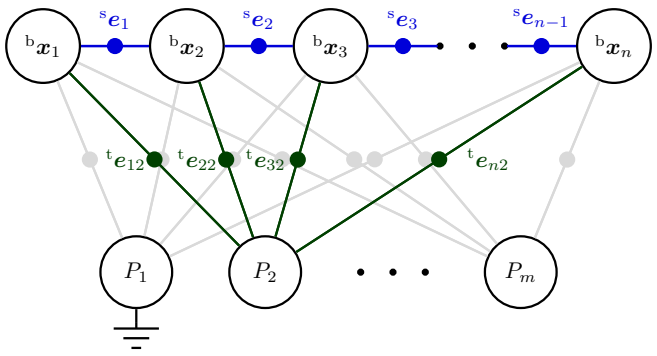


Fig. 2. The factor graph representing the relations of the variables of the self-calibration of accelerometer arrays. The nodes drawn as circles represent the variables, which are the target of the optimization, whereas the small filled nodes embody the constraints between them.

reflecting the uncertainty of the constraint between the nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Assuming all observations to be independent and their errors to be normally distributed a joint objective function F can be composed, which is the negative log likelihood F of all observations and is given by

$$F(\boldsymbol{X}) = \sum_{\{i,j\}\in\mathcal{C}} \boldsymbol{e}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij})^T \, \Sigma_{ij}^{-1} \, \boldsymbol{e}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij}) \tag{11}$$

where $\mathcal{C}$ is the set of pairs of indices $\{i, j\}$ for which there is a observation $\boldsymbol{z}_{ij}$ relating the variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Vector $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T & \dots & \boldsymbol{x}_n^T \end{bmatrix}^T$ holds all target variables. Finding the minimum of the joint error function F in the form

$$\boldsymbol{X}^* = \arg\min_{\boldsymbol{X}} F(\boldsymbol{X}) \tag{12}$$

leads to the optimal values $\boldsymbol{X}^*$, i.e., the values that maximize the likelihood of all observations [30]. After the construction of the graph the optimization is independent of the problem setting. It can be solved iteratively using nonlinear least-squares techniques such as the Gauss-Newton or the Levenberg-Marquardt algorithm. Popular examples that can be solved using a graph-optimization can be found in robotics [31] or in computer vision [32].

B. Graph Structure and Error Models

The factor graph we derived for the self-calibration of accelerometer arrays is shown in Fig. 2. In the following we will discuss its structure and show how it embeds the prior knowledge of known sensor triads and consecutive samples.

Say we have a set of n acceleration measurements recorded by m sensor triads. Therefore, the nodes representing the variables of the optimization consist of n nodes for the motion vectors ${}^{\mathrm{b}}\boldsymbol{x}_{1:n}$ and another m nodes representing the poses $P_{1:m}$ of the transducer triads. The triad with pose $P_j = \{\boldsymbol{q}_j, \boldsymbol{r}_j\}$ took an acceleration sample ${}^{\mathrm{t}}\boldsymbol{z}_{ij}$ corresponding to the motion ${}^{\mathrm{b}}\boldsymbol{x}_i$. This implies the following constraint between P_j and ${}^{\mathrm{b}}\boldsymbol{x}_i$. Given P_j and ${}^{\mathrm{b}}\boldsymbol{x}_i$ we can compute the expected measurement using three times the sensor model given in (4). Up to a certain error due to sensor noise, the expected observation must match the recorded observation ${}^{\mathrm{t}}\boldsymbol{z}_{ij}$ once P_j and ${}^{\mathrm{b}}\boldsymbol{x}_i$ are equal to the true pose and motion. Thus, we define the error function for constraints between ${}^{\mathrm{b}}\boldsymbol{x}_i$ and P_j as

$$\begin{aligned} {}^{\mathrm{t}}\boldsymbol{e}_{ij} &\equiv {}^{\mathrm{t}}\boldsymbol{e}\left({}^{\mathrm{b}}\boldsymbol{x}_i,\, P_j,\, {}^{\mathrm{t}}\boldsymbol{z}_{ij}\right) \\ &= {}^{\mathrm{t}}\boldsymbol{z}_{ij} - \begin{bmatrix} {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}_i,\, P_j, C_1) \\ {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}_i,\, P_j, C_2) \\ {}^{\mathrm{s}}h({}^{\mathrm{b}}\boldsymbol{x}_i,\, P_j, C_3) \end{bmatrix}, \end{aligned} \tag{13}$$

where the superscript t stand for triad. By describing all three sensitive axes by only one pose P_j we reduce the possible solutions in the following way: All orientations and positions are valid for each axis but there is a fixed relation between three axes, which is encoded by the sensor parameters $C_{1:3}$. To relate the results of ${}^{\mathrm{t}}\boldsymbol{e}$ with the noise of the sensors we define the information matrix ${}^{\mathrm{t}}\Sigma_{ij}^{-1}$ corresponding to ${}^{\mathrm{t}}\boldsymbol{e}$ to be the inverse of the covariance of the triad measurement ${}^{\mathrm{t}}\Sigma_{ij} = \mathrm{diag}([{}^{\mathrm{s}}\sigma_1^2 \; {}^{\mathrm{s}}\sigma_2^2 \; {}^{\mathrm{s}}\sigma_3^2])$. The variances ${}^{\mathrm{s}}\sigma_{1:3}^2$ are the measurement variances of each one of the sensitive axes. For the sensors of our prototype (Fig. 1b) we experimentally determined those to be $4 \cdot 10^{-4}\,(\mathrm{m/s^2})^2$.

As we can adjust the sampling rate we know the time Δt_i between two consecutive states ${}^{\mathrm{b}}\boldsymbol{x}_i$ and ${}^{\mathrm{b}}\boldsymbol{x}_{i+1}$. For a given Δt_i the process model ${}^{\mathrm{sa}}\boldsymbol{f}$ in (7) describes how the state evolves to the next time-step. Thus, we use it to define the error function ${}^{\mathrm{s}}\boldsymbol{e}$ for a constraint between two consecutive states as

$$\begin{aligned} {}^{\mathrm{s}}\boldsymbol{e}_i &\equiv {}^{\mathrm{s}}\boldsymbol{e}\left({}^{\mathrm{b}}\boldsymbol{x}_i, {}^{\mathrm{b}}\boldsymbol{x}_{i+1}, \boldsymbol{0}\right) \\ &= {}^{\mathrm{b}}\boldsymbol{x}_{i+1} - {}^{\mathrm{sa}}\boldsymbol{f}({}^{\mathrm{b}}\boldsymbol{x}_i) = \begin{bmatrix} {}^{\mathrm{l}}\boldsymbol{a}_{i+1} - {}^{\mathrm{l}}\boldsymbol{a}_i \\ \boldsymbol{\omega}_{i+1} - (\boldsymbol{\omega}_i + \dot{\boldsymbol{\omega}}_i \Delta t_i) \\ \dot{\boldsymbol{\omega}}_{i+1} - \dot{\boldsymbol{\omega}}_i \end{bmatrix}, \end{aligned} \tag{14}$$

where we treat any difference between ${}^{\mathrm{b}}\boldsymbol{x}_{i+1}$ and ${}^{\mathrm{sa}}\boldsymbol{f}({}^{\mathrm{b}}\boldsymbol{x}_i)$ as error. To satisfy the form given by (10) we define the observation between ${}^{\mathrm{b}}\boldsymbol{x}_i$ and ${}^{\mathrm{b}}\boldsymbol{x}_{i+1}$ to be $\boldsymbol{0}$ at all time-steps. As we use the process model ${}^{\mathrm{sa}}\boldsymbol{f}$ to construct the error function ${}^{\mathrm{s}}\boldsymbol{e}$ we can use the covariance ${}^{\mathrm{w}}\Sigma$ of the error term $\boldsymbol{w}$ to compute the information matrix ${}^{\mathrm{s}}\Sigma_i^{-1}$ by evaluating its inverse as ${}^{\mathrm{s}}\Sigma_i^{-1} = ({}^{\mathrm{w}}\Sigma)^{-1}$. The values for σ_a^2, $\sigma_{\dot{\omega}}^2$, and σ_{int}^2 of ${}^{\mathrm{w}}\Sigma$ can be regarded as tuning parameters of the self-calibration that reflect the dynamics of the motion according to the following rationals. The error functions $({}^{\mathrm{l}}\boldsymbol{a}_{i+1} - {}^{\mathrm{l}}\boldsymbol{a}_i)$ and $(\dot{\boldsymbol{\omega}}_{i+1} - \dot{\boldsymbol{\omega}}_i)$ prevent the optimization to generate large jumps in ${}^{\mathrm{l}}\boldsymbol{a}_{1:n}$ and $\boldsymbol{\omega}_{1:n}$. Still, we choose relatively large values for σ_a^2 and $\sigma_{\dot{\omega}}^2$ because values that are too small suppress any change of the motion from one time-step to another. In contrast to this, we choose σ_{int}^2 to be small because we want the optimization to trust the integration constraint as it represents the link between $\boldsymbol{\omega}$ and its time derivative $\dot{\boldsymbol{\omega}}$. For the evaluations of our method in this paper we set $\sigma_a^2 = 10^3\,(\mathrm{m/s^2})^2$, $\sigma_{\dot{\omega}}^2 = 10^3\,(\mathrm{rad/s^2})^2$, and $\sigma_{\mathrm{int}}^2 = 10^{-7}\,(\mathrm{rad/s})^2$.

All constraints we introduced so far are made between the variables without comparing to an absolute reference. Thus, the optimized graph is only consistent relatively. We can alter the values of the variables without generating an error as long as we preserve their relation. Therefore, we fix the pose of one triad, i.e., we set its position and orientation to certain values and exclude it from the optimization. In the graph of Fig. 2 this is the triad with pose P_1. As a result, the poses of the free triads are estimated relative to this fixed triad. Thus,

to fix the pose of a triad equals the definition of the body frame. At the first glance, this may appear as a drawback of the self-calibration. However, the conventional calibration involves the same issue. Independent of the type of reference system, we have to define the frame of the body as soon as we compute the reference motion. E.g., the rotation table only acquires the angles of its axes. To compute the motion of the accelerometer array we have to define the center of the body and its orientation in relation to the axes of the table. Thus, both calibrations with or without reference data generate pose estimations relative to a defined frame.

We have now completely described the graph we want to optimize in order to solve the self-calibration problem. Thus, we can refine $F(\boldsymbol{X})$ of (11) as

$$F(\boldsymbol{X}) = \sum_{i=1}^{n}\sum_{j=1}^{m} {}^{\mathrm{t}}\boldsymbol{e}_{ij}^{T}\, {}^{\mathrm{t}}\Sigma_{ij}^{-1}\, {}^{\mathrm{t}}\boldsymbol{e}_{ij} + \sum_{i=1}^{n-1} {}^{\mathrm{s}}\boldsymbol{e}_{i}^{T}\, {}^{\mathrm{s}}\Sigma_{i}^{-1}\, {}^{\mathrm{s}}\boldsymbol{e}_{i} \quad (15)$$

where n is the number of samples and m is the number of triads. Vector $\boldsymbol{X}$ is composed as

$$\boldsymbol{X} = \begin{bmatrix} {}^{\mathrm{b}}\boldsymbol{x}_{1:n} & P_{j\in\mathcal{T}} \end{bmatrix}^{T} \quad (16)$$

where $\mathcal{T}$ is the set of indices of the triads that are not fixed.

C. Optimization on Manifolds

$F(\boldsymbol{X})$ employs unit quaternions to represent the orientations of the transducer triads to circumvent singularities that result from minimal representations such as Euler angles. They can be written as a 4-dimensional vector with a unit-constraint, which is why they can be regarded as a manifold of $\mathbb{R}^4$, i.e., not all vectors in $\mathbb{R}^4$ are valid unit quaternions, only the ones that suffice the unit-constraint. However, conventional iterative least-squares optimization algorithms operate in the Euclidean space, i.e., all values are valid. When applied directly they ignore the constraints within an over-parameterized orientation parameterization like quaternions or rotation matrices, which introduces errors to the solution. An elegant way to overcome this problem is to exploit that manifolds may not be globally Euclidean but can be regarded as Euclidean locally [33]. The idea is that the optimization operates on a minimal, Euclidean parameterization of the manifold [34], [35]. After each iteration, the resulting increment $\Delta\boldsymbol{b}^*$ is applied to the current hypothesis of the manifold $\breve{\boldsymbol{b}}$. Recent frameworks for graph optimizations already implement this methodology such that the optimization of the manifolds is transparent to the user [34], [36], [37]. For an arbitrary manifold, all the user has to specify is a function that adds an increment $\Delta\boldsymbol{b}$ to the manifold $\boldsymbol{b}$. This function is called the box-plus operator $\boxplus$ and is a applied as

$$\boldsymbol{b}^* = \breve{\boldsymbol{b}} \boxplus \Delta\boldsymbol{b}^*, \quad (17)$$

where the * marks the optimized values and $\breve{}$ the hypothesis of the last iteration [31], [34]. A quaternion $\boldsymbol{q} = \begin{bmatrix} w & \boldsymbol{v}^T \end{bmatrix}^T$ consists of a scalar part w and a vector part $\boldsymbol{v}$ and, as we are using only unit-quaternions, has the constraint $||\boldsymbol{q}|| = 1$. One way of implementing the $\boxplus$ operator for unit-quaternions is to treat the increment as the vector part $\boldsymbol{v}$ of a quaternion and map the increment to the original quaternion representation by computing the real part v such that it suffices $||\boldsymbol{q}|| = 1$. For the box-plus operator this results to

$$\boldsymbol{q}^* = \breve{\boldsymbol{q}} \boxplus \Delta\boldsymbol{q}^* = \breve{\boldsymbol{q}} \circ \begin{bmatrix} \sqrt{1 - ||\boldsymbol{v}^*||^2} \\ \boldsymbol{v}^* \end{bmatrix} \quad (18)$$

where first $\Delta\boldsymbol{q}^*$ is converted to a full quaternion before both rotations are combined by the quaternion multiplication denoted by the operator $\circ$. A further option to implement the $\boxplus$ operator is to treat the increment as a axis-angle representation where the length of the vector encodes the angle of the rotation [34], [38].

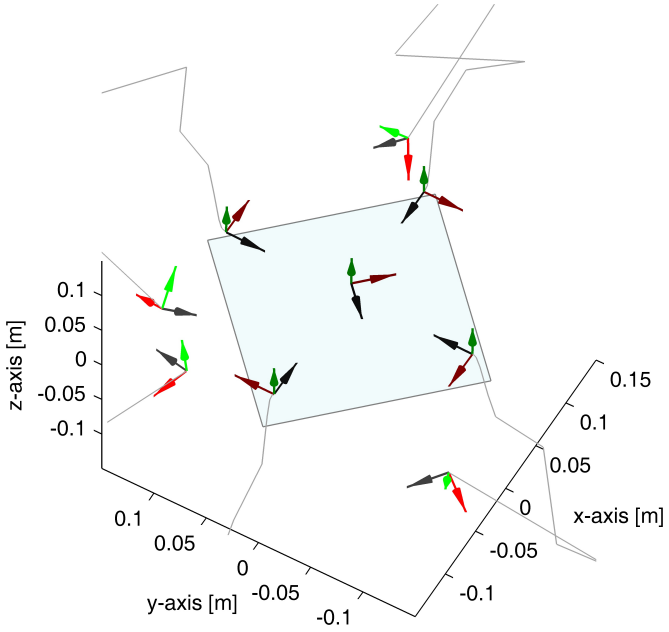


Fig. 3. The poses of the accelerometer triads during the optimization. The initial and final poses are shown in light and dark red-black-green arrow triplets. The intermediate positions are visualized as gray lines. The square area represents the metal plate on which the transducers are mounted.

D. Implementation

Our implementation of the graph optimization is based on the C++ framework g^{2}o [37]. Figure 3 and 4 depict a typical optimization of both the poses of the triads and the motion. For this example we recorded acceleration data from a motion which we imposed on the prototype using the 3-D rotation table of Fig. 1a. To minimize $F(\boldsymbol{X})$ we chose the Levenberg-Marquardt algorithm. We fixed the pose of the triad that is located at the center of the metal plate. Figure 3 shows how the sensor triads are pushed apart by the first optimization steps. At the same time, the amplitude of the motion hypothesis rises (Fig. 4). This behavior is reasonable because the observations feature high acceleration amplitudes but the initial hypothesis of the motion is zero for all dimensions. The overall error can only be reduced by either increasing the distances between the triads or by raising the amplitude of the motion. We can also observe the effect of the constraints that connect consecutive states. All hypotheses feature a smooth sequence of the angular velocity and its gradient matches $\dot{\boldsymbol{\omega}}$. A part of the path of the triads appears to be random. We reason this with the gradient

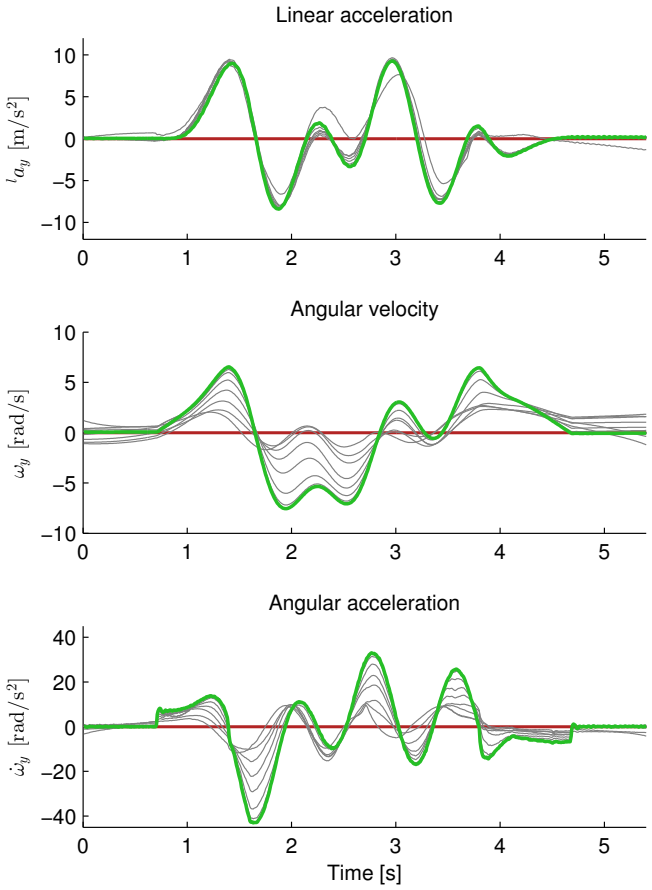


Fig. 4. The estimated motion during the optimization process. For clarity, we only show the y-axis of ${}^{\mathrm{I}}\boldsymbol{a}$, $\boldsymbol{\omega}$, and $\dot{\boldsymbol{\omega}}$. The initial hypothesis is drawn in dark red, the final, optimized motion in light green. The thin, gray lines represent the intermediate hypothesis.

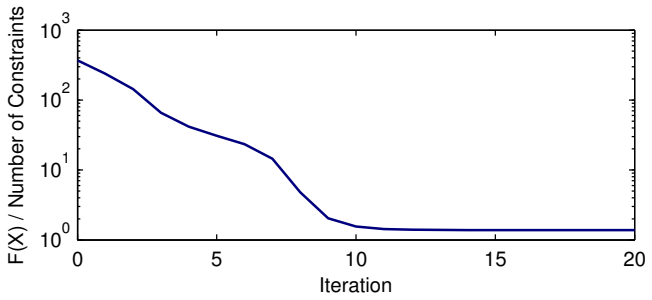


Fig. 5. The value of $F(\boldsymbol{X})$ divided by the total number of constraints over the number of iterations.

that is constructed by the Levenberg-Marquardt algorithm at every iteration. Partly, the random motion results from a small overshoot along its direction. Furthermore, it is computed based on noisy measurement data causing it to be slightly inaccurate. After a few iterations the algorithm finds an optimum for the poses of the triads as well the motion. Figure 5 shows how the value of $F(\boldsymbol{X})$ rapidly decreases during the first 10 iterations and then settles to a constant level.

The computationally crucial part of the Levenberg-Marquardt algorithm is to solve a set of linearized equations at each iteration, which has a size equal to the number of free variables. As the self-calibration estimates the triad poses and the motion jointly the number of free variables increases proportionally to the number of acceleration samples used for the self-calibration. The linearized system of equations is solved using a Cholesky decomposition. Thus, the time complexity of our method is $O((n+m)^3)$ in general. However, modern software implementations exploit the structure of the problem and achieve an acceptable run-time even for high-dimensional problems [37]. The motion of the example has a length of 5.4 s and the accelerations are sampled at 125 Hz. Thus, the optimization includes 675 states to optimize. As ${}^{\mathrm{b}}\boldsymbol{x}$ is of dimension 9 this results to 6075 free variables for the motion only. Despite the high number of variables one optimization step consumed only 0.15 s, which amounts to 3 s for the complete self-calibration (desktop computer, Athlon 64, 2 GHz, single-threaded).

V. Calibration Accuracy and Precision

In this section, we discuss the accuracy and the precision of the pose estimations generated by the self-calibration using both experimental and simulated measurement data. We also compare the estimated poses to those resulting from the conventional calibration.

Before we evaluate the experiments we briefly describe the measurement setup and methodology we used. The GF-IMU prototype (Fig. 1b) incorporates five accelerometer triads (Bosch Sensortec, BMA180). The acceleration measurements are read out by a microcontroller which is part of a mainboard at the center of an aluminum carrier plate. One of the accelerometer triads is situated at the center of the mainboard while the other four are placed on satellite boards. In addition to the acceleration sensors, the prototype contains a triple-axis gyroscope (InvenSense, ITG-3200).

We determined the reference poses of the accelerometer triads using a conventional calibration. The method we previously used to calibrate the transducer array treated all sensitive axes individually and estimated those with respect to the body frame [12]. However, we are interested in the pose of the sensor triad because we want to compare the estimates of the self-calibration to the ones resulting from the calibration with reference data. The sensor model, which we derive in Appendix A, aligns the sensitive axes with the sensor frame. By that, it eliminates the redundancy of the pose and the sensitive axes. The model describes a triad by means of its pose P and the joint parameters of its sensitive axes ${}^{\mathrm{t}}C$. Employing it for a conventional calibration allows for a direct comparison of the accuracy of the estimated poses. Thus, for each triad we estimate P and ${}^{\mathrm{t}}C$ by means of an iterative optimization in the form of

$$\underset{P,\,{}^{\mathrm{t}}C}{\arg\min} \sum_{i=1}^{n} \left({}^{\mathrm{t}}\boldsymbol{z}_i - {}^{\mathrm{t}}\boldsymbol{h}({}^{\mathrm{b}}\boldsymbol{x}_i^{\mathrm{R}},\, P,\, {}^{\mathrm{t}}C)\right)^2 \tag{19}$$

where the function ${}^{t}\boldsymbol{h}({}^{\mathrm{b}}\boldsymbol{x}_i^{\mathrm{R}},\, P,\, {}^{\mathrm{t}}C)$ computes the expected observation of all three sensitive axes given the reference motion vector ${}^{\mathrm{b}}\boldsymbol{x}_i^{\mathrm{R}}$. To maximize its accuracy we used a motion of ∼2 min recorded at 250 Hz. This generated a large data-set, which effectively suppresses errors due to sensor noise. The resulting positions of the triads are summarized in Table I.

TABLE I
REFERENCE POSITIONS OF THE ACCELEROMETER TRIADS.

Positions (cm)	Triad No. 1	2	3	4	5
x	0.07	0.28	11.93	− 0.27	− 11.94
y	− 0.24	11.89	− 0.28	− 11.94	0.29
z	3.87	2.47	2.50	2.51	2.51

To compare the estimation errors of the conventional and the self-calibration we used the same motion as for the example of the last section as an evaluation data-set, i.e., we performed both types of calibrations using this evaluation data-set and later compared it to the poses, which we estimated using the conventional calibration together with the reference data-set (Tab. I). The motion for the evaluation has a duration of 5.4 s and we used a sampling frequency of 125 Hz. Thus, the evaluation data-set has only 2.25 % of sampling points compared to the one we used to generate the reference poses.

For the self-calibration at least one triad must be fixed to define the body frame. Here, we fixed the pose of triad no. 1 located at the center of the metal plate (see Fig. 1b and 3) to the one we obtained from the reference calibration (Table I).

We compared the accuracy of the pose estimations with simulated measurements and with experimental data. To generate the synthetic measurements we used the motion provided by the rotation table and applied the observation model together with the poses P and the sensor parameters ${}^{\mathrm{t}}C$ we obtained by the reference calibration. Subsequently, we added random numbers to the expected observations drawn from a normal distribution with zero mean and a standard deviation equal to the one we experimentally determined for the sensors. Thus, the simulations resemble the experiment, i.e., the same sensor setup undergoes the same motion, but with ideal sensors without any systematic deviation from their linear model. Furthermore, the simulations allow to analyze the statistics of the errors. For that, we executed each simulation trial 100 times with new random numbers as sensor noise at each simulation run. Assuming the distribution of the errors to be a Gaussian we computed the mean as well as the standard deviation as measure of accuracy and precision.

For both the conventional and the self-calibration we evaluated the position error, i.e., the distance of estimated position to the reference position, and the orientation error, i.e., the angle between the estimated orientation to the reference orientation. To express the error of one calibration trial by only two numbers, we computed the mean error of all triads, however, without considering the fixed triad. Table II summarizes the mean errors and the standard deviations the simulations (top) as well as the errors of the experiments (bottom).

The calibrations based on experimental data result to mean errors that are by a factor of $\sim$100 higher compared to the mean errors resulting from simulated measurements. This large deviation cannot be justified by the standard deviations of the respective calibrations. We reason the deviation by an imperfect description of the sensors. Either the sensor parameters ${}^{t}C$ may contain errors or the transducers do not follow a completely linear behavior. Effects like non-linearity, cross-axes sensitivity, or bias stability are not covered by the linear model. However, besides their magnitude the error values of the experiments confirm the simulations in terms of their ranking.

TABLE II
THE ERROR OF THE ESTIMATED POSITIONS AND ORIENTATIONS OF BOTH SELF-CALIBRATION AND CONVENTIONAL CALIBRATION. WE ASSUME THE ERRORS OF THE SIMULATIONS TO BE NORMALLY DISTRIBUTED AND EVALUATE THEIR MEAN AND THEIR STANDARD DEVIATION BASED ON 100 SIMULATION TRIALS.

Simulation	Position $[10^{-6}\mathrm{m}]$	Orientation $[10^{-3\circ}]$
Self-Calibration	5.42 ($\pm$ 81.84)	1.00 ($\pm$ 14.93)
Conventional Calibration		
Table	1.43 ($\pm$ 32.43)	0.40 ($\pm$ 9.20)
Gyro - calibrated	4.17 ($\pm$ 75.66)	0.80 ($\pm$ 16.66)
Gyro - uncalibrated	31.40 ($\pm$ 1827.58)	7.34 ($\pm$ 487.46)
Experiment	**Position $[10^{-6}\mathrm{m}]$**	**Orientation $[10^{-3\circ}]$**
Self-Calibration	653.08	338.07
Conventional Calibration		
Table	556.96	200.60
Gyro - calibrated	977.58	203.63
Gyro - uncalibrated	2287.93	849.22

To compute the reference data for the conventional calibration we used either the measurements provided by the rotation table or the measurements of the center accelerometer triad together with the gyroscope. The values in Table II indicate that the calibration with reference data originating from the table feature the smallest error of both position and orientation in terms of mean and standard deviation. This makes sense because the calibration has access to motion values. In contrast to this, the self-calibration jointly estimates the triads poses and the motion. Thus, there is less information available and therefore sensor noise has a stronger effect on the pose estimates. In simulation, the standard deviation is approximately 2 times larger compared to the conventional calibration. In experiment, we obtain a positioning error of 0.56 mm and 0.65 mm and an orientation error of 0.20 ° and 0.34 ° for the conventional calibration and the self-calibration, respectively.

The conventional calibration treats the reference data as ground truth. Therefore, errors within the provided reference motion influence the accuracy of the pose estimates. To illustrate the effect we used the reference data provided by the center accelerometer triad and the gyroscope in two ways. In both cases we applied a calibration step to the data of the center accelerometer. However, for the first evaluation, we applied a calibration step on the gyroscope data and for the second evaluation we used the gyroscope data directly, without preprocessing. When we calibrate the gyroscope data we minimize the systematic errors and noise is the dominant error source. In this case, the conventional calibration and the self-calibration show approximately the same error within the pose estimation both in terms of mean and standard deviation. When we use uncalibrated gyroscope data the reference data contains systematic errors. This generates a large error within the pose estimates as we assume a different motion than there actually was for the acceleration samples.

The analysis presented here does not consider the influence

of the motion nor the geometrical sensor configuration on the estimation accuracy. To analyze their impact requires further elaborate investigations, which will be part of our future research. However, we can draw the following conclusions. The self-calibration achieves an estimation accuracy and precision comparable to the conventional calibration. If there is high quality reference data of the motion available the conventional calibration achieves betters results. However, the self-calibration shows a better performance than a conventional calibration supplied with motion data that contains systematic errors.

VI. Convergence

The derived self-calibration consists of a minimization of the objective function $F(\boldsymbol{X})$. In contrast to a linear optimization problem, $F(\boldsymbol{X})$ contains local minima because of the nonlinear models we used to construct it. Thus, there are motions and sensor poses with a minimal value of $F(\boldsymbol{X})$ compared to other possible solutions in a certain neighborhood of the search space. However, in contrast to the global minimum of $F(\boldsymbol{X})$ they do not correspond to the true motion and the true sensor poses. As we discuss in the following, whether the optimization falls into a local minimum mainly depends on the initial poses of the triads and on the motion from which the acceleration data was sampled from. For reasonable initial poses and motions convergence is not a major issue. Still, we want to give an insight into the mechanisms of the optimization and derive a selection scheme for reasonable initial values. First, we explain the different kinds of local minima before we evaluate the influences on the convergence of the estimates to the global optimum.

A. Local Minima

We separate the local minima into two different types. Local minima, which consist of an only partly erroneous motion estimate, and local minima corresponding to a completely diverged estimate. The first one is related to the direction of the estimated rotation. If the initial poses of the triads deviate severely from their true poses the optimization may misinterpret the accelerations and estimate a wrong sign for the angular velocity at the very first iterations. This can hardly be corrected by an iterative optimization algorithm because the quadratic dependency of $\boldsymbol{\omega}$ in $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r})$ does not allow for a change of the sign. Figure 6 shows an example of an optimization that is trapped in such a local minimum. Such kinds of estimation errors also affect the estimation of the triad poses. How strong this effect is depends on how often there is a sign estimation error compared to the whole motion. Single errors have a rather small impact. Typically, the resulting position errors are within a millimeter range. However, if the relation of sign errors compared to the correct estimation is too large, the entire estimation may diverge to one of the following local minima, which we classify as the second type of local minima. Table III summarizes such minima in terms of the final positions of the triads and the final angular motion, i.e., the angular acceleration and velocity. They can be easily detected after the optimization because the values of the estimates are not reasonable. This may not be the case for the first type of

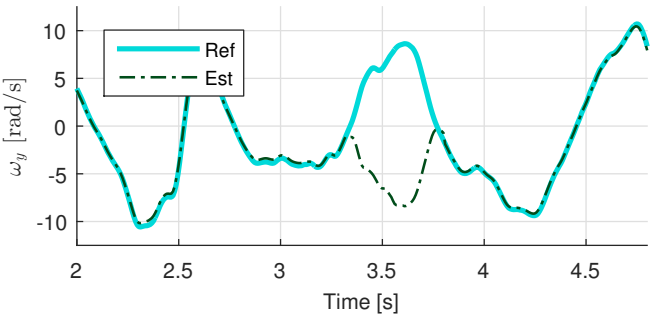


Fig. 6. Example where the motion estimate is partly trapped in a local minimum. We recorded the acceleration values while we moved the prototype by hand and used the gyroscope to compute the reference motion (Ref). The estimate (Est) results from the self-calibration using the acceleration measurements of all available triads. For clarity, only the y-axis of the angular velocity is depicted.

TABLE III
Local minima with completely diverged estimate.

	Type 1	Type 2	Type 3
Triad positions	Zero	Infinity	Zero
Angular motion	Zero	Zero	Infinity

minima featuring partially trapped motion estimates. For this type, a straight forward solution is to rerun the optimization with the estimated poses as initial values. As they are close to the true triad configuration it is very likely that the optimization now correctly estimates the direction of the rotation.

B. Initial Values

The initial values have a strong impact on whether the optimization converges to the global optimum. The linear acceleration of the body can be initialized almost arbitrarily. Because at least one triad has a fixed pose there is a unique relation from its measurements to the linear acceleration. In our experiments, we set the initial estimate of ${}^{1}\boldsymbol{a}_{1:n}$ to zero. The estimate quickly converges to the correct linear acceleration within a few iterations (cf. Fig. 4). For the angular motion there is only one reasonable choice for the initial values. If we do not have any previous knowledge about the motion both $\boldsymbol{\omega}_{1:n}$ and $\dot{\boldsymbol{\omega}}_{1:n}$ must be set to zero to avoid to predetermine the sign of the angular velocity. Otherwise, this could result in the optimization to partially converge to the local minimum of the first type or to diverge. Whereas we have clear rules for the initial values of the motion, there is no certain set of poses that safely serves as initial set for all motions and triad configurations. In general, the best choice for the initial poses is to set them as close possible to the real poses of the triads. However, the exact poses are not available. Still, most applications allow to make a reasonable guess of the poses, e.g., based on the construction plan. For the following analysis, we evaluate how much initial deviation from the true poses the optimization tolerates, i.e., how good the initial guess has to be, before it diverges.

For the evaluations we collected measurements imposing 4 different motions. The first one was a synthetic motion for which we generated segments of sinusoidal angular velocity

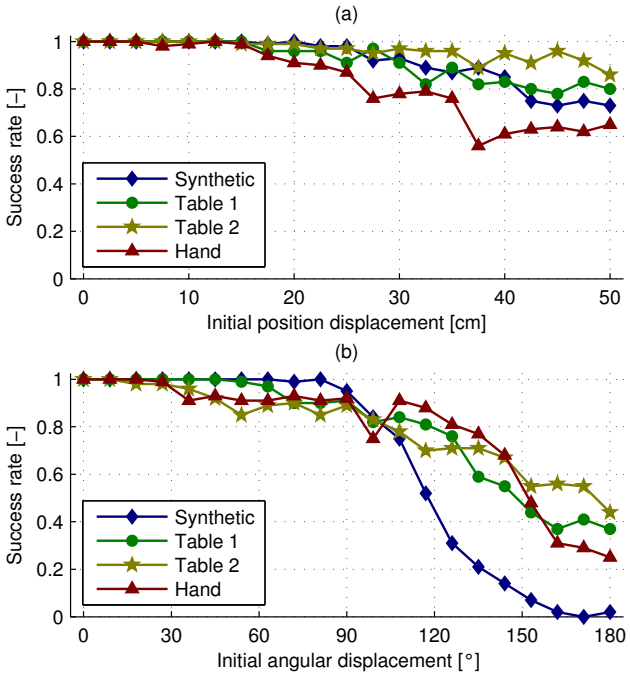


Fig. 7. Comparison of different motions regarding how the self-calibration tolerates deviations of the initial poses from the true poses. The success rate is evaluated for increasing position errors (a) and increasing angular deviations (b).

separately on each axis. "Table 1" is the same motion we used for the error evaluation of Section V, while "Table 2" is another motion we recorded using the rotation table, where again all three axes are used. At last, we captured a motion we generated by hand, i.e., we waved the prototype around trying to capture many different motions. All motions were 5.4 s long (675 samples) with the exception of "Table 2", which had a length of 20 s (2500 samples).

While the initial values of the motion were set to zero we varied the initial error of the poses of all free triads at every evaluation. On either the position or the orientation we added a certain error level while the other one was set to its true value. Concretely, we drew a random error vector from a uniform distribution, scaled this vector to a certain length, and added it to the true position of the triad. To generate an orientation error we drew a random axis through the origin of the triad and rotated the triad around this axis by an angle of a certain value. We applied a new random error with the respective magnitude for each one of the free triads. For each error level, we executed the self-calibration 100 times with new initial values for every trial and computed how often the self-calibration successfully converged. Figure 7 shows the evaluated results. The relation of successfully converged trails in relation to the total number of trials is referred to as the success rate at a certain error level.

The results of Fig. 7 must be considered in relation to the geometry of the used accelerometer array (Tab. I). E.g., the distance between two triads on satellite boards is only 16.91 cm. Thus, starting at an error of 8.45 cm a triad can have an initial position that is closer to the true position of another triad than

TABLE IV
COMPARISON BETWEEN CONVENTIONAL CALIBRATION AND SELF-CALIBRATION.

	Conventional calibration	Self-calibration
Estimated parameters / variables	Poses P Sensor parameters C	Poses P Motion ${}^{\mathrm{b}}\boldsymbol{x}$
Required data / parameters	Acceleration measurements $\boldsymbol{z}$ Motion ${}^{\mathrm{b}}\boldsymbol{x}$	Acceleration measurements $\boldsymbol{z}$ Sensor parameters C
Assumes sensor triads	No	Yes
Requires consecutive samples	No	Yes
Body frame defined by	Frame of reference system	Pose of fixed triad
Convergence depends on initial values	No	Yes
Accuracy depends on the accuracy of the reference data	Yes	No

to its own true position. However, the success rates shown in Fig. 7 illustrate a general relation between the initial pose error and the convergence of the self-calibration: The self-calibration converges safely as long as the initial error of the poses is not excessively large. The higher the initial deviation of the poses the more likely it is that the optimization drops into the local minima 2 or 3 of Tab. III.

Clearly, the tolerance of the self-calibration to initial pose errors clearly depends on the motion from which the acceleration data was sampled. Until now, we can only show, that the self-calibration works with different motions, even with arbitrary hand motion. However, the minimum requirement to a suitable motion is an open issue. The conventional calibration treats the state variables as independent. Therefore, it is straight forward to derive the minimum requirement for a suitable motion: The motion must contain samples from all dimensions of the motion space, which in this case is spanned by the linear and angular acceleration and the quadratic terms of the angular velocity [12]. In contrast to this, the state variables are no longer independent within the graph optimization of the self-calibration as they are connected by the process model. This may lead to relaxed minimum requirements. Our future research will be dedicated to investigate these requirements that guarantee for the convergence of the self-calibration.

VII. CONCLUSION

In this work, we are concerned with the calibration of accelerometer arrays. Specifically, we want to reconstruct the poses of the transducers, i.e., their positions and orientations. Conventional calibration methods rely on reference data of the imposed motion and corresponding acceleration measurements. In contrast to this, we present a method to estimate the accelerometer poses using only the measurements of the sensors themselves. No reference data of the motion is required, which is why we refer to it as self-calibration.

We achieve this by formulating the problem as a graph-optimization that estimates the sensor poses and the motion

jointly. To reduce the possible solutions to only one global optimum we introduce constraints between the free variables. Those are represented by error functions, which we derive from the process and the observation model of the control system formulation of accelerometer arrays. Furthermore, we reduce the number of free parameters as we model the used triple-axis sensors as sensor triads with known sensor parameters.

To summarize the properties of the derived method we compare it to the conventional calibration. In addition, the most important properties are collected in Table IV. The proposed self-calibration estimates the sensor poses and the imposed motion. In contrast to this, the conventional calibration does not infer the motion but is able to determine the sensor parameters. It is furthermore applicable to all types of accelerometer arrays as the transducers are modeled as single-axis sensors while the self-calibration relies on sensor triads. The self-calibration requires a set of consecutive samples whereas the ordering is not relevant for the conventional calibration. Both approaches determine the poses relative to a defined frame. The frame of the reference system defines the body frame for the conventional calibration. The pose of the fixed triad defines the body frame for the self-calibration. The convergence of the self-calibration depends on the initial error of the sensor poses as its nonlinear error function contains local minima. The conventional calibration converges for all initial values. The accuracy of the motion data has an influence on the estimation quality of the conventional calibration. This is not the case for the self-calibration as motion data is not required. Thus, our method is especially valuable if there is no accurate reference system available to capture the motion.

VIII. Outlook

The convergence of the self-calibration depends on the motion the accelerations were sampled from. In our future research we want to identify the minimum requirement for a motion to allow for a successful self-calibration. For a conventional calibration, the motion can be analyzed for its suitability as reference motion data is available. For self-calibration, the goal is to achieve autonomy from any kind of reference system that captures the imposed motion. As such, it is not possible to directly verify the suitability of the motion. The acceleration measurements themselves must be analyzed. Thus, a further goal of our future research is to derive a mathematical criterion that detects whether the acceleration measurements originate from a suitable motion and rates its quality. This would enable us to continuously measure the accelerations of an arbitrary motion and start the self-calibration once we detect a suitable acceleration set.

IX. Acknowledgments

We gratefully acknowledge the support of this work by the German Research Foundation (DFG). Parts of the presented results were achieved within the Research Training Group 1103 (Embedded Microsystems) while our current research on this topic is funded by a DFG research grant.

Appendix A
Linear Model

In this section, we derive a sensor model for accelerometer triads, i.e., a function that maps the acceleration in the sensor frame to the accelerations observed by the transducer triad. We want the sensitive axes to align with the frame of the sensor thereby eliminating the redundancy of the orientation and the sensitive axis. One common way to achieve this is to define the sensors to be located at the same position, the sensitive axes to be orthogonal with respect to each other and to be aligned with the major axes of triad frame. If we join the three sensitive axes to one sensitivity matrix S this results to

$$S = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \boldsymbol{s}_3 \end{bmatrix} = \begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & s_{3z} \end{bmatrix} \tag{20}$$

where the scalars s_{1x}, s_{2y}, and s_{3z} are the sensitivities of the accelerometers of the triad. However, with this approach it is not possible to model the misalignment of the sensitive axes as S is zero besides the diagonal. The model we propose overcomes this limitation. To align with the sensor frame it implements the following constraints on the sensitive axes.

1) The first sensitive axis $\boldsymbol{s}_1$ of the sensor triad aligns with the x-axis of the triad frame.
2) The second axis $\boldsymbol{s}_2$ lies in the xy-plane of the triad frame.

With these constraints the orientation and the sensitive axes are no longer redundant, however, without requiring them to be orthogonal. Imposing the constraints on the structure of sensitivity matrix S results to

$$S = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \boldsymbol{s}_3 \end{bmatrix} = \begin{bmatrix} s_{1x} & s_{2x} & s_{3x} \\ 0 & s_{2y} & s_{3y} \\ 0 & 0 & s_{3z} \end{bmatrix}. \tag{21}$$

As $\boldsymbol{s}_1$ is aligned with the x-axis of the triad, it has only one degree of freedom, which is its length. Thus, the first constraint is integrated in the same way as in (20) by defining $\boldsymbol{s}_1 = \begin{bmatrix} s_{1x} & 0 & 0 \end{bmatrix}^T$. As the second sensitive axis $\boldsymbol{s}_2$ lies in the xy-plane of the sensor frame it is not sensitive in the z-direction of the triad. Hence, we describe it by means of a minimal parameter set as $\boldsymbol{s}_2 = \begin{bmatrix} s_{2x} & s_{2y} & 0 \end{bmatrix}^T$. The third sensitive axis $\boldsymbol{s}_3$ does not have a constraint on its orientation within the triad. Therefore, we use a three dimensional vector without defining certain entries to be zero.

To calculate the observed accelerations we first have to compute the acceleration in the sensor frame ${}^{\mathrm{s}}\boldsymbol{a}$. As we assume the same position $\boldsymbol{r}$ for all three axes and furthermore describe them in the same sensor frame, ${}^{\mathrm{s}}\boldsymbol{a}$ is the same for all three axes. Hence, we can join the three scalar multiplications given in (4) to one matrix multiplication, which results to

$${}^{\mathrm{t}}\boldsymbol{z} = {}^{\mathrm{t}}\boldsymbol{h}({}^{\mathrm{b}}\boldsymbol{x},\, P,\, {}^{\mathrm{t}}C) = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \boldsymbol{s}_3 \end{bmatrix}^T {}^{\mathrm{s}}\boldsymbol{a} + {}^{\mathrm{o}}\boldsymbol{a}, \tag{22}$$

where ${}^{\mathrm{t}}\boldsymbol{z}$ holds the observed accelerations and ${}^{\mathrm{o}}\boldsymbol{a}$ the measurement offsets of each sensitive axis. For the calculation of ${}^{\mathrm{t}}\boldsymbol{z}$ we define the function ${}^{\mathrm{t}}\boldsymbol{h}$, which is dependent on the motion ${}^{\mathrm{b}}\boldsymbol{x}$, the pose P of the triad, and the collected sensor parameters ${}^{\mathrm{t}}C = \{s_{1x}, s_{2x}, s_{2y}, \boldsymbol{s}_3, {}^{\mathrm{o}}\boldsymbol{a}\}$.

The model covers every possible configuration of the axes as long as their positions are the same. However, if there is the

demand, the model can be easily extended to support multiple positions, e.g., by two vectors that describe the displacement of the y- and z-axis from the position of the x-axis. However, this results in ${}^{s}\boldsymbol{a}$ being different for each axis. Thus, ${}^{s}\boldsymbol{a}$ must be computed individually, which raises the computational effort. The packages of modern accelerometer triads are only a few millimeters wide. E.g., the Bosch BMA180 sensor, which we use, features a sensor housing with an outline of only 3x3x1 mm. Of course, the sensitive axes of the triad are located at different position within the housing. However, due to our experience, we expect the error generated by one common position of the axes to be small in comparison to the errors that arise from other non-ideal effects that the model does not cover, e.g., non-linearity or temperature dependence.

REFERENCES

[1] L. Klingbeil, M. Romanovas, P. Schneider, M. Traechtler, and Y. Manoli, "A modular and mobile system for indoor localization," in *Proc. IEEE Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2010, pp. 1–10.

[2] F. Höflinger, J. Müller, R. Zhang, L. Reindl, and W. Burgard, "A wireless micro inertial measurement unit (IMU)," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 9, pp. 2583–2595, Sep. 2013.

[3] S. Ovaska and S. Valiviita, "Angular acceleration measurement: A review," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 5, pp. 1211–1217, Oct. 1998.

[4] W. T. Latt, U.-X. Tan, C. N. Riviere, and W. T. Ang, "Placement of accelerometers for high sensing resolution in micromanipulation," *Sensors and Actuators A: Physical*, vol. 167, no. 2, pp. 304–316, Mar. 2011.

[5] M. Trächtler, T. Link, J. Dehnert, J. Auber, P. Nommensen, and Y. Manoli, "Novel 3-axis gyroscope on a single chip using SOI-technology," in *Proc. IEEE Sensors*, Oct. 2007, pp. 124–127.

[6] T. Northemann, M. Maurer, S. Rombach, A. Buhmann, and Y. Manoli, "A digital interface for gyroscopes controlling the primary and secondary mode using bandpass sigma-delta modulation," *Sensors and Actuators A: Physical*, vol. 162, no. 8, pp. 388–393, Aug. 2010.

[7] N. Yazdi, F. Ayazi, and K. Najafi, "Micromachined inertial sensors," *Proc. IEEE*, vol. 86, no. 8, pp. 1640–1659, Aug. 1998.

[8] N. Barbour and G. Schmidt, "Inertial sensor technology trends," *IEEE Sensors J.*, vol. 1, no. 4, pp. 332–339, Dec. 2001.

[9] P. Cappa, F. Patanè, and S. Rossi, "A redundant accelerometric cluster for the measurement of translational and angular acceleration and angular velocity of the head," *J. Med. Devices*, vol. 1, no. 1, pp. 14–22, Aug. 2007.

[10] A. Buhmann, C. Peters, M. Cornils, and Y. Manoli, "A GPS aided full linear accelerometer based gyroscope-free navigation system," in *Proc. IEEE/ION Position, Location, and Navigation Symposium*, Apr. 2006, pp. 622–629.

[11] P.-C. Lin, H. Komsuoglu, and D. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 932–943, Oct. 2006.

[12] P. Schopp, L. Klingbeil, C. Peters, and Y. Manoli, "Design, geometry evaluation, and calibration of a gyroscope-free inertial measurement unit," *Sensors and Actuators A: Physical*, vol. 162, no. 2, pp. 379–387, Aug. 2010.

[13] P. Schopp, H. Graf, M. Maurer, M. Romanovas, and Y. Manoli, "Observing relative motion with three accelerometer triads," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 12, pp. 3137–3151, Dec. 2014.

[14] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton, New Jersey: Princeton University Press, 1999.

[15] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2520–2530, Dec. 2005.

[16] S. Park, C.-W. Tan, and J. Park, "A scheme for improving the performance of a gyroscope-free inertial measurement unit," *Sensors and Actuators A: Physical*, vol. 121, no. 2, pp. 410–420, Jun. 2005.

[17] K. Parsa, T. Lasky, and B. Ravani, "Design and implementation of a mechatronic, all-accelerometer inertial measurement unit," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 6, pp. 640–650, Dec. 2007.

[18] D. Dube and P. Cardou, "The calibration of an array of accelerometers," *Trans. Can. Soc. Mech. Eng.*, vol. 35, no. 2, pp. 251–267, Jul. 2011.

[19] P. Cappa, F. Patanè, and S. Rossi, "Two calibration procedures for a gyroscope-free inertial measurement system based on a double-pendulum apparatus," *Meas. Sci. Technol.*, vol. 19, no. 5, p. 055204, Apr. 2008.

[20] I. Frosio, F. Pedersini, and N. Borghese, "Autocalibration of MEMS accelerometers," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 2034–2041, Jun. 2009.

[21] G. Panahandeh, I. Skog, and M. Jansson, "Calibration of the accelerometer triad of an inertial measurement unit, maximum likelihood estimation and Cramer-Rao bound," in *Proc. Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2010, pp. 1–6.

[22] F. Camps, S. Harasse, and A. Monin, "Numerical calibration for 3-axis accelerometers and magnetometers," in *Proc. IEEE Int. Conf. on Electro/Information Technology*, Jun. 2009, pp. 217–221.

[23] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2014, pp. 3042–3049.

[24] M. Glück, D. Oshinubi, P. Schopp, and Y. Manoli, "Real-time autocalibration of MEMS accelerometers," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 1, pp. 96–105, Jan. 2014.

[25] J.-O. Nilsson, I. Skog, and P. Handel, "Aligning the forces – eliminating the misalignments in IMU arrays," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 10, pp. 2498–2500, Oct. 2014.

[26] A. Kozlov, I. Sazonov, and N. Vavilova, "IMU calibration on a low grade turntable: Embedded estimation of the instrument displacement from the axis of rotation," in *Proc. Int. Symp. on Inertial Sensors and Systems (ISISS)*, Feb. 2014, pp. 1–4.

[27] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[28] G. Grisetti, R. Kümmerle, and K. Ni, "Robust optimization of factor graphs by using condensed measurements," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 581–588.

[29] R. Kümmerle, "State estimation and optimization for mobile robot navigation," Ph.D. dissertation, University of Freiburg, Germany, Apr. 2013.

[30] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 428–439, Sep. 2009.

[31] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Feb. 2010.

[32] K. Konolige, "Sparse sparse bundle adjustment," in *Proc. British Machine Vision Conference (BMVC)*, Aberystwyth, Wales, Aug. 2010.

[33] J. M. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. Berlin: Springer, 2003, no. 218.

[34] C. Hertzberg, "A framework for sparse, non-linear least squares problems on manifolds," Master's thesis, University of Bremen, Germany, 2008.

[35] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, Jan. 2013.

[36] R. Wagner, O. Birbach, and U. Frese, "Rapid development of manifold-based graph optimization systems for multi-sensor calibration and slam," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, CA, Sep. 2011, pp. 3305–3312.

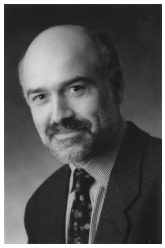
[37] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 3607–3613.

[38] A. Ude, "Filtering in a unit quaternion space for model-based object tracking," *Robotics and Autonomous Systems*, vol. 28, no. 2-3, pp. 163–172, Aug. 1999.

Patrick Schopp received the Dipl.-Ing. degree in Microsystems Engineering from the University of Freiburg, Germany, in 2008, where he is currently working towards the Ph.D. degree at the Fritz Huettinger Chair of Microelectronics. His research interests include state estimation, calibration, and machine learning.

Hagen Graf received his M.Sc. in Microsystems Engineering from the University of Freiburg, Germany, in 2011. He is now a Ph.D. student at the Fritz Huettinger Chair of Microelectronics at the same University. His research interests include orientation estimation and application specific processors.

Wolfram Burgard Wolfram Burgard studied Computer Science at the University of Dortmund and received his PhD also in computer science from the University of Bonn. Since 1999 he is a professor for computer science at the University of Freiburg and head of the research lab for Autonomous Intelligent Systems. His areas of interest lie in artificial intelligence and mobile robots. His research mainly focuses on the development of robust and adaptive techniques for state estimation and control. Over the past years he and his group have developed a series of innovative probabilistic techniques for robot navigation and control. They cover different aspects including localization, map-building, SLAM, path-planning, exploration. Wolfram Burgard published more than 300 publications. In 2010, he received the Gottfried-Wilhelm-Leibniz-Preis. He is Fellow of the IEEE, AAAI, and ECCAI.



Yiannos Manoli (M'82-SM'08) was born in Famagusta, Cyprus, in 1954. As a Fulbright scholar, he received the B.A. degree (summa cum laude) in physics and mathematics from Lawrence University in Appleton, WI, in 1978 and the M.S. degree in electrical engineering and computer science from the University of California, Berkeley, in 1980. He obtained the Dr.-Ing. Degree in electrical engineering from the Gerhard Mercator University in Duisburg, Germany, in 1987.

From 1980 to 1984, he was a research assistant at the University of Dortmund, Germany, in the field of A/D and D/A converters. In 1985, he joined the newly founded Fraunhofer Institute of Microelectronic Circuits and Systems, Duisburg, Germany, where he established a design group working on mixed-signal CMOS circuits especially for monolithic integrated sensors and application specific microcontrollers. From 1996 to 2001, he held the Chair of Microelectronics as full professor with the Department of Electrical Engineering, University of Saarland, Saarbrücken, Germany.

In 2001, he joined the Department of Microsystems Engineering (IMTEK) at the Faculty of Engineering of the University of Freiburg, Germany where he established the Chair of Microelectronics. With an endowment of the Fritz Huettinger Foundation and in memory of the founder of today's Huettinger Elektronik, the University of Freiburg renamed the Chair to "Fritz Huettinger Chair of Microelectronics", in 2010. In 2012 Professor Manoli became Dean of Engineering after having served as Associate Dean since 2008.

Since 2005, he has additionally served as a director of the "Hahn-Schickard-Institut für Mikro- und Informationstechnik" in Villingen-Schwenningen and Freiburg, Germany.

His current research interests are the design of low-voltage/low-power mixed-signal CMOS circuits, energy harvesting electronics, sensor read-out circuits as well as Analog-to-Digital converters. Additional research activities concentrate on motion and vibration energy transducers and on the field of inertial sensors and sensor fusion.

In 2000, he had the opportunity to spend half a year on a research project with Motorola (now Freescale) in Phoenix, AZ. In 2006, he spent his sabbatical semester with Intel, Santa Clara, CA, working on the read-out electronics for a high-resolution accelerometer.

Prof. Manoli and his group have received best paper awards at ESSCIRC 1988, 2009 and 2012, PowerMEMS 2006, MWSCAS 2007, and MSE-2007. The MSE-2007 award was granted for SpicyVOLTsim (www.imtek.de/svs), a web-based application for the animation and visualization of analog circuits for which Yiannos Manoli also received the Media Prize of the University of Freiburg in 2005. He was the first to receive the Best Teaching Award of the Faculty of Engineering when it was introduced in 2008. For his creative and effective contributions to the teaching of microelectronics, he has also received the Excellence in Teaching Award of the University of Freiburg and the Teaching Award of the State of Baden-Wuerttemberg, both in 2010.

Professor Manoli has served as a Distinguished Lecturer of the IEEE and as guest editor of the "Transactions on VLSI" and the "Journal of Solid-State Circuits". He is on the Senior Editorial Board of the IEEE "Journal on Emerging and Selected Topics in Circuits and Systems" and on the Editorial Board of the "Journal of Low Power Electronics". Professor Manoli has served on the committees of a number of conferences such as ISSCC, ESSCIRC, IEDM and ICCD, and was Program Chair (2001) and General Chair (2002) of the IEEE International Conference on Computer Design (ICCD). He is a member of VDE, Phi Beta Kappa, Mortar Board, and senior member of the IEEE.