FORWARD AND INVERSE STOCHASTIC FILTERING FOR INERTIAL SENSOR CALIBRATION

Joachim Fox, Laboratory of Process Automation (LPA), Saarland University, Germany, e-mail: j.fox@lpa.uni-saarland.de[†]

ABSTRACT
Extended Kalman filters have long been applied to sensor fusion in navigation tasks. They can be used to estimate both the states and the parameters of the dynamic system. In recent years, so-called sigma-point Kalman filters with an improved estimation accuracy compared to extended Kalman filters have emerged. This work shows how these filters can be applied to calibrate an inertial measurement unit used for unaided navigation. Two different filter structures are proposed: a forward filter models the whole navigation process (states and parameters) while an inverse filter performs only a parameter estimation.

KEY WORDS
sigma-point Kalman filter, intertial navigation, calibration

1 Introduction

The parameter estimation of complex nonlinear dynamical systems has long been an area of interest for researchers. If both states and parameters are to be estimated while the measurements are exposed to noise, recursive stochastic methods lend themselves to be used. Among others, the extended Kalman filter [1] is a popular algorithm for this purpose. It has many applications, especially in the field of inertial navigation.

However, extended Kalman filtering may lead to biased estimations in the presence of large nonlinearities. New approaches, called *unscented Kalman filters* (UKF) [2], *central difference Kalman filters* (CDKF) [3] or – more generically – *sigma-point Kalman filters* (SPKF) [4], yield much more accurate estimations of a system's states and/or parameters and are even easier to implement because they lack the need to calculate analytical derivatives.

This paper will show how the concept of sigma-point filtering can be used to calibrate an inertial measurement unit (IMU), consisting of three accelerometers and three gyroscopes, measuring turn rates. A standard industrial robot moves the IMU in order to generate reference accelerations and rates. However, both the IMU's measurements and the robot's poses, which are used in the calibration process, are subject to measurement noise. Applying Kalman-type filters allows to take into account the stochastic properties of the noise in order to generate parameter (and state) estimates.

[†]Inertial sensor calibration and robotics are research topics of the LPA. The author would like to thank Prof. H. Janocha, head of the LPA, for his scientific and financial support of this work.

To the author's knowledge, the work presented here and its predecessor [5] describe some of the first applications of sigma-point filters to complex navigation systems based on real measurements and not only on simulated data.

The paper is structured as follows: First an overview is given of the whole calibration process including a mathematical model of the IMU. Section 3 follows with a brief introduction into sigma-point filter theory. In section 4, two possible formulations of the navigation SPFK are introduced, one integrating the IMU's measurements and one differentiating the robot's pose data. Section 5 reviews the experimental results and compares both approaches. Conclusions and a prospect for further research complete the paper.

2 The IMU calibration process

2.1 Inertial navigation

Inertial measurement units are used to compute the pose (position and orientation) of a moving object by taking measurements of the real rates $\boldsymbol{\omega}$ and accelerations $\mathbf{a}$ (throughout the paper, vectors and matrices are bold-typed). In modern strapdown systems, $\boldsymbol{\omega}$ and $\mathbf{a}$ are measured in a coordinate frame fixed to the object. Therefore, the accelerations have to be transformed into an earth-fixed frame. Since the transformation matrix depends on the rotation rates, the system is nonlinear. There are special integration algorithms for the navigation equations which can be found in [6].

If an IMU shall be used for *unaided* navigation, i.e. no external information such as GPS data is provided, it must be assured that the sensors are very accurate. In the past, various calibration strategies for IMUs have been developed [7, 8, 9], but they suffer either from the need for expensive hardware (precision rotating tables) or from their limitation to low-cost units. In this work, a new approach shall be followed where comparingly low-cost hardware – an ordinary industrial robot – is used in combination with modern algorithms to calibrate the IMU.

2.2 The IMU's calibration parameters

Within this work, it is assumed that both the accelerometers and the rate sensors have a well-defined characteristic, which can be modeled to the needed accuracy

by a polynomial of the order $N_{pa/\omega}$ with coefficients[1] $k_{a/\omega,x/y/z,i}$ $(i = 0 \dots N_{pa/\omega})$. These coefficients may be temperature-dependent. Ideally, all sensors should be aligned exactly parallel to the coordinate axes of the body frame, however, production tolerances result in a nonorthogonal mounting. This can be compensated with a 3×3-dimensional cross-coupling matrix $\mathbf{C}_{a/\omega}$. If we fix $k_{a/\omega,x/y/z,1} = 1$, these matrices are not subject to constraints, and therefore each of these matrices contains nine independent parameters that are to be estimated.

It is impossible to mount all three acceleration sensors at the origin of the body frame. As the sensors rotate around the origin, centripetal and tangential accelerations will be measured by the accelerometers. Therefore, an excentricity correction vector has to be introduced, where $\mathbf{u}_{x/y/z}$ describes the displacement of the $x/y/z$-acceleration sensor. For the accelerometers, the full sensor model is thus

$$\mathbf{a} = \mathbf{C}_a \left(\begin{pmatrix} \sum_{i=0}^{N_{pa}} k_{a,x,i} {a'_x}^i \\ \sum_{i=0}^{N_{pa}} k_{a,y,i} {a'_y}^i \\ \sum_{i=0}^{N_{pa}} k_{a,z,i} {a'_z}^i \end{pmatrix} - \begin{pmatrix} (\mathbf{C}_a^{-1})_{1,1:3}(\boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{u}_x + \dot{\boldsymbol{\omega}} \times \mathbf{u}_x) \\ (\mathbf{C}_a^{-1})_{2,1:3}(\boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{u}_y + \dot{\boldsymbol{\omega}} \times \mathbf{u}_y) \\ (\mathbf{C}_a^{-1})_{3,1:3}(\boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{u}_z + \dot{\boldsymbol{\omega}} \times \mathbf{u}_z) \end{pmatrix} \right) \quad (1)$$

where $a'_{x/y/z}$ are the measured accelerations in the x, y, and z directions while $\mathbf{a}$ and $\boldsymbol{\omega}$ are the real accelerations and rates, respectively [8]. The model of the rate sensors is similar but usually does not need an excentricity correction.

2.3 The robot-based calibration process

Estimating an IMU's parameters requires an appropriate set of input data to the sensors to be generated, i.e. the IMU must be exposed to rates and accelerations. Most calibration methods make use of the fact that the accelerometers always sense gravity and therefore the IMU is positioned in different orientations to the gravity vector. The absolute value of the measured acceleration vector [8, 9] or all vector components [7, 8] are compared to the gravity vector. To estimate the rate sensors' parameters, however, one has to generate reference rates artificially, because the 'natural' reference rate related to the earth's rotation is not large enough to allow for an accurate calibration. Rate tables are used for this purpose; only recently, a method has been proposed to use a robot for this static calibration [8].

While the rate table based methods have been established since long, they suffer from being rather expensive. Therefore here, a new calibration method will be presented. The needed hardware is shown schematically in Fig. 1: the robot moves the IMU and log's its poses while the IMU sends the raw accelerations and rates to a calibration computer that performs all calculations. Now there are two data

[1] In the following, the index a stands for accelerometer related parameters, ω for rate-sensor related parameters.

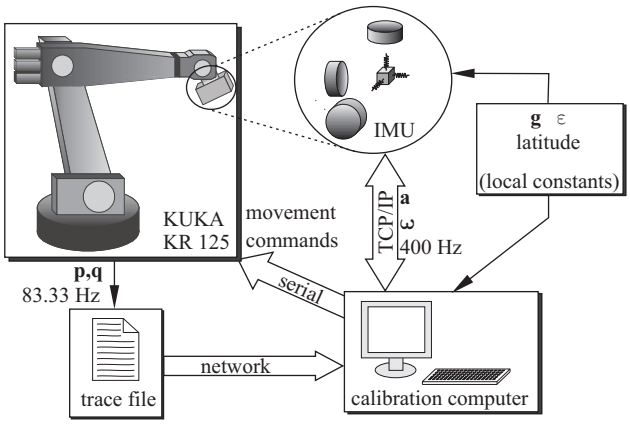

Figure 1. Hardware Configuration of the Calibration.

sets with different characteristics: the IMU's measurements provide accurate short-time but poor long-time pose information while the robot's trace data guarantees a long-time stability.

The incorporation of redundant measurements into one optimized estimation of states or parameters has since long been a field for extended Kalman filtering. In many navigation systems, Kalman filters are used to combine inertial measurements with other sensors like GPS or odometers. However, the convergence of Kalman filters is difficult to prove, especially if parameters and states shall be estimated at the same time. This becomes more and more an issue if update information is comparingly rare as is the case with GPS measurements. Therefore in real-world systems no more than a few parameters (such as accelerometer biases) are estimated during the navigation calculation.

In this paper however, it will be shown that it is possible to perform an IMU's calibration based on stochastic filtering. A new kind of stochastic filter, named sigma-point Kalman filter (SPKF), is used because it proved to be more accurate and easier to implement and control than conventional extended Kalman filters.

3 Sigma-point Kalman filters: theory review

In this section, the SPKF theory shall be reviewed in brief. Good overviews can be found in [2, 4]. The dynamic system treated in this paper can be written as a first-order Markov system with the following equations.

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \mathbf{n}_k^{\mathrm{p}} \quad (2)$$

$$\mathbf{x}'_{k+1} = \mathbf{f}'(\mathbf{x}'_k, \boldsymbol{\xi}_k, \mathbf{u}_k, k) + \mathbf{n}' s_k \quad (3)$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}'_k, \boldsymbol{\xi}_k, \mathbf{u}_k, k) + \mathbf{n}_k^{\mathrm{m}} \quad (4)$$

$\mathbf{x}'$, $\mathbf{u}$ and $\mathbf{y}$ are vectorial state, input and output variables, $\boldsymbol{\xi}$ is a vector composed of the system's parameters, k is the current sample index, $\mathbf{n}'s$, $\mathbf{n}^{\mathrm{m}}$, and $\mathbf{n}^{\mathrm{p}}$ stand for zero-mean system, measurement, and parameter noise, and $\mathbf{f}'$ and $\mathbf{g}$ symbolize the (possibly non-linear) system function

and the measurement function of the process, respectively. The state vector $\mathbf{x}'$ is of dimension n', the output is of dimension m. $\mathbf{n}'s_k$, $\mathbf{n}^{\mathrm{m}}_k$, $\mathbf{n}^{\mathrm{p}}_k$, $\boldsymbol{\xi}_k$, and $\mathbf{x}'_k$ are stochastically independent. p parameters are estimated. $\mathbf{n}^{\mathrm{p}}$ can be used to control the convergence behaviour of the parameters [5].

By using this formulation, one can introduce an augmented state[2] $\mathbf{x} = (\boldsymbol{\xi}^T, \mathbf{x}'^T)^T$ with dimension $n = n' + p$. Then, the first two equations can be combined to the following equation with the new system function $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) = (\boldsymbol{\xi}^T_k, \mathbf{f}'^T(\mathbf{x}'_k, \boldsymbol{\xi}_k, \mathbf{u}_k, k))^T$ and $\mathbf{n}^{\mathrm{s}} = (\mathbf{n}^{\mathrm{p}T}, \mathbf{n}'^{\mathrm{s}T})^T$.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{n}^{\mathrm{s}}_k \tag{5}$$

The most widely used estimator for nonlinear systems is the extended Kalman filter (EKF). It consists of two steps: an a-priori estimation $\hat{\mathbf{x}}^-_k$ before the measurement $\mathbf{y}_k$ has been taken and an a-posteriori estimation $\hat{\mathbf{x}}^+_k$ with knowledge of $\mathbf{y}_k$. This method results in the following equations where $E\{\cdot\}$ stands for the expectation[3].

$$\begin{aligned}
\hat{\mathbf{x}}^-_k &= \mathbf{f}(\hat{\mathbf{x}}^+_{k-1}, \mathbf{u}_{k-1}, k-1) \\
&\approx E\{\mathbf{x}_k | \mathbf{y}_{k-1}, \mathbf{y}_{k-2}, \ldots\} && (6)\\
\hat{\mathbf{x}}^+_k &= \hat{\mathbf{x}}^-_k + \mathbf{K}_k[\mathbf{y}_k - \hat{\mathbf{y}}_k] \\
&\approx E\{\mathbf{x}_k | \mathbf{y}_k, \mathbf{y}_{k-1}, \ldots\} && (7)\\
\hat{\mathbf{y}}_k &= \mathbf{g}(\hat{\mathbf{x}}^-_k, \mathbf{u}_k, k) \\
&\approx E\{\mathbf{y}_k | \mathbf{y}_{k-1}, \mathbf{y}_{k-2}, \ldots\} && (8)
\end{aligned}$$

The Kalman gain matrix $\mathbf{K}_k$ is computed using the matrix $\hat{\mathbf{P}}^-_k$, an estimation of the state estimation error's covariance $E\{(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T\}$. $\hat{\mathbf{P}}^{-/+}_k$ are computed by linearizing the system equations around the current estimate $\hat{\mathbf{x}}^+_{k-1}/\hat{\mathbf{x}}^-_k$:

$$\begin{aligned}
\hat{\mathbf{P}}^-_{k+1} &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \hat{\mathbf{P}}^+_k \Big(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big)^T + \mathbf{Z}_k && (9)\\
\hat{\mathbf{P}}^+_k &= \Big[\mathbf{I} - \mathbf{K}_k \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big] \hat{\mathbf{P}}^-_k && (10)\\
\mathbf{K}_k &= \hat{\mathbf{P}}^-_k \Big(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big)^T \Big[\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \hat{\mathbf{P}}^-_k \Big(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big)^T + \mathbf{W}_k\Big]^{-1} && (11)
\end{aligned}$$

where $\mathbf{Z}_k = E\{\mathbf{n}^{\mathrm{s}}_k \mathbf{n}^{\mathrm{s}T}_k\}$, $\mathbf{W}_k = E\{\mathbf{n}^{\mathrm{m}}_k \mathbf{n}^{\mathrm{m}T}_k\}$, and $\mathbf{I}$ is the identity matrix. Taking a closer look at (11), one can see that this is nothing but a linear approximation of

$$\mathbf{K}_k = \mathbf{P}^-_{xy,k} (\mathbf{P}^-_{yy,k})^{-1}, \tag{12}$$

with $\mathbf{P}^-_{yy}$ denoting the covariance matrix of the *innovation* $(\mathbf{y}_k - \hat{\mathbf{y}}_k)$ and $\mathbf{P}^-_{xy}$ the covariance matrix between the a-priori state estimate error and the innovation. (6) and (9) are referred to as the Kalman prediction step, and (7), (8), (10), and (11) as the Kalman update step. For a more detailed discussion, see for example [1].

[2] If unambiguous, the index k is omitted from now on.

[3] Throughout this paper, variables with a hat $(\hat{\cdot})$ stand for estimated values while non-hatted variables represent the true quantities. A '-' denotes a-priori estimates, a '+' a-posteriori estimates.

Since the system equations are linearized around the current estimate, the stochastic distribution of the EKF's state is disregarded. Often, this leads to a biased estimate and a significant underestimation of the state's covariance matrix, the latter being a common cause of filter divergence. New approaches, all of which belong to the group of sigma-point Kalman filters, can solve this problem [2, 4]. The expectations are no longer being approximated by means of analytical linearization, as is the case with the EKF, but are formed by means of a weighted sample covariance of a set of r representative points $\mathcal{X}_0 \ldots \mathcal{X}_{r-1}$ in the state space [5]:

$$\begin{aligned}
\mathcal{X}_i &= \mathbf{f}(\mathcal{X}^+_i, \mathbf{u}), \qquad \mathcal{Y}_i = \mathbf{g}(\mathcal{X}^-_i, \mathbf{u}) && (13)\\
\hat{\mathbf{P}}^- &= \sum_{i=0}^{r-1} \sum_{j=0}^{r-1} w^c_{ij} \mathcal{X}_i \mathcal{X}^T_j + \mathbf{Z} && (14)\\
\hat{\mathbf{P}}^-_{xy} &= \sum_{i=0}^{r-1} \sum_{j=0}^{r-1} w^{cc}_{ij} \mathcal{X}^-_i \mathcal{Y}^T_j && (15)\\
\hat{\mathbf{P}}^-_{yy} &= \sum_{i=0}^{r-1} \sum_{j=0}^{r-1} w^c_{ij} \mathcal{Y}_i \mathcal{Y}^T_j + \mathbf{W} && (16)\\
\hat{\mathbf{x}}^- &= \sum_{i=0}^{r-1} w^m_i \mathcal{X}_i\,, \quad \hat{\mathbf{y}} = \sum_{i=0}^{r-1} w^m_i \mathcal{Y}_i && (17)
\end{aligned}$$

w^c_{ij}, w^{cc}_{ij}, and w^m_i are the weights of the covariance, the cross-covariance and of the mean. It is then possible to show that computing the gain matrix according to (12) with $\hat{\mathbf{P}}^-_{yy}$ and $\hat{\mathbf{P}}^-_{xy}$ from (13) to (17) represents a numeric linearization of the process. This linearization includes r points unlike the EKF, which uses only the current best estimate for linearization. The advantage of this method is that under some assumptions, the approximation is precise at least up to the third order of the Taylor series (in contrast to the first-order EKF approximation). Moreover, the SPKF avoids the analytic computation of the partial derivatives $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ needed for the EKF.

The various implementations of the sigma-point filter differ in the choice of sigma points $\mathcal{X}_i$, as well as in the weights w^c_{ij}, w^{cc}_{ij} and w^m_i. The equations for the *central difference Kalman filter* (CDKF) are given here as an example ([3, 5]).

$$\begin{aligned}
\mathcal{X}^{+/-}_0 &= \hat{\mathbf{x}}^{+/-} && (18)\\
\mathcal{X}^{+/-}_i &= \hat{\mathbf{x}}^{+/-} + h(\sqrt{\hat{\mathbf{P}}^{+/-}})_{1:n,i},\ i = 1 \ldots n && (19)\\
\mathcal{X}^{+/-}_{i+n} &= \hat{\mathbf{x}}^{+/-} - h(\sqrt{\hat{\mathbf{P}}^{+/-}})_{1:n,i},\ i = 1 \ldots n && (20)\\
w^m_0 &= \frac{h^2 - n}{h^2}, \quad w^m_i = \frac{1}{2h^2}\ \ (i > 0) && (21)\\
w^c_{00} &= \frac{h^2 - 1}{h^4}\, n && (22)\\
w^c_{ii} &= \frac{2h^2 - 1}{4h^4} \quad (i = 1 \ldots 2n) && (23)
\end{aligned}$$

$$w_{0i}^{c} = w_{i0}^{c} = \frac{1-h^2}{2h^4} \quad (i = 1 \ldots n) \qquad (24)$$

$$w_{i,i+n}^{c} = w_{i+n,i}^{c} = -\frac{1}{4h^4} \quad (i = 1 \ldots n) \qquad (25)$$

$$w_{ii}^{cc} = w_{0,i+n}^{cc} = \frac{1}{2h^2} \quad (i = 1 \ldots n) \qquad (26)$$

$$w_{i,i+n}^{cc} = w_{0,i}^{cc} = -\frac{1}{2h^2} \quad (i = 1 \ldots n) \qquad (27)$$

All other weights w_{ij}^{c} and w_{ij}^{cc} are zero. The root of the positive definite matrices $\hat{\mathbf{P}}^{+/-}$ can be computed using the Cholesky decomposition. h is a scaling parameter determining the spread of the sigma-points around the mean. It can be selected depending on the distribution of the state; $h = \sqrt{3}$ gives the best mean and covariance approximation in the Taylor sense for the normal distribution. The CDKF has been chosen for the calibration task described below, because it features a slightly better precision than other SPKF variants [4]. Additionally, h is its only parameter which is why the CDKF is easier to tune than, for instance, the UKF which has three rather non-intuitive tuning parameters.

4 Forward and inverse navigation filtering

This section will explain how an SPKF can be used to calibrate an IMU. There exist two conceivable approaches to implement a Kalman-like filter: first, accelerations and rates can be integrated as is done in usual navigation systems; thus, the update is performed on the basis of the pose data. This method is called here *forward navigation filter*. As a second approach, the robot's pose data can be differentiated such that the measured accelerations and rates can be compared to computed ones. This is called an *inverse navigation filter*. Both approaches will be described now in more detail.

4.1 Forward navigation filter

The upper part of Fig. 2 shows the signal flow of the forward navigation filter. First, the raw measurements run through the sensor model blocks $\mathrm{P_{Fa}}$ and $\mathrm{P_{F\omega}}$ described in section 2.2. These blocks are the main interest of the calibration procedure. According to the navigation algorithms, rates and accelerations are then used to compute position and orientation (represented as quaternions, see [6]). Therefore, the SPKF state consists of the IMU's parameters, the orientation information (expressed as a 4-dimensional quaternion $\mathbf{q}_{\mathrm{IMU}}$, see [6]), the position $\mathbf{p}_{\mathrm{IMU}}$, and the velocity $\mathbf{v}_{\mathrm{IMU}}$, both given in the world frame. Since all six sensors have superimposed noise, $\mathbf{x}_{\mathrm{aug}}$ has $p + 4 + 3 + 3 + 6$ elements.

This approach is the natural way to estimate both states and parameters of the system. It is convenient to implement because the integration algorithms are usually part of an IMU. Since the SPKF has no need for analytical derivatives, it can be implemented as a black-box algorithm which uses the functions already found on the IMU. This also allows for a convenient separation of the sensor model and the navigation algorithm – if different IMUs shall be calibrated with the new approach, only the sensor model has to be changed and there is no need to recalculate this model's influence on the whole system's derivatives.

Note that the quaternion used here to represent the orientation introduces an equality constraint because $|\mathbf{q}_{\mathrm{IMU}}| = 1$. Constrained estimation with SPKFs is still a matter of research. Here, a renormalization of the quaternion has been done after every update which is also often been applied to Kalman filters. Experiments have also been conducted with a 3-degrees-of-freedom unconstrained representation of the orientation. The results showed no significant difference to the quaternion algorithm. Therefore here, the quaternion representation was used because it eases the implementation of the navigation equations.

4.2 Inverse navigation filter

When using the forward navigation filter, accelerations and rates are looked at as inputs to the system while the robot's controller measurements serve as updates. It is possible to reverse this problem by using an inverse navigation filter whose signal flow is shown in the lower half of Fig. 2. In this case, the robot's measurements are differentiated numerically to compute the rates and accelerations which should be measured by the IMU's sensors. This is done in a precomputation step and not part of the Kalman estimator. The real measurements are then used as updates for the SPKF.

While this approach diminishes largely the computing time – computing numerical derivatives can be done much more efficiently than integrating the navigation equations – it introduces a new error source into the calculations: the numerical derivatives tend to be more inaccurate than the integrations used in the forward filter. In addition to that, high-frequency information in the accelerations and rates will be lost; this effect can be seen clearly if one compares the measured accelerations with the numerically computed ones. For the numerical differentiation, an algorithm is used which uses five points spread symmetrically around the current sample. This leads to a smoothing of the data which has to be taken into account by applying an appropriate filter to the raw accelerations and rates, too. This is particularly important because the acceleration sensors show a rather large noise level and therefore unsmoothed update measurements would lead to very small innovations in the filter because the relevant parts of $\mathbf{W}$ exceed those of $\mathbf{P}_{xy}^{-}$ by far (cf. (12) and (16)).

The inverse filter opens further possibilities to improve the results: since the numerical derivatives are computed before the filter is run, the parameter estimation is no longer a dynamical system but rather a static map between the inputs (robot's accelerations, velocities and rates) and outputs (IMU's measurements). Therefore, one can store the input-output pairs beforehand and then shuffle them be-

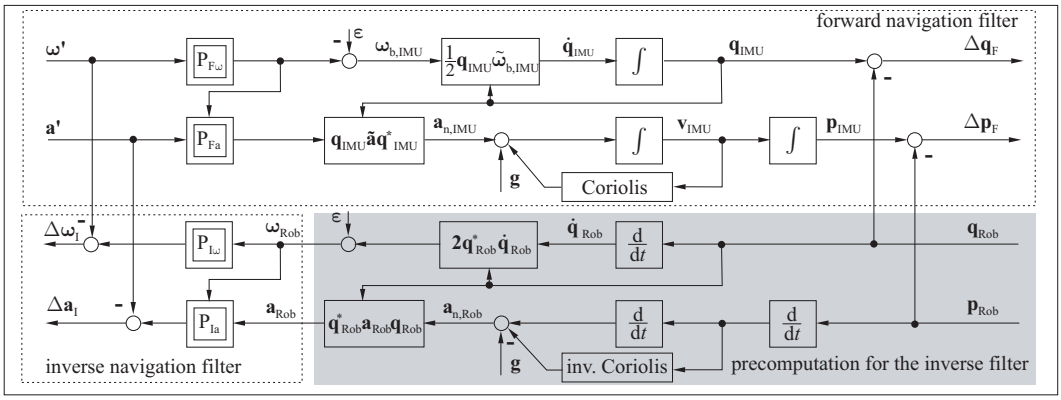

Figure 2. SIGNAL FLOW OF THE IMU CALIBRATION

fore feeding them to the SPKF algorithm. This destroys any correlation between consecutive data, the filter is driven by inputs which resemble white noise more closely than the real path data does, and thus persistent excitation is guaranteed. By using this shuffling technique, the convergence speed of the method is improved largely as can be seen in section 5. Note that this approach cannot be taken with the forward filter because there, the states have to be estimated too.

Note that the inverse filter is not applied in addition to the forward filter but replaces it. Therefore, the filters use either the positions or the differentiated positions as updates, not both – using both would not provide additional information to the system and would therefore make no sense. The basic difference between the filters lies in the location of the updates: the forward filter uses updates 'far' (i.e. two integrations) away from the paramters to be calibrated while the inverse filters updates the sensor measurements directly.

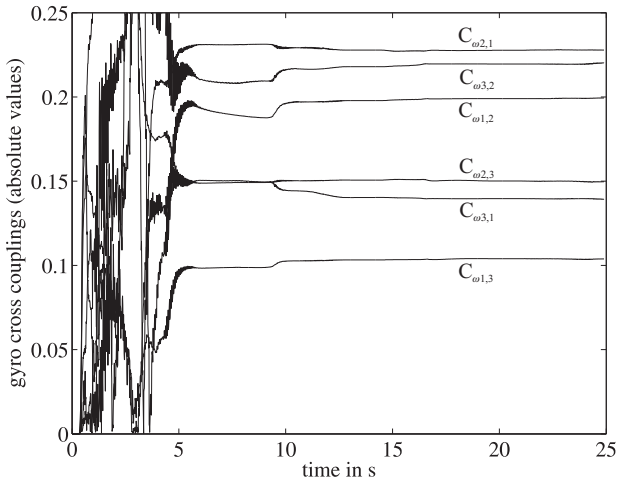

Figure 3. Estimated Gyro Cross Couplings, Off-Diagonal Elements (Forward Filter)

5 Experimental results

Experiments have been conducted on a high-end IMU originally designed at our laboratory for industrial robot calibration [10]. It consists of three servo accelerometers (resolution: $10^{-6}g$) and three ring-laser gyroscopes (resolution: $5 \cdot 10^{-6}\,\mathrm{rad}$). Throughout the experiments the data acquisition rate was 400 Hz (the robot's movements' spectral parts ended at about 50 Hz). The IMU was moved by a KUKA KR125 industrial robot which is able to log its poses with 12 ms sample time. However, only the commanded poses are logged, not the true poses, so this data is unreliable; the resulting error is modeled as white noise.

First, the results of the forward navigation filter will be examined. In Fig. 3, one sees the estimated gyroscope cross couplings. After a few seconds, the parameters are close to their true values and change much more slowly. One can observe major changes in the parameters at about 9 s which correspond to a different kind of motion of the robot. It is clear that here, the lack of persistent excitation may be an issue, and therefore, one has to chose the robot paths carefully. The estimated parameters have been found to be accurate to the third (this accuracy has been verified by adding known artificial errors to the sensor measurements and identifying both the 'real' errors and the additional ones).

Fig. 4 shows the estimated gyroscope cross couplings when applying the inverse navigation filter. The solid lines indicate the parameter estimations when using the shuffling algorithm while the dashed lines are generated by the conventional, 'ordered' (non-shuffled) algorithm. One can see clearly that both filters are able to estimate the parameters correctly. The shuffling algorithm however converges faster because the input data are closer to white noise. The same behaviour can be seen in Fig. 5 where the estimated x-axis accelerometer's excentricities (i.e. the distances between the sensor and the robot's tool center point) are plotted. However, the accelerometer's parameters show a slower convergence than the gyro's. This is mainly due to

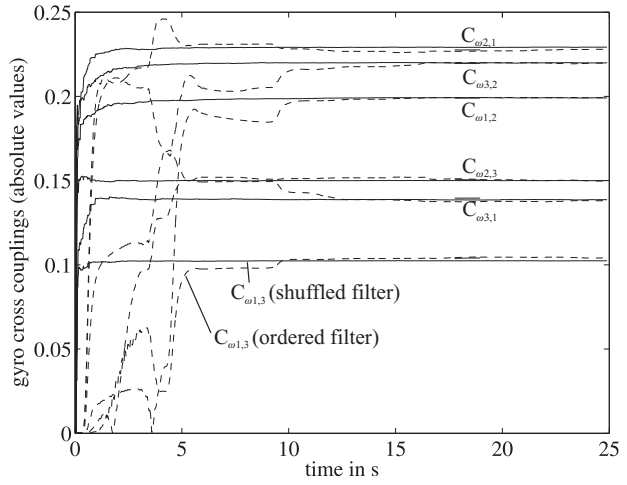

Figure 4. Estimated Gyro Cross Couplings, Off-Diagonal Elements (Inverse Filter). Solid lines: shuffled filter. Dashed lines: ordered filter.

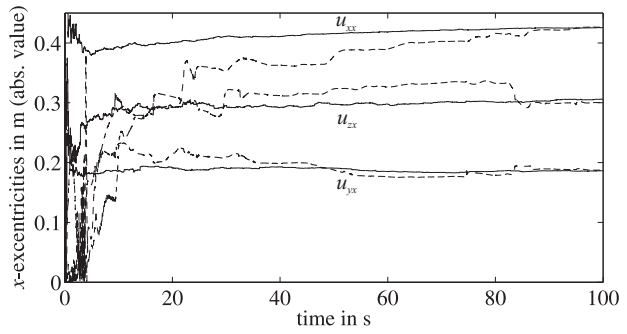

Figure 5. Estimated x-Accelerometer Excentricities (Inverse Filter). Solid lines: shuffled filter. Dashed lines: ordered filter.

a much higher noise level on the accelerometer's measurements and the error that comes from the numerical differentiation of the rates in order to correct the excentricities.

Since the inverse filter estimates only the parameters of the system while the forward filter also has to estimate the states, and because the inverse filter has a much simpler system function, the inverse filter needs much less computing time than the forward filter while the estimation accuracy is on the same order. Therefore, a clear preference can be given here for the inverse filter with data shuffling.

6 Conclusions and future works

In this paper, two approaches have been presented to estimate an inertial measurement unit's parameters when the unit is moved by an industrial robot. Since no exact pose information is known and additionally the IMU's measurements are noisy, a stochastic estimation scheme based on the recently emerged sigma-point Kalman filters is employed. While robustness and speed of convergence of this method are satisfying, there is still work to be done to improve the calibration accuracy. High-end IMUs require an accuracy $< 10^{-5}$ on some parameters, which, at the moment, cannot be achieved with the proposed algorithms, so the method is limited to lower-precision IMUs.

On a more theoretical basis, some questions regarding SPKFs remain open. For example the estimation of (equality or inequality) constrained states is still unresolved. In a mobile robot navigation context, for example, this would be of interest when moving in a known world where walls and other objects constitute regions with a probability of zero for the IMU's position. This topic is part of our current research.

Another topic which is being investigated is how the synchronization between the robot and the IMU can be estimated by a Kalman filter. See the companion paper [11] for further information.

References

[1] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*: John Wiley & Sons, New York, Chichester, 2001

[2] S. J. Julier and J. K. Uhlmann, Unscented filtering and nonlinear estimation: *Proceedings of the IEEE*, 92(3):401–421, March 2004

[3] M. Nørgaard, N. K. Poulsen, and O. Ravn, New developments in state estimation for nonlinear systems: *Automatica*, 36:1627–1638, 2000

[4] R. v. d. Merwe, *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*: PhD thesis, Oregon Health & Science University, Portland, USA, 2004 *http://www.cslu.ogi.edu/publications/*

[5] J. Fox and H. Janocha, A rare-update sigma-point Kalman filter as parameter estimator: In *Proceedings of the IASTED International Conference on Modelling, Identification, and Control (MIC'05)*, pages 190–195, Innsbruck, Austria, February 2005

[6] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology*: IEE Radar, Sonar, Navigation and Avionics Series 5. Peter Peregrinus Ltd., London, UK, 1997

[7] A. Lawrence, *Modern Inertial Technology*: Springer-Verlag, New York, Berlin, 1998.

[8] J. Fox and H. Janocha, Industrial robots as a reference: Static calibration of inertial measurement systems: In *Robotik 2004, VDI-Berichte 1841*, pages 187–194, Munich, Germany, June 2004 paper in German with English abstract.

[9] E.-H. Shin and N. El-Sheimy, A new calibration method for strapdown inertial navigation systems: *zfv – Zeitschrift für Vermessungswesen*, 127(1):41–50, 2002

[10] E. v. Hinüber and H. Janocha, Inertial measurement system for calibration and control of robots: *Industrial Robot*, 20(2):20–27, 1993

[11] J. Fox, Synchronization of input and output measurements using a kalman filter: In *Proceedings of the IASTED International Conference on Modeling, Identification, and Control (MIC'06)*, Lanzarote, Spain, February 2006