A RARE-UPDATE SIGMA-POINT KALMAN FILTER AS PARAMETER ESTIMATOR

Joachim Fox
Laboratory of Process Automation
Saarland University
Saarbrücken, Germany
email: j.fox@lpa.uni-saarland.de

Hartmut Janocha
Laboratory of Process Automation
Saarland University
Saarbrücken, Germany
email: janocha@lpa.uni-saarland.de

ABSTRACT
Extended Kalman Filters (EKF) have widely been used as state estimators in non-linear dynamic systems. In recent years, a new variant of Kalman filtering has emerged. So-called sigma-point Kalman filters (SPKF) exhibit a better accuracy and require less analytical calculations during the design than conventional Kalman filters. As has been done with EKFs, SPKFs can be used to estimate parameters of a system as well. In this work, it will be shown why the application of a SPKF is highly advantageous compared to the EKF in systems where only few update measurements are available while the inputs occur at high frequencies. The calibration of an inertial measurement unit serves as an example for the parameter estimation.

KEY WORDS
identification, estimation, sigma-point Kalman filter, inertial navigation

1 Introduction

The extended Kalman filter (EKF) has long been established as a method for estimating the state of a discrete, non-linear system [1]. The EKF is a suboptimal estimator which is based on the linearisation of the system equations around the current best estimate. Due to its recursive composition, it is suitable for real-time implementation. However, linearisation of the system equations around only one point can result in a biased estimation. A range of variants of the EKF has been suggested in the past. They aimed at improving the filter's estimation accuracy at the expense of the computing time by including higher order terms in the approximation of the system equations. These variants feature a considerable demand on the filter design, since they require the analytic computation of higher derivatives of the system equations. Due to these disadvantages they are rarely used in practice.

The sigma-point Kalman filter (SPKF) is a relatively young class of estimators. One of these filters, the *unscented Kalman filter* (UKF), was first published by Julier and Uhlmann. An overview of the different filters is given in [2, 3]. The SPKF computes a representative set of points in the state space. The system is then propagated using each of these points as a starting point of the trajectories. The spread of the propagated points in the state space allows to determine how the stochastic distribution (described by mean and covariance) of the state has developed as a result of the non-linear transformation represented by the system function. Similar to the Kalman update step, mean and covariance are corrected when new measurements are available. Therefore, these filters feature similarities to the Kalman filter as well as to Monte Carlo methods. Compared to the Monte Carlo methods, their advantage is that the number of points they require is smaller by several orders of magnitude: usually only $2n + 1$ sigma points are needed, whereby n is the dimension of the state vector. Compared to EKFs, their design involves less work, since they do not require analytic computation of the derivatives and still feature a higher precision than EFKs [3]. While the SPKF can be implemented such that the algorithms are of the order $\mathcal{O}(n^3)$, the same as for square-root EKFs, the computing time of an SPKF is generally higher than that of an EKF. This is due to the fact that the system and measurement equations must be computed $2n + 1$ times, once for each sigma point. This disadvantage and the fact that the filters are relatively unknown are probably the reasons why SPKFs have only been used to date in special fields such as speech signal processing [4].

However, some estimation tasks are less time critical: system parameters which are constant over time may be estimated in an identification process prior to system operation and therefore need not be computed online. EKFs have been used for parameter estimation in a wide range of applications but they are known to have convergence problems when too many parameters have to be estimated. In contrast, parameter estimation with SPKFs has shown to be more reliable.

In this work, a special case of parameter estimation is examined where the underlying system is dynamic and non-linear and where measurements that serve as a reference for the estimation (i.e. Kalman update measurements) are rare compared to the dynamic system's inputs. It will be shown that the application of an EKF would require a much more complex filter structure than that of a SPKF.

The following section compares the theoretical bases of EKF and SPKF. In section 3, the SPKF-based parameter estimation with rare updates is explained. Section 4 describes an example for this technique: the SPKF is used to identify the parameters of an inertial measurement unit. Measurement results are presented in section 5. The paper ends with a summary and an outlook.

2 Extended Kalman filters vs. sigma-point filters

This paper deals with a discrete dynamic system which can be described as a first-order Markov system with the following equations:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{n}_k^s & (1) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{n}_k^m & (2) \end{aligned}$$

$\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ are vectorial state, input and output variables, k is the current sample, $\mathbf{n}^s$ and $\mathbf{n}^m$ stand for zero-mean system and measurement noise, and $\mathbf{f}$ and $\mathbf{h}$ 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. $\mathbf{n}_k^s$, $\mathbf{n}_k^m$ and $\mathbf{x}_k$ are independent. Stochastic state estimation is used to compute the current state of the system from the known input and output data, whereby only certain characteristics (e.g. mean and covariance) are given for $\mathbf{n}^s$ and $\mathbf{n}^m$.

The most widely used estimator for nonlinear systems is the EKF. It consists of two steps: an a-priori estimation $\hat{\mathbf{x}}_{k+1}^-$ before a measurement has been taken and an a-posteriori estimation $\hat{\mathbf{x}}_{k+1}^+$ with knowledge of the measurement[1]. This method results in the following equations where $E\{\cdot\}$ stands for the expectation:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, k-1) \\ &\approx E\{\mathbf{x}_k | \mathbf{y}_{k-1}, \mathbf{y}_{k-2}, \ldots\} & (3) \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - \hat{\mathbf{y}}_k] \\ &\approx E\{\mathbf{x}_k | \mathbf{y}_k, \mathbf{y}_{k-1}, \ldots\} & (4) \\ \hat{\mathbf{y}}_k &= \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{u}_k, k) & (5) \end{aligned}$$

The Kalman gain matrix $\mathbf{K}_k$ is computed using the matrix $\hat{\mathbf{P}}_k^-$, an estimation of the state's covariance, given by

$$\hat{\mathbf{P}}_k \approx E\Big\{(\mathbf{x}_k - E\{\mathbf{x}_k\})(\mathbf{x}_k - E\{\mathbf{x}_k\})^T\Big\}. \quad (6)$$

$\hat{\mathbf{P}}_k^{-/+}$ is computed by linearizing the system equations around the current estimate $\hat{\mathbf{x}}_k^-$:

$$\begin{aligned} \hat{\mathbf{P}}_{k+1}^- &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \hat{\mathbf{P}}_k^+ \Big(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big)^T + \mathbf{Z}_k & (7) \\ \hat{\mathbf{P}}_k^+ &= \Big[\mathbf{I} - \mathbf{K}_k \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\Big] \hat{\mathbf{P}}_k^- & (8) \\ \mathbf{K}_k &= \hat{\mathbf{P}}_k^- \Big(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\Big)^T \Big[\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \hat{\mathbf{P}}_k^- \Big(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\Big)^T + \mathbf{W}_k\Big]^{-1} & (9) \end{aligned}$$

where $\mathbf{Z}_k = E\{\mathbf{n}_k^s \mathbf{n}_k^{sT}\}$, $\mathbf{W}_k = E\{\mathbf{n}_k^m \mathbf{n}_k^{mT}\}$ and $\mathbf{I}$ is the identity matrix. Taking a closer look at (9), one can see that this is nothing but a linear approximation of the following equation:

$$\mathbf{K}_k = \mathbf{P}_{xy,k}^- (\mathbf{P}_{yy,k}^-)^{-1} \quad (10)$$

with $\mathbf{P}_{yy}^-$ denoting the covariance matrix of the *innovation* $(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k-1}^-, \mathbf{u}_{k-1}, k-1))$ and $\mathbf{P}_{xy}^-$ the covariance matrix between the error of the a-priori estimate and the output error. Eq. (3) and (7) are referred to as the Kalman prediction step, and (4), (8) and (9) as Kalman update step. For a more detailed discussion, see for example [1].

Since the system equations are linearized around the current estimate, the stochastic distribution of the state of the EKF is disregarded. This can lead to a significant underestimation of the state's covariance matrix and even to a biased estimate: imagine a ball rolling on the ridge of a mountain – since this path is 'unstable', the ball will soon leave the ridge and roll down on the one or the other side. If the starting position of the ball was not known precisely, a Monte Carlo analysis would result in some trajectories on the one side and some on the other. Therefore the covariance of the position would increase largely over time. The EKF however choses only one trajectory and therefore decides for one side, disregarding the other!

New approaches, all of which belong to the group of sigma-point Kalman filters, can solve this problem [2, 3]. The expectation in (6) is no longer being approximated by means of linearisation, as is the case with the EKF, but is 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 (the time index k is omitted from now on)[2]:

$$\begin{aligned} \hat{\mathbf{P}}^- &= \sum_{i=0}^{r-1}\sum_{j=0}^{r-1} w_{ij}^c \mathcal{X}_i \mathcal{X}_j^T + \mathbf{Z} & (11) \\ \hat{\mathbf{P}}_{xy}^- &= \sum_{i=0}^{r-1}\sum_{j=0}^{r-1} w_{ij}^{cc} \mathcal{X}_i \mathcal{Y}_j^T & (12) \\ \hat{\mathbf{P}}_{yy}^- &= \sum_{i=0}^{r-1}\sum_{j=0}^{r-1} w_{ij}^c \mathcal{Y}_i \mathcal{Y}_j^T + \mathbf{W} & (13) \\ \hat{\mathbf{x}}^- &= \sum_{i=0}^{r-1} w_i^m \mathcal{X}_i\,, \quad \hat{\mathbf{y}} = \sum_{i=0}^{r-1} w_i^m \mathcal{Y}_i & (14) \\ \mathcal{Y}_i &= \mathbf{h}(\mathcal{X}_i, \mathbf{u}) & (15) \end{aligned}$$

w_{ij}^c, w_{ij}^{cc} and w_i^m 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 equation (10) with $\hat{\mathbf{P}}_{yy}^-$ and $\hat{\mathbf{P}}_{xy}^-$ from equation (11) to (15) represents a numeric linearization of the process. This linearization, however, includes now r points. The advantage of this method is that the approximation is precise at least up to the second order of the Taylor series (in contrast to

[1] Throughout this paper, variables with a hat $(\hat{\cdot})$ stand for estimated values while non-hatted variables represent the true values.

[2] It has been claimed in the literature ([3, 5]) that both the UKF and the CDKF (see below) can be formulated like the weighted statistical linear regression. However, this does not apply to CDKF if one uses the regression given in [3]. Therefore here, a more general form has been derived which is valid for both filters. This form will be analyzed in detail in upcoming publications.

the first-order EKF approximation). Moreover, the SPKF avoids the analytic computation of the partial derivatives $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ needed for the EKF. If the filter equations are formulated with a root form [4] (analogue to the square-root form of the EKF), the algorithms are of the same order of magnitude as the EKF but still are slower in practice because the system and measurement equations (1) and (2) have to be computed for every sigma point which usually takes longer than one single computation of these equations together with the computation of the derivatives.

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_{ij}^c, w_{ij}^{cc} and w_i^m. The equations for the *central difference Kalman filter* (CDKF) are given here as an example [6][3]:

$$\begin{aligned}
\mathcal{X}_0 &= \bar{\mathbf{x}} \\
\mathcal{X}_i &= \bar{\mathbf{x}} + h(\sqrt{\hat{\mathbf{P}}})_{1:n,i}, \quad (i = 1 \dots n) \\
\mathcal{X}_{i+n} &= \bar{\mathbf{x}} - h(\sqrt{\hat{\mathbf{P}}})_{1:n,i}, \quad (i = 1 \dots n) \\
w_0^m &= \frac{h^2 - n}{h^2}, \quad w_i^m = \frac{1}{2h^2} \quad (i > 0) \\
w_{00}^c &= \frac{h^2 - 1}{h^4} n \\
w_{ii}^c &= w_{ii}^{cc} = \frac{2h^2 - 1}{4h^4} \quad (i = 1 \dots 2n) \\
w_{0i}^c &= w_{i0}^c = \frac{1 - h^2}{2h^4} \quad (i = 1 \dots n) \\
w_{i,i+n}^c &= w_{i+n,i}^c = -\frac{1}{4h^4} \quad (i = 1 \dots n) \\
w_{ij}^{cc} &= -\frac{1}{4h^4} \quad (i, j = 1 \dots 2n,\ i \neq j) \\
w_{0i}^{cc} &= \frac{n - h^2}{2h^4} \quad (i = 1 \dots 2n)
\end{aligned}$$

All other weights w_{ij}^c and w_{ij}^{cc} are zero. The CDKF is based on Stirling's interpolation formula which is used to approximate the system function unlike the EKF which uses a first-order Taylor series. The root of the positive definite matrix $\hat{\mathbf{P}}$ 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 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 [3]. Additionally, h is its only parameter which is why it is easier to tune the CDKF than, for instance, the UKF [4] which has three rather non-intuitive tuning parameters.

[3]The weights given here cannot be found in [3, 6] because there, a different form is used, see footnote 2. Both formulations can be shown to be equivalent.

3 Parameter estimation with rare updates

The EKF has since long been used not only as a state estimator but also as a parameter estimator since parameters can be interpreted as states that are constant over time. If the system and measurement functions $\mathbf{f}$ and $\mathbf{h}$ depend on a set of parameters $\boldsymbol{\xi}$, the system (1)–(2) can be written as

$$\begin{aligned}
\boldsymbol{\xi}_{k+1} &= \boldsymbol{\xi}_k + \mathbf{n}_k^p & (16) \\
\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k, k) + \mathbf{n}_k^s & (17) \\
\mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k, k) + \mathbf{n}_k^m . & (18)
\end{aligned}$$

$\mathbf{n}^p$ is a small noisy input which can be interpreted as a change rate of the parameters. This offers two approaches: first, the state vector $\mathbf{x}$ could be augmented with the parameters $\boldsymbol{\xi}$ which is the standard approach and has been done e.g. in [7, 8]. However, if a SPKF is used, this approach demands $2(n + p) + 1$ sigma points although only p parameters have to be estimated, i.e. the computing time will increase.

Here, a second approach is used. If the states are directly measured, i.e. $\mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k, k) = \mathbf{x}_k$ and $\mathbf{n}^m$ is small such that

$$E\{|\mathbf{n}_{k+1}^m|\} \ll E\{|\mathbf{f}(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, \hat{\boldsymbol{\xi}}_k^+, k) - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}, k) - \mathbf{n}_k^s|\} \quad (19)$$

holds, one can reformulate the parameter estimation problem:

$$\begin{aligned}
\boldsymbol{\xi}_{k+1} &= \boldsymbol{\xi}_k + \mathbf{n}_k^p & (20) \\
\mathbf{y}_{k+1} &= \mathbf{f}(\mathbf{y}_k - \mathbf{n}_k^m, \mathbf{u}_k, \mathbf{n}_k^s, \boldsymbol{\xi}_k, k) & (21)
\end{aligned}$$

This leads to the following filter equations:

$$\begin{aligned}
\hat{\boldsymbol{\xi}}_{k+1}^+ &= \hat{\boldsymbol{\xi}}_{k+1}^- + \mathbf{K}_k[\mathbf{y}_{k+1} - \mathbf{f}(\mathbf{y}_k, \mathbf{u}_k, \boldsymbol{\xi}_k, k)] & (22) \\
\hat{\boldsymbol{\xi}}_{k+1}^- &= \hat{\boldsymbol{\xi}}_k^+, \qquad \hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- = \mathbf{y}_{k+1} & (23) \\
\mathbf{K}_k &= \hat{\mathbf{P}}_{\xi y}^- (\hat{\mathbf{P}}_{yy}^-)^{-1} & (24)
\end{aligned}$$

Since $\boldsymbol{\xi}$ now plays the role of the state in the filter model, $\hat{\mathbf{P}}_{\xi y}^-$ and $\hat{\mathbf{P}}_{\xi\xi}^{-/+}$ correspond to $\hat{\mathbf{P}}_{xy}^-$ and $\hat{\mathbf{P}}^{-/+}$ respectively. But in which cases does (19) hold? Usually, $\mathbf{f}$ is a discrete-time approximation of a differential equation which describes the physical laws the state is subject to, i.e.

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}, k) \approx \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}, t)\, \mathrm{d}t. \quad (25)$$

(here, the variables without index k are the continuous-time correspondents of the discrete-time variables). Now imagine a system where measurements of the input $\mathbf{u}$ are taken at much higher frequencies than of the output $\mathbf{y}$, i.e. updates are *rare* compared to the underlying sample time. One can see that generally, the longer the time interval $t_{k+1} - t_k$ is, the more inaccuarate the numerical integration will be and will at last fulfill condition (19). Therefore in the case of rare updates, it is possible to estimate the parameters of a dynamical system without having to estimate the states as well.

If such an approach is chosen, an EKF can no longer be applied. This is because the derivative of $\mathbf{f}$ in eq. (25) needed to calculate $\hat{\mathbf{P}}_{\xi\xi}^{-/+}$ and $\mathbf{K}$ is actually $\partial\mathbf{f}/\partial\boldsymbol{\xi}$ and depends highly on the inputs $\mathbf{u}$. Since the trajectories of $\mathbf{u}$ are usually not known beforehand and $\mathbf{u}$ is only discretely measured, the computation of these derivatives would have to be performed numerically requiring a great effort which seems not to be beneficial. Unlike this, if one uses the SPKF, the derivatives are calculated inherently which means no changes to the algorithm have to be developed.

Remember that the system noise $\mathbf{n}^s$ has been assumed to be additive. This means that in the new filter, it will be part of the measurement noise $\mathbf{n}^m$ while the new system noise is $\mathbf{n}^p$. Thus the matrix $\mathbf{W}$ has to be replaced by $\mathbf{W}+\mathbf{Z}$ and $\mathbf{Z}$ in (11) is replaced by $\mathbf{Z}_\xi$, a covariance matrix which influences the parameter change rate and which can be used as a tuning parameter. If however, $\mathbf{n}^s$ is non-additive, a strict adherence to the SPKF formalism would require the state $\boldsymbol{\xi}$ to be augmented by the noise[6]. This however is not feasible since $M \cdot n$ dimensions would have to be added where M is the number of time steps between two updates. This would largely increase the state vector and thus the number of sigma-points. Therefore, if non-additive system noise is to be considered, one can only resort to applying the 'additive' filter and using $\mathbf{Z}$ as another tuning parameter.

4 Example: calibration of an inertial measurement unit

In this section, an application of the above described SPKF parameter estimator is outlined. It consists of the calibration of an inertial measurement unit (IMU). Such units are applied e.g. to the navigation of airplanes, submarines and other vehicles. They need at least six sensors to solve the navigation equations: three acceleration sensors as well as three rotation rate sensors. Since the sensors of modern IMUs are fixed to the moving object (so-called strapdown technology) the measured accelerations have to be converted from the body coordinate system (index b) into the so-called navigation system (index N) with axes pointing to North, East and downwards. This is achieved by premultiplying the accelerations with the rotation matrix $\mathbf{R}_N^b$. Because the object rotates with the rates $\boldsymbol{\omega}_b$, $\mathbf{R}_N^b$ changes over time:

$$\dot{\mathbf{R}}_N^b = \mathbf{R}_N^b \boldsymbol{\Omega}_b \tag{26}$$

$\boldsymbol{\Omega}_b$ is a skew symmetric matrix, generated by $\boldsymbol{\omega}_b$. Taking gravity and effects resulting from the earth's rotation rate ε^N into account, the measured rotation rates $\boldsymbol{\omega}_b$ and accelerations $\mathbf{a}_b$ lead to the following equations which display the object's velocity $\mathbf{v}_e^N$ over ground [9]:

$$\begin{aligned}\dot{\mathbf{v}}_e^N &= \mathbf{R}_N^b\mathbf{a}^b - (2\varepsilon^N + \boldsymbol{\omega}_{eN}^N)\times\mathbf{v}_e^N + \mathbf{g}^N &(27)\\ \boldsymbol{\omega}_{eN}^N &= \left[\frac{v_E}{R_0+h}, -\frac{v_N}{R_0+h}, -v_E\tan\frac{L}{R_0+h}\right]^T &(28)\end{aligned}$$

L is the latitude, h the height above sea level and R_0 is the reference radius of the earth. v_N and v_E denote the northern and eastern component of $\mathbf{v}_e^N$. $\mathbf{g}^N$ is the local vector of gravity which also includes the centripetal force of the earth rotation. $\boldsymbol{\omega}_{eN}^N$ describes the rotation of the navigation system compared to a system placed in the center of the earth. In stationary applications such as the IMU calibration procedure below, it is approx. zero. Therefore the state of the navigation system is given by $\mathbf{x} = (\mathbf{R}_N^b, \mathbf{p}^N, \mathbf{v}_e^N)$ (where $\mathbf{p}^N$ is the position), it's inputs $\mathbf{u}$ are accelerations and rates. $\mathbf{f}$ is given by

$$\mathbf{f}(\boldsymbol{\xi}_{k+1}) = \begin{pmatrix} \mathbf{R}_{N,k}^b + \int_{t_k}^{t_{k+1}} \dot{\mathbf{R}}_N^b(\tau)\mathrm{d}\tau \\ \mathbf{p}_k^N + \int_{t_k}^{t_k+1}\left(\mathbf{v}_{e,k}^N + \int_{t_k}^{\tau}\dot{\mathbf{v}}_e^N(\tau')\mathrm{d}\tau'\right)\mathrm{d}\tau \\ \mathbf{v}_{e,k}^N + \int_{t_k}^{t_{k+1}}\dot{\mathbf{v}}_e^N(\tau)\mathrm{d}\tau \end{pmatrix} \tag{29}$$

The integrations have to be carried out numerically. Usually, instead of $\mathbf{R}_N^b$, an equivalent quaternion is used [9]. The system noise $\mathbf{n}^s$ is found in the inputs which cannot be exactly measured, i.e. $\mathbf{u} = \mathbf{u}_{\mathrm{true}} + \mathbf{n}^s$.

In [10] a parametric model for the sensors has been developed. It includes error sources such as non-linearities of a single sensor, cross-correlation between the sensors and errors induced by the excentricity of the accelerometers. The parameters of the sensors are collected in the vector $\boldsymbol{\xi}$. In this application, at least 33 different IMU parameters had to be calibrated (more are possible, depending on the order of the sensor models). In addition to that, the positioning errors of the robot used for the calibration can (and should) be estimated. They are also treated as system parameters.

Usually, rate tables are used to excite the IMU's sensors with defined stationary rates and accelerations (by making use of the gravity) during the calibration process. This method requires special, expensive equipment. In this work, a new calibration method is proposed which uses a standard industrial robot to move the IMU. Update measurements for the SPKF are collected from the robot's controller.

This is a typical example for the above described rare-update filter. Though the robot theoretically could trace the end effector's pose with frequencies close to the IMU's data acquisition rate, standard industrial robots have a low absolute dynamic accuracy and show errors of 10 mm and more. However, during standstill, the accuracy is on the order of 1 mm and moreover, its repeatability can be ten times better. Therefore, updates are taken only when the robot stands still.

The calibration process is schematically depicted in Fig. 1. In a first step, all measurements are taken, i.e. the robot moves the IMU along N_m successive measurement paths. At the end of each path, the robot waits a short while to allow for vibrations to settle, then it sends its actual position to the calibration computer. After all measurements have been collected, the SPKF is run, i.e. a set of $2N_\xi + 1$ sigma points is calculated and the same number of navigation algorithms is started in the pose that was recorded

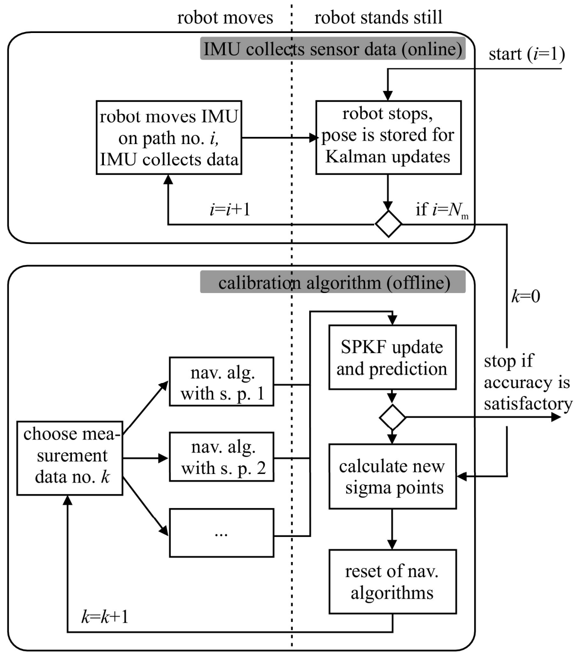

Figure 1. Calibration workflow.

from the robot's control. A SPKF update is performed with the new standstill pose, a prediction step follows immediately. Then, a new set of sigma points can be calculated and the navigation algorithms are initialized for the following path. The algorithm stops if all paths have been used by the SPKF or if the accuracy of the estimated parameters is good enough which can be seen either from the parameters' standard deviation over the last few updates or from the error between the end position of the navigation algorithm based on sigma point $\mathcal{X}_0$ (the current best estimate of the parameters) and the pose where the robot's controller assumes the IMU to be.

Equation (29) displays both the disadvantages and the advantages of the application of the SPKF in comparison to the EKF. Since the measurement equation (21) must be computed for the total of $2N_\xi + 1$ sigma points, the entire navigation computation has to be executed that many times. This is also the reason why the proposed calibration method usually would not be used online. If, on the other hand, an EKF was used, the navigation computation would have to be executed just once: for the current best estimate of the parameters. However, as outlined above, the EKF would have to include orientation, position and velocity into the esitmation because the derivative $\partial \mathbf{f} / \partial \boldsymbol{\xi}$ cannot be computed, so it is not possible to compare the EKF directly to the SPKF. Also, the EKF is known to work well only on systems which are almost linear between two updates but here there are large time periods in between the Kalman updates, ($t_{k+1} - t_k$ covers a time of several seconds), whereas the robot performs, as the case may be, very complex trajectories – thus near-linearity is not to be expected.

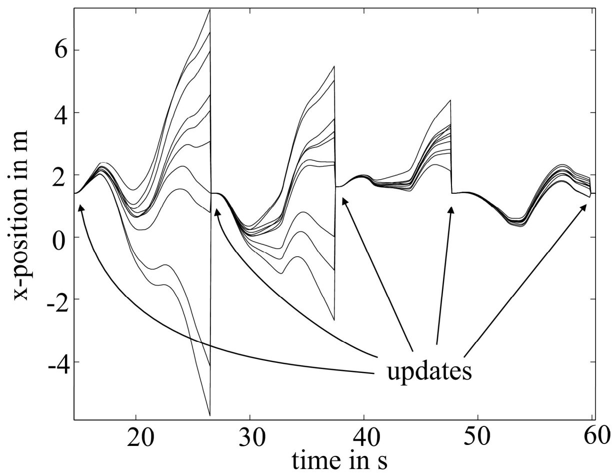

Figure 2. x-coordinate estimation of the IMU based on 10 different sigma points.

5 Experimental results

The above calibration algorithm has been implemented and tested on a high-precision IMU with servo accelerometers from AlliedSignal and ring-laser gyros from Honeywell. The IMU was moved for some 12 minutes by a KUKA industrial robot. Acceleration and rate data was collected with sample rates ranging from 100 Hz to 400 Hz. The different sampling periods did not have a noticeable influence on the calibration results. The robot moved the IMU along 54 paths of 6–20 s duration. Three different update poses were used. The polynomial correction of the sensor data was calculated up to order $N_p = 1$, i.e. biases and scaling factors were estimated.

Figure 2 depicts the parallel computation of the navigation algorithms. It shows the estimated trajectory of the coordinate x of the IMU during the movement of the robot (only four paths can be seen here whilst the whole calibration process consists of 54 paths). The trajectories of 10 different sigma points are plotted (in fact, there are more than 70 sigma points). Note that the true path of the robot is not known (if it were, the updates wouldn't be rare!). At each update point, the trajectories are being reset to the pose delivered by the robot's controller, see eq. (23). Because of the different sensor parameters which are determined by the sigma points, the trajectories drift apart over the course of time. It is obvious that at the end of a path, the error induced by wrong parameters is much larger (at first in the range of meters) than the accuracy of the robot used for updates (in the range of millimeters), i.e. condition (19) holds. As can be seen clearly, the estimated trajectories converge and the position error in the update point gets smaller as the paramterer estimates improve with every update step.

In Fig. 3, the results of the accelerometer scaling factor estimation can be seen. The factors converge but it takes about 150 iterations until they reach the steady state. Since only 54 different paths were measured, the calibration algorithm had been run on the same measurements multiple times to have more iterations.

The convergence behavior of the calibration algorithm shows a clear dependence on the choice of the SPKF matrices $\mathbf{P}_{\xi\xi 0}$ (initial parameter uncertainty), $\mathbf{Z}_\xi$ (parameter change rate) and $\mathbf{Z}$ (noise induced during the calculation of $\mathbf{f}$). While $\mathbf{P}_{\xi\xi 0}$ is mostly known from the sensor specifications, $\mathbf{Z}_\xi$ is a parameter which can be used to tune the filter. Very small values of $\mathbf{Z}_\xi$ were found to give the best convergence behavior in this application but it is to be expected that in applications where the parameters change noticably over time, higher values for this matrix should be used.

The choice of $\mathbf{Z}$ is more sophisticated. As has been outlined above, $\mathbf{Z}$ has to contain not only the uncertainty of the measurement (i.e. the robot's pose repeatability) but also errors that result from noisy accelerometer and gyro measurements. To complicate things further, these noisy measurements are integrated over a long time and along an unknown path. To be exact, one would have to incorporate numerical errors of the navigation algorithms as well. Therefore, as stated earlier, $\mathbf{Z}$ was viewed as an additional tuning parameter. Indeed, it proved that the choice of $\mathbf{Z}$ largely influences the performance of the algorithms. Up to this time, a heuristical approach has been chosen to tune $\mathbf{Z}$, but research focusses on methods to obtain a more precise value of this covariance matrix.

The experiments showed that the proposed SPKF estimation algorithm is capable in principle of calibrating an IMU's parameters. However, the performance of the algorithm is expected to increase largely if more information is provided about the robot's path. Although rather inaccurate, the robot's own measurements (12 ms sampling period) could be used as update information during the first few paths. As soon as the $2N_\xi + 1$ navigation algorithms' paths are close enough to the robot's measurements, one proceeds with the rare-update filtering. This has to be done because the robot's path measurements errors are mostly deterministic (but unknown) and not zero-mean – the application of a Kalman type filter requires zero-mean noise. With thus improved algorithms, it seems promising that high-precision IMUs can be calibrated with the required accuracy in the future by using an SPKF.

6 Conclusion

This paper has shown how a sigma-point Kalman filter can be used for parameter estimation. It has been pointed out that if update measurements are rare compared with the frequency of the inputs, this filter allows the estimaton of parameters without having to estimate the system's states as well. Therefore, the filter's dimension and computing time are reduced. This approach would not be possible with an extended Kalman filter.

Current research focusses on the convergence behavior of the filters. Therefore, variable filter parameters, such

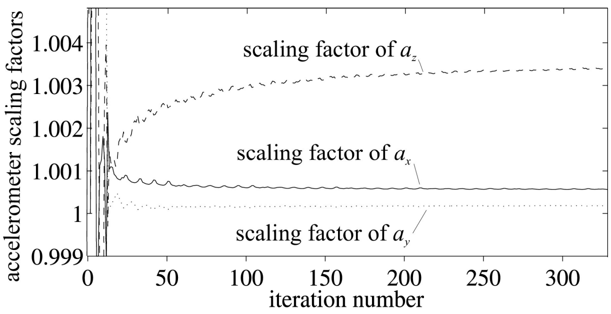

Figure 3. Estimated scaling factors.

as $\mathbf{Z}$, $\mathbf{Z}_\xi$ $\mathbf{P}_{\xi\xi 0}$ and h are investigated. Particularly the choice of a suitable matrix $\mathbf{Z}$ requires that not only the robot's repeatability is known, but also the integration errors which may result, for instance, from the random walk of the sensors. On the other hand, new filter structures are being developed, e.g. a filter which does not generate the sigma points using a Cholesky decomposition, but with a stochastic algorithm in order to avoid a periodic behavior of the estimated parameters in a quasi-stationary state.

The calibration of an inertial measurement unit will be investigated in more detail. The combination of different filter structures, including the here presented, promises faster and more accurate calibration results.

References

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

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

[3] R. v. d. Merwe and E. A. Wan, Sigma-point Kalman filters for probabilistic inference in dynamic state-space models, *Proc. of the Workshop on Advances in Machine Learning*, Montreal, Canada, 2003.

[4] R. v. d. Merwe and E. A. Wan, The square-root unscented Kalman filter for state and parameter-estimation, *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, USA, 2001.

[5] T. Lefebvre, H. Bruyninckx, and J. De Schutter, Comment on 'A new method for the nonlinear transformation of means and covariances in filters and estimators', *IEEE Transactions on Automatic Control*, 47(8), 2002, 1406–1409.

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

[7] L. W. Nelson, The simultaneous on-line estimation of parameters and states in linear systems, *IEEE Transactions on Automatic Control*, February, 1976, 94–98.

[8] E. A. Wan, R. v. d. Merwe, and A. T. Nelson, Dual estimation and the unscented transformation, *Advances in Neural Information Processing Systems*, 12, 2000, 666–672.

[9] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology* (London: Peter Peregrinus Ltd., UK, 1997).

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