

BAYESIAN NONLINEAR FILTERING USING QUADRATURE AND CUBATURE RULES APPLIED TO SENSOR DATA FUSION FOR POSITIONING

Carles Fernández-Prades*, Jordi Vilà-Valls**

* Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)

Av. Carl Friedrich Gauss 7, 08860 Castelldefels, Spain. E-mail: carles.fernandez@cttc.cat

** GIPSA-Lab, Grenoble Institute of Technology (INPG)

BP 46 – 38402 Saint Martin d’Hères, France. E-mail: jordi.vilavalls@gipsa-lab.grenoble-inp.fr

ABSTRACT

This paper shows the applicability of recently-developed Gaussian nonlinear filters to sensor data fusion for positioning purposes. After providing a brief review of Bayesian nonlinear filtering, we specially address square-root, derivative-free algorithms based on the Gaussian assumption and approximation rules for numerical integration, namely the Gauss-Hermite quadrature rule and the cubature rule. Then, we propose a motion model based on the observations taken by an Inertial Measurement Unit, that takes into account its possibly biased behavior, and we show how heterogeneous sensors (using time-delay or received-signal-strength based ranging) can be combined in a recursive, online Bayesian estimation scheme. These algorithms show a dramatic performance improvement and better numerical stability when compared to typical nonlinear estimators such as the Extended Kalman Filter or the Unscented Kalman Filter, and require several orders of magnitude less computational load when compared to Sequential Monte Carlo methods, achieving a comparable degree of accuracy.

Index Terms— Bayesian estimation, Gauss-Hermite quadrature, cubature rules, nonlinear filtering, square-root filters, sensor data fusion, positioning.

I. INTRODUCTION TO BAYESIAN FILTERING

The problem under study, concerns the derivation of efficient methods for online estimation and prediction (filtering), of the time-varying unknown state of a system (state could refer to position, velocity, attitude and so on), along with the continuous flow of information (observations) from the system. A positioning system, is nonlinear and dynamic by nature, and thus the objective is to recursively compute estimates of states at time index k , $\mathbf{x}_k \in \chi$, given measurements $\mathbf{y}_{1:k} \in \mathcal{Y}$ and measured inputs $\mathbf{u}_k \in \mathcal{U}$, where χ , \mathcal{Y} and \mathcal{U}

are their respective supports, not necessarily with the same dimension. State equation models the evolution of target states (on which measurements depend on), as a discrete-time stochastic model. Following a first-order Markov chain approach, this can be modeled by

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_{k-1}), \quad (1)$$

where \mathbf{f}_{k-1} is a known, possibly nonlinear, function of the state \mathbf{x}_k , and \mathbf{v}_{k-1} is referred to as process noise, which gathers any mismodeling effect or disturbances in the state characterization. The relation between the measurements and the state is modeled by

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{n}_k) \quad (2)$$

where \mathbf{h}_k is a known, possibly nonlinear, function and \mathbf{n}_k is referred to as measurement noise. The combination of (1) and (2) leads to a Markov chain whose state is not directly observable but indirectly inferred through measurements, an approach known as Hidden Markov Model. A generic recursive Bayesian filter is depicted in Algorithm 1, where $\mathbb{E}\{\cdot\}$ stands for the statistical expectation operator. The state transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is given by (1), and the old posterior density at time $k-1$, $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1})$, is obtained in the previous update step. From (5) to (6) we have used $p(\mathbf{x}_k|\mathbf{y}_{k-1}, \mathbf{u}_{k-1}, \mathbf{u}_k) = p(\mathbf{x}_k|\mathbf{y}_{k-1}, \mathbf{u}_{k-1})$, which is equivalent to state that the input \mathbf{u}_k can be inferred using $\hat{\mathbf{x}}_{k|k-1}$. This allows the recursive relationship, since this *a priori* density is computed in the prediction step. The likelihood function $p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{u}_k)$ is given by (2).

In this paper, we present the nonlinear Bayesian estimation problem, and we summarize the recently-developed Sigma-Point methods: the Quadrature Kalman Filter (QKF) [1] and the Cubature Kalman Filter (CKF) [2]. We propose a motion model, following a new approach, which includes the biases introduced by the Inertial Measurement Unit, into the state. We consider two different measurement models and we combine them into a nonlinear Bayesian filter. This approach can be useful when having different types of sensors for the same application. These methods, from the

This work has been partially supported by the Spanish Science and Technology Commission: CENIT2007-2002 (TIMI) and TEC2008-02685/TEC (NARRA), and by the European Commission in the framework of the COST Action IC0803 (RFCSET) and the FP7 Network of Excellence in Wireless COMMunications NEWCOM++ (contract n. 216715).

best of our knowledge, has never been used for data fusion for positioning.

II. GAUSSIAN NONLINEAR FILTERING

Algorithm 1 provides an estimation framework that is optimal in the Bayesian sense¹ and enjoys a recursive structure, which is a desirable property for online estimation. In the general case, however, this algorithm cannot be used in practice because the multi-dimensional integrals involved in (4) and (6) are intractable. For Markovian, nonlinear and non-Gaussian state-space models, Sequential Monte Carlo (SMC) methods [3] (also known as Particle Filters), provide a framework to obtain numerical approximations of such integrals by a stochastic sampling approach. This broad suitability comes at the expense of a high computational load, that makes this solution difficult to embed in digital light processors or real-time applications.

An alternative to the SMC methods, to solve the nonlinear filtering problem, are the so-called Sigma-Point Kalman filters (SPKF) [4], [2]. The key assumption here, is to consider that both, process noise \mathbf{v}_k and measurement noise \mathbf{n}_k , are independent random additive processes, with Gaussian distribution. This leads to reckon the state transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ and the measurement likelihood function $p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{u}_k)$ as Gaussian densities, which in turn reverts to a Gaussian posterior density $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k})$. If the state transition function $\mathbf{f}(\cdot)$ and the measurement function $\mathbf{h}(\cdot)$ are both linear, this approach leads to the Kalman filter. In case of dealing with slightly non-linear systems, we can resort to the Extended Kalman Filter (based on the linearization of these functions around the predicted and updated estimates), but this approach fails when nonlinearities are moderate or severe. As shown in [5], for an arbitrary non-linear system and considering the Gaussian assumption, we can derive a recursive algorithm by equating the Bayesian formulae with respect to the first and second moments of the distributions, that is, means and covariances of the conditional densities involved. Since all the appearing multi-dimensional integrals are of the form

$$I = \int_{\mathcal{X}} \text{nonlinear function} \cdot \text{Gaussian density} \cdot d\mathbf{x}, \quad (3)$$

the problem of Bayesian estimation under the Gaussian assumption reduces to a numerical evaluation of such integrals. Possible ways to approximate the definite integral of a function such as (3) are the Gauss-Hermite quadrature rules [6] or the cubature rules [7]. These approaches are based on a weighted sum of function values, at specified (*i.e.*, deterministic) points within the domain of integration, as opposite to the stochastic sampling performed by SMC

¹The characterization of the posterior distribution allows us to compute the Minimum Mean-Squared Error (MMSE), the Maximum a Posteriori (MAP) or the median of the posterior (Minimax) estimators, addressing optimality in many senses.

Algorithm 1 Optimal recursive Bayesian filter structure

1: **for** $k = 1$ to ∞ **do**

2: **Prediction:** estimate the predicted state

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbb{E} \{ \mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1} \} \\ &= \int_{\mathcal{X}} \mathbf{x}_k p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_k \end{aligned}$$

where:

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) &= \int_{\mathcal{X}} p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_{k-1} \\ &= \int_{\mathcal{X}} p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_{k-1} \end{aligned} \quad (4)$$

3: **Update:** update the state estimation with the new measurement and input at time k

$$\hat{\mathbf{x}}_{k|k} = \mathbb{E} \{ \mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k} \} = \int_{\mathcal{X}} \mathbf{x}_k p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}) d\mathbf{x}_k$$

where, applying Bayes' rule:

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}) &= p(\mathbf{x}_k | \mathbf{y}_k, \mathbf{u}_k, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) \\ &= \frac{p(\mathbf{y}_k | \mathbf{x}_k, \mathbf{u}_k) p(\mathbf{x}_k | \mathbf{u}_k, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{u}_k, \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1})} \end{aligned} \quad (5)$$

$$= \frac{p(\mathbf{y}_k | \mathbf{x}_k, \mathbf{u}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1})}{\int_{\mathcal{X}} p(\mathbf{y}_k | \mathbf{x}_k, \mathbf{u}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{u}_{1:k-1}) d\mathbf{x}_k} \quad (6)$$

4: **end for**

methods. A well known method within this family, is the Unscented Kalman Filter (UKF) [8], that can be expressed as a particular case of a quadrature rule if parameters are tuned adequately [5].

A further refinement of these schemes comes from the fact that, when we propagate the covariance matrix through a nonlinear function, the filter should preserve the properties of a covariance matrix, namely, its symmetry, and positive-definiteness. In practice, however, due to lack of arithmetic precision, numerical errors may lead to a loss of these properties. To circumvent this problem, a square-root filter is introduced to propagate the square root of the covariance matrix instead of the covariance itself. Even more, it avoids the inversion of the updated covariance matrix, entailing additional computational saving. Although this idea is not new [9], it has been recently applied to the UKF [10] (although it does not guarantee positive definiteness of the covariance matrix) and, more successfully, to the Square-Root Quadrature Kalman Filter (SQKF)[11] and the Square-Root Cubature Kalman Filter (SCKF) [2].

The resulting procedure is sketched in Algorithm 2, where we present a general framework for square-root, derivative-free nonlinear Bayesian filtering. In steps 8 and 17, $\mathbf{S} = \mathbf{Tria}(\mathbf{A})$ denotes a general triangularization algorithm (for instance, the QR decomposition), where $\mathbf{A} \in \mathbb{R}^{p \times q}$, $p < q$,

Algorithm 2 Square-root, derivative-free nonlinear Kalman Filter

Require: $\mathbf{y}_{1:K}$, $\mathbf{u}_{0:K}$, $\hat{\mathbf{x}}_0$, $\Sigma_{x,0} = \mathbf{S}_{x,0|0} \mathbf{S}_{x,0|0}^T$, $\Sigma_{v,0}$, $\Sigma_{n,0}$.

Initialization:

- 1: Define sigma-points and weights $\{\xi_i, \omega_i\}_{i=1, \dots, L}$ by using Algorithms 3, 4 or any other rule.
- 2: Set $\mathbf{W} = \text{diag}(\sqrt{\omega_i})$

Tracking:

- 3: **for** $k = 1$ to ∞ **do**
 - 4: **Time update:**
 - 5: Evaluate the sigma points:
 $\mathbf{x}_{i,k-1|k-1} = \mathbf{S}_{x,k-1|k-1} \xi_i + \hat{\mathbf{x}}_{k-1|k-1}$, $i = 1, \dots, L$.
 - 6: Evaluate the propagated sigma points:
 $\tilde{\mathbf{x}}_{i,k|k-1} = \mathbf{f}(\mathbf{x}_{i,k-1|k-1}, \mathbf{u}_{k-1})$.
 - 7: Estimate the predicted state:
 $\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^L \omega_i \tilde{\mathbf{x}}_{i,k|k-1}$.
 - 8: Estimate the square-root factor of the predicted error covariance:
 $\mathbf{S}_{x,k|k-1} = \text{Tria} \left(\left[\tilde{\mathcal{X}}_{k|k-1} \ ; \ \mathbf{S}_{\Sigma_{v,k-1}} \right] \right)$, where:
 $\mathbf{S}_{\Sigma_{v,k-1}}$ is a square-root factor of $\Sigma_{v,k-1}$ such that
 $\Sigma_{v,k-1} = \mathbf{S}_{\Sigma_{v,k-1}} \mathbf{S}_{\Sigma_{v,k-1}}^T$, and
 $\tilde{\mathcal{X}}_{k|k-1} = [\tilde{\mathbf{x}}_{1,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \ \cdots \ \tilde{\mathbf{x}}_{L,k|k-1} - \hat{\mathbf{x}}_{k|k-1}] \mathbf{W}$.
 - 9: **Measurement update:**
 - 10: Evaluate the sigma points:
 $\mathbf{x}_{i,k|k-1} = \mathbf{S}_{x,k|k-1} \xi_i + \hat{\mathbf{x}}_{k|k-1}$, $i = 1, \dots, L$.
 - 11: Evaluate the propagated sigma points:
 $\tilde{\mathbf{y}}_{i,k|k-1} = \mathbf{h}(\mathbf{x}_{i,k|k-1}, \mathbf{u}_k)$.
 - 12: Estimate the predicted measurement:
 $\hat{\mathbf{y}}_{k|k-1} = \sum_{i=1}^L \omega_i \tilde{\mathbf{y}}_{i,k|k-1}$.
 - 13: Estimate the square-root of the innovation covariance matrix:
 $\mathbf{S}_{y,k|k-1} = \text{Tria} \left(\left[\mathcal{Y}_{k|k-1} \ ; \ \mathbf{S}_{\Sigma_{n,k}} \right] \right)$, where:
 $\mathbf{S}_{\Sigma_{n,k}}$ denotes a square-root factor of $\Sigma_{n,k}$ such that
 $\Sigma_{n,k} = \mathbf{S}_{\Sigma_{n,k}} \mathbf{S}_{\Sigma_{n,k}}^T$, and
 $\mathcal{Y}_{k|k-1} = [\tilde{\mathbf{y}}_{1,k|k-1} - \hat{\mathbf{y}}_{k|k-1} \ \cdots \ \tilde{\mathbf{y}}_{L,k|k-1} - \hat{\mathbf{y}}_{k|k-1}] \mathbf{W}$.
 - 14: Estimate the cross-covariance matrix
 $\Sigma_{xy,k|k-1} = \mathcal{X}_{k|k-1} \mathcal{Y}_{k|k-1}^T$, where:
 $\mathcal{X}_{k|k-1} = [\mathbf{x}_{1,k|k-1} - \hat{\mathbf{x}}_{k|k-1} \ \cdots \ \mathbf{x}_{L,k|k-1} - \hat{\mathbf{x}}_{k|k-1}] \mathbf{W}$.
 - 15: Estimate the Kalman gain
 $\mathbf{K}_k = \left(\Sigma_{xy,k|k-1} / \mathbf{S}_{y,k|k-1}^T \right) / \mathbf{S}_{y,k|k-1}$.
 - 16: Estimate the updated state
 $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})$.
 - 17: Estimate the square-root factor of the corresponding error covariance:
 $\mathbf{S}_{x,k|k} = \text{Tria} \left(\left[\mathcal{X}_{k|k-1} - \mathbf{K}_k \mathcal{Y}_{k|k-1} \ ; \ \mathbf{K}_k \mathbf{S}_{\Sigma_{n,k}} \right] \right)$.
 - 18: **end for**
-

and \mathbf{S} is a lower triangular matrix.

In Algorithms 3 and 4, we present the way to compute the deterministic samples using, respectively, the Gauss-Hermite quadrature rule and the third-degree spherical-radial cubature rule. The symbol \otimes is the Kronecker tensor product, and we

Algorithm 3 Generation of Sigma-Points and weights for multi-dimensional Gauss-Hermite quadrature rule

- 1: Set the number of points per dimension α , and $L = \alpha^{\dim(\mathbf{x})}$
 - 2: Set $J_{i,i+1} = \sqrt{\frac{i}{2}}$, where $i = 1, \dots, (\alpha - 1)$
 - 3: Compute λ_i , the eigenvalues of \mathbf{J}
 - 4: Set $\xi_i = \sqrt{2} \lambda_i$
 - 5: Set $\omega_i = (\nu_i)_1^2$, where $(\nu_i)_1$ is the first element of the i -th normalized eigenvector of \mathbf{J}
 - 6: **if** $\dim(\mathbf{x}) > 1$ **then**
 - 7: Set $\zeta_{1,:} = \xi_i$ and $\varpi_{1,:} = \omega_i$
 - 8: **for** $m = 2$ to $\dim(\mathbf{x})$ **do**
 - 9: Set $\Xi_{1:m-1,:} = \zeta \otimes \mathbf{1}_{1 \times \alpha}$
 - 10: Set $\Xi_{m,:} = \mathbf{1}_{1 \times \alpha} \otimes \zeta_{m-1,:}$
 - 11: Set $\Omega_{1:m-1,:} = \varpi \otimes \mathbf{1}_{1 \times \alpha}$
 - 12: Set $\Omega_{m,:} = \mathbf{1}_{1 \times \alpha} \otimes \varpi_{m-1,:}$
 - 13: Set $\zeta = \Xi$ and $\varpi = \Omega$
 - 14: **end for**
 - 15: $\xi_i = \Xi_{:,i}$, where $i = 1, \dots, L$
 - 16: $\omega_i = \prod_{l=1}^{\dim(\mathbf{x})} \Omega_{l,i}$
 - 17: **end if**
-

Algorithm 4 Generation of Sigma-Points and weights for third-degree spherical-radial cubature rule

- 1: Set $M = \dim(\mathbf{x})$ and $L = 2M$.
 - 2: Set the cubature points $\xi_i = \sqrt{M} \left[\mathbf{I}_{M \times M} \ ; \ -\mathbf{I}_{M \times M} \right]_i$, where $[\cdot]_{i=1, \dots, L}$ indicates the i -th column.
 - 3: Set the cubature weights $\omega_i = \frac{1}{2M}$, $i = 1, \dots, L$.
-

have used the MATLAB notation for matrices: $\mathbf{A}_{1:i,:}$ refers to the first i rows of matrix \mathbf{A} .

III. MOTION MODELS

The choosing of a motion model will depend on the targeted application, the availability of different technologies and the computational capability. In this paper, we propose a new motion model, inspired in the one presented in [12], based on an intrinsic coordinate system (i.e., the heading angle and the distance moved along the path), and focused on the thrusts applied when turning a vehicle. In an intrinsic coordinate system, applied forces can be represented relative to the heading of the object, rather than relative to the more standard Cartesian or polar fixed coordinate frame. Distance traveled along the path of motion is denoted s , while angle of the path relative to horizontal is denoted ψ . We assume that a piecewise constant thrust, relative to the direction of heading, is applied between any two times τ_k and τ_{k+1} , with tangential component F_T and perpendicular component F_P . We also assume a damping term $\lambda \frac{ds}{dt}$, applied in the opposite direction to the heading. Thus:

$$F_T = \lambda \frac{ds}{dt} + m \frac{d^2 s}{dt^2}, \quad F_P = m \frac{ds}{dt} \frac{d\psi}{dt} \quad (7)$$

where m is the mass of the object and λ the coefficient of resistance. Accelerations tangential to and perpendicular to

the motion are then given by standard dynamics of a point mass:

$$a_T(t) = \frac{d^2 s}{dt^2}, \quad a_P(t) = \frac{ds}{dt} \frac{d\psi}{dt} = \frac{1}{R} \left(\frac{ds}{dt} \right)^2 \quad (8)$$

where R is the instantaneous radius of curvature of the path.

The tangential equation is readily integrated from time τ_k to τ_{k+1} to give the speed $v(t)$ along the path at time $t = \tau_k + T$:

$$v(\tau_k + T) = v_{k+1} = \frac{1}{\lambda} \left(F_{T_k} - (F_{T_k} - \lambda v(\tau_k)) e^{-\frac{T\lambda}{m}} \right) \quad (9)$$

Using (9), the perpendicular equation can be integrated from τ_k to $\tau_k + T$, obtaining

$$\psi(\tau_k + T) = \psi_{k+1} = \psi_k + \frac{F_{P_k}}{F_{T_k}} \left(\frac{\lambda T}{m} - \log \left| \frac{v(\tau_k)}{v(\tau_k + T)} \right| \right) \quad (10)$$

The Cartesian position can be computed by an Euler approximation on a fine time grid, so the changes in x and y coordinates over a time interval T can be expressed as

$$x_{k+1} \approx x_k + v_{k+1} T \cos(\psi_k) \quad (11)$$

$$y_{k+1} \approx x_k + v_{k+1} T \sin(\psi_k) \quad (12)$$

Inertial Measurement Units (IMUs) are known to deliver biased estimations, and these biases are an underlying source of error. We propose to include those biases in the model:

$$\tilde{a}_{T_k} = a_{T_k} + \delta a_{T_k}, \quad \tilde{a}_{P_k} = a_{P_k} + \delta a_{P_k}, \quad \tilde{\psi}_k = \psi_k + \delta \psi_k \quad (13)$$

The state space is then defined as

$$\mathbf{x}_k = [x_k \quad y_k \quad v_k \quad \delta a_{T_k} \quad \delta a_{P_k} \quad \delta \psi_k]^T \quad (14)$$

and the input signal

$$\mathbf{u}_k = [\tilde{a}_{T_k} \quad \tilde{a}_{P_k} \quad \tilde{\psi}_k]^T, \quad (15)$$

and thus the state transition $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is completely defined by equations (7)-(12), plus the assumptions $\delta a_{T_k} = \delta a_{T_{k-1}}$, $\delta a_{P_k} = \delta a_{P_{k-1}}$, and $\delta \psi_k = \delta \psi_{k-1}$.

An advantage of this model is that we can interpolate position, speed, and angle at any time between τ_k and τ_{k+1} , by substituting T by $\Delta\tau$ in the above equations (where $0 < \Delta\tau < T$), thus easing time alignment between inertial measurements and other measurements coming from wireless systems.

The reason for including the path angle bias $\delta \psi_k$ and the acceleration biases δa_{T_k} and δa_{P_k} in the state vector, and the measured angle ψ_k and accelerations a_{T_k} and a_{P_k} in the input signal is due to the well-known behavior of the gyroscopes and accelerometers of an IMU, and the high impact of such biases in dead reckoning methods used for position computation. Inertial measurement units are typically biased, and thus this effect is included in the model.

IV. MEASUREMENT MODEL

We assume a receiver equipped with a set of wireless system interfaces. Measurements taken from these systems (mainly time delay or received power strength) are nonlinearly related to the device position.

If we use technologies in which the natural measurement is time delay, it can be converted to range multiplying by the speed of light c , but taking into account the desynchronization between emitter and receiver (time stamp of measures has been omitted for clarity):

$$y = c \underbrace{(t_{Rx} - t_{Tx})}_d + c(\Delta t_{Rx} - \Delta t_{Tx}) + e \quad (16)$$

where $d = \sqrt{(r_{Tx_x} - r_{Rx_x})^2 + (r_{Tx_y} - r_{Rx_y})^2 + (r_{Tx_z} - r_{Rx_z})^2}$ is the geometric distance, Δt refers to the device's clock bias with respect to an agreed time framework, and e is a random variable modeling noise.

In case of technologies in which we can easily have access to the received signal strength, we will assume a stochastic model for the strength loss:

$$y = P_{Tx} - \bar{P}_L(d_0) - 10n \log_{10} \left(\frac{d}{d_0} \right) + \epsilon, \quad (17)$$

where P_{Tx} is the transmitted power, d_0 is the reference distance, $\bar{P}_L(d_0)$ is the mean loss at d_0 , n is the slope of the loss (depending on the scenario) and ϵ is a random variable modeling noise.

V. COMPUTER SIMULATIONS

In order to provide illustrative numerical results, we have particularized the system model to the following network deployment. Let us assume a 2D field \mathcal{R} , a square centered at $(0, 0)$ with sides of 500 m length, where we have a total number of $N_p = 20$ beacons of a power-based ranging technology (with the measurement function of equation (17)) and $N_t = 5$ beacons of time-based ranging (equation (16)), all with known positions and a coverage of 30 m. We have assumed a random, uniformly distributed beacon deployment over \mathcal{R} . The chosen values for the simulations have been $P_{Tx} = 0$ dBm, $d_0 = 1$ m, $\bar{P}_L(d_0) = 30$ dB, $n = 3$ (a typical value for urban area cellular radio) and ϵ is a random variable with a log-normal distribution with zero mean and $\sigma_\epsilon = 3$ dBm. The noise e is Gaussian with zero mean and $\sigma_e = 1$ m. We have simulated $F_{T_k} \sim \mathcal{N}(\mu_T + m \cdot \delta a_{T_k}, \sigma_T^2)$ and $F_{P_k} \sim \mathcal{N}(m \cdot \delta a_{P_k}, \sigma_P^2)$. The parameters of the motion model and the IMU are $m = 500$ kg, $\lambda = 0.3$, $\mu_T = 200$ N, $\sigma_T = 5000$ N, $\sigma_P = 3000$ N, and $T = 1$ s. Results are averaged over 500 independent computer simulations, each one with a different network deployment and target trajectory. Both filters have been initialized with $\hat{\mathbf{x}}_0 = [0 \ 0 \ 2 \ 0.5 \ 0.5 \ 0.1]^T$, $\Sigma_{x,0} = \text{diag}([10 \ 10 \ 5 \ 1 \ 1 \ 0.1])$, $\Sigma_{v,0} = \text{diag}([1 \ 1 \ 1 \ 0.1 \ 0.1 \ 0.1])$, and $\Sigma_{n,0} = \text{diag}([\sigma_\epsilon \mathbf{I}_{N_p \times N_p} \ \sigma_e \mathbf{I}_{N_t \times N_t}])$.

We compare the results with a Particle Filter (PF) using a residual resampling step [3]. We note that we are using

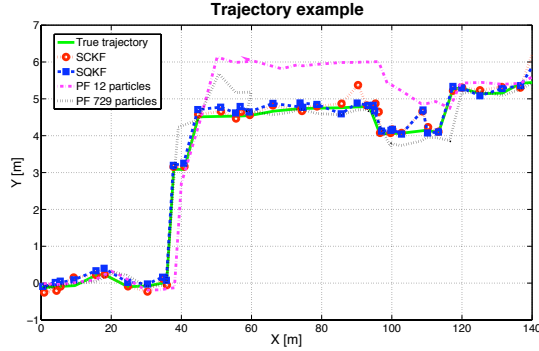


Fig. 1. Example of a true trajectory, and the estimate for a SQKF, a SCKF, and a PF using 12 and 729 particles

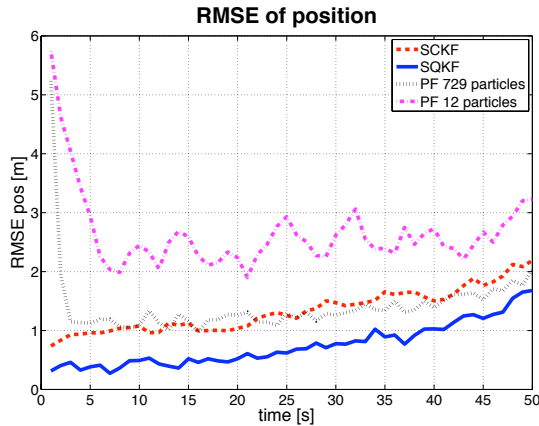


Fig. 2. RMSE for the estimation of the position, for the SQKF, the SCKF and the PF using 729 and 12 particles

$L = 2\dim(\mathbf{x}) = 12$ deterministic samples for the SCKF, and $L = 3^{\dim(\mathbf{x})} = 729$ deterministic samples for the SQKF, which considers 3 samples per dimension. To have a coherent comparison, we consider two PFs, using 12 and 729 particles, respectively. In figure 1, we plot a trajectory example in \mathcal{R} , and the estimated trajectories using both methods. We can see that using both Sigma-point methods, and the PF with 729 particles, we obtain a correct estimation of the trajectory. Using the PF with 12 particles, we obtain a worse estimation, because the PF needs more particles to obtain good results. In figure 2, we plot the RMSE of the estimation of the position. We can see that the deterministic methods obtain better results than the PF. If we have no computational constraints, we will prefer the SQKF, otherwise, the SCKF will be the choice for real-time applications.

VI. CONCLUSIONS

This paper has presented the applicability of Gaussian nonlinear filters to the data fusion of heterogeneous sensors for positioning. We have proposed a state-space approach, which includes the biases introduced by the into

the state, to compensate measurement errors. Our numerical results indicate that the square-root versions of both, the Gauss–Hermite quadrature Kalman filter and the third-degree spherical–radial cubature Kalman filter, achieve good performance in scenarios where the square-root Unscented Kalman filter fails, while maintaining numerical stability. We have also compared the results with a SMC method, which give poorer performances while being computationally heavier. The quadrature filter attains a lower estimation error, while the cubature filter requires much less computational load, being the algorithm of choice for real-time implementations.

VII. REFERENCES

- [1] I. Arasaratnam, S. Haykin, and R. J. Elliot, “Discrete-time nonlinear filtering algorithms using gauss-hermite quadrature,” *Proc. of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.
- [2] I. Arasaratnam and S. Haykin, “Cubature Kalman Filters,” *IEEE Trans. Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.
- [3] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo in Practice*, Springer, 2001.
- [4] R. Van der Merwe, E. A. Wan, and S. J. Julier, “Sigma-Point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation,” in *Proc. of the AIAA GNCC*, Providence, RI, August 2004.
- [5] K. Ito and K. Xiong, “Gaussian filters for nonlinear filtering problems,” *IEEE Trans. on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.
- [6] G. H. Golub, “Some modified matrix eigenvalue problems,” *SIAM Rev.*, vol. 15, no. 2, pp. 318–334, April 1973.
- [7] A. H. Stroud, *Approximate calculation of multiple integrals*, Prentice–Hall, Englewood Cliffs, NJ, 1971.
- [8] S. J. Julier and J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Proc. of the Int. Symp. ADSSC*, 1997, vol. 3.
- [9] J. E. Potter and R. G. Stern and, “Statistical filtering of space navigation measurements,” in *Proc. of the AIAA GNCC*, Cambridge, MA, August 1963.
- [10] R. Van der Merwe and E. A. Wan, “The square-root unscented Kalman filter for state and parameter estimation,” in *Proc. of ICASSP’01*, Salt Lake City, UT, May 2001, vol. 6, pp. 3461–3464.
- [11] I. Arasaratnam and S. Haykin, “Square-Root Quadrature Kalman Filtering,” *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2589–2593, June 2008.
- [12] S. Godsill, J. Vermaak, W. Ng, and J. Li, “Models and algorithms for tracking of maneuvering objects using variable rate particle filters,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 925–952, May 2007.