A Quaternion-based Unscented Kalman Filter for Orientation Tracking

Edgar Kraft

Physikalisches Institut, University of Bonn,
Nussallee 12, 53115 Bonn, Germany
kraft@physik.uni-bonn.de

Abstract − *This paper describes a Kalman filter for the real-time estimation of a rigid body orientation from measurements of acceleration, angular velocity and magnetic field strength. A quaternion representation of the orientation is computationally effective and avoids problems with singularities. The nonlinear relationship between estimated orientation and expected measurement prevent the usage of a classical Kalman filter. This problem is solved by an Unscented Kalman filter which allows nonlinear process and measurement models and is more accurate and less costly than the common Extended Kalman filter. Several extensions to the original Unscented Kalman filter are necessary to treat the inherent properties of unit quaternions. Results with simulated and measured data are discussed.*

Keywords: Tracking, filtering, estimation, Quaternions, Unscented Kalman Filter.

1 Introduction

The determination of a rigid body orientation from various types of measurements is one of the basic problems of all object tracking applications. Yet the algorithmic solutions to this problem still vary widely in terms of accuracy, stability and computational effectiveness. This article describes an approach which combines the benefits of two different key ingredients, quaternions and the Unscented Kalman filter.

The body's orientation is represented by a quaternion q, which is a number with four real components $(q_0, q_1, q_2, q_3) \in \mathbb{R}$.

$$q \equiv q_0 + \mathrm{i}\, q_1 + \mathrm{j}\, q_2 + \mathrm{k}\, q_3 \qquad \text{with} \qquad (1)$$

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} \equiv -1 \qquad \text{and} \qquad (2)$$

$$i \neq j \neq k \qquad (3)$$

i,j and k are three different square roots of -1 (*imaginary unities*) similar to the i known from the complex numbers.

The set of quaternions $\mathbb{H}$ is a superset of the complex numbers $\mathbb{C}$ and the elements can be used to describe spatial rotations similarly to the way complex numbers describe planar rotations. Quaternions offer a singularity-free description (as opposed to Euler angles) and rotations are computed more effectively compared to rotation matrices. A description of quaternion algebra can be found at [1].

The Unscented Kalman filter (UKF) is an extension of the classical Kalman filter to nonlinear process and measurement models. The main difference to the well known Extended Kalman Filter (EKF) is that the UKF approximates the Gaussian probability distribution by a set of sample points whereas the EKF linearises the (nonlinear) model equations. This leads to results which are usually both more accurate (because the original equations are used) and less costly to compute (because no Jacobi matrices need to be calculated) [2].

2 Filter Concept

The filter concept presented here is quite general and applicable to a wide range of possible sensor systems. Its key features are illustrated with the setup used in an early prototype of the BlueTrak user tracking system (now referred to as *tracker*), which is under development at the Silicon Lab of Bonn University[1]. The system comprises sensor groups for acceleration, angular rate and magnetic field strength, each consisting of three sensors at orthogonal angles – a common setup for sourceless orientation measurements. The output of this system is a set of spatial vectors for these quantities, measured with respect to the local (tracker) reference frame. Given this data, the filter computes an estimate of the system state vector.

2.1 State vector

The state vector x of the Kalman filter combines the estimates of the orientation q and the angular velocity $\vec{\omega}$:

$$x = \begin{pmatrix} q \\ \vec{\omega} \end{pmatrix} \qquad (4)$$

It has seven components, four from q (q_0, q_1, q_2, q_3) and three from ω $(\omega_x, \omega_y, \omega_z)$. q is a *unit* quaternion, meaning that it is of size one:

$$||q|| \equiv \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \qquad (5)$$
$$\mathbb{H}_1 \equiv \{q \in \mathbb{H} \mid ||q|| = 1\} \qquad (6)$$

This is denoted by $q \in \mathbb{H}_1$. The constraint on the size simplifies the handling and deprives the quaternion of one degree of freedom. The remaining three degrees of freedom are sufficient to provide a representation for any spatial rotation (actually there are two, because q and $-q$ cause identical rotations). Another side effect of the constraint is that the first four components of the state vector are no longer independent of each other. This causes a conflict with the concept underlying a Kalman filter and the way how noise is treated.

2.2 Process model

The process model A predicts the evolution of the state vector x and describes the influence of the random variable (*process noise*) w.[2] In the generalisation of the classical Kalman filter, the process model equation is given by

$$x_{k+1} = A(x_k, w_k) \qquad (7)$$

A is an arbitrary function of x and w. The process noise w does neither have to be additive, nor does it have to have the same dimension as the state vector x. Equation (7) offers a wide range of possible models and is the basis of the process model of the filter described here. The simplest process model, a static orientation, is only a crude approximation to the expected behaviour of the tracked object (e.g. a human head). Thus the second simplest one, motion with constant angular velocity, is chosen.

$$\omega_{k+1} = \omega_k. \qquad (8)$$

Experimental results show this choice is sufficient for our application to provide a robust prediction, but more elaborate models can easily be implemented.

Given $\vec{\omega}_k$ of the previous state estimate and the length Δt of the time interval, the differential rotation during this interval has the

$$\text{angle:} \qquad \alpha_\Delta = |\vec{\omega}_k| \cdot \Delta t \qquad \text{and the} \qquad (9)$$

$$\text{axis:} \qquad \vec{e}_\Delta = \frac{\vec{\omega}_k}{|\vec{\omega}_k|}. \qquad (10)$$

---

[2]Please note that the filter described here limits itself to pure tracking applications and does thus not consider any control inputs.

$|\vec{\omega}_k|$ is the total angular velocity in $°/\sec$. The corresponding quaternion is given by:

$$q_\Delta = \left[\cos\left(\frac{\alpha_\Delta}{2}\right), \vec{e}_\Delta \sin\left(\frac{\alpha_\Delta}{2}\right)\right] \qquad (11)$$

This notation separates the components of the quaternion into a scalar part (q_0) and a vectorial part (q_1, q_2, q_3), illustrating the similarity between a quaternion and the angle/axis notation of the orientation. q_Δ is used to calculate the new orientation q_{k+1} from the previous one by multiplying:

$$q_{k+1} = q_k q_\Delta \qquad (12)$$

This *quaternion multiplication* is performed according to equations (1) to (3). It is different from a simple product of the corresponding components [1]. The resulting rotation q_{k+1} is equivalent to the rotation q_Δ followed by the rotation q_k.

Equations (8) through (12) define the process model $A()$ for the undisturbed state vector. It has already been noted that the state vector has only got six degrees of freedom. Therefore we choose to describe the influence of the process noise with a six-dimensional noise vector w_k. Its first three components (denoted by $\vec{w}_q$) affect the orientation, the last three components ($\vec{w}_\omega$) affect the angular velocity:

$$w_k = \begin{pmatrix} \vec{w}_q \\ \vec{w}_\omega \end{pmatrix} \qquad (13)$$

Despite the values of w_k change with every time step, the subscript k is omitted in the q- and ω- components for matters of readability. The covariance Q of the random variable w is a measure of the rate at which the uncertainty of the system state estimate increases with time. In the simplest case, the $\vec{w}_\omega$-part can be assumed to be additive, meaning that the uncertainty in the angular velocity increases for example at a rate of $10°/s$ per time interval. Similarly, $\vec{w}_q$ causes an increase of the uncertainty in the orientation (degrees per time interval). Since $\vec{w}_q$ is a three dimensional noise vector, it can not simply be added to the four component quaternion. It has to be converted into a unit quaternion.

Let the random variable $\vec{w}_q$ follow a normal distribution with covariance Q_q (a 3,3-matrix) and mean 0. The vector $\vec{w}_q$ can be regarded as a *rotation vector*. This means that $\vec{w}_q$ represents a random rotation with the

$$\text{angle:} \qquad \alpha_w = |\vec{w}_q| \qquad \text{and the} \qquad (14)$$

$$\text{axis:} \qquad \vec{e}_w = \frac{\vec{w}_q}{|\vec{w}_q|}. \qquad (15)$$

The quaternion representation q_w of this rotation is:

$$q_w = \left[\cos\left(\frac{\alpha_w}{2}\right), \vec{e}_w \sin\left(\frac{\alpha_w}{2}\right)\right] \qquad (16)$$

This *noise quaternion* q_w is the equivalent to the noise vector $\vec{w}_q$. It can be applied to the original quaternion component q_k of the state vector estimate, resulting in a *disturbed* quaternion $\widetilde{q_k}$:

$$\widetilde{q_k} = q_k \, q_w \tag{17}$$

The respective term for the disturbed angular velocity part $\widetilde{\vec{\omega}_k}$ is given by:

$$\widetilde{\vec{\omega}_k} = \vec{\omega}_k + \vec{w}_\omega \tag{18}$$

Combining both equations, the disturbed state vector is:

$$\widetilde{x_k} = \left(\begin{array}{c} \widetilde{q_k} \\ \widetilde{\vec{\omega}_k} \end{array} \right) = \left(\begin{array}{c} q_k \, q_w \\ \vec{\omega}_k + \vec{w}_\omega \end{array} \right) \tag{19}$$

The full process model equations can be obtained by replacing the components of the undistorted vector in equations (12) and (8) with those of the disturbed state vector:

$$q_{k+1} = \widetilde{q_k} \, q_\Delta = q_k q_w q_\Delta \tag{20}$$
$$\vec{\omega}_{k+1} = \widetilde{\vec{\omega}_k} = \vec{\omega}_k + \vec{w}_\omega \tag{21}$$

This implies that we choose to apply the process noise *before* the process model. Finally, the process model $A()$ is given by

$$x_{k+1} = A(x_k, w_k) = \left(\begin{array}{c} q_k q_w q_\Delta \\ \vec{\omega}_k + \vec{w}_\omega \end{array} \right) \tag{22}$$

with q_w from equations (14) to (16) and q_Δ as defined in (9) to (11).

2.3 Measurement Model

The measurement model H relates the measurement value z to the value of the state vector x and describes the influence of a random variable v (measurement noise) on the measured value. The generalised form of the model equation is (in analogy to (7)):

$$z_k = H(x_k, v_k) \tag{23}$$

Since the tracking system produces three different types of measurements, three models $H_1..H_3$ are defined corresponding to the measurements ($z_{rot}, z_{acc}, z_{mag}$). The angular velocity $\vec{\omega}$ is already part of the state vector, leading to the simplest possible model:

$$H_1: \qquad \vec{z}_{rot} = \vec{\omega}_k + \vec{v}_{rot} \tag{24}$$

$\vec{v}_{rot}$ is the measurement noise of the angular velocity.

Let $\vec{g}$ be the vector of the gravitational field ("down") and $\vec{b}$ be the vector of the magnetic field ("north"). The expected measurements of these fields are given by the transformation of $\vec{g}$ and $\vec{b}$ to the tracker coordinate system. In order to express this notion in terms of quaternions, we first have to introduce the identification of three dimensional vectors with so called *vector quaternions*. The vector

$$\vec{g} = \left(\begin{array}{c} g_x \\ g_y \\ g_z \end{array} \right) \tag{25}$$

will be identified with the corresponding vector quaternion

$$g \equiv (0, [g_x, g_y, g_z]) \equiv (0, \vec{g}) \tag{26}$$

and vice versa. A vector quaternion is a quaternion with a scalar value of zero. There is no restriction on the size of the other components. Using this identity, the transformation of $\vec{g}$ and $\vec{b}$ from the global coordinate system to the tracker coordinate system $(\vec{g'}, \vec{b'})$ can be written as:

$$g' = q_k \, g \, q_k^{-1} \tag{27}$$
$$b' = q_k \, b \, q_k^{-1} \tag{28}$$

g, g', b and b' are vector quaternions. All multiplications are quaternion multiplications. These equations are typical examples of the computation of spatial rotations using quaternions [1]. The measurement models for acceleration and magnetic field strength are hence:

$$H_2: \qquad \vec{z}_{acc} = \vec{g'} + \vec{v}_{acc} \tag{29}$$
$$H_3: \qquad \vec{z}_{mag} = \vec{b'} + \vec{v}_{mag} \tag{30}$$

$\vec{v}_{acc}$ and $\vec{v}_{mag}$ are the measurement noise variables of the acceleration and magnetic field. Note that all three functions H_1, H_2 and H_3 are functions of the state vector x_k, even though H_1 uses only the ω_k-component and H_2 and H_3 use only the q_k-component.

2.4 Problematic Properties

Although the described models are reasonably simple, there are two major problems which prevent an implementation of these models with a classical Kalman filter:

The first problem is that H_2 and H_3 are nonlinear functions of the state vector x_k (more precisely, the q_k-component). Even though $\vec{g'}$ of equation (27) is a linear function of $\vec{g}$ (the rotation could also be expressed with a rotation matrix), there is generally no matrix H which fulfills

$$g' = H \, q_k \qquad \text{or} \qquad g' = H \, x_k. \tag{31}$$

This means that $\vec{g'}$ is not a linear function of the *rotation q_k*.

The second problem is the mismatch of state vector dimension and its number of degrees of freedom. This leads to a non-additive, six-dimensional process noise that cannot be described by the process equation of the classical linear Kalman filter.

The solution to these problems is the usage of an extension to the classical Kalman filter, namely the Unscented Kalman filter. This filter bases on the generalised equations (7) and (23), allows nonlinear measurement and process models and offers the necessary flexibility in dealing with the random variables v and w.

3 The Unscented Kalman Filter

From an abstract point of view, the Kalman filter operates on a probability distribution in the n-dimensional state vector space. This distribution is characterised by its first two statistical moments: mean and covariance. The process and measurement models transform this distribution. They project the distribution ahead in time or into the m-dimensional measurement vector space. New values for mean and covariance are computed to parameterise the transformed distributions. Finally, the residual between the predicted measurement and the measured value is used to compute an update of the probability distribution. The extend of this update is determined by the Kalman gain K, a m,n-matrix whose components depend on the relative size of the covariances of estimate and measurement. K performs two functions: it weighs the update and transforms the residual from measurement space to state vector space.

Both the classical and the Unscented Kalman filter follow this underlying scheme, yet they do so in a different manner. A detailed discussion of the differences and similarities can be found in [2] and lies beyond the focus of this paper.

3.1 Sigma Points

W, P is built from rotation vector

In the beginning of every UKF recursion the previous estimates of the state vector $\hat{x}_{k-1}$ and its covariance P_{k-1} are known. The n,n-matrix P_{k-1} is transformed into a set $\{\mathcal{X}_i\}$ of 2n *sigma* points, a set of state vectors which has the same mean and covariance as P_{k-1}. This is done by computing a matrix S with the property

$$P_{k-1} = S^T S. \tag{32}$$

S is called "square root" of P_{k-1}, because if P_{k-1} is just a number (a 1,1-matrix), then S equals the usual $\sqrt{P_{k-1}}$. Since P_{k-1} is a covariance matrix, it happens to be symmetric and positive definite. This allows us to use a *Cholesky Decomposition* to compute S. The n column vectors of S are multiplied by $\pm\sqrt{2n}$ and form the set $\{\mathcal{W}_i\}$. This set has $2n$ elements, which are distributed around the mean value zero (positive and the square root cancel each other) with the covariance P_{k-1}. We can shift the mean to the desired value (in this case $\hat{x}_{k-1}$) by adding it to every element, thus creating the set of sigma points:

$$\mathcal{X}_i = \hat{x}_{k-1} + \mathcal{W}_i \tag{33}$$

The original mean value $\hat{x}_{k-1}$ can be included into the set of sigma points (with an arbitrary weighting factor κ) without affecting the mean value or the covariance. The effect of κ on the filter performance is summarised in [2]. It is omitted here for matters of clarity.

3.2 Quaternion Sigma Points

Note that the elements of $\{\mathcal{W}_i\}$ are similar to the process noise random vector w of equation (13), which also

has a mean of zero but a covariance of Q. If we interpret the sigma points as a set of *disturbed* state vectors, then equation (33) describes how these disturbed state vectors are build from the set $\{\mathcal{W}_i\}$. In our specific case (unit quaternion in state vector) we already know a simple addition of the random variable to the state vector is not possible because the dimensions do not match. Instead, the sigma points have to be calculated in the way described for the process noise, similar to equation (19).

$$\mathcal{X}_i = \left(\begin{array}{c} q_{k-1}\, q_{\mathcal{W}} \\ \vec{\omega}_{k-1} + \vec{\omega}_{\mathcal{W}} \end{array} \right) \tag{34}$$

$q_{\mathcal{W}}$ is the quaternion corresponding to the first three components of $\mathcal{W}_i$, $\vec{\omega}_{\mathcal{W}}$ denotes the angular velocity vector built from the remaining three components, q_{k-1} and $\vec{\omega}_{k-1}$ are part of the previous estimate $\hat{x}_{k-1}$.

This treatment of the noise has some effect on the dimensionality of the vectors and covariances involved. The vectors $\mathcal{W}_i$ are six-dimensional, because they have the same dimension as the process noise vector. Their covariance P_k is hence a 6,6-matrix. The sigma points $\mathcal{X}_i$, on the other hand, are state vectors with a quaternion component and are therefore seven dimensional. Equation (34) performs the transition between the six and the seven dimensional set. This transition bases on the conversion of a rotation vector representation to a quaternion representation, as given in equations (14) to (16).

As described in section 2.2, we choose to apply the process noise (with covariance Q) *before* the process model. Equations (34) and (19) show that similar steps occur during the transformation of the state vector covariance P_k into the sigma points and in the application of the process noise to an undisturbed state vector. Therefore we can combine both steps and include the process noise into the set of sigma points by adding Q to P_k *before* the sigma points are calculated. This is denoted by

$$S = \sqrt{P_{k-1} + Q} \qquad \text{and} \tag{35}$$
$$\mathcal{W}_{i,i+n} = \text{columns}(\pm\sqrt{2n \cdot (P_{k-1} + Q)}). \tag{36}$$

A more elaborate treatment of noise using augmented state vectors is described in [2].

3.3 Transformations of the sigma points

After the sigma points $\{\mathcal{X}_i\}$ are obtained, the process model is used to project each point ahead in time, resulting in a different set of state vectors $\{\mathcal{Y}_i\}$.

$$\mathcal{Y}_i = A(\mathcal{X}_i, 0) \tag{37}$$

Note that no additional noise vector is being considered in the equation above (denoted by the second argument being 0), because the influence of the process noise is already represented in the distribution of the sigma points. Equation (37) is equal to the left part of the process model equations

(20) and (21), with $\mathcal{X}_i$ replacing the disturbed state vector $\widetilde{x_{k-1}}$.

The set $\{\mathcal{Y}_i\}$ samples the probability distribution of the *a priori* estimate. $\hat{x}_k^-$ is defined as the mean value of this distribution and P_k^- is the covariance. Sections 3.4 and 3.5 describe how these values are computed from the set.

$$\hat{x}_k^- = \text{mean}(\{\mathcal{Y}_i\}) \qquad (38)$$
$$P_k^- = \text{covariance}(\{\mathcal{Y}_i\}) \qquad (39)$$

Usually the a priori estimate is computed when a new measurement occurs. A measurement update step requires both the a priori estimate and an estimate z_k^- of the measurement. The set $\{\mathcal{Y}_i\}$ is thus transformed further by the measurement model H, resulting in a set $\{\mathcal{Z}_i\}$ of projected measurement vectors.

$$\mathcal{Z}_i = H(\mathcal{X}_i, 0) \qquad (40)$$

The second argument is zero, since zero is the expectation value of the random variable v. The mean value of the set is the expected measurement vector z_k^-.

$$z_k^- = \text{mean}(\{\mathcal{Z}_i\}) \qquad (41)$$
$$P_{zz} = \text{covariance}(\{\mathcal{Z}_i\}) \qquad (42)$$

In a classical Kalman filter, the measurement prediction is simply $z_k^- = H\hat{x}_k^-$. The UKF uses the mean value of the projected distribution instead, so that generally

$$z_k^- \neq H(\hat{x}_k^-). \qquad (43)$$

The explanation for this is, that the mean of the projected distribution is generally unequal to the projected mean of the original distribution. Both are identical if $H()$ is linear - like in the classical Kalman filter. The *innovation* ν_k is the difference between the actual measurement z_k and its predicted value z_k^-.

$$\nu_k = z_k - z_k^- \qquad (44)$$

Its expected covariance $P_{\nu\nu}$ is the sum of the projected state vector covariance (the uncertainty in the measurement caused by the uncertainty in the state vector prediction) P_{zz} and the measurement noise covariance R (the additional uncertainty induced by the measurement process).

$$P_{\nu\nu} = P_{zz} + R \qquad (45)$$

The *a posteriori* estimate $\hat{x}_k$ is finally computed by adding the a priori estimate to the innovation multiplied by the Kalman gain K_k

$$\hat{x}_k = \hat{x}_k^- + K_k\nu_k. \qquad (46)$$

These transformations are illustrated in figure 1.

3.4   Computation of the mean

If the state vector is element of a vector space (which is the case in the original concepts of Kalman filter and UKF, but not in our application), the mean value is simply the sum over all elements of the set divided by the number of addends ($2n$). This is called the *barycentric* mean.

$$\overline{\mathcal{Y}} = \frac{1}{2n} \sum_{i=1}^{2n} \mathcal{Y}_i \qquad (47)$$

It can be used for the set of measurement vectors $\{\mathcal{Z}_i\}$ and for the angular velocity component of $\{\mathcal{Y}_i\}$.

$$z_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \mathcal{Z}_i \qquad (48)$$

The orientation component of $\{\mathcal{Y}_i\}$ is more difficult, because orientations are periodic. In other words, they are members of a homogenous Riemannian manifold (the four dimensional unit sphere) but not of a vector space. Equation (47) does not yield correct results, as can be seen from an example with two rotations around the x-axis: The calculation with $\{-178°, 180°\}$ gives $1°$, but the expected result is $-179°$. A different argument is that the mean of two unit quaternions has to be a unit quaternion, too. The result of equation (47) with the quaternions $\{(0,0,0,-1),(0,0,0,1)\}$ does obviously not fulfill this requirement.

Our approach to this problem uses the intrinsic gradient descent algorithm described in [3]. The key ingredient is the definition of a new metric which describes the distance between two elements. For orientations, we use the angle θ of the rotation which turns one orientation into the other. Given two quaternions q_1, q_2, the rotation q_{12} which fulfills

$$q_2 = q_{12}\, q_1 \qquad (49)$$

is simply given by

$$q_{12} = q_2\, q_1^{-1}. \qquad (50)$$

θ can be calculated from the scalar part of q_{12}, see (11).

$$\theta = 2 \cdot \arccos(q_{\text{w}}) \qquad (51)$$

Starting with an arbitrary orientation, the estimate of the mean orientation $\overline{q}$ is iterated (iteration steps are denoted by the index t). In each step, the so called *error vectors* are computed for every set element. An error vector $\vec{e}_i$ is the rotation vector corresponding to the relative rotation between the set element q_i and the estimated mean of the last iteration $\overline{q}_t$. The quaternion representation e_i of $\vec{e}_i$ is

$$e_i = q_i\, \overline{q}_t^{-1} \qquad \text{so that} \qquad (52)$$
$$q_i = e_i\, \overline{q}_t. \qquad (53)$$

In other words: e_i rotates the mean into the set element q_i. The barycentric mean $\vec{e}$ of all error vectors

$$\vec{e} = \frac{1}{2n} \sum_{i=1}^{2n} \vec{e}_i \qquad (54)$$

is a measure of the deviation between the estimated mean and the real mean orientation. $\vec{e}$ is a rotation vector that points in the direction of the real mean. The corresponding quaternion e can hence be used to calculate a better estimate for the next iteration step:

$$\overline{q}_{t+1} = e\,\overline{q}_t \qquad (55)$$

Since $\vec{e}$ is used to adjust the estimated mean from iteration to iteration it is called the *adjustment vector*. Its value is zero if $\overline{q}_t$ is equal to the real mean of the set of orientations. The size of the adjustment vector can consequently be used to stop the iteration once a satisfactory precision is achieved. Note that the result of equation (54) does not suffer from the periodicity problem if $\overline{q}_t$ is close to the real mean.

$$\overline{q} = \lim_{t \to \infty} \overline{q}_t \qquad (56)$$

The error vectors $\vec{e}_i$ are introduced solely for the iterative mean finding algorithm. Interestingly, the set $\{\vec{e}_i\}$ of the final iteration finds more use during the computation of the covariance (see there).

Switching back and forth between error vectors and their quaternion representation could be avoided by replacing the sum of equations (54) by a series of quaternion multiplications. Yet even though the above described method might seem complex or confusing, it is expected to be computationally more effective since the sum of vectors is cheaper to compute than a series of quaternion multiplications.

The starting value of the iterations is arbitrary, but it determines the number of iterations needed to reach the desired precision. It is therefore feasible to include the previous state vector estimate $\hat{x}_{k-1}$ into the set of sigma points (as mentioned above) and use its quaternion as a start value.

3.5 Computation of the Covariance

Let $\{\mathcal{X}_i\}$ be a set of $2n$ (vector space) elements with a mean $\overline{x}$. The covariance of this set is given by

$$P = \frac{1}{2n} \sum_{i=1}^{2n} [\mathcal{X}_i - \overline{x}][\mathcal{X}_i - \overline{x}]^T \qquad (57)$$

The term $[\mathcal{X}_i - \overline{x}]$ is the difference between the sigma point and the mean of the distribution. If $\{\mathcal{X}_i\}$ is the untransformed set of equation (33), there are the correspondences

$$\overline{x} \quad \leftrightarrow \quad \hat{x}_{k-1} \qquad \text{and} \qquad (58)$$
$$[\mathcal{X}_i - \overline{x}] \quad \leftrightarrow \quad W_i \qquad \text{and} \qquad (59)$$
$$P \quad \leftrightarrow \quad P_{k-1}. \qquad (60)$$

3.5.1 A Priori State Vector Covariance

The first matrix which has to be calculated from a set of sigma points is the a priori state vector covariance P_k^-.

$$P \leftrightarrow P_k^- \qquad (61)$$

P_k^- is a 6,6-matrix and is the covariance of the set $\{\mathcal{Y}_i\}$ in the same sense in which P_{k-1} is the covariance of the untransformed set $\{\mathcal{X}_i\}$. The corresponding mean value of this set is the a priori state vector

$$\overline{x} \leftrightarrow \hat{x}_k^-. \qquad (62)$$

In analogy to (59) we can replace the term $[\mathcal{X}_i - \overline{x}]$ by a six-dimensional vector $\mathcal{W}_i'$. The prime sign denotes the fact that this vector relates to the transformed set $\{\mathcal{Y}_i\}$.

$$[\mathcal{X}_i - \overline{x}] \leftrightarrow \mathcal{W}_i' \qquad (63)$$

Using this notation, equation (57) becomes

$$P_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \mathcal{W}_i' \, \mathcal{W}_i'^T \qquad (64)$$

Similar to $\mathcal{W}_i$, $\mathcal{W}_i'$ has a rotation vector component $\vec{r}_{\mathcal{W}'}$ and an angular velocity vector component $\vec{\omega}_{\mathcal{W}'}$

$$\mathcal{W}_i' = \begin{pmatrix} \vec{r}_{\mathcal{W}'} \\ \vec{\omega}_{\mathcal{W}'} \end{pmatrix} \qquad (65)$$

$\vec{\omega}_{\mathcal{W}'}$ is the (standard vectorial) difference of the angular velocity components of $\mathcal{Y}_i$ and $\hat{x}_k^-$, denoted by

$$\vec{\omega}_{\mathcal{W}'} = \vec{\omega}_i - \overline{\vec{\omega}} \qquad (66)$$

$\vec{r}_{\mathcal{W}'}$ is a representation of the rotation which turns the orientation part of $\hat{x}_k^-$ into $\mathcal{Y}_i$. The corresponding quaternion $r_{\mathcal{W}'}$ of this rotation is

$$r_{\mathcal{W}'} = q_i \, \overline{q}^{-1} \qquad (67)$$

Comparing this equation to equation (52), we find that $\vec{r}_{\mathcal{W}'}$ was already calculated in form of the error vector $\vec{e}_i$ during the last iteration of the mean finding algorithm.

3.5.2 Measurement Estimate Covariance

The uncertainty P_{zz} of the predicted measurement z_k^- originates in the uncertainty P_k^- of the predicted state vector. It is the covariance of the set $\{\mathcal{Z}_i\}$. Since the measurement vectors are elements of a vector space, their covariance can be calculated similarly to (57):

$$P_{zz} = \frac{1}{2n} \sum_{i=1}^{2n} [\mathcal{Z}_i - z_k^-][\mathcal{Z}_i - z_k^-]^T \qquad (68)$$

As stated in equation (45), the covariance $P_{\nu\nu}$ of the innovation is the sum of P_{zz} and the measurement noise covariance R. $P_{\nu\nu}$ is not specific to the UKF, even though it is not computed explicitly in the classical Kalman filter.

	$P_{\nu\nu}$	$=$			
Unscented KF:			P_{zz}	$+R$	(69)
classical KF:			$HP_k^- H^T$	$+R$	

3.5.3 Cross correlation matrix

The cross correlation matrix P_{xz} is used in the UKF for the calculation of the Kalman gain K_k (see section 3.6). It relates the noise in the state vector to the noise in the measurement and is computed from the sets $\{\mathcal{W}_i'\}$ and $\{\mathcal{Z}_i\}$ according to

$$P_{xz} \;=\; \frac{1}{2n}\sum_{i=1}^{2n}[\mathcal{W}_i'][\mathcal{Z}_i - z_k^-]^T \tag{70}$$

$$\;\hat{=}\; \frac{1}{2n}\sum_{i=1}^{2n}[\mathcal{Y}_i - \hat{x}_k^-][\mathcal{Z}_i - z_k^-]^T \tag{71}$$

The latter equation can be used if the elements of $\{\mathcal{Y}_i\}$ are elements of a vector space. In our application the term in the first brackets would produce invalid results, consequently $\mathcal{W}_i'$ is used. P_{xz} corresponds to the matrix $P_k^- H^T$ of the classical Kalman filter.

3.6 Kalman Gain and Update Equations

In the UKF, the Kalman gain K_k is given by

$$K_k = P_{xz}\, P_{\nu\nu}^{-1}. \tag{72}$$

Note that the corresponding equation of the classical Kalman filter is very similar:

$$K_k = P_k^- H^T \left(H P_k^- H^T + R\right)^{-1} \tag{73}$$

The update equation of the *a posteriori* estimate is identical for both filters:

$$\hat{x}_k = \hat{x}_k^- + K_k \nu_k \tag{74}$$

The covariance update equation is also equivalent in both filters, as can easily be shown with the correspondences described above.

$$P_k \;=\; P_k^- - K_k\, P_{\nu\nu}\, K_k^T \tag{75}$$
$$\;\hat{=}\; (I - K_k\, H)\, P_k^- \tag{76}$$

Equations (74) and (75) are the final steps of an UKF filter cycle. The updated values $\hat{x}_k$ and P_k become the basis of the next cycle. This concludes the description of the key steps of an UKF for the orientation estimation problem.

3.7 Summary

The filter cycle of a Quaternion-Based Unscented Kalman Filter comprises the following steps:

1. The sum of previous estimate error covariance P_{k-1} and process noise covariance Q is transformed into a set $\{\mathcal{W}_i\}$ of $2n$ six-dimensional vectors. This set is distributed around zero with the covariance $P_{k-1} + Q$.

2. The previous state estimate $\hat{x}_{k-1}$ is applied to $\{\mathcal{W}_i\}$, resulting in the set $\{\mathcal{X}_i\}$ of $2n$ seven-dimensional state vectors (sigma points).

3. The process model $A()$ transforms $\{\mathcal{X}_i\}$ into $\{\mathcal{Y}_i\}$.

4. The a priori estimate $\hat{x}_k^-$ is computed as the mean of the transformed sigma points $\{\mathcal{Y}_i\}$.

5. The set $\{\mathcal{Y}_i\}$ is transformed into the six-dimensional set $\{\mathcal{W}_i'\}$ by first *removing* the mean vector $\hat{x}_k^-$ from each element and then converting the quaternion part into a rotation vector.

6. The a priori process covariance P_k^- is computed from $\{\mathcal{W}_i'\}$. This concludes the time update step ("prediction").

7. One of the three measurement models H_1, H_2 or H_3 is used to project the sigma points $\{\mathcal{Y}_i\}$ into the three-dimensional measurement space, yielding the set $\{\mathcal{Z}_i\}$.

8. The mean of $\{\mathcal{Z}_i\}$ is computed, giving the measurement estimate z_k^-. This is compared to the actually measured value z_k, their difference being ν_k, the *innovation*.

9. The innovation covariance $P_{\nu\nu}$ is determined by adding the measurement noise R to the covariance P_{zz} of the set $\{\mathcal{Z}_i\}$.

10. The cross correlation matrix P_{xz} is computed from the sets $\{\mathcal{W}_i'\}$ and $\{\mathcal{Z}_i\}$.

11. The Kalman gain K_k is first computed from P_{xz} and $P_{\nu\nu}$ and then used to calculate the a posteriori estimate $\hat{x}_k$ and its estimate error covariance P_k, which concludes the measurement update step ("correction").
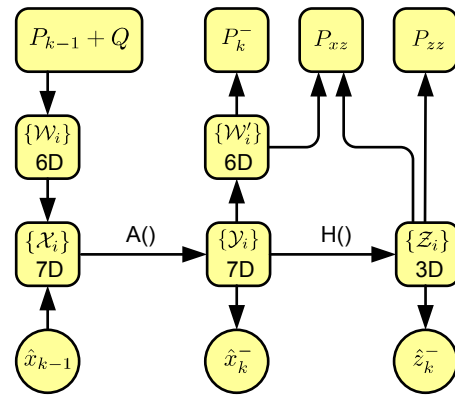

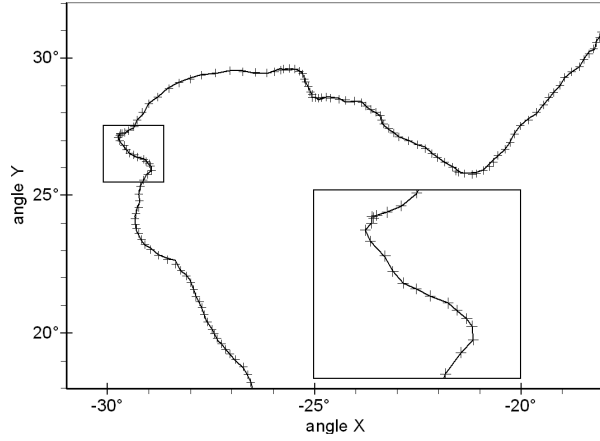Figure 1: Schematic view of the described filter.

Figure 2: Comparison between simulated path (crosses) and reconstructed path (line).

4 Results

Figure 2 shows the result of a simulated random walk (crosses) in comparison to the path reconstructed by the described filter (black line). The simulated path is produced by integrating a disturbed angular velocity vector

$$\vec{\omega}_{sim} = \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}. \tag{77}$$

Here, the components of $\vec{\omega}$ are interpreted as *Euler angles* denoting a rotation around the global x-axis followed by a rotation around the global y-axis. $\vec{\omega}_{sim}$ is disturbed by two random variables v_x, v_y which have a zero mean gaussian distribution with the variance Q_{sim}/f. f is the frequency with which the simulated path is sampled (here: 50 Hz).

$$\begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}_{t+\Delta t} = \begin{pmatrix} \omega_x \\ \omega_y \\ 0 \end{pmatrix}_t + \begin{pmatrix} v_x \\ v_y \\ 0 \end{pmatrix} \tag{78}$$

The resulting motion is similar to the movement which occurs if the user of the head tracking systems watches a fly on its erratic path around the room.

At each sample point, the measurement models are used to compute measurements of acceleration, angular velocity and magnetic field from the current orientation. These "perfect" simulated measurements are subsequently disturbed by additive, gaussian noise variables with the covariances R_{acc}, R_{rot} and R_{mag}. The Kalman filter uses the disturbed measurements to compute a new estimate of the orientation which is then decomposed into the components around the global x- and y-axes. Ideally, these components should match the simulated orientation given in Euler angles.

Figure 2 shows an interval of about three seconds from a ten seconds simulation. In this example, the reconstructed path follows the simulated points closely. The covariances of the filter were chosen equal to the corresponding covariances of the simulated measurement noise. Like expected, a higher measurement noise causes a higher reconstruction error. Despite both the type of simulation and the specific parameter set for this example are designed to suit the filter, the results demonstrate the general validity of the described filter concept.

Operation of the filter with real measurements shows the desired performance as well. A set of filter parameters which produces smooth, responsive and stable results is easily found. Comparisons with other filter concepts in terms of performance, stability or accuracy have not been made.

5 Conclusion

A quaternion-based Unscented Kalman Filter for the estimation of a rigid body attitude has been described. The problems caused by the inherent properties of orientations were discussed and solutions to these problems presented. Even though the original UKF can already handle nonlinear models and noise of arbitrary dimension, some extension were necessary to deal with state vectors which are not elements of a vector space and whose components are subject to constraints. These extensions were made for the example application of a sourceless head tracking system. The described filter shows the desired performance on both simulated and measured data.

References

[1] E. Dam, M. Koch, M. Lillholm, *Quaternions, Interpolation and Animation*, Technical Report DIKU-TR-98/5, Department of Computer Science, Univ. Copenhagen, Denmark, 1998.
(http://www.diku.dk/research/published/98-5.ps.gz)

[2] S. J. Julier and J. K. Uhlmann, *A new extension of the Kalman filter to nonlinear systems*, in International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, USA, 1997.

[3] X. Pennec, *Computing the mean of geometric features - Application to the mean rotation*, Institut National de Recherche en Informatique et en Automatique (INRIA), Le Chesnay, France, 1998.
(http://www.inria.fr/rrrt/rr-3371.html)

[4] E. A. Wan, R. van der Merwe, *The Unscented Kalman Filter*, in Kalman Filtering and Neural Networks, pp. 221-280, Wiley Publishing, 2001.

[5] T. Levebvre, H. Bruyninckx, 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, Vol. 47, pp. 1406 - 1408, 2002.