ION GNSS 2013, September 16-20, Nashville, TN

OVERCOMING ERRORS IN PROCESSING NONLINEAR MEASUREMENTS: A NEW EXTENSION FOR THE KALMAN FILTER

Alexandr Draganov
Argon ST, a wholly owned subsidiary of the Boeing Company
Fairfax, USA
alexandr.b.draganov@boeing.com

Abstract - The Predictor-Corrector Unscented Kalman filter is designed to compute the measurement update for linear or nonlinear measurements. It is backward-compatible with other Kalman filter formulations (e.g., EKF or UKF) in a sense that the filter state is described by the mean and covariance only. At a small additional cost in computational load, this filter delivers a better (sometimes, much better) accuracy than alternatives.

Keywords: Kalman filtering, nonlinearity, nonlinear estimation

I. INTRODUCTION

The list of sensors that are used for navigation applications continues to expand. This brings a new challenge to the filtering algorithm: it must process a wide variety of measurements, including nonlinear and non-Gaussian ones. For example, a GPS pseudorange usually is an almost linear measurement, but ranging off a beacon or a peer user is nonlinear, if the position uncertainty is on the order of the distance to the signal source. Applying an Extended Kalman filter to such measurements would produce an error in the state estimate. There are several known nonlinear extensions to the classic Kalman filter algorithm, which improve the result to some degree, but still may not meet requirements for a particular application. Nonlinear variants for the Kalman filter include the Extended Kalman Filter [e.g., see 1] (which is nothing more than a linearization), the 2nd order filter [2], the Iterated Kalman Filter [e.g., see 3], the Unscented Kalman Filter [4], Gauss-Hermite Filter [5,6], the Cubature Filter [7], etc.

A Kalman filter or its extension characterizes both the state and a measurement using only the mean and covariance, i.e. the first two moments or cumulants of the statistical distribution. It has been long recognized that nonlinear measurements warp the joint distribution of measurement and state errors, making it non-Gaussian even in the case of Gaussian noise statistics [4]. This gives rise to higher order cumulants, which must be truncated for processing by the filter. This is the reason for errors that a classic Extended Kalman filter produces when processing nonlinear measurements. In particular, a nonlinear measurement may produce a bias in the state estimate [4]. A bias is especially dangerous as it is not removed by processing multiple measurements from the same source. For example, repeated processing of ranging measurements from an RF beacon over some time period. Random measurement errors are averaged out by the filter; however the bias due to nonlinearity is not removed by the Extended Kalman filter and thus corrupts the solution. This bias increases with the increase in the uncertainty in the user solution or for a close-by beacon.

The popular Unscented Kalman Filter algorithm [4] and its variations (such as the Gauss-Hermite [5, 6] or Cubature algorithm [7]) account for the biases induced by measurement or time update nonlinearity. They compute the mean and covariance for the distribution, which is "warped" by nonlinearity. This leads to a more accurate solution in the statistical sense. As a result, posteriori errors are not biased, but still can be substantial.

Paper [8] showed that known Kalman extensions fall short of processing all types of nonlinear measurements. It also derived an analytical formulation, which is universally applicable for different types of nonlinearities. In this paper, we extend that idea and present a computer algorithm, which delivers robust, efficient and highly accurate results for processing nonlinear measurements.

In our algorithm we make use of the fact that nonlinear errors are a function of the user uncertainty, and this uncertainty is always smaller for posteriori state than for a priori one. Hence, using the posteriori state for computing nonlinear corrections should produce a more accurate result. As a result, we apply a predictor-corrector scheme. At the predictor stage, posteriori state and covariance are roughly estimated using the Unscented Kalman filter (or its variant). Corrector uses this rough estimate to compute and apply nonlinear corrections, again using the Unscented Kalman filter. The result is at least as good as for the standard UKF application; however if the Kalman gain is relatively high, the predictor-corrector algorithm is much more accurate.

This reasoning is confirmed by computer simulations. The user determines its position on a plane by processing range measurements from two beacons, which are located at known locations above the plane. We compare results from the predictor-corrector algorithm with those from EKF, Iterated Extended Kalman filter, and the Unscented Kalman filter. The nonlinearity corrupts the EKF estimate by producing a position-dependent non-zero mean error, as expected. Depending on the measurement quality, IEKF may produce a much better estimate, but it can still be biased. UKF removes the bias, but the error (even though it is zero mean) persists. The predictor-corrector algorithm achieves the best of both results: the bias is removed and the magnitude of the error is reduced, in some cases by a large factor.

The new algorithm is easy to implement and efficient to run. It retains the main attractiveness of the Kalman filter: the user state is described only by the mean and covariance. This is a simple and economical approach compared to other parameterizations for the state, such as those used by particle filters [e.g., see 9]. Our algorithm can be applied in all applications that currently use a Kalman filter or one of its extensions. It delivers more accurate (in some cases, substantially more accurate) results than previously known comparable algorithms.

II. Nonlinear Errors and Approaches for Their Mitigation

A Kalman filter typically estimates the state from indirect measurements. This means that we do not have the luxury of measuring the state vector itself; rather, we measure the value of a quantity that is a function of the state vector. A measurement function is often nonlinear, and the Extended Kalman filter linearizes it at a priori state. It is this linearization that often causes errors in the result. Below we consider a qualitative, intuitive example that shows the nature of the nonlinear errors in EKF. This example will help us to understand relative strengths and weaknesses of different flavors of nonlinear extensions of the Kalman filter and ultimately to design an improved and more accurate algorithm.

Consider a filter that estimates a one-dimensional (scalar) state x, where we measure the value of some nonlinear function $A(x)$. Let us assume that a scalar measurement z has a very low noise variance, so that it essentially produces an accurate value of $A(x)$. For this example, an EKF performs the following procedure:

1. It linearizes $A(x)$ at a priori value $\tilde{x}$. This produces a slope $A'(\tilde{x})$. This is known as "computing the partials" in the Kalman filter jargon.
2. It computes the measurement residual $z - A(\tilde{x})$
3. Finally, EKF updates the a priori estimate as follows: $\hat{x} = \tilde{x} + \dfrac{z - A(\tilde{x})}{A'(\tilde{x})}$

Of course, EKF equations are more general than this rudimentary algorithm: they use matrix-vector formulation for multidimensional states (and often, for multidimensional measurements) and they allow for uncertainty in the measurement, which leads to a weighted sum of a priori state and the measurement. However, our simple example is useful because it captures the essential mechanism for producing nonlinear errors.

Indeed, Figure 1 shows that the state that is estimated via linearized procedure is different from the one, which can possibly produce measurement z. The difference between the estimated state and the true value is entirely due to the curvature of the measurement function $f(x)$, which is not accurately approximated by its linearization

$$f(x) \approx f(\tilde{x}) + (x - \tilde{x}) \cdot f'(\tilde{x}) \quad (1)$$

From this simple example we can draw two important observations. First, the nonlinear error is a function of the difference between a priori and the true state. Second, if measurement residuals for two different measurements have different signs (i.e., one is positive and another is negative), the nonlinear error does not necessarily flips its sign. This latter property causes the biggest concern for applying EKF to nonlinear measurements. Indeed, even if the measurement noise is zero mean, the nonlinear error is not, and may introduce a persistent bias in each measurement update. Such bias will not be averaged out by processing more measurements, corrupting the solution.

At this point, we consider two different broad approaches that were used by various flavors of nonlinear Kalman filters for mitigating the nonlinear error.

A. Approach 1

The first approach recognizes that a priori state estimate has fundamental uncertainty, and estimates the nonlinear error statistically. Since the Kalman filter keeps track of the state covariance, we can integrate the expected nonlinear error over the entire distribution and compute its mean value. Then posteriori estimate can be corrected for this mean nonlinear error. This method does not guarantee to remove the nonlinear error for each individual measurement, but it does remove the nonlinear bias in the statistical sense. Hence, there is hope that over many measurements the nonlinear errors will not have

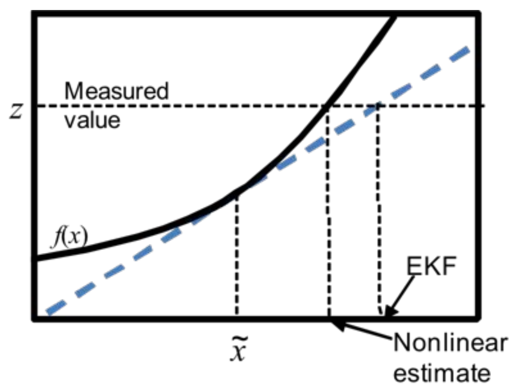

Figure 1. A linearized (EKF) and a nonlinear estimate for a one-dimensional state.

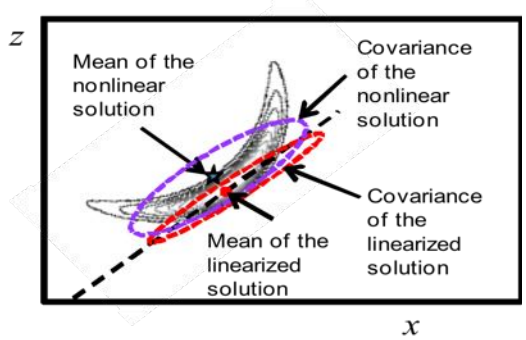

Figure 2. Mean and covariance for a linearized and a nonlinear joint distribution

a large detrimental effect on the solution. This approach is taken by such algorithms as the 2nd order filter, the Unscented Kalman Filter, the Gauss-Hermite Filter and the Cubature Filter. They are similar in spirit but differ essentially by the algorithm that is used to integrate the nonlinear error over the entire a priori distribution.

Another way to look at this approach is to consider the augmented state that includes the measurement noise in a joint distribution function. Here we no longer assume that the measurement has very small variance. A notional example of the joint distribution function is shown in Figure 2. The sickle-shaped distribution is decidedly non-Gaussian. Yet for simplicity and to retain the classic Kalman filtering paradigm, this approach characterizes the state distribution by the mean and covariance only. In practice, this can be viewed as truncating the set of distribution cumulants beyond the second order, which would be equivalent to approximating the joint distribution by a Gaussian. Figure 2 also shows a Gaussian approximation, which has the same mean and covariance as the joint distribution. The mean (the gravity center of the sickle-shaped region) is shifted compared to a priori value; this is exactly the nonlinear bias that is predicted by the Unscented Kalman Filter.

B. Approach 2

The second approach recognizes that we do know more about the posteriori user state than about a priori one. Indeed, the covariance of the user state is always shrunk by a measurement update. In a limit case of low variance measurements, we may have a much more accurate user state estimate after processing a measurement. From Figure 1 we can infer that it may be possible to estimate the nonlinear correction if we have a rough estimate of the state from an EKF. This suggests the idea of the Iterated Extended Kalman Filter: the first iteration is used to roughly estimate the state, and this knowledge is used during a second iteration of the filter to correct the nonlinear error at the state predicted by the first iteration.

C. Relative strengths and weaknesses of different nonlinear Kalman filter extensions

From the above description it is clear that these two broad approaches are complementary. The first approach removes the bias due to uncertainty in a priori user state, but it does not benefit from the information in posteriori state. The second approach uses posteriori information, but it fails to account for a bias that is due to state uncertainty. Each approach has its pluses and minuses.

Removal of nonlinear bias is important for many applications, especially when multiple measurements of the same type are processed over time. A bias will not be averaged out by processing and will corrupt the solution. However, in practice the residual bias may still remain despite our best efforts. The reason for this is that the computed bias correction is based on a priori state covariance (as the nonlinear error is integrated over state uncertainty). In a typical application, the state covariance computation does not use measured data; instead, it relies on models for measurement variances and for the process noise. Such models are not always accurate; thus, the state covariance may be under- or overestimated. From the above description it is evident that this would correspondingly lead to under- or overestimation of the nonlinear bias.

Approach 2 is more accurate if the posteriori state uncertainty is small, but it fails to capture the overall bias effect if the measurement update does not drastically shrink the state uncertainty. As we discussed above, the residual bias is the primary danger in many applications.

The above analysis suggests a straightforward repair to both problems. The nonlinear correction should be computed at posteriori mean (Approach 2) plus the bias due to uncertainty in the posteriori state (Approach 1, but used for posteriori state). This idea is at the foundation of a new prediction-correction filter that is presented in this paper.

D. Relationship to the IMRE Kalman Filter

The complementary nature of existing nonlinear Kalman filter extensions was first demonstrated in [8]. That paper also presented an analytical formulation for a filter that integrates strengths of existing filters; this formulation was called the IMRE Kalman filter. It is instructive to see how the IMRE Kalman filter fits the paradigm of Approach 1 and 2 to treating nonlinearities. The final IMRE Kalman expression for posteriori estimate is as follows. The IMRE Kalman filter uses the measurement update that applies the conventional EKF first, similarly to the first iteration of IEKF. This produces the preliminary state estimate $\hat{x}$ and the new covariance matrix $\hat{P}$. Then the updated state is corrected for nonlinear effects:

$$\begin{aligned}\hat{x}_I &= \hat{x} - \frac{1}{2\sigma_z^2}\left(\Delta x^T G \Delta x\right)\cdot \hat{P}H \\ &- \frac{1}{\sigma_z^2}\left(A + H\Delta x - z\right)\hat{P}G\Delta x \\ &- \frac{1}{2\sigma_z^2}\left[Tr\left(\hat{P}G\right)\hat{P}H + 2\hat{P}G\hat{P}H\right] + \frac{\hat{P}GH}{H^2}\end{aligned} \tag{2}$$

The notations are as follows:

$\hat{x}_I$ - IMRE Kalman filter estimate for the state,

$\hat{x}$ – posteriori EKF estimate for the state,

$\hat{P}$ – posteriori EKF estimate for the state covariance matrix,

σ_z^2 - measurement variance,

$\Delta x = \hat{x} - \tilde{x}$ - the EKF state update, where

$\tilde{x}$ - a priori state,

A, H, G – correspondingly values of the measurement function, its gradient and Hessian, computed at a priori state $\tilde{x}$.

Below we analyze this formula in view of qualitative approaches to treat nonlinear errors that are discussed above. Let us present the EKF state update as

$$\Delta x = \Delta x_t + \delta x$$

where Δx_t is the (unknown) difference between the true and a priori and state, and δx is the estimation error. We do not know the value of the estimation error, but we do know its statistics: it is described by the posteriori covariance matrix. Let us compute an estimate for the following quantity (compare this to Equation (2)):

$$\begin{aligned}\chi &= -\frac{1}{2\sigma_z^2}\left(\Delta x_t^T G \Delta x_t\right)\cdot \hat{P}H - \\ &\frac{1}{\sigma_z^2}\left(A + H\Delta x_t - z\right)\hat{P}G\Delta x_t\end{aligned} \tag{3}$$

Since Δx_t is unknown, the best we can do is to compute the expected value of χ. By substituting $\Delta x_t = \Delta x - \delta x$ and taking the expected value, we get:

$$\begin{aligned}\langle\chi\rangle &= -\frac{1}{2\sigma_z^2}\left(\Delta x^T G \Delta x\right)\cdot \hat{P}H - \\ &\frac{1}{\sigma_z^2}\left(A + H\Delta x - z\right)\hat{P}G\Delta x - \\ &\frac{1}{2\sigma_z^2}\left\langle \delta x^T G \delta x\right\rangle \cdot \hat{P}H - \\ &\frac{1}{\sigma_z^2}\left\langle\left(A + H\delta x - z\right)\hat{P}G\delta x\right\rangle\end{aligned} \tag{4}$$

In the component notation,

$$\left\langle \delta x^T G \delta x \right\rangle = \left\langle \sum_{i,j} \delta x_i G_{ij} \delta x_j \right\rangle =$$
$$\sum_{i,j} \left\langle \delta x_i \delta x_j \right\rangle G_{ij} = \sum_{i,j} \hat{P}_{ji} G_{ij} = Tr\left(\hat{P}G\right) \tag{5}$$

and

$$\left\langle \left(A + H\delta x - z\right)\hat{P}G\delta x \right\rangle_k = \left\langle H\delta x \hat{P}G\delta x \right\rangle_k =$$
$$\left\langle \sum_{jl} H_j \delta x_j \left(\hat{P}G\right)_{kl} \delta x_l \right\rangle =$$
$$\sum_{jl} H_j \left\langle \delta x_j \delta x_l \right\rangle \left(\hat{P}G\right)_{kl} =$$
$$\sum_{jl} \left(\hat{P}G\right)_{kl} \hat{P}_{lj} H_j = \left(\hat{P}G\hat{P}H\right)_k$$
(6)

By comparing these expressions with the IMRE Kalman filter formula (2), we can see that the first four terms for the nonlinear correction there can be interpreted as the expected value of the nonlinear correction, computed at the true state. The fifth term $\dfrac{\hat{P}GH}{H^2}$ does not fit into this scheme and describes additional nonlinear effects.

This is generally consistent with the qualitative argument put forth earlier. Namely, nonlinear effects arise due to the fact that the true value of the state is different from the estimated value. If we knew the state accurately, we could have estimated the nonlinear correction accurately as well. However, we do not know the state accurately (after all, estimating the state is the very goal of the filter processing). For each measurement or set of measurements, the best knowledge of the state is achieved after these measurements have been processed; it is the posteriori mean and covariance. A better way to account for nonlinear effects is thus to compute them at the posteriori mean and then average over posteriori uncertainty.

III. PREDICTOR-CORRECTOR UNSCENTED KALMAN FILTER

The new algorithm computes first a rough approximation to the solution (the predictor step), and then makes a fine correction to it (the corrector step). In this sense it is similar to the Iterated Extended Kalman Filter and IMRE Kalman filter. It is an improvement from the previous algorithms in the following respects:

1. The Iterated Extended Kalman Filter uses the classic EKF as the underlying engine; due to this, it does not capture the bias in the estimate that is associated with state uncertainty.
2. The IMRE Kalman filter is an analytical formulation that requires knowledge of second derivatives of the measurement function (the Hessian). In practice, computation of second derivatives may not be easy or even accurate. The Predictor-Corrector algorithm uses sigma-points. This is analogous to the benefits of Unscented Kalman Filter [4] compared to the 2nd order filter [2]: the earlier algorithm was complex and required second derivatives; it was only after introduction of sigma points that nonlinear Kalman filter processing received wide acceptance.
3. The Unscented Kalman Filter does not take advantage of posteriori information to perform nonlinear processing. Posteriori state always has smaller uncertainty (and is statistically more accurate). Hence, using it helps to reduce the nonlinear error.

Below we provide algorithm details for the Predictor-Corrector filter. For simplicity, the formulation is limited to the case of scalar measurements. Also, for this implementation, we selected a set of $2N$ sigma-points following [4].

A. First iteration (predictor)

The first iteration is a standard UKF filter. We follow [4]; explicit algorithm is provided here for completeness only. Even though we use one particular implementation of UKF, any other implementation or another similar algorithm (such as the Gauss-Hermite algorithm) can be used in its place. A priori state starts from mean x_0 and covariance P_0. The Predictor steps are as follows.

1. Set sigma points:
$$x_0^{(j)} = x_0 + \left(\sqrt{NP_0}\right)_j$$
$$x_j^{(j)} = x_0 - \left(\sqrt{NP_0}\right)_j \tag{7}$$
$$W_j = \frac{1}{2N}$$
where $\left(\sqrt{NP_0}\right)_j$ is the $j-$th row of the square root of NP_0. In practice, one may offset sigma points from the mean in the direction of eigenvectors of P_0 and at the distance of the square root of the product of N and the corresponding eigenvalue of P_0.
2. Compute expected measurements by using the measurement function $A\left(x_0^{(j)}\right)$
$$y_0^{(j)} = A\left(x_0^{(j)}\right);\ y_0 = \sum_j W_j y_0^{(j)} \tag{8}$$

3. Compute innovation covariance:

$$S_0 = \sum_j W_j \left(y_0^{(j)} - y_0\right)^2 + \sigma_z^2 \tag{9}$$

where σ_z^2 is the measurement variance.

4. Compute cross covariance matrix

$$P_0^{xy} = \sum_i W_j \left(x_0^{(i)} - x_0\right)\left(y_0^{(j)} - y_0\right)^T \tag{10}$$

5. Finally, perform the Kalman filter update for measurement z:

$$\begin{aligned} \mathbf{W}_0 &= P_0^{xy} S_0^{-1} \\ \nu_0 &= z - y_0 \\ P_1 &= P_0 - \mathbf{W}_0 S_0 \mathbf{W}_0^T \\ x_1 &= x_0 + \mathbf{W}_0 \nu_0 \end{aligned} \tag{11}$$

where subscript 1 refers to the results of the Predictor step.

B. Second iteration (corrector):

The second iteration also uses UKF as the underlying engine, but the original measurement is corrected to form the "effective measurement". It uses two sets of sigma points: one based on the result of the first iteration and another set (referred to as hybrid below) which combined the mean from the first iteration x_1 and the a priori covariance matrix P_0. The steps are as follows.

1. Form sigma-points for the result of the first iteration:

$$\begin{aligned} x_1^{(j)} &= x_1 + \left(\sqrt{NP_1}\right)_j \\ x_1^{(j)} &= x_1 - \left(\sqrt{NP_1}\right)_j \end{aligned} \tag{12}$$

2. Form a hybrid set of sigma-points which uses the mean from the first iteration and the covariance from a priori state:

$$\begin{aligned} x_H^{(j)} &= x_1 + \left(\sqrt{NP_0}\right)_j \\ x_H^{(j)} &= x_1 - \left(\sqrt{NP_0}\right)_j \end{aligned} \tag{13}$$

3. Compute expected measurements for both sets of points:

$$\begin{aligned} y_1^{(j)} &= A\left(x_1^{(j)}\right);\ y_1 = \sum_j W_j y_1^{(j)} \\ y_H^{(j)} &= A\left(x_H^{(j)}\right);\ y_H = \sum_j W_j y_H^{(j)} \end{aligned} \tag{14}$$

4. Compute the effective measurement value:

$$\begin{aligned} \delta x &= x_1 - x_0 \\ \Delta x_H^{(j)} &= x_H^{(i+N)} - x_H^{(i)};\ i = 1 \ldots N \\ \Delta y_H^{(j)} &= y_H^{(i+N)} - y_H^{(i)};\ i = 1 \ldots N \\ \delta y &= \sum_{i=1}^{N} \frac{\Delta y_H^{(j)} \left(\Delta x_H^{(j)} \cdot \Delta \delta x\right)}{\left(\Delta x_H^{(j)}\right)^2} \\ z_1 &= z - \delta y \end{aligned} \tag{15}$$

5. Compute the innovation covariance for the hybrid set of sigma points:

$$S_H = \sum_j W_j \left(y_H^{(j)} - y_H\right)^2 + \sigma_z^2 \tag{16}$$

6. Compute cross covariance matrix for the hybrid set:

$$P_H^{xy} = \sum_i W_j \left(x_H^{(i)} - x_H\right)\left(y_H^{(j)} - y_H\right)^T \tag{17}$$

7. Finally, perform the Kalman filter update:

$$\begin{aligned} \mathbf{W}_H &= K_H^{xy} S_H^{-1} \\ \nu_1 &= z_1 - y_1 \\ P_2 &= P_0 - \mathbf{W}_H S_H \mathbf{W}_H^T \\ x_2 &= x_0 + \mathbf{W}_H \nu_1 \end{aligned} \tag{18}$$

where subscript 2 refers to the results of the Corrector step.

Note two features of the Corrector step. First, the Kalman filter update (Step 7) uses the expected measurement y_1 from the first iteration. This value is different from a priori state in two respects: the state itself is changed by δx and the covariance matrix is tighter. Therefore, using y_1 accomplishes the main goal of the new algorithm: it makes advantage of the better knowledge of posteriori state (both in the sense of new state value and tighter covariance). Second, the Corrector step is still applied to a priori state x_0 and its covariance P_0. Simply using a priori state x_0 and posteriori expected measurement y_1 would be inconsistent; this is fixed by computing the "effective" measurement z_1, which differs from the original measurement by the amount needed to remove such inconsistency.

IV. SIMULATION RESULTS

We assess performance of the predictor-corrector filter vs. that of other nonlinear Kalman filter extensions by a computer simulation. Here filters are applied to a navigation application, where a user is located on a plane and determines its position by measuring ranges to two beacons at known locations.

Measurement equations for this application are well known: they are similar to those in GPS positioning. However, the distance to a GPS satellite is typically much larger than the uncertainty of the user position. Hence, the direction to a satellite is approximately uniform across the uncertainty in the user position. This makes GPS pseudorange equations approximately linear for vast majority of applications. In our simulations, beacons are deliberately placed at a distance, which is on the order of user position uncertainty. This setup is the source of nonlinearity in the measurements and of associated errors in the solution.

In each simulation, we fix the estimated user position and the location of the beacons, and cycle through the "true" user position on the plane. For each "true" user location, we run the filters under test and determine

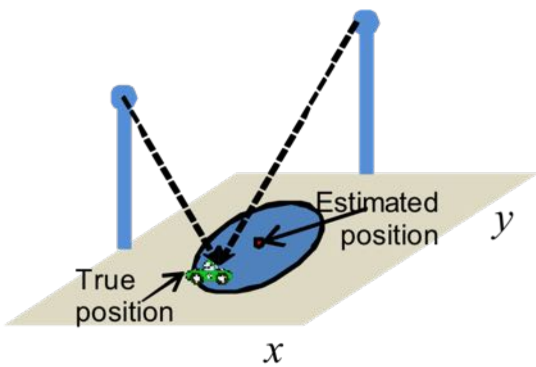

Figure 3. Simulation setup. A user processes range measurement from two beacons

posteriori position error. The beacons are located in orthogonal directions (x and y) at the altitude of 400 m above the user plane and at the horizontal distance of 400 m from the estimated user position. This setup is notionally illustrated in Figure 3. A priori standard deviation of the user position is 100×100 m. The measurement variance value used as an input by each filter is 0.1 m^2. However, range values processed by the filter did not have any noise added to the true range. This was done to focus the results of the simulation on the errors due to nonlinearity and not to obfuscate them by noise effects. Below, we present results for two simulations: (1) when the user processes ranges from both beacons and (2) when only the range in the x direction is processed.

A. Simulation 1

Figure 4 shows the magnitude of the user position error after processing both range measurements for each user location using the standard EKF. (The position error is a 2D vector, so it is difficult to display except as a magnitude.) Note that the error is zero if the true user location happens to coincide with the a priori location, and grows if a priori location is inaccurate.

Figure 5 shows the magnitude of the user position error for UKF. The error is generally lower than that for EKF, except in the center, where a priori location is close to the true location. A look at the numbers shows that the 2D vector error in the central part of the plot has the direction, which is generally opposite to that for the error in the periphery. Unfortunately, the error magnitude in the figure does not show this flip in the direction, but it is precisely the effect of UKF processing that subtracts the overall bias from the solution. This leads to better results on the periphery (large a priori errors), but overcompensates the bias in the center (small a priori errors).

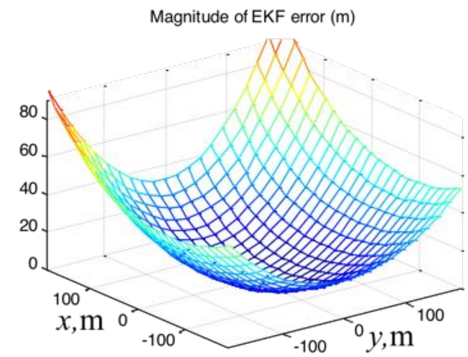

Figure 4. Magnitude of the user position error for EKF filter

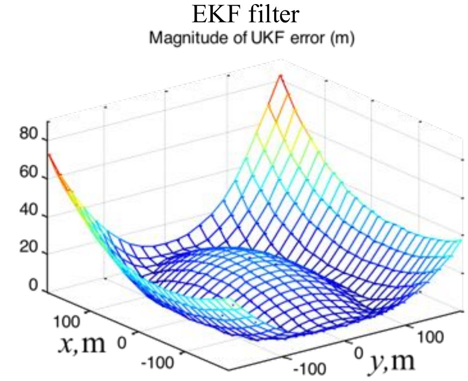

Figure 5. Magnitude of the user position error for UKF filter

Figures 6 and 7 show results for the same simulation, but for the Iterated Extended Kalman Filter (IEKF) and the Predictor-Corrector Unscented Kalman filter (PC-UKF). Both results are uniformly better than those for EKF or UKF, and over the large portion of the distribution they are much better. Note that IEKF and PC-UKF accuracies are comparable in this simulation. This is due to the fact that we assume quite accurate ranging (0.31 m error standard deviation) in our simulation, and hence posteriori user position uncertainty is much smaller than a priori one. As we discussed above, IEKF and PC-UKF are able to take advantage of this fact to correct nonlinear errors in the solution.

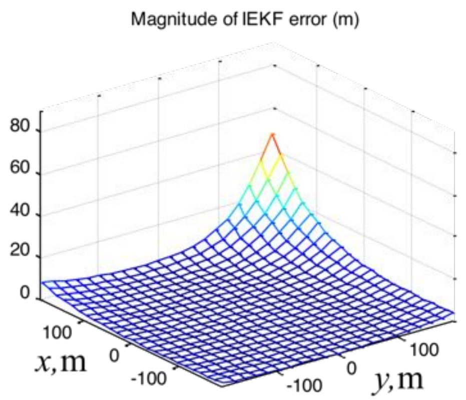

Figure 6. Magnitude of the user position error for IEKF filter

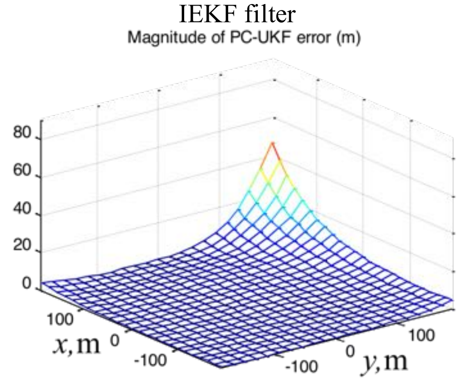

Figure 7. Magnitude of the user position error for PC-UKF filter

Table 1 presents results for the position error, which is integrated over a priori user distribution for all four filters. UKF lowers the integrated magnitude of the error only moderately (13.1 m for UKF vs. 15.9 m for EKF). However, the main benefit of UKF is in decreasing the bias; this can be seen if we look at the average error for individual components x and y and if we retain the error sign in the calculation (i.e., compute the average of the actual error component, not of its magnitude). Specifically, the average error in the x component is -14.5 m for EKF and 4.2 m for UKF. On Figure 5 one can see that the error is nulled

Filter	Mean of abs error	Mean of the x error	Mean of the y error
EKF	15.9	-14.5	-5.06
UKF	13.1	4.23	1.51
IEKF	0.85	-0.49	-0.67
PC-UKF	1.21	-0.45	-0.91

Table 1. Estimation errors averaged over a priori distribution for Simulation 1

on an ellipse in x, y plane; the overall bias in the 2D vector error is overcompensated within this ellipse and undercompensated outside of it. Table 1 shows that IEKF and PC-UKF outperform both the EKF and UKF in this simulation; this is also evident from looking at Figures 4 through 7.

While these results are encouraging, they raise a question: why do we need the PC-UKF algorithm if IEKF delivers a comparable performance? The answer to this question is that IEKF performs well only when posteriori state uncertainty is small (as it is the case in this simulation); otherwise, IEKF solution suffers from a nonlinear bias. This is illustrated by results from Simulation 2, presented below.

B. Simulation 2

In this simulation we use the same setup, but process only the range measurement from the beacon in the x direction. This leaves a substantial position uncertainty, and a priori error in the y direction is not corrected by the filter. Similarly to Simulation 1, we compare results from four different filters: EKF, UKF, IEKF, and PC-UKF. To separate the effect of the nonlinear error from the residual a priori error in the y direction, we present results only for the x error component. This error is a scalar, and now we have the opportunity to display the actual error on the plots, and not its magnitude.

Figures 8 through 11 show posteriori x error component for EKF, UKF, IEKF and PC-UKF. Again, we can see that EKF error is the largest and is

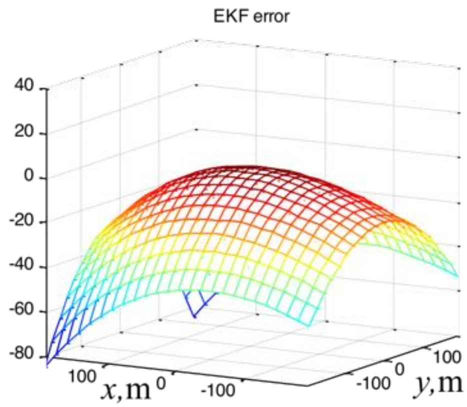

Figure 8. X component of the user position error for EKF filter

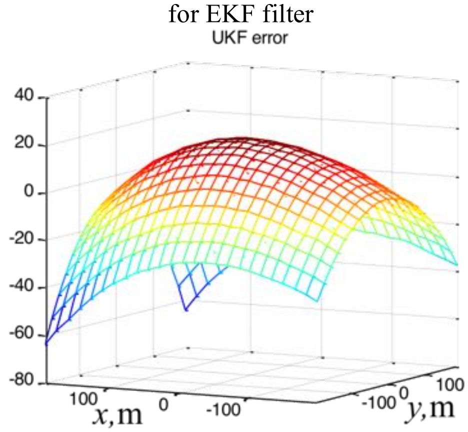

Figure 9. X component of the user position error for UKF filter

Filter	Mean of the x error component
EKF	-14.5
UKF	4.23
IEKF	-9.9
PC-UKF	3.09

Table 2. Estimation errors averaged over a priori distribution for Simulation 2

always negative, which creates a bias in the solution. Compared to EKF, the UKF error appears shifted upwards, which largely compensates for the bias. IEKF error is also biased, which is an important drawback for this filter. Finally, PC-UKF error is similar in shape to the IEKF error, but is shifted in such

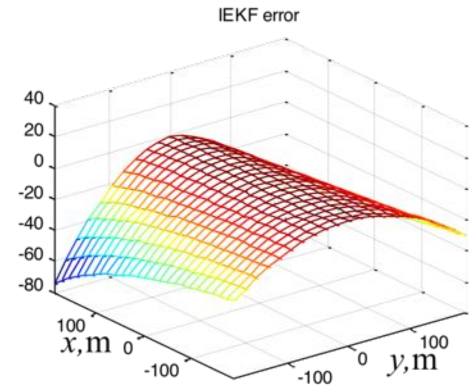

Figure 10. X component of the user position error for IEKF filter

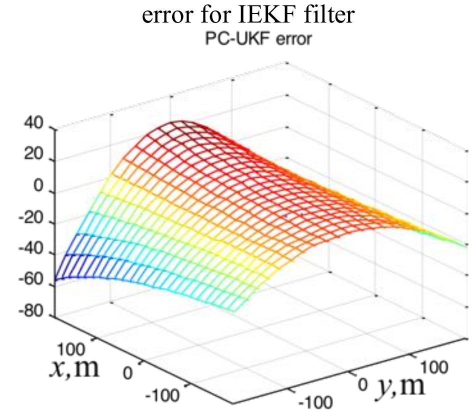

Figure 11. X component of the user position error for PC-UKF filter

way that the bias is largely compensated. Table 2 shows the x error component which is integrated over the entire distribution. These values show the benefit of the PC-UKF filter.

As we discussed before, the bias in the EKF or IEKF solution is dependent on the state uncertainty. However, the bias in the EKF solution is a function of a priori state uncertainty, whereas the bias in IEKF solution is a function of posteriori state uncertainty. Simulations 1 and 2 have very different posteriori state uncertainties. In Simulation 1, we processed two measurements, and posteriori state uncertainty is small compared to a priori one. In Simulation 2, the posteriori RMS user position uncertainty is decreased approximately by the factor of $\sqrt{2}$ from a priori uncertainty. Comparison of Tables 1 and 2 shows an agreement with our qualitative argument: IEKF bias in Simulation 1 is very small, and in Simulation 2 it is roughly $\sqrt{2}$ times smaller than the EKF bias.

V. SUMMARY AND FUTURE WORK

In this paper we described a new nonlinear Kalman filter extension, Predictor-Corrector Unscented Kalman Filter. Following [8], we presented analysis that previously existing nonlinear Kalman filter extensions (the 2nd order filter, IEKF, UKF, the Gauss-Hermite filter, and the Cubature filter) are complementary. Namely, IEKF makes advantage of better posteriori knowledge about user state to reduce nonlinear errors, but does not account for the bias that is associated with posteriori state uncertainty. Conversely, other filters in the list do account for the bias due to state uncertainty, but they do it at a priori state, when the uncertainty (and the nonlinear error) is larger. This analysis opened an opportunity to develop a filter that combines the best of both approaches. At the predictor stage, this filter estimates the state roughly. At the corrector stage, this estimate is used to correct for the nonlinear error associated with both the new state and new, smaller uncertainty.

We also presented results of a simulation that compares PC-UKF to EKF, IEKF and UKF. In our implementation, we used UKF as underlying filters for both predictor and corrector stages. It is straightforward to use any other nonlinear filter from the same family (i.e., the 2nd order, the Gauss-Hermite, or the Cubature filter) as underlying filters. We expect the results to be qualitatively similar to our UKF-based analysis: the predictor-corrector filter works at least as well as the underlying filtering algorithm, and works much better if the Kalman gain is large.

PC-UKF algorithm does not deviate from the main premise of a Kalman filter, which is using only the mean and covariance to describe the state statistics. While a more accurate model for state distribution may potentially lead to a more accurate estimate, it would also be more complex and computationally intensive. An example of such approach is the particle filter [9], which may potentially offer better accuracy, but suffers from the so-called curse of dimensionality for applications with large numbers of state dimensions. In contrast, the computational requirement for PC-UKF is roughly only twice of that for UKF. Also, since PC-UKF uses only the mean and covariance, it can be used wherever the EKF or any of its nonlinear extension (such as UKF) are being used.

As presented in this paper, PC-UKF applies to processing linear and nonlinear measurements by a filter. In practice, a filter may alternate the measurement update and the time update steps. A time update propagates the state and its covariance between measurements. Such propagation is often nonlinear, which is also a source of errors for the classic EKF. UKF and related filters elegantly treat the time and the measurement update steps in a similar way; they recognize that the culprit for nonlinear errors in both cases is the distortion of probability distribution due to nonlinear mapping.

This opens an intriguing possibility of extending the PC-UKF idea to the time update step as well. Currently, PC-UKF may use a standard UKF filter for time update (not shown in the algorithm description or in simulations above). Therefore, the accuracy of the time update is limited by the accuracy of UKF at this step. (Alternatively, the predictor-corrector filter may also use the Gauss-Hermite or a similar filter.) The paradigm for UKF at the time update is the same as for the measurement update: a distribution, warped by nonlinear transformation, is characterized by a mean and covariance. Similar to the measurement update, the nonlinear error is a function of the state covariance. Hence, nonlinear errors at the time update step can be reduced compared to UKF if we manage to use posteriori state instead of a priori one. This also suggests a predictor-corrector scheme, but in this case one would have to perform a nonlinear (e.g., UKF) time update, then process the measurement or measurements that come immediately afterwards and that reduce the state uncertainty, and then use that information to perform the same time update again in a corrector step. By analogy with PC-UKF, this processing sequence, while more complex than the standard one, may produce better accuracy for nonlinear time updates. It will be a topic of future research.

REFERENCES

[1] G. J. Bierman, Factorization Methods for Discrete Sequential Estimation, Dover, 2006, 241 pp.

[2] M. Athans, R. P. Wishner, and A. Bertolini, Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements, IEEE Trans. on Automatic Control, vol. AC-13, no. 5, Oct 1968, pp. 504-514.

[3] R. P. Wishner, J. A. Tabaczynski, and M. Athans, A Comparison of Three Non-Linear Filters, Automatica, vol. 5, 1969, pp. 487-496.

[4] S. J. Julier, and J. K. Uhlmann, Unscented Filtering and Nonlinear Estimation, Proc. IEEE, vol. 92, no. 3, Mar 2004, pp. 401-422.

[5] K. Ito and K. Xiong, Gaussian Filters for Nonlinear Filtering Problems, IEEE Trans. on Automatic Control, vol. 45, no. 5, May 2000, pp. 910-927.

[6] I. Arasaratnam, S. Haykin, and R. J. Elliott, Discrete-Time Nonlinear Filtering Algorithms Using Gauss–Hermite Quadrature, Proc. IEEE, vol. 95, no. 5, Mar 2007, pp. 953-977.

[7] I. Arasaratnam, and S. Haykin, Cubature Kalman Filters, IEEE Trans. on Automatic Control, vol. 54, no. 6, June 2009, pp. 1254-1269.

[8] A. Draganov, L. Haas, and M. Harlacher, The IMRE Kalman FILTER – a new Kalman Filter Extension for Nonlinear Applications, PLANS 2012

[9] A. Doucet, S. Godsill, and C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing, 2000, v. 10, pp. 197–208.