A Rao-Blackwellised Unscented Kalman Filter

Mark Briers
QinetiQ Ltd.
Malvern Technology Centre
Malvern, UK.
m.briers@signal.qinetiq.com

Simon R. Maskell
QinetiQ Ltd.
Malvern Technology Centre
Malvern, UK.
s.maskell@signal.qinetiq.com

Robert Wright
QinetiQ Ltd.
Malvern Technology Centre
Malvern, UK.
r.wright@signal.qinetiq.com

Abstract – The Unscented Kalman Filter offers significant improvements in the estimation of non-linear discrete-time models in comparison to the Extended Kalman Filter [12]. In this paper we use a technique introduced by Casella and Robert [2], known as Rao-Blackwellisation, to calculate the tractable integrals that are found in the Unscented Kalman Filter. We show that this leads to a reduction in the quasi-Monte Carlo variance, and a decrease in the computational complexity by considering a common tracking problem.

Keywords: Tracking, filtering, estimation, unscented, Rao-Blackwellisation.

1 Introduction and Motivation

The Unscented Kalman Filter (UKF) has been shown to offer significant improvements in the estimation of non-linear discrete-time models in comparison to the Extended Kalman Filter (EKF) [12] in moderately nonlinear environments. The UKF deterministically samples Sigma Points and propagates these points through the true models. In this paper we propose a method to reduce the quasi-Monte Carlo variance by only resorting to this scheme for the intractable integrals that are present.

Approaches have been proposed to reduce the number of sigma points needed to be sampled in a given dimensional space in order to reduce computation and storage expense (see [5], for example). This paper complements that previous research primarily by reducing the dimensionality of the state vector to be sampled. We therefore reduce the quasi-Monte Carlo variance and increase the efficiency of the algorithm.

The outline of this paper is as follows. In the following section, we introduce the general tracking problem formulated in a Rao-Blackwellised Unscented Kalman Filter (RBUKF) framework. In Section 3, we show that exploiting certain structures in the models allows us to reduce the amount of approximation necessary. In Section 4, we motivate the ideas through a simple example and compare computational time and variances of the state estimates. Section 5 concludes the discussion.

2 Sequential Bayesian inference and the UKF

2.1 Introduction

It is convenient to define $k$ to index time and $x_k$ as the state of the system at time $k$. The measurement received at time $k$ is $y_k$ and the set of measurements received up to time $k$ is then $y_{1:k}$.

We consider three kinds of model for the evolution of the $x_k$:

\[
\begin{align*}
x_k &= \begin{cases} 
Fx_k + \epsilon_k & \text{Linear Gaussian} \\
 f_1(x_{k-1}) + \epsilon_k & \text{Non-Linear Gaussian} \\
 f_2(x_{k-1}, \epsilon_k) & \text{Non-Linear Non-Gaussian}
\end{cases}
\end{align*}
\]

where $\epsilon_k \sim N(0, Q_k)$. and where $N(x; m, P)$ is a Gaussian density with argument $x$, mean $m$ and covariance $P$.

We also consider three kinds of measurement model describing the relationship between $y_k$ and $x_k$:

\[
\begin{align*}
y_k &= \begin{cases} 
Hx_k + v_k & \text{Linear Gaussian} \\
h_1(x_k) + v_k & \text{Non-Linear Gaussian} \\
h_2(x_k, v_k) & \text{Non-Linear Non-Gaussian}
\end{cases}
\end{align*}
\]

where $v_k \sim N(0, R_k)$ is Gaussian noise that is independent of $\epsilon_k$.

2.2 The Kalman filter

Should both models be linear Gaussian then the Kalman filter is an optimal recursive technique that exactly calculates the parameters of $p(x_k|y_{1:k})$ [4].

\[
x_k = \begin{cases} 
F \hat{x}_{k-1} + \epsilon_k & \text{Linear Gaussian} \\
 f_1(\hat{x}_{k-1}) + \epsilon_k & \text{Non-Linear Gaussian} \\
 f_2(\hat{x}_{k-1}, \epsilon_k) & \text{Non-Linear Non-Gaussian}
\end{cases}
\]

\[
y_k = \begin{cases} 
H \hat{x}_k + v_k & \text{Linear Gaussian} \\
h_1(\hat{x}_k) + v_k & \text{Non-Linear Gaussian} \\
h_2(\hat{x}_k, v_k) & \text{Non-Linear Non-Gaussian}
\end{cases}
\]
The Kalman filter algorithm can then be viewed as the following:

\[
p(x_{k-1}|y_{1:k-1}) = N(x_{k-1}; M, C) \tag{3}
\]

\[
p(x_k; y_k|y_{1:k-1}) = N \left( \begin{bmatrix} x_k \\ y_k \end{bmatrix} ; M', C' \right) \tag{4}
\]

\[
p(x_k|y_{1:k}) = N(x_k; m_{k|k}, P_{k|k}) \tag{5}
\]

where

\[
M = m_{k-1|k-1} \tag{6}
\]

\[
C = P_{k-1|k-1} \tag{7}
\]

\[
M' = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \tag{8}
\]

\[
C' = \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix} \tag{9}
\]

\[
m_{k|k} = \mu_x + P_{xy}P_{yy}^{-1}(y_k - \mu_y) \tag{10}
\]

\[
P_{k|k} = P_{xx} - P_{xy}P_{yy}^{-1}P_{yx} \tag{11}
\]

with:

\[
\mu_x = F_km_{k-1|k-1} \tag{12}
\]

\[
P_{xx} = Q_k + F_kP_{k-1|k-1}F_k^T \tag{13}
\]

\[
\mu_y = H_k\mu_x \tag{14}
\]

\[
P_{yy} = H_kP_{xx}H_k^T + R_k \tag{15}
\]

\[
P_{xy} = P_{xx}H_k^T. \tag{16}
\]

In the above equations, the transpose of a matrix \( M \) is denoted by \( M^T \).

The assumptions of the Kalman filter restrict its use in ‘real’ situations. Often one approximates (1) and (2) to both be (locally) linear Gaussian and so obtain the Extended Kalman Filter (EKF). As an alternative to the EKF, one can use a sample based approximation to \( p(x_{k-1}|y_{1:k-1}) \) and then a Gaussian approximation to \( p(x_k, y_k|y_{1:k-1}) \). In so doing one never has to approximate the models, but only \( p(x_k, y_k|y_{1:k-1}) \).

### 2.3 The Unscented Kalman filter

The unscented transform has been used to conduct this approximate scheme[7, 12]. The resulting filter, the Unscanted Kalman Filter (UKF), considers a set of points that are deterministically selected from the Gaussian approximation to \( p(x_{k-1}|y_{1:k-1}) \). These points are all propagated through the true models and the parameters of \( p(x_k, y_k|y_{1:k-1}) \) are then estimated from the transformed samples. The UKF algorithm begins by drawing samples:

\[
\begin{bmatrix} x_{(i)}^{(k-1)} \\ \epsilon_{(i)}^{k-1} \\ y_k^{(i)} \end{bmatrix} \sim N \left( \begin{bmatrix} x_{k-1} \\ \epsilon_{k-1} \\ y_k \end{bmatrix} ; M, C \right), \tag{17}
\]

where

\[
M = \begin{bmatrix} m_{k-1|k-1} \\ 0 \\ 0 \end{bmatrix} \tag{18}
\]

\[
C = \begin{bmatrix} P_{k-1|k-1} & 0 & 0 \\ 0 & Q_{k-1} & 0 \\ 0 & 0 & R_k \end{bmatrix} \tag{19}
\]

with associated weights \( w^{(i)} \). The sampling is chosen according to an appropriate deterministic procedure [11], which is elaborated on in section 2.4.

The update stage is the same as the Kalman filter equations, (10) and (11), where now:

\[
\mu_x \approx \sum_{i=1}^{N_s} w^{(i)}A
\]

\[
P_{xx} \approx \sum_{i=1}^{N_s} w^{(i)}[A - \mu_x][A - \mu_x]^T \tag{20}
\]

\[
\mu_y \approx \sum_{i=1}^{N_s} w^{(i)}h_2(A, v_k^{(i)}) \tag{21}
\]

\[
P_{yy} \approx \sum_{i=1}^{N_s} w^{(i)}[h_2(A, v_k^{(i)}) - \mu_y][h_2(A, v_k^{(i)}) - \mu_y]^T \tag{22}
\]

\[
P_{xy} \approx \sum_{i=1}^{N_s} w^{(i)}[A - \mu_x][h_2(A, v_k^{(i)}) - \mu_y]^T \tag{23}
\]

\[
A = f_2(x_k^{(i), e_k^{(i)}}) \tag{24}
\]

### 2.4 The choice of sigma points

The examples and notations presented in this paper use the standard sampling and weighting procedures associated with the Unscented transform as presented in [7, 12]. However, this is not the only sampling scheme available. Here we summarise several possible ways of choosing sigma points.

Julier and Uhlmann [7] originally proposed placing sigma points on the faces of a hypercube, one can achieve a second order unscented transform. However,
to achieve a fourth order unscented transform, and therefore match the kurtosis, further sigma points must be added at the corners of the hypercube [8].

Van Zandt [15] suggests greater robustness can be achieved by choosing points in a higher dimensional space and projecting them down: the greater number of points yielding improved accuracy. An alternative to this is the so-called minimal skew or simplex sigma point solution. By placing the sigma points on a simplex [9] one can match the mean and covariance whilst minimising the skew. This method only requires \( n + 1 \) sigma points (which can be calculated recursively) for an \( n \)-dimensional random variable as opposed to the \( 2n + 1 \) used in the symmetric and scaled schemes.

Intuition tells us that if more sigma points are used then a more accurate estimate is achievable. However, the position of these sigma points is clearly important. If the same accuracy of estimate can be achieved with fewer but better placed sigma points then computational expense is kept to a minimum.

Alternative choices of sigma point location is an active area of research [9] [5] with all methods applicable to both the UKF and the RB-UKF.

3 Rao-Blackwellisation and the RBUKF

3.1 Overview

As explained earlier, the Kalman filter is optimal when the models are linear and Gaussian. When the Kalman filter assumptions do not hold, the UKF is an appropriate sub-optimal technique that can improve performance over the EKF. When tracking with linear state or measurement evolution models, one can conceptually use the UKF for the non-linear non-Gaussian (hard) portion of the problem and a Kalman filter for the remainder. Wan[13] proposes a method for Rao-Blackwellisation in the presence of additive Gaussian noise. Although using the same concept, the approach presented here differs somewhat in that it only samples once, and so reduces the Monte-Carlo variance.

Table 1 displays the different combinations of models and their associated sampling procedures; references to the appropriate sections in this paper appear in brackets after the state that needs to be sampled in each case. We advocate the use of particle filtering (PF) in scenarios for which the models are entirely non-linear and non-Gaussian[3].

The following sections define the equations used in the prediction and update stages in the Kalman filter framework, thereby replacing equations (12) to (16). We label each section with the definition of the system being considered in Table 1.

3.2 Algorithm 1:

\[
x_k = F x_{k-1} + \epsilon_{k-1}, \quad y_k = h_1(x_k) + v_k
\]

\( \mu_x \) and \( P_{xx} \) can be calculated exactly using (12) and (13). Then:

\[
x^{(i)} \sim N(x_k; \mu_x, P_{xx})
\]

\[
\mu_y \approx \sum_{i=1}^{N_s} w(i) h_1(x^{(i)}_k)
\]

\[
P_{yy} \approx \sum_{i=1}^{N_s} w(i) [h_1(x^{(i)}_k) - \mu_y][h_1(x^{(i)}_k) - \mu_y]^T + R_k
\]

\[
P_{xy} \approx \sum_{i=1}^{N_s} w(i)[x^{(i)}_k - \mu_x][h_1(x^{(i)}_k) - \mu_y]^T
\]

3.3 Algorithm 2:

\[
x_k = F x_{k-1} + \epsilon_{k-1}, \quad y_k = h_2(x_k, v_k)
\]

Again, \( \mu_x \) and \( P_{xx} \) can be calculated exactly using (12) and (13). One then uses the following scheme:

\[
\begin{bmatrix}
x_k^{(i)} \\
v_k^{(i)}
\end{bmatrix} \sim N\left( \begin{bmatrix} x_k \\ v_k \end{bmatrix}; \begin{bmatrix} \mu_x & 0 \\ 0 & R_k \end{bmatrix} \right)
\]

\[
\mu_y \approx \sum_{i=1}^{N_s} w(i) h_2(x^{(i)}_k, v_k)
\]

\[
P_{yy} \approx \sum_{i=1}^{N_s} w(i) h_2(x^{(i)}_k, v_k) - \mu_y][h_2(x^{(i)}_k, v_k) - \mu_y]^T
\]

\[
P_{xy} \approx \sum_{i=1}^{N_s} w(i)[x^{(i)}_k - \mu_x][h_2(x^{(i)}_k, v_k) - \mu_y]^T
\]

3.4 Algorithm 3:

\[
x_k = f_1(x_{k-1}) + \epsilon_{k-1}, \quad y_k = H x_k + v_k
\]

In this case:

\[
x^{(i)}_{k-1} \sim N(x_{k-1}; m_{k-1|k-1}, P_{k-1|k-1})
\]

\[
\mu_x \approx \sum_{i=1}^{N_s} w(i) f_1(x^{(i)}_{k-1})
\]

\[
P_{xx} \approx \sum_{i=1}^{N_s} w(i)[f_1(x^{(i)}_{k-1}) - \mu_x][f_1(x^{(i)}_{k-1}) - \mu_x]^T + Q_{k-1}
\]

\( \mu_y, P_{yy} \) and \( P_{xy} \) can then be calculated exactly using (14) to (16).
Table 1: Sampled variables and section references

<table>
<thead>
<tr>
<th>$H x_k + v_k$</th>
<th>$f(x_{k-1} + \epsilon_{k-1})$</th>
<th>$f(x_{k-1}, \epsilon_{k-1})$</th>
<th>$f(x_{k-1}, \epsilon_{k-1})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman Filter (2.2)</td>
<td>$x_{k-1}$ (3.4)</td>
<td>$x_{k-1}$</td>
<td>$x_{k-1}$ (3.6)</td>
</tr>
<tr>
<td>$h(x_k) + v_k$</td>
<td>$x_{k-1}$ (3.2)</td>
<td>$x_{k-1}$, $\epsilon_{k-1}$ (3.5)</td>
<td>$x_{k-1}$, $\epsilon_{k-1}$ (3.7)</td>
</tr>
<tr>
<td>$h(x_k, v_k)$</td>
<td>$x_{k-1}$, $\epsilon_{k-1}$ (3.3)</td>
<td>$x_{k-1}$, $\epsilon_{k-1}$, $v_k$ (3.8)</td>
<td>UKF (2.3)/ Particle Filter [3]</td>
</tr>
</tbody>
</table>

3.5 Algorithm 6:

$x_k = f_1(x_{k-1}) + \epsilon_{k-1}, \quad y_k = h_1(x_k) + v_k$

The advantage in this case is that some of the integrals can be calculated exactly:

$$
\left[ \begin{array}{c}
\bar{x}_{k-1}^{(i)} \\
\bar{\epsilon}_{k-1}^{(i)} \\
\end{array} \right] \sim N \left( \left[ \begin{array}{c}
\bar{x}_{k-1} \\
\bar{\epsilon}_{k-1} \\
\end{array} \right] ; M, C \right)
$$

where

$$
M = \left[ \begin{array}{cc}
m_{k-1} & 1 \\
0 & 0 \\
\end{array} \right] \\
C = \left[ \begin{array}{cc}
P_{k-1} & 1 \\
0 & Q_{k-1} \\
\end{array} \right]
$$

$$
\mu_y \approx \sum_{i=1}^{N_x} w^{(i)} h_1 B
$$

$$
P_{yy} \approx \sum_{i=1}^{N_x} w^{(i)} [h_1 B - \mu_y][h_1 B - \mu_y]^T + R_k
$$

$$
P_{xy} \approx \sum_{i=1}^{N_x} w^{(i)} [B - \mu_x][h_1 B - \mu_y]^T
$$

3.6 Algorithm 5:

$x_k = f_2(x_{k-1}), \quad y_k = H x_k + v_k$

Here the samples are drawn as in (37) and then the (normal UKF) approximations in (20) and (21) are used to calculate $\mu_x$ and $P_{xx}$. $\mu_y$, $P_{yy}$ and $P_{xy}$ can then be calculated exactly using (14) to (16).

3.7 Algorithm 6:

$x_k = f_2(x_{k-1}), \quad y_k = h_1(x_k) + v_k$

Again the samples are drawn as in (37); $\mu_x$ and $P_{xx}$ are calculated approximately using (20) and (21). The remaining parameters are calculated as follows:

3.8 Algorithm 7:

$x_k = f_1(x_{k-1}) + \epsilon_{k-1}, \quad y_k = h_2(x_k, v_k)$

Samples are drawn as in (17). $\mu_x$ and $P_{xx}$ are calculated approximately as in (35) and (36). The remaining relations are then:

$$
\mu_y \approx \sum_{i=1}^{N_x} w^{(i)} h_2 (E, v_k)
$$

$$
P_{yy} \approx \sum_{i=1}^{N_x} w^{(i)} [h_2(E, v_k) - \mu_y][h_2(E, v_k) - \mu_y]^T
$$

$$
P_{xy} \approx \sum_{i=1}^{N_x} w^{(i)} [E - \mu_x][h_2(E, v_k) - \mu_y]^T
$$

3.9 Comment

It is worth considering where the Rao-Blackwellised Unscented Kalman filter algorithms fit into the wide range of filters readily available in the tracking literature. Clearly it is closely related to the Kalman filter and the traditional Unsecented Kalman filter. The UKF has been likened to the Linear Regression Kalman filter [10] and we believe that the non-linear Gaussian Rao-Blackwellised UKF algorithms are indeed special cases of the Linear Regression Kalman filter. Any of the various methods of choosing sigma points (2.4) can be applied to the RB-UKF in the same way as they would be applied to the UKF.
4 Example

A single target moves in two-dimensions and measurements of range and bearing are generated. The true range of the target varies between approximately 10km and 40km. The standard deviation of the range errors is 10m. Two cases were considered; a ‘low noise’ scenario for which the bearing errors had $0.5^\circ$ standard deviation and a ‘high noise’ case for which the errors had a standard deviation of $3^\circ$ (and for which the nonlinearity is more pronounced). The target exhibits almost straight-line motion for long periods before performing sharp turns.

\[ x_{k|k}(i) \text{ and } P_{k|k}(i) \text{ are respectively the true state, estimated state and associated covariance at time } k \text{ in run } i. \] In this example, \( N_r = 100 \) was used for each noise level for each of the Extended Kalman filter, Unscented Kalman filter and the first Rao-Blackwellised Unscented Kalman filter algorithm\(^1\).

The measurements for one exemplar Monte Carlo run for both noise levels are shown, projected down into the positional space, in figure 1. Average RMS statistics for the two cases are shown in figures 2(a) and 2(b). Average NEES statistics for are shown in figures 3(a) and 3(b). Note that the lower the RMS errors, the better the measure of performance, whereas the closer the NEES statistic is to the dimensionality of the measurement space (2 here), the better the measure of performance.

\[ \bar{\varepsilon}_{\text{RMS}} = \sqrt{\frac{1}{N_r} \sum_{i=1}^{N_r} (x_{k}(i) - m_{k|k}(i))^2} \quad (52) \]

\[ \bar{\varepsilon}_{\text{NEES}} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left[ \left( x_{k}(i) - m_{k|k}(i) \right)^T P_{k|k}(i)^{-1} \left( x_{k}(i) - m_{k|k}(i) \right) \right] \quad (53) \]

where \( N_r \) is the number of Monte Carlo runs and \( x_k(i) \),

---

\(^1\)Since the measurements are a nonlinear Gaussian function of the state and the state evolution is linear Gaussian, the appropriate RBUKF algorithm is the first algorithm, described in section 3.2.
In terms of RMS errors, only at the higher noise level is the EKF out-performed by both the UKF and the RBUKF (as observed in [12, 14]). There is no discernible difference between the UKF and the RBUKF in terms of RMS errors. The NEES statistic for the three filters is similar in the low noise case, except during the turns when the EKF’s performance is slightly inferior. In the high noise case, the NEES statistic indicates that the performance of the EKF deteriorates significantly; it drastically underestimates the covariance. Also, the RBUKF then generally offers an improvement in NEES in comparison over that of the UKF. Averaging the NEES value over time (as well as over the Monte-Carlo runs) gives values of 2.01 for the RBUKF and 1.55 for the UKF; the UKF overestimates the covariance.

In this example, the non-linearity is evaluated 21 times per iteration by the UKF. The Rao-Blackwellised UKF only has to evaluate the same function 9 times each iteration. The improvement in computational efficiency is clear.

5 Conclusions

In this paper we have developed a technique whose advantages are twofold; namely the reduction in quasi-Monte Carlo variance, and the decrease in computational expense for high-dimensional problems. A Rao-Blackwellised UKF (RBUKF) was developed and the algorithmic detail was outlined. The algorithm was demonstrated on a common tracking problem; non-linear (polar) Gaussian measurements of a system with linear Gaussian dynamics. The RBUKF takes advantage of the linear system model reducing the sampling requirements of the UKF and offering the potential for an improvement in performance. We demonstrate that a consistent covariance estimate is achieved via the NEES metric. This offers a marked improvement over both the Extended Kalman filter and the Unscented Kalman filter.

5.1 Applications

It has been shown that improvements to the tracking performance of the Unscented Kalman filter can be achieved by only using the sampling scheme for the intractable integrals that are present. The authors believe that the impact of this approach will be most pronounced in scenarios where the reduction in dimensionality of the space inhabited by the sigma points is significant.

Acknowledgments

This research was sponsored by the UK MOD Corporate Research Programme.

References


