ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 205-210

# Efficient Derivative-Free Kalman Filters for Online Learning

Rudolph van der Merwe and Eric A. Wan *

Oregon Graduate Institute of Science and Technology
20000 NW Walker Road, Beaverton, Oregon 97006, USA
{rvdmerwe,ericwan}@ece.ogi.edu

**Abstract.** The *extended Kalman filter* (EKF) is considered one of the most effective methods for both nonlinear *state estimation* and *parameter estimation* (*e.g.*, learning the weights of a neural network). Recently, a number of derivative free alternatives to the EKF for state estimation have been proposed. These include the *Unscented Kalman Filter* (UKF) [1, 2], the *Central Difference Filter* (CDF) [3] and the closely related *Divided Difference Filter* (DDF) [4]. These filters consistently outperform the EKF for state estimation, at an equal computational complexity of $\mathcal{O}(L^3)$. Extension of the UKF to parameter estimation was presented by Wan and van der Merwe in [5, 6]. In this paper, we further develop these techniques for parameter estimation and neural network training. The extension of the CDF and DDF filters to parameter estimation, and their relation to UKF parameter estimation is presented. Most significantly, this paper introduces efficient square-root forms of the different filters. This enables an $\mathcal{O}(L^2)$ implementation for parameter estimation (equivalent to the EKF), and has the added benefit of improved numerical stability and guaranteed positive semi-definiteness of the Kalman filter covariances.

## 1. Introduction

The EKF has been applied extensively to the field of nonlinear estimation for both *state estimation* and *parameter estimation*. The focus of this paper is on parameter estimation (*i.e.*, system identification or machine learning), which involves determining a nonlinear mapping $\mathbf{y}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{w})$, where $\mathbf{x}_k$ is the input, $\mathbf{y}_k$ is the output, and the nonlinear map (*e.g.*, neural network), $\mathbf{G}(\cdot)$, is parameterized by the vector $\mathbf{w}$. Typically, a training set is provided with sample pairs consisting of known input and desired outputs, $\{\mathbf{x}_k, \mathbf{d}_k\}$. The error of the machine is defined as $\mathbf{e}_k = \mathbf{d}_k - \mathbf{G}(\mathbf{x}_k, \mathbf{w})$, and the goal of learning involves solving for the parameters $\mathbf{w}$ in order to minimize the expectation of some given function of the error. While a number of optimization approaches exist (*e.g.*, gradient descent, backpropagation, and Quasi-Newton methods), parameters can be efficiently estimated on-line by writing a *state-space* representation

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{r}_k \quad (1)$$

$$\mathbf{d}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{e}_k, \quad (2)$$

where the parameters $\mathbf{w}_k$ correspond to a stationary process with identity state transition matrix, driven by process noise $\mathbf{r}_k$ (the choice of variance determines convergence and tracking performance). The output $\mathbf{d}_k$ corresponds to a nonlinear observation on $\mathbf{w}_k$. The EKF can then be applied directly as an efficient "second-order" technique for estimating the parameters [7, 8]. In this paper, we present new derivative-free implementations of Kalman filtering for this purpose.

*This work was sponsored in part by NSF under grant ECS-0083106, and DARPA under grant F33615-98-C-3516.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 205-210

## 2. Derivative-free Nonlinear Filters

The EKF involves the recursive estimation of the mean and covariance of the state (*i.e.*, parameters) under a Gaussian assumption. The inherent flaws are due to its linearization approach for calculating the statistics of a random variable, $\mathbf{x}$, which undergoes a nonlinear transformation, $\mathbf{y} = \mathbf{f}(\mathbf{x})$. This linearization can be viewed as a truncation of the Taylor-series expansion of the nonlinear function around the mean $\bar{\mathbf{x}}$ (to simplify notation we give the scalar expansion),

$$y = f(x) = f(\bar{x} + \Delta_x) \tag{3}$$

$$= f(\bar{x}) + \Delta_x f^{'}(x)|_{x=\bar{x}} + \frac{1}{2!}\Delta_x^2 f^{''}(x)|_{x=\bar{x}} + \dots \tag{4}$$

where the zero mean random variable $\Delta_x$ has the same covariance, $\mathbf{P}_x$, as $x$. The "first-order" mean and covariance used in the EKF is thus given by $\bar{y} = f(\bar{x}), \mathbf{P}_y = f'(\bar{x})^T \mathbf{P}_x f'(\bar{x})$, which often introduces large errors relative to the true posterior mean and covariance.

The UKF [1, 2, 5, 9], is an improved derivative-free approach to Kalman filtering. Essentially, $2L + 1$, *sigma* points ($L$ is the state dimension), are chosen based on a square-root decomposition of the prior state covariance. These sigma points are propagated through the true nonlinearity, without approximation, and then a weighted mean and covariance is taken. A simple illustration of the approach is shown in Figure 1 for a 2-dimensional system: the left plot shows the true mean and covariance propagation using Monte-Carlo sampling; the center plots show the results using a linearization approach as would be done in the EKF; the right plots show the performance of the UKF approach (note only 5 sigma points are required). This approach results in approximations that are accurate to the third order (Taylor series expansion) for Gaussian inputs for all nonlinearities. For non-Gaussian inputs, approximations are accurate to at least the second-order [2]. (Full specification UKF equations for the square-root implementation will be given in the next section).

To relate this sample-based approach to the EKF, we consider expanding the nonlinear function, $\mathbf{y} = \mathbf{f}(\mathbf{x})$, by polynomial approximations based on *interpolating formulas*. One such formula is *Sterling's* interpolation formula, which, if we limit ourselves to a

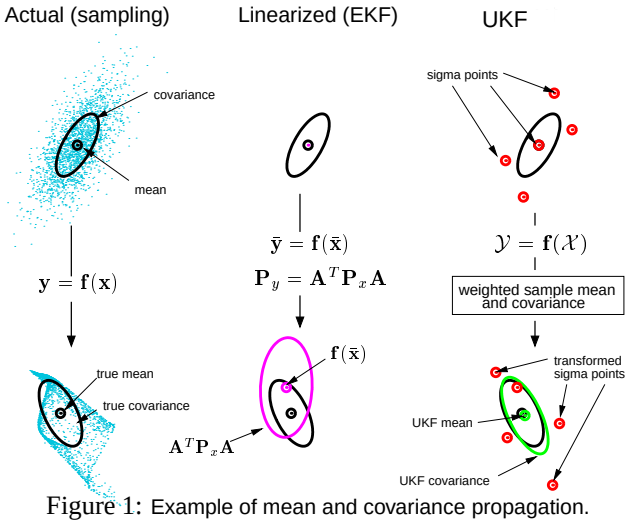


Figure 1: Example of mean and covariance propagation.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 205-210

*second order* expansion gives the following approximation

$$y \approx f(\bar{x}) + \Delta_x f'_{CD}(x)|_{x=\bar{x}} + \frac{1}{2!}\Delta_x^2 f''_{CD}(x)|_{x=\bar{x}} \tag{5}$$

$$f'_{CD}(x) = \frac{f(x+h) - f(x-h)}{2h} \quad \text{and} \quad f''_{CD}(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}.$$

One can thus interpret Eqn. 5 as a second order Taylor series expansion where the derivatives are replaced by *central differences* which only rely on functional evaluations. Expanding this approximation to higher-dimensions, is achieved by first *stochastically decoupling* the prior random variable $\mathbf{x}$ by the following transformation, $z = \mathbf{S}_{\mathbf{x}}^{-1}\mathbf{x}$, where $\mathbf{S}_{\mathbf{x}}$ is the Cholesky factor of the covariance matrix of $\mathbf{x}$, $\mathbf{P}_{\mathbf{x}}$, such that $\mathbf{P}_{\mathbf{x}} = \mathbf{S}_{\mathbf{x}}\mathbf{S}_{\mathbf{x}}^T$. This allows for the application of the central differencing operations independently to the eigen-axes of the random variables covariance-subspace.

This formulation was the basis of Norgaard's [4] recent derivation of the *divided difference filter* as well as Ito and Xiong's [3] *central difference filter*. These two filters are essentially identical and will henceforth be refered to jointly as the CDF. While the UKF was not developed in this manner, a careful analysis of the Taylor series expansion of both the CDF and the UKF approximations, show that both approaches are essentially the same (*i.e.*, square-root decompositions of the covariance, functional evaluations around the prior mean, and weighted sample means and covariances for the posterior estimates). Both filters actually calculate the posterior mean exactly the same. The difference between the two approaches, however, lie in the approximation of the posterior covariance term. In both filters, the cross-terms in the higher order expansions of the covariance are ignored to avoid combinatorial explosion and keep computational cost down. The CDF and the UKF simply retain a different subset of these terms. The CDF does, however, have a smaller absolute error in the fourth order term and also guarantees positive semi-definiteness (PSD) of the posterior covariance (a single scale factor $h$ is used[1] ). In contrast, the UKF may result in a non-positive semi-definite covariance, which is compensated for using two additional heuristic scaling parameters [2].

Both the CDF and UKF provide substantial performance increase over the EKF in state estimation problems, and are in general $\mathcal{O}(L^3)$. However, for *parameter estimation*, the specific form of the state-space equations allow the EKF to be implemented in $\mathcal{O}(L^2)$. In the following section, we present new *square-root* implementations of the derivative-free Kalman filters that are also $\mathcal{O}(L^2)$.

## 3. Efficient Square-Root Implementation

In the standard Kalman implementation, the state (parameter) covariance $\mathbf{P}_{\mathbf{w}}$ is recursively calculated. The UKF requires taking the matrix square-root, $\mathbf{S}_{\mathbf{w}}\mathbf{S}_{\mathbf{w}}^T = \mathbf{P}_{\mathbf{w}}$, at each time step, which is $\mathcal{O}(L^3/6)$ using a Cholesky factorization. In the *square-root UKF* (SR-UKF) and *square-root CDF* (SR-CDF) implementations, $\mathbf{S}_{\mathbf{w}}$ (as well as the Cholesky factor, $\mathbf{S}_{\mathbf{d}}$, of the observation-error covariance) will be propagated directly, avoiding the need to refactorize at each time step. This and the special state-space formulation of parameter estimation allows for an $\mathcal{O}(L^2)$ implementation[2]. The complete specification of the new square-root filters are given in Algorithm 3.1.

[1] The scale $h$ is optimally set equal to the *kurtosis* of the prior RV [4, 3]. For Gaussians, $h = \sqrt{3}$.

[2] Although both Norgaard and Ito [4, 3] also implemented the CDF and DDF in a square-root form, this was in the framework of state estimation and did not exploit efficient Cholesky updates or the special form of the parameter estimation state-space that leads to reduction in computational cost.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 205-210

Initialize with: $\hat{\mathbf{w}}_0 = E[\mathbf{w}]$, $\mathbf{S}_{\mathbf{w}_0} = \text{chol}\left\{E[(\mathbf{w}-\hat{\mathbf{w}}_0)(\mathbf{w}-\hat{\mathbf{w}}_0)^T]\right\}$
For $k \in \{1, \ldots, \infty\}$,
Time update and sigma point calculation:

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1} \tag{6}$$

$$\mathbf{S}_{\mathbf{w}_k}^- = \lambda_{RLS}^{-1/2}\mathbf{S}_{\mathbf{w}_{k-1}} \quad \text{or} \quad \mathbf{S}_{\mathbf{w}_k}^- = \mathbf{S}_{\mathbf{w}_{k-1}} + \mathbf{D}_{\mathbf{r}_{k-1}} \tag{7}$$

$$\boldsymbol{\mathcal{W}}_{k|k-1} = \left[\hat{\mathbf{w}}_k^- \quad \hat{\mathbf{w}}_k^- + \gamma\mathbf{S}_{\mathbf{w}_k}^- \quad \hat{\mathbf{w}}_k^- - \gamma\mathbf{S}_{\mathbf{w}_k}^-\right] \tag{8}$$

$$\boldsymbol{\mathcal{D}}_{k|k-1} = \mathbf{G}[\mathbf{x}_k, \boldsymbol{\mathcal{W}}_{k|k-1}] \tag{9}$$

$$\hat{\mathbf{d}}_k = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{D}_{i,k|k-1} \tag{10}$$

Measurement update equations:
`if` **SR-UKF**:

$$\mathbf{S}_{\mathbf{d}_k} = \text{qr}\left\{\left[\sqrt{W_1^{(c)}}\left[\boldsymbol{\mathcal{D}}_{1:2L,k} - \hat{\mathbf{d}}_k\right] \quad \sqrt{\mathbf{R}^{\mathbf{e}}}\right]\right\} \tag{11}$$

$$\mathbf{S}_{\mathbf{d}_k} = \text{cholupdate}\left\{\mathbf{S}_{\mathbf{d}_k}\ ,\ \mathcal{D}_{0,k} - \hat{\mathbf{d}}_k\ ,\ W_0^{(c)}\right\} \tag{12}$$

$$\mathbf{P}_{\mathbf{w}_k\mathbf{d}_k} = \sum_{i=0}^{2L} W_i^{(c)}[\mathcal{W}_{i,k|k-1} - \hat{\mathbf{w}}_k^-][\mathcal{D}_{i,k|k-1} - \hat{\mathbf{d}}_k]^T \tag{13}$$

`elseif` **SR-CDF**:

$$\mathbf{S}_{\mathbf{d}_k} = \text{qr}\left\{\left[\sqrt{W_1^{(c)}}\left[\boldsymbol{\mathcal{D}}_{1:L,k} - \boldsymbol{\mathcal{D}}_{L+1:2L,k}\right] \ \sqrt{W_2^{(c)}}\left[\boldsymbol{\mathcal{D}}_{1:L,k} + \boldsymbol{\mathcal{D}}_{L+1:2L,k} - 2\mathcal{D}_{0,k}\right] \ \sqrt{\mathbf{R}^{\mathbf{e}}}\right]\right\}$$

$$\mathbf{P}_{\mathbf{w}_k\mathbf{d}_k} = \mathbf{S}_{\mathbf{w}_k}^-\left[\sqrt{W_1^{(c)}}\left[\boldsymbol{\mathcal{D}}_{1:L,k} - \boldsymbol{\mathcal{D}}_{L+1:2L,k}\right]\right]^T \tag{14}$$

`end`

$$\mathcal{K}_k = (\mathbf{P}_{\mathbf{w}_k\mathbf{d}_k}/\mathbf{S}_{\mathbf{d}_k}^T)/\mathbf{S}_{\mathbf{d}_k} \tag{15}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathcal{K}_k(\mathbf{d}_k - \hat{\mathbf{d}}_k) \tag{16}$$

$$\mathbf{U} = \mathcal{K}_k\mathbf{S}_{\mathbf{d}_k} \tag{17}$$

$$\mathbf{S}_{\mathbf{w}_k} = \text{cholupdate}\left\{\mathbf{S}_{\mathbf{w}_k}^-\ ,\ \mathbf{U}\ ,\ \text{-1}\right\} \tag{18}$$

$\mathbf{R}^{\mathbf{e}}$=measurement noise covariance (this can be set to an arbitrary value, *e.g.*, $.5\mathbf{I}$.). $\mathbf{D}_{\mathbf{r}_{k-1}} = -Diag\left\{\mathbf{S}_{\mathbf{w}_{k-1}}\right\} + \sqrt{Diag\left\{\mathbf{S}_{\mathbf{w}_{k-1}}\right\}^2 + Diag\left\{\mathbf{R}_{k-1}^{\mathbf{r}}\right\}}$. qr$\{\cdot\}$ denotes the QR decomposition of a matrix where only the upper triangular part of $\mathbf{R}$ is returned (this equals the transpose of the Cholesky factor of $\mathbf{P} = \mathbf{A}\mathbf{A}^T$). $\mathbf{S} = \text{cholupdate}\{\mathbf{S}, \mathbf{u}, \pm\nu\}$ denotes the $M$ consecutive rank-1 updates (or downdates), $\mathbf{P} \pm \sqrt{\nu}\mathbf{u}\mathbf{u}^T$, using the $M$ columns of $\mathbf{u}$. $\mathbf{A}/\mathbf{b}$ denotes the least squares solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ (solved using a QR decomposition with pivoting).
**SR-UKF parameters** : $\{W_i\}$ is a set of scalar weights, $W_0^{(m)} = \lambda/(L+\lambda)$ , $W_0^{(c)} = \lambda/(L+\lambda) + (1-\alpha^2+\beta)$ , $W_i^{(m)} = W_i^{(c)} = 1/\{2(L+\lambda)\}\,(i = 1, \ldots, 2L)$. $\lambda = \alpha^2(L+\kappa) - L$ , $\gamma = \sqrt{(L+\lambda)}$, $\kappa$ and $\alpha$ are scaling parameters.
**SR-CDF parameters** : $\{W_i\}$ is a set of scalar weights, $W_0^{(m)} = (h^2 - L)/h^2$, $W_i^{(m)} = 1/2h^2\,(i = 1, \ldots, 2L)$, $W_1^{(c)} = 1/4h^2$ , $W_2^{(c)} = (h^2-1)/4h^4$, $h \geq 1$ is the scalar central difference step size ($\gamma = h$ in Eqn. 8).

**Algorithm 3..1:** SR-UKF and SR-CDF for parameter estimation.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 205-210

Key elements of the algorithm are as follows: The time-update of the state covariance (for the special parameter estimation case) is given simply by $\mathbf{P}_{\mathbf{w}_k}^- = \mathbf{P}_{\mathbf{w}_{k-1}} + \mathbf{R}_{k-1}^{\mathbf{r}}$. In the square-root filters, $\mathbf{S}_{\mathbf{w}_k}$ may thus be updated directly in Eqn 7 using one of two options: 1) $\mathbf{S}_{\mathbf{w}_k}^- = \lambda_{RLS}^{-1/2}\mathbf{S}_{\mathbf{w}_{k-1}}$, corresponding to a an exponential weighting on past data[3]. 2) $\mathbf{S}_{\mathbf{w}_k}^- = \mathbf{S}_{\mathbf{w}_{k-1}} + \mathbf{D}_{\mathbf{r}_{k-1}}$, where the diagonal matrix $\mathbf{D}_{\mathbf{r}_{k-1}}$, is chosen to approximate the effects of annealing a diagonal process noise covariance $\mathbf{R}_{k-1}^{\mathbf{r}}$ [4] or a Robbins-Monro derived noise estimate [10]. Next, in Eqn. 14 the Cholesky factor, $\mathbf{S}_{\mathbf{d}_k}$, is calculated using a QR decomposition of the compound matrix containing the weighted observed sigma points and the matrix square-root of the additive measurement noise covariance. This step differs slightly in the SR-UKF and is broken up into two steps: Firstly, because the zeroth weight, $W_0^{(c)}$, may be negative in the SR-UKF, the first observed sigma point is not included in the QR decomposition. Instead, it is incorporated by a subsequent Cholesky update (or downdate) in Eqn. 12 depending on the sign of $W_0^{(c)}$. These steps are $\mathcal{O}(LM^2)$, where $M$ is the observation dimension. In contrast to the way the Kalman gain is calculated in the standard UKF, we make use of Norgaard's method based on two nested inverse (or *least squares*) solutions to the following expansion of Eqn. 15, $\mathcal{K}_k(\mathbf{S}_{\mathbf{d}_k}\mathbf{S}_{\mathbf{d}_k}^T) = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k}$. Since $\mathbf{S}_{\mathbf{d}}$ is square and triangular, efficient "back-substitutions" can be used to solve for $\mathcal{K}_k$ directly without the need for a matrix inversion. Finally, the posterior measurement update of the Cholesky factor of the state covariance is calculated in Eqn. 18 by applying $M$ sequential Cholesky downdates to $\mathbf{S}_{\mathbf{w}_k}^-$, which requires $\mathcal{O}(L^2M)$ computations.

## 4. Experimental Results

The improvement in error performance of the UKF and CDF over that of the EKF for state estimation is well documented [2, 5, 9, 4]. For parameter estimation, the performance of the different filters are expected to be more comparable (this is because the state-transition function is linear, Eqn. 1, while the nonlinearity arises only in the observation equation, Eqn. 2). The main advantage of the derivative-free forms is in avoiding the need to evaluate Jacobians or Hessians. The focus of this section will be to verify the error performance of the SR-UKF and the SR-CDF compared to the EKF and UKF, and show the reduction in computational cost achieved by the efficient square-root forms.

For the first experiment, we train a 2-12-2 MLP neural network on the well known *Mackay-Robot-Arm*[5] benchmark problem of mapping robotic joint angles to Cartesian hand coordinates. The learning curves of the different filters are shown in Figure 2a. As expected, the performance of all filters are comparable. In the next example, we consider training a 2-10-10-4 network on a benchmark pattern classification problem having four interlocking regions (see [7] for details). Figure 2b illustrates learning curves for the different filters, and again shows the equivalent performance of the square-root derivative free approaches relative the EKF. Finally, Figure 3 shows how the computational complexity of the different filters scale as a function of the number of parameters (MLP weights). Clearly, the EKF and all square-root filters are $\mathcal{O}(L^2)$.

[3] This is identical to the approach used in weighted recursive least squares (W-RLS). $\lambda_{RLS}$ is a scalar weighting factor chosen to be slightly less than 1, *i.e.* $\lambda_{RLS} = 0.9995$.

[4] This updates ensures the main diagonal of $\mathbf{P}_{\mathbf{w}_k}^-$ is exact. However, additional off-diagonal cross-terms $\mathbf{S}_{\mathbf{w}_{k-1}}\mathbf{D}_{\mathbf{r}_{k-1}}^T + \mathbf{D}_{\mathbf{r}_{k-1}}\mathbf{S}_{\mathbf{w}_{k-1}}^T$ are also introduced (though the effect appears negligible).

[5] http://wol.ra.phy.cam.ac.uk/mackay

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 205-210

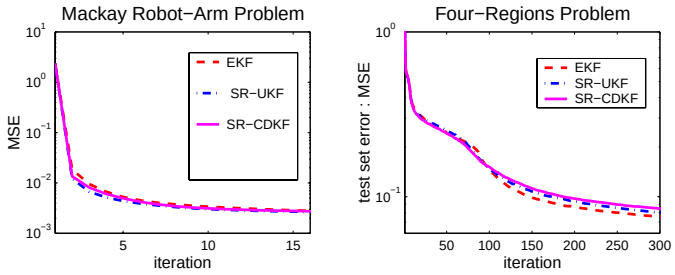


Figure 2: Learning curves for NN training (MSE vs. epoch). Settings: $[\alpha = 1, \beta = 2, h = \sqrt{3}]$.

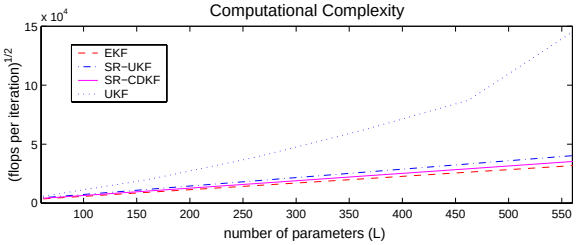


Figure 3: Computational complexity: flops/epoch vs. # of parameters (Mackay-Robot-Arm).

## 5. Conclusions

In this paper, we showed how different derivative-free filters can be related, and introduced square-root forms, specifically applied to parameter estimation for training neural networks. The square-root forms result in efficient $\mathcal{O}(L^2)$ implementations, have better numerical properties than their non square-root forms, and provide similar performance relative to EKF parameter estimation without the need to analytical calculate Jacobians.

## References

[1] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the American Control Conference*, 1995, pp. 1628–1632.

[2] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proc. of AeroSense: The 11th Int. Symp. A.D.S.S.C.*, 1997.

[3] Kazufumi Ito and Kaiqi Xiong, "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, may 2000.

[4] M. Nørgaard, N. K. Poulsen, and O. Ravn, "Advances in Derivative-Free State Estimation for Nonlinear Systems," Tech. Rep. IMM-REP-1998-15, Tech. Univ. of Denmark, 2000.

[5] E. Wan, R. van der Merwe, and A. T. Nelson, "Dual Estimation and the Unscented Transformation," in *Neural Information Processing Systems 12*. 2000, pp. 666–672, MIT Press.

[6] Rudolph van der Merwe, Nando de Freitas, Arnaud Doucet, and Eric Wan, "The unscented particel filter," in *Advances in Neural Information Processing Systems 13*, Nov 2001.

[7] S. Singhal and L. Wu, "Training multilayer perceptrons with the extended Kalman filter," in *Advances in Neural Information Processing Systems 1*, San Mateo, CA, 1989, pp. 133–140.

[8] G.V. Puskorius and L.A. Feldkamp, "Decoupled Extended Kalman Filter Training of Feedforward Layered Networks," in *IJCNN*, 1991, vol. 1, pp. 771–777.

[9] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Proc. of IEEE Symposium 2000 (AS-SPCC)*, Lake Louise, Alberta, Canada, Oct. 2000.

[10] E. A. Wan and R. van der Merwe, "Chapter 7 : The Unscented Kalman Filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed., Wiley Publishing, (in press) 2001.