Bayesian updating for nonlinear dynamics problems using machine learning

W. van Dijk, W.B.J. Hakvoort, B.V. Rosić
University of Twente, Department of Mechanics of Solids, Surfaces & Systems,
P.O. box 217, 7500AE, Enschede, The Netherlands
e-mail: **w.vandijk@utwente.nl**

Abstract
The probabilistic parameter and state estimation of highly nonlinear dynamical system can be quite challenging due to presence of both nonlinearity and non-Gaussianity. In this paper we show a computationally efficient approach to the previously mentioned estimation. By setting the problem in a Bayesian framework, we show that the posterior mean can be directly estimated given measurement data if the optimal map representing the conditional expectation of the quantity to be estimated given the observation variable is known. As an example, we explore the simple nonlinear feedforward neural network as an approximation of conditional expectation. The approach is tested on a highly nonlinear Lorenz-63 system by employing Monte Carlo sampling.

1 Introduction

The predictive modelling of physical systems requires careful model identification given noisy measurement data. Assuming that the physical phenomena can be accurately modelled by first principles, the model identification eventually collapses to the problem of state and parameter identification. However, the system dynamics is often nonlinear and non-invertible, and the measurement data represent the direct noisy observation of only couple of the system states. Hence, the estimation is known to be ill-posed in a sense of Hadamard, and the appropriate regularisation approach is needed. In a probabilistic setting this matches with the a priori expert's description of unknown quantities. By describing the unknown states and parameters as uncertain, one may model them as random variables in a probabilistic space, and further assimilate them with the measurement data under the Bayes rule. The Bayesian framework then offers the decision making between the experts knowledge and the measurement data embodied in an estimated posterior distribution.

In case of simultaneous state and parameter estimation under nonlinear system dynamics, the relation between the parameters and the measurements is often non-linear. This makes the estimation of the likelihood function difficult and computationally expensive. The main reason for this is the uncertainty propagation of a priori uncertainty through the nonlinear system dynamics. However, as often the interest lies not in estimating the full posterior but its mean, few attempts are made in the direct estimation of posterior statistics. Such an approach is known as Kalman filtering and its corresponding versions [1, 2, 3].

The classical Kalman filter (KF) [1] is an optimal-variance filter for linear systems under the Gaussian assumption, and therefore fails when non-linear dynamics are encountered. Instead, the extended Kalman filter (EKF) [2] is proposed with the aim of linearizing the measurement operator by a first order Taylor series expansion. The EKF filter can be further improved by iterating over the linearisation points [4] or including higher-order terms in the Taylor series expansion. However, the previously mentioned filter is focusing on the linearization aspect, and not on the proper approximation of non-Gaussianity. Therefore, the Ensemble Kalman filter (EnKF) [3] is introduced in which the uncertainties are represented by a sample of sufficient size. The EnKF however requires a linear observation map, which limits its effectiveness for state and parameter estimation of highly non-linear dynamics.

A generalization of the Kalman filters mentioned above is a filter derived from the conditional expectation

[5]. This general filter is free from Gaussianity and linearity assumptions on the system dynamics and the observation function, and is therefore suitable as a starting point for the design of new filters. Under their respective assumptions, this generalization reduces to the classical KF, the EKF, and the EnKF, among others.

In this paper, we propose machine learning approach to the general filter proposed in [6]. We investigate the ensemble implementation of the general filter form in combination with a feed forward neural network and compare it against its linear optimal variant on a Lorenz-63 example.

The remainder of the paper is structured as follows. In Section 2, the problem setting and the method are introduced. In Section 3, the numerical evaluation of the proposed filter performance is made. In Section 4 the discussion and the concluding remarks are drawn.

2 Problem setting

Consider the following non-linear initial value problem

$$\frac{\mathrm{d}\boldsymbol{x}(t,\boldsymbol{q})}{\mathrm{d}t} = f(\boldsymbol{x}(t),\boldsymbol{q}), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1}$$

in which $\boldsymbol{x}(t,\boldsymbol{q})$ denotes the state-vector of the dynamical system and $\boldsymbol{q}$ is a vector containing parameters describing system properties. This system is observed at discrete time moments t_k by

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k), \tag{2}$$

where $h(\cdot)$ denotes a possibly non-linear observation operator. Typically, the observations are corrupted by sensory noise such that

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k^{true}) + \boldsymbol{e}_k, \tag{3}$$

holds, where $\boldsymbol{x}_k^{true}$ denotes the true state and $\boldsymbol{e}_k$ is a realization of the sensory noise. For notational simplicity the time index is dropped in the notation. Thus, at the time instance t_k, the observation is $\boldsymbol{y} = h(\boldsymbol{x})$ and the measurement is $\boldsymbol{z} = h(\boldsymbol{x}^{true}) + \boldsymbol{e}$.

Typically, the states and/or parameters in Eq. (1) are unknown and are to be estimated given the measurement data in Eq. (3). As such an estimation is known to be ill-posed in a Hadamard sense, we restrain ourselves to Bayesian approach in which unknown parameters/states are described by prior expert's knowledge. The overachieving goal is therefore to combine the information present in measurements with the prior available information on the states and parameters, and in doing so, to improve our knowledge about the uncertain states and/or parameters.

2.1 Estimation by conditional expectation

In a Bayesian setting the model parameters and states are modelled a priori as random variables:

$$\boldsymbol{\kappa}_f := [\boldsymbol{x}_f(t,\omega_f), \boldsymbol{q}_f(\omega_f)],$$

belonging to a probability space $(\Omega_f, \mathcal{F}_f, \mathbb{P}_f)$ in which the index "f" denotes "the forecast". Here Ω_f denotes the sample space, $\mathcal{F}_f$ denotes the σ-algebra, and $\mathbb{P}_f$ is a probability measure.

Incorporating prior information in Eq. (1) results in a stochastic ordinary differential equation of the following form:

$$\frac{\mathrm{d}\boldsymbol{x}_f(t,\omega_f)}{\mathrm{d}t} = f(\boldsymbol{x}_f(t,\omega_f), \boldsymbol{q}_f(\omega_f)), \tag{4}$$

with the initial condition being a random variable $\boldsymbol{x}_0(\omega_f)$. As a consequence of Eq. (4), the measurement forecast becomes a random variable too, i.e.,

$$\boldsymbol{y}_f(\omega_f) := h(\boldsymbol{x}_f(\omega_f)) + \boldsymbol{\varepsilon}(\omega_\varepsilon), \tag{5}$$

in which $\boldsymbol{\varepsilon}(\omega_\varepsilon)$ is the predicted measurement noise belonging to $(\Omega_\varepsilon, \mathcal{F}_\varepsilon, \mathbb{P}_\varepsilon)$ and is typically described by a zero-mean Gaussian random variable with a covariance matrix C_ε, e.g. $\boldsymbol{\varepsilon}(\omega_\varepsilon) \sim \mathcal{N}(\mathbf{0}, C_{\boldsymbol{\varepsilon}})$. Assuming independence between the measurement noise $\boldsymbol{\varepsilon}$ and $\boldsymbol{\kappa}_f$, the overall probability space is described by the triplet $(\Omega := \Omega_f \times \Omega_\varepsilon, \mathcal{F} := \sigma(\mathcal{F}_f \times \mathcal{F}_\varepsilon), \mathbb{P} := \mathbb{P}_f \mathbb{P}_\varepsilon)$, and an elementary event ω. Thus, $\boldsymbol{\kappa}_f(\omega_f)$ is further denoted as $\boldsymbol{\kappa}_f(\omega)$, and $\boldsymbol{\varepsilon}(\omega_\varepsilon)$ as $\boldsymbol{\varepsilon}(\omega)$.

To assimilate the prior knowledge with a measurement, a general filter form

$$\boldsymbol{\kappa}_a(\omega) = \boldsymbol{\kappa}_f(\omega) + \mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{z}] - \mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{y}_f(\omega))], \tag{6}$$

derived in [6, 7] is used. Thus, the posterior mean $\boldsymbol{\kappa}_a(\omega)$ ("a" stands for "assimilated") linearly depends both on the prior knowledge $\boldsymbol{\kappa}_f(\omega)$ and the innovation term $\mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{z}] - \mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{y}_f(\omega))]$. Here, $\mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{z}]$ and $\mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{y}_f(\omega)]$ denote the conditional expectations of $\boldsymbol{\kappa}_f$ given the measurement $\boldsymbol{z}$ or the predicted measurement $\boldsymbol{y}_f(\omega)$, respectively.

By Doob-Dynkin lemma the conditional expectation $\mathbb{E}[\boldsymbol{\kappa}_f|\cdot]$ can be approximated by a map $\varphi_{\boldsymbol{\kappa}}(\cdot)$ parametrized by $\boldsymbol{\beta}$, which can be estimated given following optimality condition:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \mathbb{E}[\|\boldsymbol{\kappa}_f(\omega) - \varphi_{\boldsymbol{\kappa}}(\boldsymbol{y}_f(\omega), \boldsymbol{\beta})\|^2], \tag{7}$$

as shown in [7]. This further leads to

$$\boldsymbol{\kappa}_a(\omega) = \boldsymbol{\kappa}_f(\omega) + \varphi_{\boldsymbol{\kappa}}(\boldsymbol{z}, \boldsymbol{\beta}^*) - \varphi_{\boldsymbol{\kappa}}(\boldsymbol{y}_f(\omega), \boldsymbol{\beta}^*). \tag{8}$$

Based on the problem at hand, one can pick a suitable model structure for the map $\varphi_{\boldsymbol{\kappa}}$. In this work, under appropriate differentiability assumption we compare two different types of maps: a simple linear map like in a classical Kalman filter setting, and a nonlinear map in the form of a non-linear feedforward neural network.

2.2 Optimal linear map

In a linear form the conditional expectation reads:

$$\mathbb{E}[\boldsymbol{\kappa}_f(\omega)|\boldsymbol{y}_f(\omega))] \approx \varphi_{\boldsymbol{\kappa}}(\boldsymbol{y}_f(\omega)) = K\boldsymbol{y}_f(\omega) + \mathbf{b}, \tag{9}$$

such that Eq. (7) rewrites to

$$\arg\min_{K,\boldsymbol{b}} \|\boldsymbol{\kappa}_f(\omega) - (K\boldsymbol{y}_f(\omega) + \mathbf{b})\|^2. \tag{10}$$

The analytical solution of the previous optimization problem as shown by [8] reads:

$$K = C_{\boldsymbol{\kappa}_f \boldsymbol{y}_f} (C_{\boldsymbol{y}_f})^{-1}, \tag{11}$$

$$\boldsymbol{b} = \mathbb{E}[\boldsymbol{\kappa}_f(\omega) - K\boldsymbol{y}_f(\omega)], \tag{12}$$

in which $C_{\boldsymbol{\kappa}_f \boldsymbol{y}_f}$ denotes the cross-covariance between the forecast $\boldsymbol{\kappa}_f(\omega)$ and the observation forecast $\boldsymbol{y}_f(\omega)$ and $C_{\boldsymbol{y}_f}$ denotes the covariance of the measurement forecast.

By substituting the optimal linear parameters from Eq. (11) into Eq. (8), one obtains the Gauss-Markov-Kalman filter:

$$\boldsymbol{\kappa}_a(\omega) = \boldsymbol{\kappa}_f(\omega) + K(\boldsymbol{z} - \boldsymbol{y}_f(\omega)), \tag{13}$$

which reduces to the well-known Kalman equation under Gaussianity assumption. The latter assumption is not present in this work, meaning that Eq. (13) can also be used in a non-Gaussian case.

When evaluating Eq. (13), the random variables can be discretised by Monte Carlo sampling. By collecting N independent samples of $\boldsymbol{\kappa}_f(\omega)$ in the matrix:

$$\boldsymbol{Q}_f = [\boldsymbol{\kappa}_f(\omega_1), \boldsymbol{\kappa}_f(\omega_2), \ldots, \boldsymbol{\kappa}_f(\omega_N)], \tag{14}$$

and similarly for $\boldsymbol{h}_f(\omega)$ and $\boldsymbol{e}(\omega)$:

$$\boldsymbol{Y}_f = [\boldsymbol{y}_f(\omega_1), \boldsymbol{y}_f(\omega_2), \ldots, \boldsymbol{y}_f(\omega_N)], \tag{15}$$

one may to rewrite Eq. (13) to

$$\boldsymbol{Q}_a = \boldsymbol{Q}_f + \boldsymbol{K}(\boldsymbol{Z} - \boldsymbol{Y}_f), \tag{16}$$

in which $\boldsymbol{Z} = \boldsymbol{z} \otimes \boldsymbol{1}_N$ with $\boldsymbol{1}_N$ being the unit vector of length N. Here, the matrix $\boldsymbol{K}$ from Eq. (11) is evaluated using the ensemble covariances:

$$C_{\boldsymbol{\kappa}_f \boldsymbol{y}_f} \approx \frac{\tilde{\boldsymbol{Q}}_f \tilde{\boldsymbol{Y}}_f^\intercal}{N-1}, \qquad C_{\boldsymbol{y}_f} \approx \frac{\tilde{\boldsymbol{Y}}_f \tilde{\boldsymbol{Y}}_f^\intercal}{N-1}, \tag{17}$$

in which $\tilde{\boldsymbol{Q}}_f$ and $\tilde{\boldsymbol{Y}}_f$ denote the variational parts of $\boldsymbol{Q}_f$ and $\boldsymbol{Y}_f$ respectively, i.e.,

$$\tilde{\boldsymbol{Q}}_f = \boldsymbol{Q}_f - \bar{\boldsymbol{\kappa}}_f \boldsymbol{1}_N, \qquad \tilde{\boldsymbol{Y}}_f = \boldsymbol{Y}_f - \bar{\boldsymbol{y}}_f \boldsymbol{1}_N. \tag{18}$$

Finally, $\bar{\boldsymbol{\kappa}}_f$ and $\bar{\boldsymbol{y}}_f$ denote the mean estimates, i.e.,

$$\bar{\boldsymbol{\kappa}}_f = \frac{1}{N} \sum_n^N \boldsymbol{\kappa}_f(\omega_n), \qquad \bar{\boldsymbol{y}}_f = \frac{1}{N} \sum_n^N \boldsymbol{y}_f(\omega_n). \tag{19}$$

Under the assumptions stated above, this implementation is equivalent to the Ensemble Kalman filter (EnKF) [9].

2.3 Optimal neural network map

When the observation operator is non-linear or if one of the random variables in Eq. (8) is non-Gaussian, the linear map is not optimal anymore. Therefore, in this work a feedforward neural network is used to approximate the conditional expectations. Note that there exist many more models structures that are potentially equally or more suitable. For simplicity they are not considered in this paper.

The neural network used to map $\boldsymbol{y}_f(\omega)$ to $\boldsymbol{\kappa}_f(\omega)$, here denoted by $g(\boldsymbol{y}_f(\omega), \boldsymbol{w})$, is parametrized on a set of weights collected in $\boldsymbol{w}$. These weights are trained by minimizing an objective function, in this application taken to be in a form given in Eq. (7). The objective is thus to minimize:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathbb{E}[\|\boldsymbol{\kappa}_f(\omega) - g(\boldsymbol{y}_f(\omega), \boldsymbol{w})\|^2], \tag{20}$$

which is typically solved by sampling $\boldsymbol{\kappa}_f(\omega)$ and $\boldsymbol{y}_f(\omega)$ and using gradient-based methods as explained in Section 2.3.1.

After computing the optimal weights, the update equation in (8) transforms to:

$$\boldsymbol{\kappa}_a(\omega) = \boldsymbol{\kappa}_f(\omega) + g(\boldsymbol{z}, \boldsymbol{w}^*) - g(\boldsymbol{y}_f(\omega), \boldsymbol{w}^*). \tag{21}$$

Similar to Section 2.2, the random variables are discritized by Monte Carlo sampling such that Eq. (21) reads:

$$\boldsymbol{Q}_a = \boldsymbol{Q}_f + g(\boldsymbol{z}, \boldsymbol{w}^*) - g(\boldsymbol{Y}_f, \boldsymbol{w}^*). \tag{22}$$

An overview of the nonlinear filter is shown, together with the linear filter, in Algorithm 1.

2.3.1 Training

The feedforward network used in this work contains four hidden layers with $[64, 64, 32, 16]$ nodes and a linear output layer. All hidden layers have rectified linear (ReLu) activation functions and during training Gaussian noise with a covariance matrix $C_{\boldsymbol{\varepsilon}}$ is added to the training inputs.

(a) (b)

Figure 1: Model structure during (a) training and (b) prediction.

The update equation in Eq. (21) contains the neural network after training, therefore the weights must be re-trained before every update. At time t_k, the weights are trained using a dataset $\mathcal{D} := \{d_1, \ldots, d_N\}$ containing pairs of input/output data, $d_n = (h(\boldsymbol{x}_f(\omega_n)), \boldsymbol{\kappa}_f(\omega_n))$, available from the forecast ensembles. The weights are trained for 100 epochs using the Adam optimizer [10] with a learning rate of $\lambda = 0.0001$ and a batchsize of 32. In accordance with the optimality condition in Eq. (20), a mean squared error (MSE) is minimized during training. An overview of the network during training and prediction can be seen in Figure 1.

Algorithm 1: Ensemble implementation of the linear and nonlinear filters

Input: $\boldsymbol{X}_f^{t_0}, \boldsymbol{Q}_f^{t_0}, C_{\boldsymbol{\varepsilon}}, \boldsymbol{w}$
$k = 1$
for $k < k_{max}$ **do**
  Forecast
  State forecast
  $\boldsymbol{X}_f^{t_k} = odeSolve(\boldsymbol{Q}_a^{t_{k-1}}, \Delta t, [t_{k-1}, t_k])$; ▷ integrate ensemble
  $\boldsymbol{Q}_f^{t_k} \leftarrow [\boldsymbol{X}_f^{t_k}, \boldsymbol{Q}_a^{t_{k-1}}]$; ▷ collect state and parameter forecast
  Measurement forecast
  $\boldsymbol{Y}_f^{t_k} = h(\boldsymbol{X}_f^{t_k}) + \boldsymbol{E}^{t_k}$
  Approximate conditional expectation
  $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathbb{E}[\|\boldsymbol{Q}_f(\omega) - g(\boldsymbol{Y}_f, \boldsymbol{w})\|^2]$ (non-linear)
  or
  $K, \boldsymbol{b} = \arg\min_{K,\boldsymbol{b}} \mathbb{E}[\|\boldsymbol{Q}_f(\omega) - (K\boldsymbol{Y}_f + \boldsymbol{b})\|^2]$ (linear)
  Measurement:
  $z = h(\boldsymbol{q}_{true}) + \boldsymbol{e}$
  Assimilate:
  $\boldsymbol{Q}_a^{t_k} = \boldsymbol{Q}_f^{t_k} + g(\boldsymbol{z}) - g(\boldsymbol{Y}_f^{t_k}, \boldsymbol{w}^*)$ (non-linear)
  or
  $\boldsymbol{Q}_a^{t_k} = \boldsymbol{Q}_f^{t_k} + K(\boldsymbol{z} - \boldsymbol{Y}_f^{t_k})$ (linear)
  Update:
  $k = k + 1$
end

3 Numerical example

In this section the proposed filter is applied on the Lorenz-63 model, a coupled nonlinear ordinary differential equation introduced in [11]. This problem is well-studied and used frequently in the data assimilation community, see for example [12, 3, 13, 14] and particularly for state and parameter estimation [15, 16, 17].

The governing equations of the model are

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= -\sigma x + \sigma y, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \rho x - xz - y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z. \end{aligned} \tag{23}$$

Here, x, y, and z are the states collected in $\boldsymbol{x} = [x, y, z]$, and σ, ρ, and β are the parameters collected in $\boldsymbol{q} = [\sigma, \rho, \beta]$ such that it resembles Eq. (1).

The parameters are typically chosen as $\boldsymbol{q} = [10, 28, 8/3]$ and the true initial state is chosen as $\boldsymbol{x}_0 = [1.508870, -1.531271, 25.46091]$ taken from [12]. The reference solution is obtained by integrating Eq. (23) forward in time using a fourth order Runge-Kutta scheme with fixed time intervals of $\Delta t = 0.01$. The system is observed at time intervals of Δt_k and the noisy measurements are obtained by adding Gaussian noise with a variance of σ_e^2 to the reference solution at the observation instances.

3.1 Experiment A

In this experiment the states are observed every two time units, $\Delta t_k = 1.00$, equivalent to 100 integration steps. The magnitude of this observation interval ensures that highly non-linear dynamics are present between updates (see Figure 2b). Direct measurements are performed, i.e. $h(\boldsymbol{x}) = \boldsymbol{x}$, and the measurement noise variance is set to $\sigma_n^2 = 2$. Note that the parameters are not measured directly, i.e. they are non-linearly present in the measurements.

The initial state $\boldsymbol{x}_0$ and the parameters $\boldsymbol{q}$ are uncertain and are therefore modelled as independent normal and uniform distributed random variables

$$\boldsymbol{x}_0(\omega) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}_0}, \sigma_{\boldsymbol{x}_0}^2 I), \qquad \boldsymbol{q}(\omega) \sim \mathcal{U}(\boldsymbol{q}_{min}, \boldsymbol{q}_{max}), \tag{24}$$

respectively. Here $\boldsymbol{\mu}_{\boldsymbol{x}_0}$ and $\boldsymbol{\sigma}_{\boldsymbol{x}_0}^2$ are the prior mean and variance of the initial states, whereas $\boldsymbol{q}_{min}$ and $\boldsymbol{q}_{max}$ defines the interval of the prior uniform distribution of the parameters. The prior mean and variances of the initial state are chosen according to

$$\boldsymbol{\mu}_{\boldsymbol{x}_0} = \boldsymbol{x}_0, \qquad \boldsymbol{\sigma}_{\boldsymbol{x}_0}^2 = [2, 2, 2],$$

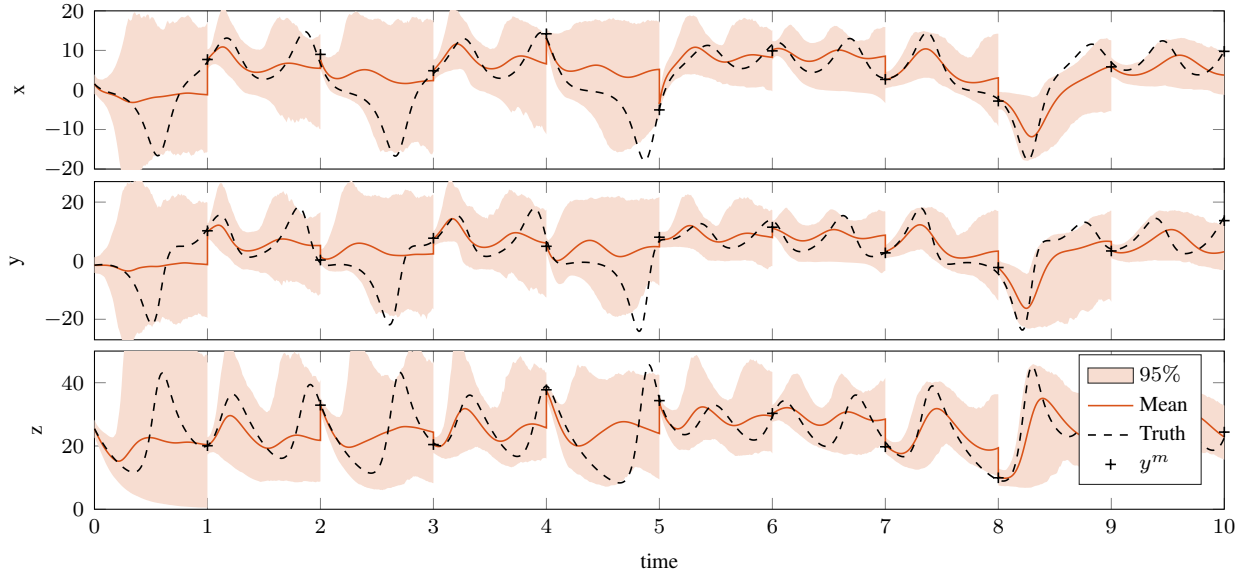
and the interval over which the parameters are uniformly distributed is:

$$\boldsymbol{q}_{min} = [1, 1, 1], \qquad \boldsymbol{q}_{max} = [30, 44.8, 5.3].$$

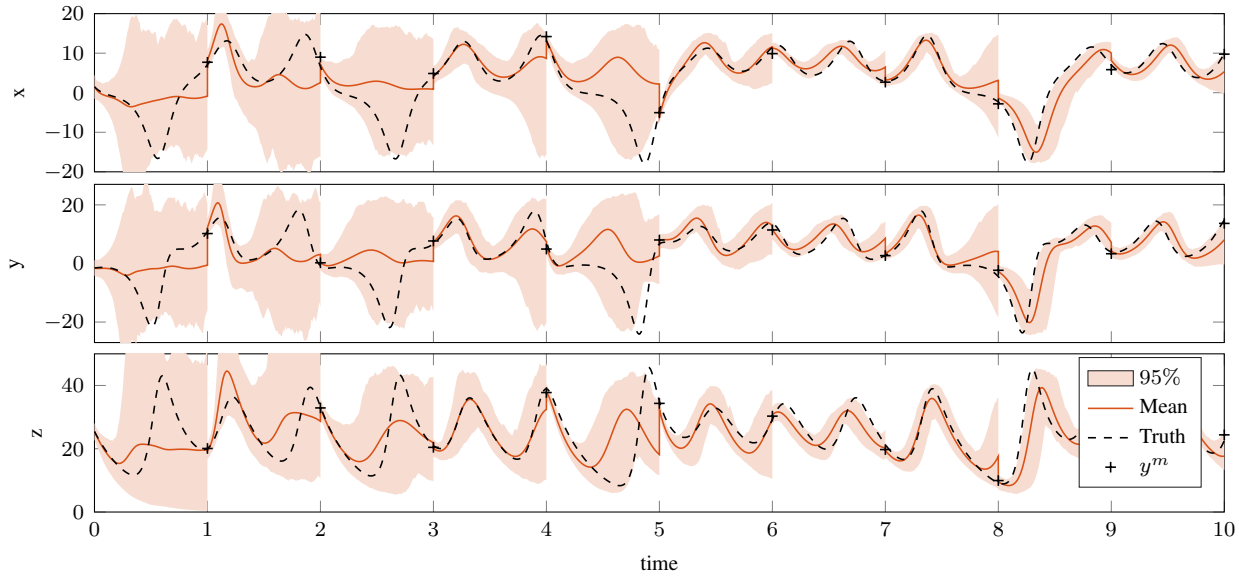
The two filters are applied on the Lorenz-63 system in the time interval $t = [0, 10]$, equivalent to 10 assimilation steps. The filters are implemented with an ensemble size of $N = 256$ and $N = 1024$ to compare the effects of sample size on the performance. Each experiment is carried out ten times with different realisations of the measurement noise and for direct comparison of the filters on each experiment, the initial ensembles are drawn using the same seed.

The state estimations of the linear and non-linear filters for a single experiment with $N = 1024$ can be seen in Figure 2a and Figure 2b respectively. The parameter estimates for the same experiment can be seen in Figure 3b and 3a respectively.

Clearly, the linear filter has difficulties incorporating information from the measurements in the assembled parameters. This is in contrast with the non-linear filter, where the mean of the assembled parameters converge to the truth. This difference is also visible in the state estimates on the interval $t = [5, 10]$. The state forecasts on this interval become better as parameter estimates of the non-linear filter converge to the truth. This is not clearly visible for the linear filter.



(a)



(b)

Figure 2: State estimates of (a) the linear filter ($N = 1024$) and (b) the non-linear filter ($N = 1024$) for a single experiment.

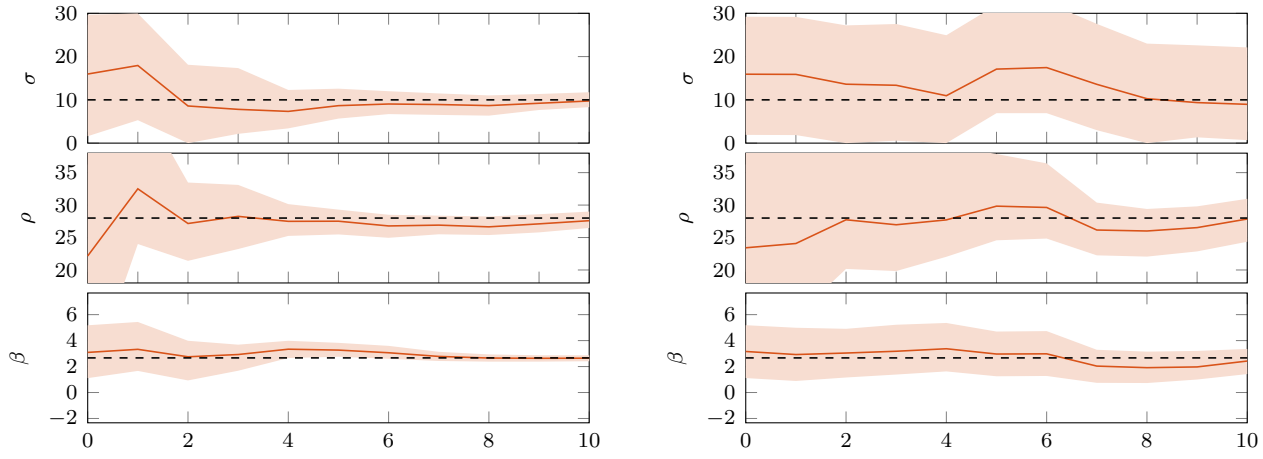


Figure 3: Parameter estimates of (a) the nonlinear filter ($N = 1024$) and (b) the linear filter ($N = 1024$) for experiment A. Truth (- - -); Estimate (——); 95% percentile on estimates (▭).

Table 1: Mean RMS error of the estimated states over 10 simulations

Filter	Linear		Nonlinear	
Ensemble size	256	1024	256	1024
x	2.97	2.66	2.14	**1.88**
y	4.06	3.70	2.85	**2.71**
z	4.50	4.23	3.32	**2.63**
all	3.84	3.53	2.77	**2.41**

Subsequently, the RMS errors of the estimated mean states are calculated over the interval $t = [5, 10]$. This allows for the algorithm to assimilate information from the measurements in the parameter estimates in the first half of the time interval. The average RMS errors of the filters over these 10 simulations are shown in Table 1. The nonlinear filters perform significantly better than their linear counterparts the best performance is obtained by the nonlinear filter with a sample size of $N = 1024$.

4 Conclusion

In this paper, a filter for concurrent state and parameter identification on highly non-linear dynamics is presented. This filter combines a general filter form with a feed-forward neural network to capture nonlinear relations between the states and parameters and the measurements.

The performance of the filter is evaluated on the Lorenz-63 benchmark problem for the task of state and parameter estimation and is compared with its optimal linear variant. It is shown that the nonlinear filter can assimilate more information from the measurements into the uncertain states and parameters than the linear variant.

Since the neural network used in this work is trained between assimilation steps, the nonlinear filter is computationally demanding. This in combination with Monte Carlo sampling for discritizing the random variables, makes the nonlinear filter not suitable for real-time applications. Therefore, in future work, improved training and sampling strategies will be used to speed up the training procedure.

References

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] A. H. Jazwinski, *Stochastic processes and filtering theory*. San Diego, CA: Academic Press, 1970.

[3] G. Evensen, "Advanced data assimilation for strongly nonlinear dynamics," *Monthly weather review*, vol. 125, no. 6, pp. 1342–1354, 1997.

[4] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[5] H. G. Matthies, E. Zander, B. V. Rosić, and A. Litvinenko, "Parameter estimation via conditional expectation: a Bayesian inversion," *Advanced modeling and simulation in engineering sciences*, vol. 3, no. 1, pp. 1–21, 2016.

[6] B. V. Rosić, A. Litvinenko, O. Pajonk, and H. G. Matthies, "Sampling-free linear Bayesian update of polynomial chaos representations," *Journal of Computational Physics*, vol. 231, no. 17, pp. 5761–5787, 2012.

[7] B. V. Rosić, A. Kučerová, J. Sỳkora, O. Pajonk, A. Litvinenko, and H. G. Matthies, "Parameter identification in a probabilistic setting," *Engineering Structures*, vol. 50, pp. 179–196, 2013.

[8] C. D. Rodgers, *Inverse methods for atmospheric sounding: theory and practice*. World scientific, 2000, vol. 2.

[9] G. Evensen, *Data assimilation: the ensemble Kalman filter*. Springer, 2009, vol. 2.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.

[11] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[12] R. N. Miller, M. Ghil, and F. Gauthiez, "Advanced data assimilation in strongly nonlinear dynamical systems," *Journal of Atmospheric Sciences*, vol. 51, no. 8, pp. 1037–1056, 1994.

[13] G. Evensen and P. J. Van Leeuwen, "An ensemble Kalman smoother for nonlinear dynamics," *Monthly Weather Review*, vol. 128, no. 6, pp. 1852–1867, 2000.

[14] J. Li and D. Xiu, "A generalized polynomial chaos based ensemble Kalman filter with high accuracy," *Journal of computational physics*, vol. 228, no. 15, pp. 5454–5469, 2009.

[15] G. Kivman, "Sequential parameter estimation for stochastic systems," *Nonlinear Processes in Geophysics*, vol. 10, no. 3, pp. 253–259, 2003.

[16] J. Annan and J. Hargreaves, "Efficient parameter estimation for a highly chaotic system," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 56, no. 5, pp. 520–526, 2004.

[17] J. T. Ambadan and Y. Tang, "Sigma-point Kalman filter data assimilation methods for strongly nonlinear systems," *Journal of the Atmospheric Sciences*, vol. 66, no. 2, pp. 261–285, 2009.