

Model Error Correction in Data Assimilation by Integrating Neural Networks

Jiangcheng Zhu, Shuang Hu, Rossella Arcucci, Chao Xu, Jihong Zhu, and Yi-ke Guo*

Abstract: In this paper, we suggest a new methodology which combines Neural Networks (NN) into Data Assimilation (DA). Focusing on the structural model uncertainty, we propose a framework for integration NN with the physical models by DA algorithms, to improve both the assimilation process and the forecasting results. The NNs are iteratively trained as observational data is updated. The main DA models used here are the Kalman filter and the variational approaches. The effectiveness of the proposed algorithm is validated by examples and by a sensitivity study.

Key words: data assimilation; deep learning; neural networks; Kalman filter; variational approach

1 Introduction and Motivation

Data Assimilation (DA) is an uncertainty quantification technique by which measurements (see Eq. (1), Algorithm 1) and model predictions (see Eq. (2), Algorithm 1) are combined to obtain an accurate representation of the state of the modeled system^[1]. DA was firstly proposed in atmospheric models and then widely applied in climatology, geophysics, aerodynamics, and economic models in the past decades. Recently, DA has also been applied in numerical simulations of geophysical applications^[2] and medical and biological sciences^[3,4] for improving

the reliability of the numerical simulations. Today, there are many DA algorithms. Those which have gained acceptance as powerful methods in the last ten years include the variational DA approach and the Kalman Filter. These approaches assume that the two sources of information, forecasts and observations, have errors that are adequately described by stationary error covariance matrices.

Algorithm 1: $\mathcal{A}(\text{DA})$

Input: A time step t_k . Assimilation window A . A numerical discretization of a dynamic system $F_k : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$, where $x_k \in \mathbb{R}^{n_x}$ be a state estimate vector and an initial condition x_0 .

M vectors of noisy measurements $y_j \in \mathbb{R}^{n_y}$ in an assimilation window time $[t_k, t_{k+A}]$ such that:

$$y_j = H_j(x_j) + v_j, \quad \forall k \leq j \leq k + A \quad (1)$$

where $H_j : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ is the first order linearization of the observation function.

1 Model Integration. Compute

$$x_k = F_{0 \rightarrow k}(x_0) + \xi_k \quad (2)$$

where $F_{0 \rightarrow k} = F_{k-1 \rightarrow k} \circ F_{k-2 \rightarrow k-1} \circ \dots \circ F_{0 \rightarrow 1}$

2 Define the covariance matrices R_k and B_k

3 Assimilate x_k with y_j (for $j = 1, \dots, A$), i.e., compute

$$x_k^{\text{DA}} = \underset{x}{\operatorname{argmin}} \left\{ \|x - x_k\|_{B_k^{-1}} + \sum_{j=k}^{k+A} \|H_j F_j(x) - y_j\|_{R_j^{-1}} \right\} \quad (3)$$

* Jiangcheng Zhu and Chao Xu are with State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China.

* Shuang Hu and Jihong Zhu are with Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

* Rossella Arcucci and Yi-ke Guo are with Data Science Institute, Imperial College London, London SW7 2AZ, UK. E-mail: y.guo@imperial.ac.uk.

† Jiangcheng Zhu and Shuang Hu contribute equally to this paper. This work was done when they were visiting researchers in Data Science Institute, Imperial College London, London SW7 2AZ, UK.

* To whom correspondence should be addressed.

Manuscript received: 2018-07-05; accepted: 2018-09-19

The solution gained by minimizing the cost function in Eq. (3) is the result of the Kalman Filter^[5]. If the solution is instead using variational Δx , it is called the variational approach. In recent years, variational approaches and Kalman Filters become more sophisticated to better fit their application requirements and circumvent their implementation issues. Nevertheless, these approaches are incapable of overcoming fully their unrealistic assumptions, particularly linearity, normality, and zero error covariances. The first step in error analysis (also called sensitivity analysis) is to understand the errors that arise at the different stages of the solution process, namely, the uncertainty in the mathematical model, in the model's solution, and in the measurements. These are the errors intrinsic to the DA inverse problem. Moreover, there are the approximation errors introduced by linearization, discretization, and model reduction. These errors occur when infinite-dimensional equations are replaced by a finite-dimensional system, or when simpler approximations to the equations are developed. Finally, there are rounding errors introduced by working in a finite precision arithmetic during implementation of the algorithm.

Machine learning algorithms are capable of assisting or replacing the aforementioned traditional methods in assimilating data and making forecasts, without the assumptions of the conventional methods. As NNs can approximate any linear or nonlinear functions, it has been integrated with DA as a supplement in various applications.

In this paper, we consider integrating the NNs into the conventional DA (hereafter named as "DA+NN"). In recent years, deep learning shows great advantage in function approximations which have unknown model and strong nonlinearity. Here, we use NNs to characterize the structural model uncertainty. The NN is implemented in an End-to-End (E2E) approach and its parameters are iteratively updated with coming observations by applying the DA methods.

This paper is structured as follows. In Section 2, we discuss the related works about machine learning and DA. We proposed a new methodology in Section 3 where we integrate NNs into DA. In Section 4, we give a brief discussion on our method. In Section 5, numerical examples are given to implement and verify the proposed algorithm. Based on these discussions, we give an outlook on this field. Finally, conclusions are drawn and future works are discussed.

2 Related Work and Contribution of the Present Study

Sensitivity of the four-dimensional variational (4D-Var) DA model has been studied in Ref. [6] where an adjoint modeling is used to obtain first- and second-order derivative information and a reduced-order approach is formulated to alleviate the computational cost associated with the sensitivity estimation. This method makes rerunning less expensive; however, the parameters must still be selected a priori, and, consequently, important sensitivities may be missed^[7]. It is proved that Algorithm 1 obtains $\tilde{x}_k^{\text{DA}} = x_k^{\text{DA}} + \delta_k$ where δ_k is

$$\|\delta_k\|_\infty = C_{\mathcal{A}(\text{DA})} \|\xi_k\|_\infty \quad (4)$$

where $C_{\mathcal{A}(\text{DA})}$ is a parameter that depends on the condition numbers of the DA numerical model and the numerical forecasting model implemented in Algorithm 1^[8] and ξ_k is the error in the numerical forecasting model in Eq. (2). Approaches to reduce the error propagation are previously proposed^[6,8] and studies based on adjoint analysis are provided^[7]. In this paper, we study the problem of reducing ξ_k by estimating the errors using machine learning.

Machine learning algorithms are capable of assisting or replacing the aforementioned traditional methods in assimilating data and making forecasts, without the assumptions of the conventional methods. As NNs can approximate any linear or nonlinear functions, they have been integrated with DA as a supplement in various applications. Babovic et al.^[9–11] applied neural network for error correction in forecasting. In these literatures, a neural network is trained by the error of past several steps. This network compensates the forecasting result from a deterministic model. Similar approaches are implemented on hydraulic applications^[12–15] and financial applications^[16]. While machine learning capability in approximating any nonlinear or complex system is promising, it is a black-box approach, which lacks the physical meanings of the actual system structure and its parameters, as well as their impacts on the system. To resolve this problem, we introduce in Section 3 a neural network model trained on DA results. This introduces a novelty in sensitivity analysis techniques used to determine which inputs in NN are significant. In Ref. [17], a series of seven different sensitivity analysis methods were reviewed. As we show in the next section, our approach can be related to the weights method^[18]. The difference

between our method and the classical weights method is that, in our approach, the connection weight among the nodes is provided by the error covariance matrices computed in the DA process.

3 Data Assimilation with Neural Networks

The overall process of DDA is described as Fig. 1 and the algorithm description of DDA is as Algorithm 2. As depicted in Eq. (2), the error ξ_k causes the error in DA. To compensate such error from model uncertainty, we introduce an NN \mathcal{G}_ω (as depicted in Fig. 2), and alternate the model F by a composition $\mathcal{G}_\omega \circ F$. The training target is

$$x_k = \mathcal{G}_\omega(F(x_{k-1})) \quad (5)$$

While NN capability in approximating complex systems is promising, it is often a black-box approach, which lacks the physical meanings of the actual system structure and its parameters, as well as their impacts on the system. To face this problem, in Eq. (5), we use the values of $F(x_{k-1}^{\text{DA}})$ and x_k^{DA} as input and output of network \mathcal{G}_ω , respectively. However, the real value of state x is unobtainable. To train the \mathcal{G}_ω , we exploit the result of DA $\{(x_{k-1}^{\text{DA}}, x_k^{\text{DA}})\}$.

The \mathcal{G}_ω model can be the iterative form $\mathcal{G}_{\omega,i} \circ \mathcal{G}_{\omega,i-1} \circ \dots \circ \mathcal{G}_{\omega,1} \circ F$ where $\mathcal{G}_{\omega,i}$ is defined as the network trained in the i -th stage. Let \mathcal{M}_i be the model implemented in the i -th stage:

$$\mathcal{M}_i = \mathcal{G}_{\omega,i} \circ \mathcal{G}_{\omega,i-1} \circ \dots \circ \mathcal{G}_{\omega,1} \circ F \quad (6)$$

from which $\mathcal{M}_i = \mathcal{G}_{\omega,i} \circ \mathcal{M}_{i-1}$ and $\mathcal{M}_0 = F$.

We define the result of DA and DA+NN as x^{DA} and $x^{\text{DA+NN}}$, respectively. For iteration $i > 1$, the training set for $\mathcal{G}_{\omega,i}$ is $\{(\mathcal{M}_i(x_{k-1}^{\text{DA+NN}}), x_k^{\text{DA+NN}})\}$, which exploits the $\{(x_{k-1}^{\text{DA+NN}}, x_k^{\text{DA+NN}})\}$ from the last updated model \mathcal{M}_i . As for loss function in Eq. (10), it is defined as in Theorem 1 described in Section 4 and the network is updated by the gradient loss function on its parameter:

$$\omega_{i+1} \leftarrow \omega_{i+1} + \alpha \frac{\partial \mathcal{L}_i}{\partial \omega} \quad (7)$$

The Fully-Connected (FC) network is the earliest and most basic NN structure. It means that the output of each neuron in the upper layer becomes the input of each neuron in the next layer, and it has a structure with corresponding weight values. There is no cycle or loop inside the FC network. FC NNs are widely cited in the fitting of functions, especially the output of continuous value functions.

The two-layer FC network contains a hidden layer and an output layer. If w_1 is the weight of the hidden layer, b_1 is the bias of the hidden layer. w_2 is the weight of the output layer and b_2 is the bias of the output layer. The hidden layer has one activation function (tanh function or relu function) per neuron, and the output layer does not have an activation function. The input

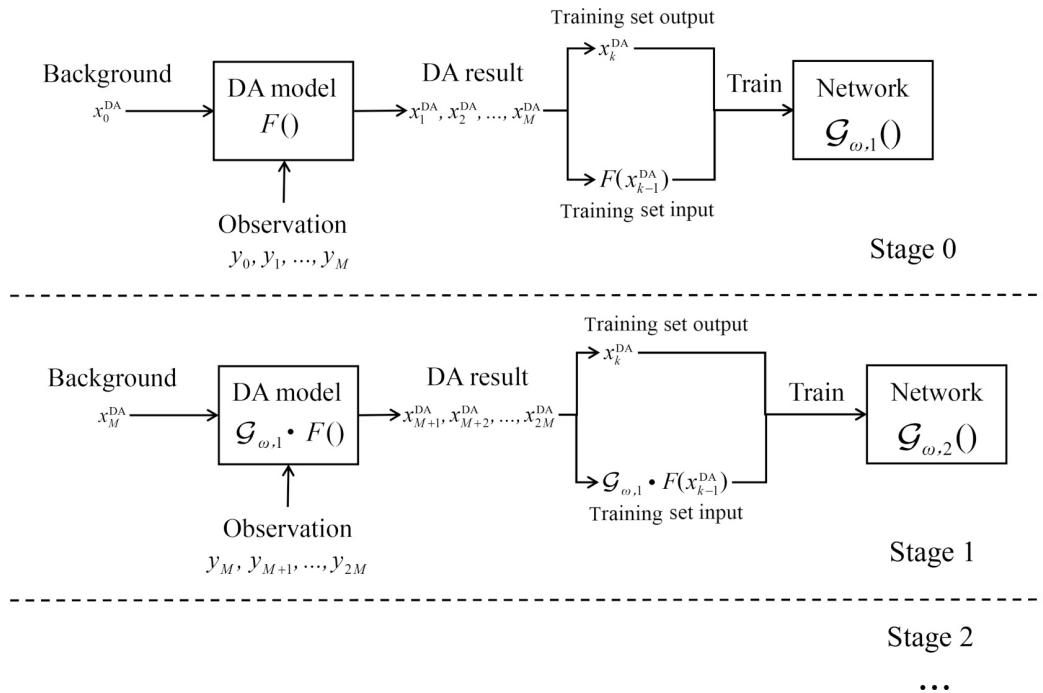


Fig. 1 Schematic diagram of DA+NN.

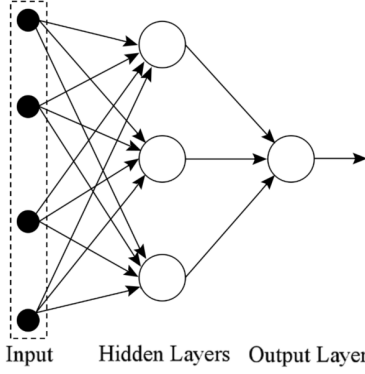


Fig. 2 Fully-connected neural network.

and output relations of this neural network are

$$y = w_2(\tanh(w_1x + b_1)) + b_2 \quad (8)$$

In training, in the framework of data assimilation with neural network as previously described, the training set is the DA result over a period of time. Taking the first cycle ($i = 0$) as an example, the training set is $x_0^{\text{DA}}, x_1^{\text{DA}}, x_2^{\text{DA}}, \dots, x_M^{\text{DA}}$ (consecutive time series results), and the model is generated from this set of data. The next estimate is $\{\hat{x}_k^{\text{DA}} = \mathcal{M}(x_{k-1}^{\text{DA}}) | k = 1, 2, \dots, M\}$. The input of NN is \hat{x}_k^{DA} and the output tag value is x_k^{DA} (DA result). This resulted in a total of M training sets. Since this network is a feedforward neural network and cannot handle time series, we consider that the M group training sets are independent of each other.

In the execution, taking the stage $i = 1$ as an example, we have already trained the feedforward NN $\mathcal{G}_{\omega,1}$ as described in the previous paragraph. Therefore, during the DA of period $i = 1$, we will modify the model to the modified neural network model $\mathcal{M}_1 = \mathcal{G}_{\omega,1} \cdot \mathcal{M}_0$. Then we need to use the model \mathcal{M}_1 to update the system when each step DA needs a model update.

4 Mathematical Analysis

We can then prove the improvement in DA results introduced by our proposed algorithm by Theorem 1 and Theorem 2.

Theorem 1 Let $\{(\mathcal{M}_i(x_{k-1}^{\text{DA}}), x_k^{\text{DA}})\}$ be the training set for $\mathcal{G}_{\omega,i}$, which exploits the $\{(x_{k-1}^{\text{DA}}, x_k^{\text{DA}})\}$ from the last updated model \mathcal{M}_i . The loss function by the back-propagation, defined as the Mean Square Error (MSE) for the DA training set, is such that

$$\mathcal{L}_k(\omega) = \|B_k D_k^T A_k^{-1} d_k - \mathcal{G}_{\omega,k}(B_{k-1} D_{k-1}^T A_{k-1}^{-1} d_{k-1})\|_2 \quad (9)$$

where $d_k = y_k - H_k x_k$ is the so-called misfit between observed and background data, $D_k = \text{diag}\{\mathcal{M}_k H_k,$

$\dots, \mathcal{M}_{k+A} H_{k+A}\}$ and $A_k = D_k B_k D_k^T + R_k$ is the Hessian of the DA system in Eq. (3).

Proof: The loss function by back-propagation, defined by the MSE, is as

$$\mathcal{L}_k = \|x_k^{\text{DA}} - \mathcal{G}_{\omega,i} \circ \mathcal{M}_i(x_{k-1}^{\text{DA}})\|_2 \quad (10)$$

Let be $D_k = \text{diag}\{\mathcal{M}_k H_k, \dots, \mathcal{M}_{k+M} H_{k+M}\}$, the solution of the DA system in Eq. (3) is^[11]

$$x_k^{\text{DA}} = B_k D_k^T (D_k B_k D_k^T + R_k)^{-1} (y_k - H_k x_k) \quad (11)$$

Let $d_k = y_k - H_k x_k$ and $A_k = D_k B_k D_k^T + R_k$, Eqs. (10) and (11) give Eq. (9).

We estimate the error ξ_k in Eq. (2) by the deep NN. Even if some error's sources cannot be ignored (for example, the round-off error), the introduction of the proposed method which exploits the benefit from both NN and DA methods allows us to reduce ξ_k . Let \hat{x}_k be the solution of the dynamic system (Eq. (2)) modeled by the NN, it is $\hat{x}_k = \mathcal{M}_k(x_0) + \hat{\xi}_k$ where $\hat{\xi}_k$ denotes the error in the NN model such that

$$\|\hat{\xi}_k\|_\infty \leq \|\xi\|_\infty \quad (12)$$

This introduces the possibility to reduce the error in the dynamic system by deep NN impact on the solution of the DA. The following result held:

Theorem 2 Let δ_k be the error in the DA solution as defined in Eq. (4), and let $\hat{\delta}_k$ be the error in the solution of proposed methodology:

$$\hat{x}_k^{\text{DA+NN}} = x_k^{\text{DA+NN}} + \hat{\delta}_k,$$

it is

$$\|\hat{\delta}_k\|_\infty \leq \|\delta_k\|_\infty \quad (13)$$

Proof: Let $\mathcal{C}_{A(\text{DA})}$ and $\mathcal{C}_{A(\text{DA+NN})}$ be the error propagation parameters which depend on the numerical methods implemented in DA and DA+NN algorithms, respectively, as defined in Eq. (4). It can be proved that

$$\mathcal{C}_{A(\text{DA})} \simeq \mathcal{C}_{A(\text{DA+NN})} \quad (14)$$

In fact $\mathcal{C}_{A(\text{DA})} \simeq \mu(A_{\text{DA}})$, where A_{DA} is the Hessian of the DA system (Eq. (3)) and $\mathcal{C}_{A(\text{DA+NN})} \simeq \mu(A_{\text{DA+NN}})$ where $A_{\text{DA+NN}}$ is the Hessian of the DA system (Eq. (3)). As the condition number of both Hessian matrices A_{DA} and $A_{\text{DA+NN}}$ are proportional to the condition number of the background error covariance matrix B_k (as in Ref. [19]) then Eq. (14) holds. From Eqs. (12) and (14) it follows that

$$\|\hat{\delta}_k\|_\infty = \mathcal{C}_{A(\text{DA+NN})} \|\hat{\xi}_k\|_\infty \leq \mathcal{C}_{A(\text{DA})} \|\xi_k\|_\infty = \|\delta_k\|_\infty \quad (15)$$

5 Numerical Examples

In this section, we address two different applications of DA+NN in case of known or unknown system model

inaccuracy ξ . Our interest is the modeling uncertainty effect in real applications. Uncertainty widely exists in the modeling of natural processes such as turbulence in fluid dynamics or aircraft design. Also, mis-modelling of chemical or biological process has not been clearly investigated. These processes can be interpreted as a general dynamic system. In these situations, DA is not sufficient in reducing the modelling uncertainty in approaching the true models. The DA+NN algorithm we have proposed is applicable in solving this problem.

5.1 Double-integral mass dot system

We first use a double-integral particle system, which is widely used for the physical representation of a simplified motion system. The system can be understood as the motion of a particle under the influence of a controlled acceleration. The affected state includes position and velocity. Due to the existence of a Proportion-Integral-Derivative (PID) controller in the feedback control system, the position as a controlled state eventually converges upon a relatively stable value. The output of the controller is the acceleration acting on the system, or it can be understood as a force that linearly relates to the acceleration.

A double-integral system is a mathematical abstraction of Newton's Second Law which is a simplified model of many controlled systems. It can be represented as a continuous model as

$$\begin{aligned}\dot{x} &= Ax + Bu + \xi, \\ y &= Cx + v\end{aligned}\quad (16)$$

where the state $x = [x_1, x_2]^T$ is a two-dimensional vector containing position x_1 and velocity x_2 information, and $u = \dot{x}_2$ is the controlled input. The coefficient matrices A, B , and C are time-invariant system and observation matrices. v is the observation noise, which is a two-dimensional Gaussian. w is the system disturbance including two aspects: random system noise and structural model uncertainty.

To implement DA on this system, model (16) is discretized to a discrete form F, G , and H as Eqs. (2) and (1). We introduce a disturbance in system Eq. (16) so that

$$\xi_k = \xi_{\text{smu}}(x_k) + \xi_{\text{iid},k} \quad (17)$$

in which $\xi_{\text{smu}}(\cdot)$ is the structural model uncertainty and ξ_{iid} is the i.i.d random disturbance.

We test two kinds of model uncertainty $\xi_{\text{smu}}(\cdot)$. They represents two typical structural error compositions: parallel and serial. Please note error $\xi_{\text{smu}}(\cdot)$ is added

in the real system, which is continuous.

The error from parallel composition is

$$\xi_{\text{smu}}(x_k) = M \frac{e^{x_k+1}}{1 + e^{x_k+1}} \quad (18)$$

where $M = [m_1, m_2]$ is the amplitude vector which can be adjusted according to the reality.

The error from serial composition is

$$\xi_{\text{smu}}(x_k) = (M - I)Ax_k \quad (19)$$

which describes the condition that the real system is $\dot{x} = M(A(x) + B(u))$, while the model in DA is $\dot{x} = A(x) + B(u)$.

To achieve the model simulation, we design a cascade controller to avoid the system from divergence. The tracking signal is set as a step function. We obtain 1000 samples from a 10-second simulation with the sample frequency of 100 Hz. This simulation is implemented on the Simulink of Matlab 2016a.

First, we run a Kalman filter based on a discretized model F, G , and H . The algorithm of DA+NN is as Algorithm 2. The training set is built by the time series $\{F(x_{k-1}^{\text{DA}}), x_k^{\text{DA}}\}$.

For the dynamic system Eq. (16), we first run the system and record the observation y , control input u . A corresponding Kalman filter runs along the system updates, and outputs a sequence of prediction x^{DA} . In the first stage (the left area divided by vertical dashline) of Fig. 3, there is an error between the black line and blue line. The network $\mathcal{G}_{\omega,1}$ is trained by the dataset of

Algorithm 2: $\mathcal{A}(\text{DA} + \text{NN})$

Input: Observation y_j for $n \cdot M$ temporal steps, $F, H, \mathcal{G}_{\omega}$, assimilation window A

- 1 Initialize the $\mathcal{M}_0 = F$ by the DA modelling with the training set $\mathcal{D}_i = \{F(x_{k-1}^{\text{DA}}), x_k^{\text{DA}}\}$
- 2 Initialize stage $i = 0$
- 3 **while** $i < n$ **do**
- 4 Run the DA for the first M steps with model \mathcal{M}_i and record estimation $\{x_k^{\text{DA+NN}} | k \in [M \cdot i, M \cdot (i + 1)]\}$

$$x_k^{\text{DA+NN}} = \underset{x}{\operatorname{argmin}} \left\{ \|x - x_k\|_{B_k^{-1}} + \sum_k^{k+A} \|H_j \mathcal{M}_{j,i}(x) - y_j\|_{R_j^{-1}} \right\} \quad (20)$$
- 5 Generate the training set $\mathcal{D}_i = \{F(x_{k-1}^{\text{DA+NN}}), x_k^{\text{DA+NN}}\}$ with $k \in [i \cdot M, (i + 1) \cdot M]$
- 6 Train a neural networks $\mathcal{G}_{\omega,i+1}$ with \mathcal{D}_i
- 7 Put trained $\mathcal{G}_{\omega,k}$ in the model:
 $\mathcal{M}_{i+1} \leftarrow \mathcal{G}_{\omega,i+1} \circ \mathcal{M}_i$
- 8 Count up i for the next stage
- 9 **end**

Output: $x_k^{\text{DA+NN}}$

these K steps.

After that, the system and Kalman filter continue to update from the $(K + 1)$ -th step to the $2K$ -th step. Now the system-update step in Kalman filter integrates the trained NN $\mathcal{G}_{\omega,1}$ as

$$x_k = \mathcal{G}_{\omega,1}(F x_{k-1} + G u_k) \quad (21)$$

Figure 3a is the result of these numerical examples. Two vertical dashed lines are at point which the NN is trained. We can see that after the first training, the Kalman filter with NN outperforms the standard one. After the second training, its performance is even further improved.

Figure 3b shows the result and comparison of this algorithm. By training twice, the result of DA+NN got close to the real value. However, biased estimation still exists and overfitting is also likely to occur in this case.

5.2 Lorenz system

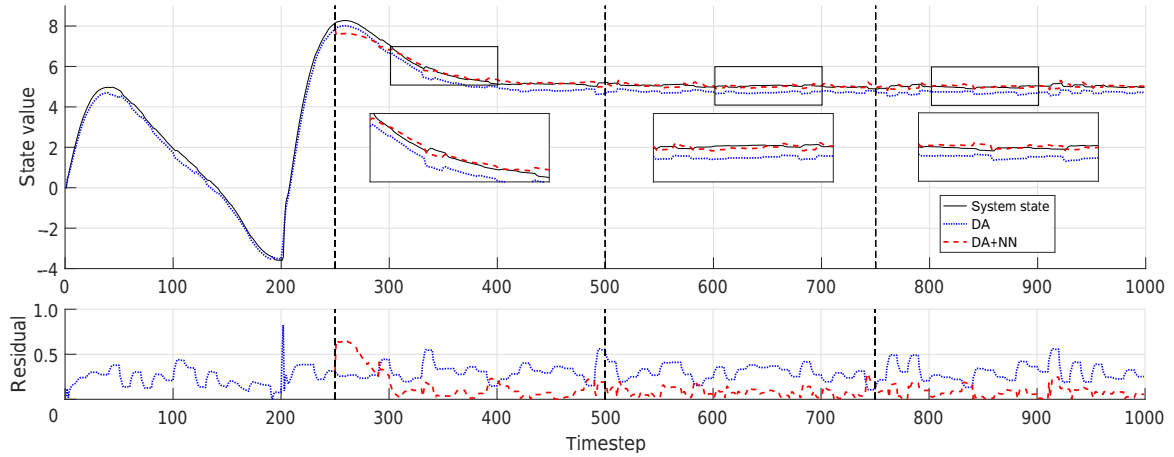
To describe the atmosphere behavior in a simple mathematical model, Lorenz^[20] developed a simplified

mathematical model for atmospheric convection. It is a popular test case for DA algorithms. The model is a system of three ordinary differential equations (the Lorenz equations). It is notable for exhibiting a chaotic behaviour for certain parameter values and initial conditions. The Lorenz equations are given by the nonlinear system:

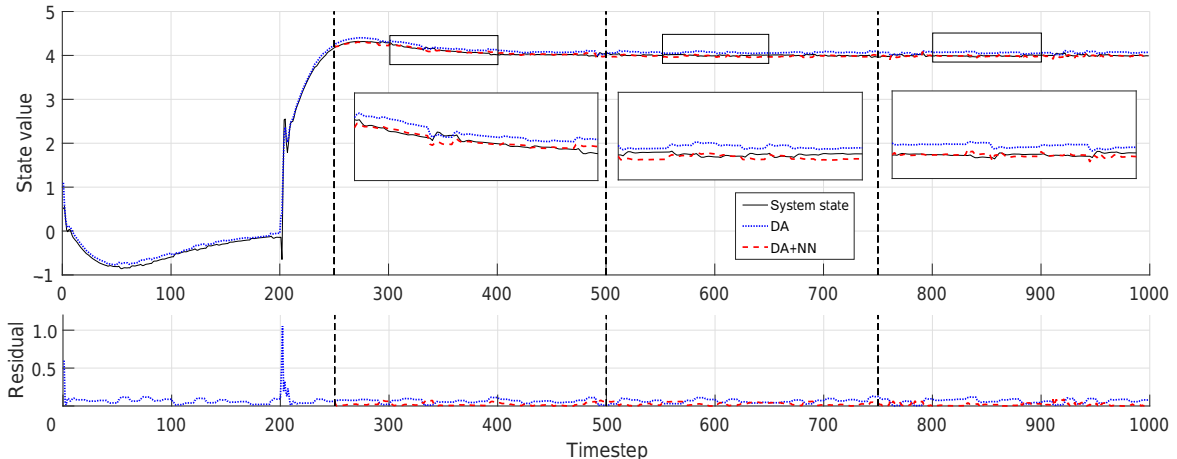
$$\begin{aligned} \frac{dp}{dt} &= -\sigma(p - q), \\ \frac{dq}{dt} &= \rho p - q - pr, \\ \frac{dr}{dt} &= pq - \beta r \end{aligned} \quad (22)$$

where p , q , and r are coordinates, and σ , ρ , and β are parameters (in this test case, $\sigma = 10$, $\rho = 8/3$, $\beta = 28$). This Lorenz system can be discretized by the second order Runge-Kutta method as in Ref. [21].

Given a state $x_k = [p_k, q_k, r_k]^T$ and the discrete function $F_k(\cdot)$, we define a Lorenz system with structural model uncertainty as



(a) Parallel composition error



(b) Serial composition error

Fig. 3 Simulation result of DA+NN on double-integral system. The vertical dash-lines refer to the training windows.

$$x_k = F_k(x_{k-1}) + \xi_{\text{snu},k} \quad (23)$$

in which the structural model uncertainty is $\xi_{\text{snu},k} = Mx_{k-1}$ (in this case $M = 0.01$), and $F_k(\cdot)$ is

$$\begin{aligned} p_{k+1} &= p_k + \sigma \frac{\Delta t}{2} [2(q_k - p_k) + \\ &\quad \Delta t(\rho p_k - q_k - p_k r_k) - \sigma \Delta t(q_k - p_k)], \\ q_{k+1} &= q_k + \frac{\Delta t}{2} [\rho p_k - q_k - p_k r_k + \rho(p_k + \\ &\quad \sigma \Delta t(q_k - p_k)) - q_k - \Delta t(\rho p_k - q_k - p_k r_k) - \\ &\quad (p_k + \sigma \Delta t(q_k - p_k))(r_k + \Delta t(p_k q_k - \beta r_k))], \\ r_{k+1} &= r_k + \frac{\Delta t}{2} [p_k q_k - \beta r_k + (p_k + \Delta t \sigma(q_k - p_k)) \cdot \\ &\quad (q_k + \Delta t(\rho p_k - q_k - p_k r_k)) - \\ &\quad \beta r_k - \Delta t(p_k q_k - \beta r_k)]. \end{aligned}$$

For the dynamic system as Eq. (22), we firstly run the system and record the system true value x . Then, the observation y is generated by adding Gaussian noise on the true value x . After that, a 4D-Var algorithm is applied to generate a sequence of DA result x^{DA} with length K . The training set is built by the time series $\{F_k(x_{k-1}^{\text{DA}}, x_k^{\text{DA}})\}$. Then the NN \mathcal{G}_ω is trained.

The forecasting results based on the DA model and DA with the NN model are then compared. For DA, the system update model is formed as

$$x_k = F_k(x_{k-1}) \quad (24)$$

For DA+NN, the system update model is

$$x_k = \mathcal{G}_\omega(F_k(x_{k-1})) \quad (25)$$

We compare the forecasting results in Fig. 4. The training set generated by $k \in [0, 1000]$ is not plotted in complete. The forecasting begins at $k = 1000$. We can see the forecasting errors of DA model in all three axes are big. The trajectories of DA model cannot track the real state, while the trajectories of DA+NN model track the real value very well.

Figure 4b is a 3D plot of this Lorenz system forecasting part. It shows the DA model fails to forecast the right hand part of this butterfly. In this test, the DA+NN model outperforms the DA model in forecasting, for the trajectory of the right wing of this butterfly.

6 Conclusion and Future Work

In this article, we discuss a new methodology that integrates NN into data assimilation and validates the algorithm by numerical examples of model uncertainty.

So far, the effectiveness of this algorithm still remains to be verified in various specific domains having huge state space and more complex internal and external mechanisms.

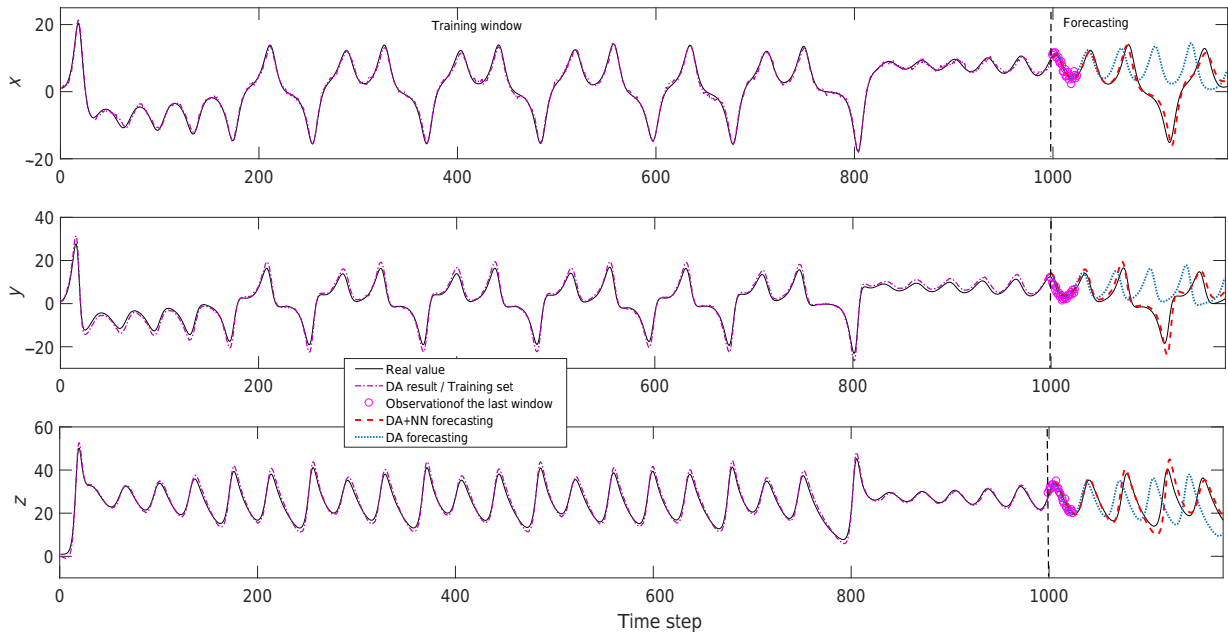
Future work should include the implementation of a DA+NN under more general conditions. For example, we will investigate the combinatorial case of parametric model inaccuracy and structural model uncertainty. Also, modern deep learning tools (e.g., CNN, RNN) can be introduced into data assimilation to improve the adaption and performance in more divergent conditions.

Acknowledgment

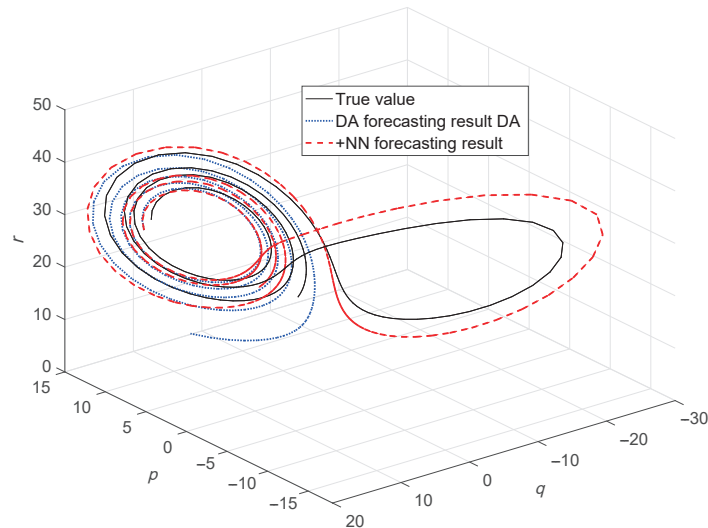
This work was supported by the EPSRC Grand Challenge grant ‘‘Managing Air for Green Inner Cities’’ (MAGIC) EP/N010221/1.

References

- [1] E. Kalnay, *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge, MA, USA: Cambridge University Press, 2003.
- [2] J. Blum, F. X. De Dimet, and I. M. Navon, Data assimilation for geophysical fluids, in *Hand-book of Numerical Analysis*. Elsevier, 2005.
- [3] M. D’Elia, M. Perego, and A. Veneziani, A variational data assimilation procedure for the incompressible navier-stokes equations in hemodynamics, *J. Sci. Comput.*, vol. 52, no. 2, pp. 340–359, 2012.
- [4] R. Potthast and P. B. Graben, Inverse problems in neural field theory, *SIAM J. Appl. Dyn. Syst.*, vol. 8, no. 4, pp. 1405–1433, 2009.
- [5] R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [6] D. Daescu and I. Navon, Sensitivity analysis in nonlinear variational data assimilation: Theoretical aspects and applications, in *Advanced Numerical Methods for Complex Environmental Models: Needs and Availability*, I. Farago, Z. Zlatev, eds. Sharjah, United Arab Emirates: Bentham Science Publishers, 2013.
- [7] D. Cacuci, I. Navon, and M. Ionescu-Bujor, *Computational Methods for Data Evaluation and Assimilation*. Boca Raton, FL, USA: CRC Press, 2013.
- [8] R. Arcucci, L. D’Amorea, J. Pistoia, R. Toumi, and A. Murli, On the variational data assimilation problem solving and sensitivity analysis, *Journal of Computational Physics*, vol. 335, pp. 311–326, 2017.
- [9] V. Babovic, M. Keijzer, and M. Bundzel, From global to local modelling: A case study in error correction of deterministic models, in *Proceedings of the Fourth International Conference on Hydro Informatics*, Iowa City, IA, USA, 2000.
- [10] V. Babovic, R. Cañizares, H. R. Jensen, and A. Klinting, Neural networks as routine for error updating of numerical models, *Journal of Hydraulic Engineering*, vol. 127, no. 3, pp. 181–193, 2001.
- [11] V. Babovic and D. R. Fuhrman, Data assimilation of local model error forecasts in a deterministic model, *International Journal for Numerical Methods in Fluids*, vol. 39, no. 10, pp. 887–918, 2002.
- [12] Y. Sun, V. Babovic, and E. S. Chan, Artificial neural networks as routine for error correction with an application



(a) Trajectories of the 4D-Var and DA with neural network on three axes.

(b) Trajectories of DA and DA with NN forecasting. It is the 3D-plot of Fig. 4a for timestep $\in (1000, 1176)$.**Fig. 4 Result of data assimilation with NN on Lorenz system.**

- in Singapore regional model, *Ocean Dynamics*, vol. 62, no. 5, pp. 661–669, 2012.
- [13] O. Makarynsky, Improving wave predictions with artificial neural networks, *Ocean Engineering*, vol. 31, nos. 5&6, pp. 709–724, 2004.
- [14] D. I. Gopinath and G. Dwarakish, Wave prediction using neural networks at new Mangalore port along west coast of india, *Aquatic Procedia*, vol. 4, pp. 143–150, 2015.
- [15] P. Jain and M. Deo, Artificial intelligence tools to forecast ocean waves in real time, *Open Ocean Engineering Journal*, vol. 1, pp. 13–20, 2008.
- [16] A.-S. Chen and M. T. Leung, Regression neural network for error correction in foreign exchange forecasting and trading, *Computers & Operations Research*, vol. 31, no. 7, pp. 1049–1068, 2004.
- [17] M. Gevrey, I. Dimopoulos, and S. Leka, Review and comparison of methods to study the contribution of variables in artificial neural network models, *Ecological Modelling*, vol. 160, no. 3, pp. 249–264, 2003.
- [18] Y. W. Foo, C. Goh, and Y. Li, Machine learning with sensitivity analysis to determine key factors contributing to energy consumption in cloud data centers, in *2016 International Conference on Cloud Computing Research and Innovations (ICCCRI)*, 2016, pp. 107–113.
- [19] N. Nichols, Mathematical concepts of data assimilation, in *Data Assimilation*, W. Lahoz, B. Khattatov, and R. Menard, eds. Springer, 2010.
- [20] E. N. Lorenz, Deterministic nonperiodic flow, in *the Theory of Chaotic Attractors*, B. R. Hunt, T. Li, J. A. Kennedy, H. E. Nusse, eds. Springer, 2004.
- [21] A. S. Lawless, Data assimilation with the Lorenz equations, University of Reading, UK, 2002.



Jiangcheng Zhu received the BEng degree from Zhejiang University (ZJU), China in 2012, and the MS degree in intelligent system, robotics and control from University of California, San Diego (UCSD), USA in 2014. He is currently a PhD candidate in ZJU. He was a visiting researcher in the Data Science Institute

(DSI) of Imperial College London (ICL), UK during October 2017 to July 2018. He led ZJU Micro Aerial Robotics Team (ZMART) won the 7th champion of the International Aerial Robotics Competition (IARC) from 2014 to 2018. His current research interests include reinforcement learning and deep learning in robotics.



Shuang Hu received the BE degree from Beijing University of Posts and Telecommunications in 2013. He is currently a PhD student in the Department of Computer Science and Technology, Tsinghua University. He had been a visiting researcher of the Data Science Institute at Imperial College London,

UK, during October 2017 to April 2018. His current research interests include system identification, data assimilation, and aerodynamic modeling.



Chao Xu received the PhD degree of mechanical engineering from Lehigh University in 2010. Currently, he is the TRUTH Associate Professor and Associate Director in the Institute for Cyber-Systems & Control (CSC), Zhejiang University. Dr. Xu also serves as the Managing-Editor of *Journal of Industrial and Management*

Optimization (JIMO), the Associate Editor of the *Numerical Algebra, Control and Optimization (NACO)*, the Associate Editor of *Information & Control Technology* (a Chinese Journal). His main research interests include bio-inspired locomotion & robotics, optimal control, and PDE techniques in informatics.



Jihong Zhu received the PhD degree in automatic control from Nanjing University of Science and Technology in 1995. He is currently a professor in the Department of Computer Science and Technology, Tsinghua University, China. He won the title of Changjiang Distinguished Professor in 2014. Prof.

Zhu was awarded several prizes including one Second-Class of State Technological Invention Award, one First-Class of MOE Technological Invention Award, one First-Class and one Second-Class of Military Science and Technology Progress

Awards, one Grand Prize and four-Second Class of Provincial Science and Technology Progress Awards. His current research interests include unmanned aerial vehicle, flight control, and avionics.



Rossella Arcucci received the master degree (cum laude) in mathematics in 2008 from the University of Naples Federico II (UNINA), Italy, and the PhD in computational and computer science from the same university in 2012. She is a Research Associate of the Data Science Institute (DSI) at Imperial College

London (ICL) in UK. Her area of expertise is in numerical analysis, scientific computing and development of methods, and algorithms and software for scientific applications on high performance architectures including parallel and distributed computing. She works on numerical and parallel techniques for accurate and efficient data assimilation by exploiting the power of machine learning models. She also achieved computing efficiency by designing models specifically to massive parallel computers and graphics processing units. During her post doc at UNINA, she coordinated the H2020 project “iNnovative Approaches for Scalable Data Assimilation in oCeanography” (NASDAC) as PI until September 2017, when she joined the DSI. She received the acknowledgement of Marie Skłodowska-Curie fellow from European Commission Research Executive Agency in Brussels on the 27th of November 2017.



Yi-ke Guo received a first-class honours degree in computing science from Tsinghua University, China, in 1985 and received the PhD degree in computational logic from Imperial College in 1993 under the supervision of Professor John Darlington. He founded InforSense, a software company for life science and

health care data analysis, and served as CEO for several years before the company’s merging with IDBS, a global advanced R&D software provider, in 2009. He is founding director of DSI at ICL as well as leading the Discovery Science Group in the department. Professor Guo also holds the position of CTO of the tranSMART Foundation, a global open source community using and developing data sharing and analytics technology for translational medicine. His research area is in data analysis and e-science. The projects he has contributed have been internationally recognised, including winning the “Most Innovative Data Intensive Application Award” at the Supercomputing 2002 conference for Discovery Net, and the Bio-IT World “Best Practices Award” for U-BIOPRED in 2014. He is a Senior Member of the IEEE, a Fellow of the British Computer Society, and a Fellow of the Royal Academy of Engineering.