Water Level Simulation in River Network by Data Assimilation Using Ensemble Kalman Filter

Yifan Chen 1,2, Feifeng Cao 2,*, Xiangyong Meng 3 and Weiping Cheng 4

1 Zhejiang Provincial Key Laboratory of Hydraulic Disaster Prevention & Mitigation, Hangzhou 310020, China
2 College of Civil Engineering, Zhejiang University of Technology, Hangzhou 310023, China
3 Zhejiang Institute of Hydraulic & Estuary, Hangzhou 310020, China
4 College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China
* Correspondence: cff@zjut.edu.cn

Abstract: Water level simulation for complex water river networks is complex, and existing forecasting models are mainly used for single-channel rivers. In this paper, we present a new data assimilation model based on the ensemble Kalman filter (EnKF) for accurate water level simulation in complex river networks. The EnKF-based data model was tested on simulated water level data from a river network hydrodynamic model and optimized through parameter analysis. It was then applied to a real mountainous single-channel river and plain river network and compared with a data assimilation model based on the extended Kalman filter (EKF). The results showed that the EnKF-based model, with a medium ensemble sample size of 100–150, normal observation noise of 0.0001–0.01 m, and a high standard deviation of 0.01–0.1 m, outperformed the EKF-based model, with a 49% reduction in simulation errors, a 45% reduction in calculation cost, and a 43% reduction in filtering time. Furthermore, the EnKF-based data assimilation model predicted the water level in the plain river network better than the mountainous single-channel river. Around 5 to 8 h were required for data assimilation; afterwards, the model could make accurate predictions covering 20 to 30 h. The EnKF-based data assimilation model offers a potential solution for water level predictions in river networks.

Keywords: water level simulation; ensemble Kalman filter; extended Kalman filter; data assimilation; water river network; parameter analysis

1. Introduction

River networks play an important role in watershed landscapes due to their contributions to the watershed’s hydrological processes. Compared to single open-channel rivers, they are more complex and more sophisticated hydraulic simulation models are required in order to solve unsteady flow problems in river networks. In addition, due to human activity, the river network structure of basins has changed significantly, causing spatial and temporal differences in hydrological connectivity [1]. The complex nature of river networks produces challenges in flow modeling, particularly in flood forecasting for a river network. As a result, the theories and corresponding algorithms for simulating and forecasting hydraulic flow in river networks have not been well established. Although flood forecasting with machine learning or deep learning models has been attempted in relation to flood simulations [2–5], a large amount of data are required as inputs for the sufficient training of model parameters. The large number of parameters and the overfitting of parameters are also great concerns when attempting to realize accurate real-time predictions [6].
The existing one-dimensional Saint-Venant equations, which include a series of nonlinear partial differential equations [7], are generally used to solve the hydrodynamic process by means of the river network hierarchical differential method [8]. However, many factors, such as the uncertainty of the model parameters and the complex boundary conditions, limit the application of the one-dimensional Saint-Venant equations for river networks; the corresponding models fail to adjust modeling errors with the data measured in real time. The use of Kalman filter (KF) real-time correction technology to improve the reliability of water simulations and the forecast accuracy of hydrodynamic models has become an important research topic in the current approach towards flood forecasting [9–13]. To improve the accuracy of the model, many studies have combined KF with other models. For example, Madsen and Skotner [14] developed a river flood correction model by introducing the form of the spatial distribution of the gain information into the model's error correction process, which was used to improve the form of the dynamic noise interfering matrix of the original system. Ziliani et al. [15] combined the ensemble Kalman filter (EnKF) with a shallow water equation (SWE) to reconstruct and predict a dam break event that was experimentally reproduced. Their study indicated that filter results could be made more accurate via the estimation of parameters alongside the state. Furthermore, some studies have adopted different types of parameters to improve flood forecasting. Meng et al. [16] established a data assimilation scheme based on the EnKF and a smoother containing two parameters: soil moisture and streamflow, particularly considering the time delay. By exploiting the benefits of soil moisture and streamflow, the model showed great potential for short-term flood forecasting by merging observations from ground measurements and remote-sensing data retrieval.

In addition, EKF and EnKF have been used to estimate both states and parameters in stochastic differential mixed-effect models with observation noise. These algorithms are efficient, easy to implement, and require less computation time than similar methods, such as the SAEM algorithm [17]. Given that the forecasting accuracy of a hydraulic model for river networks is mainly influenced by hydraulic states, such as discharge and model parameters such as bed roughness, the combination of state and parameter assimilation has been used in hydrological data assimilation studies [12,17]. However, it is unlikely that the joint estimation of state and parameters will fully replace batch calibration.

Due to the advantages of EnKF and EKF, they are also commonly used in many other fields for numerical weather prediction, stochastic signal processing, and Bayesian state estimation. These fields often face high-dimensional problems and require scalable algorithms to process and make sense of the data. A comparison showed that EKF was more accurate and faster in optimizing the magnetic particle tracking (MPT) method in particle tracking [18]. Similarly, a modified EnKF was established to deal with nonlinearity in the multiple-target tracking (MTT) and multi-sensor fusion (MSF) scenarios, which was more accurate than traditional KF [19]. In stochastic signal processing (SP), high-dimensional problems are becoming more relevant in applications such as simultaneous localization and mapping (SLAM) and extended target tracking, where the state vector can contain many parameters. EnKF can be applied to these problems as it can handle nonlinear and non-Gaussian state-space models [20]. EnKF has also been applied in reservoir engineering, where the pressure, saturation, and temperature at different grid cells in the reservoir represent the state of the system. EnKF estimates the state of the system by integrating measurements from various sources such as production data, seismic data, and well-log data. EnKF has the benefit of allowing for the prediction of reservoir performance by running the simulator in a forward direction from the latest data assimilation time (analysis step) with each ensemble member [13].

One of the key aspects in river network research is flood forecasting. Flood simulation and forecasting in plain river network areas are greatly affected by the multi-level spatial river network structure, the complex water movement, the intervention of an upper dam or gate regulation, as well as the influence of uncertain factors such as boundary input conditions, riverbed variations, and model parameters. There are still more difficulties and
challenges in such areas compared to single open-channel rivers in terms of calculation efficiency, the data treatment (e.g., data assimilation) process and degree, and forecasting periods (a few hours to days ahead). To date, very few studies have been conducted to investigate the performance of KF or its variants on the water flow simulation of river networks and, where available, they have simplified the river networks with the use of limited river branches and nodes [21], or short-term flow forecasting was carried out [9]. For example, Munier et al. [9] developed a lumped hydrological model to propagate the discharge through the Seine river basin in France and KM was used to assimilate the discharge data to improve the simulation accuracy of flow forecasting within 72 h. However, the simulation results could be significantly influenced by the degree of river network spatialization and the spatial distribution.

In this study, an EnKF algorithm was developed for performing data assimilation on simulated water level data from a river network hydraulic model for real-time water level prediction. A new data assimilation scheme was proposed for applying EnKF to simulated water level data to improve the simulation capacity, followed by a parameter analysis conducted on a simulated river network in order to understand and optimize the key parameter values. Based on the optimized parameter data, an experimental evaluation was conducted on a real single-channel river and a real plain river network with an area of 325 km$^2$. In addition, the data assimilation results obtained with EnKF were compared to those obtained with the classic Kalman filter for hydraulic simulation, i.e., the extended Kalman filter (EKF), with regard to their simulation accuracy, calculation cost, and data assimilation efficiency. In this study, we aimed to provide a data assimilation scheme using EnKF to improve the real-time simulation capacity of a river network hydraulic model.

2. One-Dimensional Hydrodynamic Model of a River

Saint-Venant equations are often used to describe one-dimensional flow, as follows. The continuous equation:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_L$$

(1)

The momentum equation:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \frac{aQ^2}{A} \right) + gA \frac{\partial Z}{\partial x} + gA \frac{Q}{K} = q_L v_x$$

(2)

where $A$ is the cross-section area; $t$ is time; $Q$ is the average flow of the cross-section; $x$ is displacement along the water flow; $q_L$ is the lateral streamflow per unit length; $a$ is the momentum correction coefficient; $g$ is the acceleration of gravity; $Z$ is the water level of the section; $K$ is the flow modulus, $K = AR^{2/3}/n$, where $R$ is the hydraulic radius of the river and $n$ is the Manning coefficient; and $v_x$ is the velocity component of the lateral stream flow along the water flow, usually taken as $v_x = 0$.

Saint-Venant equations are hyperbolic partial differential equations for which analytical solutions are not yet available. The finite difference method is an effective numerical discretization method for solving the Saint-Venant equations and is divided into two schemes: the explicit difference scheme and the implicit difference scheme. Although the explicit difference scheme does not require the joint solution of equations, the time step and space step are limited by strict adherence to the “Courant condition” in order to maintain stability, whereas the implicit difference scheme is more flexible in the selection of the time and space step and is therefore popularly used. In this study, the Preissmann four-point implicit difference scheme were adopted to discretize the equations, and the space-time discretization schemes of the continuous and momentum equations were obtained as follows:

Difference form for the continuous equations:

$$\frac{\theta(\Delta B_{i+1} + \Delta B_i) + (B_i' + B_{i+1}')}{2} \cdot \frac{\Delta Z_{i+1} + \Delta Z_i}{2\Delta t} + \left( \frac{\theta \Delta Q_{i+1} - \Delta Q_i}{\Delta x} + \frac{Q_i' - Q_{i+1}'}{\Delta t} \right)$$

$$= \frac{\theta(\Delta B_{i+1} + \Delta B_i) + (B_i' + B_{i+1}')}{2} \cdot \frac{\Delta Z_{i+1} + \Delta Z_i}{2\Delta t} + \left( \frac{\theta \Delta Q_{i+1} - \Delta Q_i}{\Delta x} + \frac{Q_i' - Q_{i+1}'}{\Delta t} \right)$$

(3)
Difference form for the momentum equations:

\[
\frac{\Delta Q_i - \Delta Q_i}{2\Delta t} + \frac{g}{\Delta x} \left( \frac{(Q'_i + \Delta Q_i)^2}{A_i + \Delta A_i} - \frac{(Q'_i + \Delta Q_i)^2}{A_i} \right) + \frac{1}{\Delta x} \left( \frac{\Delta Z_i}{A_i} + \Delta Z_i \right) + \frac{1}{\Delta x} \left( \frac{\Delta Z_i}{A_i} + \Delta Z_i \right) - \frac{1}{\Delta x} \left( \frac{\Delta Z_i}{A_i} \right) + \frac{1}{\Delta x} \left( \frac{\Delta Z_i}{A_i} \right) + \frac{1}{\Delta x} \left( \frac{\Delta Z_i}{A_i} \right) + \frac{1}{\Delta x} \left( \frac{\Delta Z_i}{A_i} \right) \]

where \( \theta \) is the Prandtl number; supscripts \( t \) and \( t+1 \) and subscripts \( i \) and \( i+1 \) represent the time and one-dimensional spatial layers, respectively; and \( B \) is the width.

Thus, the following linear equations, expressed in increments, are obtained as follows:

\[
\begin{align*}
\Delta Q_{i+1} &= a_{1i} \Delta Q_i + b_{1i} \Delta Z_i + c_{1i} \Delta Q_{i+1} + d_{1i} \Delta Z_{i+1} = e_{1i} \\
\Delta Q_i &= a_{2i} \Delta Q_i + b_{2i} \Delta Z_i + c_{2i} \Delta Q_{i+1} + d_{2i} \Delta Z_{i+1} = e_{2i}
\end{align*}
\]

where

\[
\begin{align*}
a_{1i} &= -\frac{4 \theta \Delta t}{\Delta x_i} \beta_i^2 + \frac{1}{B_{i+1}^t}; b_{1i} = 1; c_{1i} = \frac{4 \theta \Delta t}{\Delta x_i} \beta_i + \frac{1}{B_{i+1}^t}; d_{1i} = 1; \\
e_{1i} &= 2 \Delta t \left( q_{1i} + q_{1,i+1} \right) - 4 \Delta t \frac{Q_{i+1} - Q_i}{\Delta x_i} B_{i+1}^t + 2 \theta \Delta t \frac{A_{i+1}^t |Q_i|^2}{(K_i^t)^2} \\
b_{2i} &= \frac{\theta \Delta t}{\Delta x_i} \left( \frac{(Q_i)^2 B_i^t}{(A_i)^2} - g(A_i' + A_{i+1}') + g(Z_i' + Z_{i+1}') B_i^t \right) + g \theta \Delta t \frac{Q_{i+1}' |Q_i|^2}{(K_i^t)^2} (B_i^t - 2 A_{i+1}^t dK_i^t); \\
c_{2i} &= 1 + 4 \frac{\theta \Delta t}{\Delta x_i} A_{i+1}^t + 2 \theta \Delta t \frac{A_{i+1}^t |Q_{i+1}^t|^2}{(K_{i+1}^t)^2} \\
d_{2i} &= \frac{\theta \Delta t}{\Delta x_i} \left[ -2 \left( \frac{(Q_{i+1})^2}{A_{i+1}^t} \right)^2 + g(A_i' + A_{i+1}') + g(Z_i' + Z_{i+1}') B_{i+1}^t \right] + g \theta \Delta t \frac{Q_{i+1}' |Q_i|^2}{(K_i^t)^2} (B_{i+1}^t - 2 A_{i+1}^t dK_{i+1}^t); \\
e_{2i} &= \frac{\Delta t}{\Delta x_i} \left[ -2 \left( \frac{(Q_{i+1})^2}{A_{i+1}^t} \right)^2 + 2 \frac{(Q_i)^2}{A_i^t} - g(A_i' + A_{i+1}')(Z_i' + Z_{i+1}') \right] - \frac{\Delta t}{\Delta x_i} \left[ A_{i+1}^t |Q_{i+1}^t| |Q_i|^2 + A_{i+1}^t |Q_i|^2 |Q_{i+1}^t| \right]
\end{align*}
\]

Equations (4) and (5) are the expressions for the known quantities in time layer \( t \). After the elimination of the elements, we can obtain

\[
\begin{align*}
\Delta Z_{i+1} &= a_{1i} \Delta Z_i + b_{1i} \Delta Q_i + c_{1i} \\
\Delta Q_{i+1} &= a_{2i} \Delta Z_i + b_{2i} \Delta Q_i + c_{2i}
\end{align*}
\]

where

\[
\begin{align*}
a_{1i} &= c_2 b_{1i} - c_1 b_{2i}; b_{1i} = c_2 a_{1i} - c_1 a_{2i}; c_{1i} = c_1 b_{2i} - c_2 b_{1i}; c_{2i} = c_1 c_2 b_{1i} - c_2 c_1 b_{2i}; \\
a_{2i} &= d_2 b_{1i} - d_1 b_{2i}; b_{2i} = d_2 a_{1i} - d_1 a_{2i}; c_{2i} = d_1 c_2 b_{1i} - d_2 c_1 b_{2i}
\end{align*}
\]

The corresponding section recurrence equations are obtained by sequentially working backwards from the first section of the river to any section:

\[
\begin{align*}
\Delta Z_{i+1} &= EMT(2i + 1, 1) \Delta Z_i + EMT(2i + 1, 2) \Delta Q_i + EMT(2i + 1, 3) \\
\Delta Q_{i+1} &= EMT(2i + 2, 1) \Delta Z_i + EMT(2i + 2, 2) \Delta Q_i + EMT(2i + 2, 3)
\end{align*}
\]
where, when $i = 0$,

$$
\begin{align*}
\{ & EMT(1, 1) = 1, \quad EMT(1, 2) = 0, \quad EMT(1, 3) = 0 \\
& EMT(2, 1) = 0, \quad EMT(2, 2) = 1, \quad EMT(2, 3) = 0
\end{align*}
$$

and when $i > 0$,

$$
\begin{align*}
\{ & EMT(2i + 1, 1) = a'_{1i}EMT(2i - 1, 1) + b'_{1i}EMT(2i, 1) \\
& EMT(2i + 1, 2) = a'_{1i}EMT(2i - 1, 2) + b'_{1i}EMT(2i, 2) \\
& EMT(2i + 1, 3) = a'_{1i}EMT(2i - 1, 3) + b'_{1i}EMT(2i, 3) + c'_{1i} \\
& EMT(2i + 2, 1) = a'_{2i}EMT(2i - 1, 1) + b'_{2i}EMT(2i, 1) \\
& EMT(2i + 2, 2) = a'_{2i}EMT(2i - 1, 2) + b'_{2i}EMT(2i, 2) \\
& EMT(2i + 2, 3) = a'_{2i}EMT(2i - 1, 3) + b'_{2i}EMT(2i, 3) + c'_{2i}
\end{align*}
$$

For a river with $m$ reaches, the number of cross sections is $m + 1$. Based on Equation (7), we can obtain

$$
\begin{align*}
\{ & \Delta Q_{m+1} = a\Delta Z_1 + b\Delta Z_{m+1} + c \\
& \Delta Q_1 = d\Delta Z_1 + e\Delta Z_{m+1} + f
\end{align*}
$$

where

$$a = EMT(2m + 2, 1) - \frac{EMT(2m + 1, 1) * EMT(2m + 2, 2)}{EMT(2m + 1, 2)}; \quad b = \frac{EMT(2m + 2, 2)}{EMT(2m + 1, 2)};
$$

$$c = EMT(2m + 2, 3) - \frac{EMT(2m + 1, 3) * EMT(2m + 2, 2)}{EMT(2m + 1, 2)}; \quad d = -\frac{EMT(2m + 1, 1)}{EMT(2m + 1, 2)};
$$

$$e = \frac{1}{EMT(2m + 1, 2)}; \quad f = -\frac{EMT(2m + 1, 3)}{EMT(2m + 1, 2)}
$$

According to the above equations, we can see that the flow increment at the first and last cross-sections of the river can be expressed in terms of the increment of the water level. Therefore, the corresponding equations using the non-increment method can be obtained by expanding Equation (12). The equations are:

$$
\begin{align*}
Q'^{t+1}_{m+1} &= aZ'^{t+1}_{1} + bZ'^{t+1}_{m+1} - aZ'^{t}_{1} - bZ'^{t}_{m+1} + Q'^{t}_{m+1} + c \\
Q'^{t+1}_{1} &= dZ'^{t+1}_{1} + eZ'^{t+1}_{m+1} - dZ'^{t}_{1} - eZ'^{t}_{m+1} + Q'^{t}_{1} + f
\end{align*}
$$

In addition, in this study we adopted a three-stage joint solution method based on a large sparse matrix solver to solve the river network. The method divides the solution of the equations into three levels—the micro-segments, river segments, and nodes—and calculates and back-substitutes them step by step, in order to compress the size of the matrix and improve the calculation speed.

3. General Form of a River Network Hydrodynamic System

The general form of a one-dimensional hydrodynamic equation formed using the model described above can be expressed as

$$F(X, t) = 0$$

where $F$ is defined as the hydrodynamic equation. $X$ is the hydraulic state variable, including the discharge, water depth, or water level, and $t$ is time.

Given the existence of white noise in relation to hydraulic state variables, the following discretized nonlinear hydraulic state equation can be obtained:

$$X_k = \Phi_k(X_{k-1}, k - 1) + \Gamma_k(X_{k-1}, k - 1) W_k$$

where $W_k$ is a random variable representing the white noise.
where $\Phi_k$ refers to the nonlinear hydraulic state functional from the state at time $k - 1$ to the state at $k$, including the hydrodynamic model functions written in Section 2, and $\Gamma_k$ is the weighting matrix of state noise, which is normally taken as a unit matrix. $W_k$ is a dynamic noise with a mean value equal to zero.

In addition, the equation for the measured river network state variables $Y_k$ can be expressed as:

$$Y_k = H_k(X_k) + V_k$$

where $H_k$ is the linear measurement function at time $k$; $V_k$ is the measurement noise with a mean value equal to zero.

By combining Equations (2) and (3), we obtain the equations describing the hydrodynamic nonlinear dynamic system of a river network:

$$\begin{align*}
X_k &= \Phi_k(X_{k-1}, k-1) + \Gamma_k(X_{k-1}, k-1)W_k \\
Y_k &= H_k(X_k) + V_k
\end{align*}$$

### 4. Application of Ensemble Kalman Filter

In this study we established a new scheme based on EnKF and applied it to a river network hydrodynamic system. A three-stage joint solution method was used to solve the hydrodynamic model of the river network, and the water level was selected as the state variable of the nonlinear dynamic system of the river network. Before the start of the filtering treatment, the Box–Muller transform method was used to generate an initial sample ensemble of nodal water level randomly. The data followed the normal distribution $N(\mu, \sigma^2)$. To ensure the diversity of the sampling data, with reasonable dispersion, the average value $\mu$ was determined based on the simulated water level at the start of filtering, and the standard deviation was also set as a certain proportion of the nodal water depth at the start of filtering.

The initial sample ensemble of the nodal water level was generated by means of the Box–Muller transform method. Its average value $\overline{X}$ is shown as:

$$\overline{X} = \frac{1}{m} \sum_{i=1}^{m} X_i,$$

where $m$ represents the number of samples, i.e., the size of the ensemble. The sample values obey a normal distribution and have a standard deviation of $\sigma$.

The disturbance term $\Delta X_i$ for the $i$th ensemble member is

$$\Delta X_i = X_i - \overline{X},$$

and $\Delta X$ is defined as the ensemble of disturbance terms:

$$\Delta X = (\Delta X_1, \Delta X_2, \ldots, \Delta X_m).$$

Then, the error variance matrix can be expressed as follows:

$$P = \frac{1}{m-1} \Delta X (\Delta X)^T,$$

The data assimilation process for river networks using the EnKF algorithm is shown in Figure 1. The details of the algorithm’s development are shown below.

When filtering, the first step is to calculate the state simulation values of each sample set, that is, the prediction values $\hat{X}_{k/k-1}$:

$$\hat{X}_{k/k-1} = \Phi_k(\hat{X}_{k-1})$$
Figure 1. Flowchart of the EnKF-based data assimilation process.

Then, the prediction error variance matrix and gain matrix are calculated as in Equations (24) and (25), respectively. Calculating the gain matrix requires the introduction of the observation noise level, that is, the observation error. The Box–Muller method is also used to discretize the observation data to obtain the observation set, which is fed into the ensemble Kalman filtering algorithm, together with the forecast water level set for assimilation, to obtain a set of assimilated water levels, whose mean value is the optimal solution for data assimilation.

\[
P_{k/k-1} = \frac{1}{m-1} \Delta \hat{X}_{k/k-1} (\Delta \hat{X}_{k/k-1})^T, \tag{24}
\]

\[
K_k = P_{k/k-1} H_k^T (H_k P_{k/k-1} H_k^T + R_k)^{-1}, \tag{25}
\]

where \(R_k\) is the observation error covariance; \(P_{k/k-1} H_k^T\) and \(H_k P_{k/k-1} H_k^T\) are calculated according to the following equations:

\[
P_{k/k-1} H_k^T = \frac{1}{m-1} \sum_{i=1}^{m} \left[ \hat{X}_{i/k-1} - \bar{X}_{k/k-1} \right] \times H_k \left[ \hat{X}_{i/k-1} - H_k \left( \bar{X}_{k/k-1} \right) \right]^T, \tag{26}
\]

\[
H_k P_{k/k-1} H_k^T = \frac{1}{m-1} \sum_{i=1}^{m} \left[ H_k \left( \hat{X}_{i/k-1} \right) - H_k \left( \bar{X}_{k/k-1} \right) \right] \times \left[ H_k \left( \hat{X}_{i/k-1} \right) - H_k \left( \bar{X}_{k/k-1} \right) \right]^T, \tag{27}
\]

where

\[
\bar{X}_{k/k-1} = \frac{1}{m} \sum_{i=1}^{m} \hat{X}_{i/k-1}, \tag{28}
\]

\[
\bar{H}_k \left( \bar{X}_{k/k-1} \right) = \frac{1}{m} \sum_{i=1}^{m} H_k \left( \hat{X}_{i/k-1} \right), \tag{29}
\]
Finally, update each ensemble member of the water level variable based on the gain matrix:

$$
\hat{X}_i^k = X_i^{k/k-1} + K_k [Y_i^k - H_k (X_i^{k/k-1})] ,
$$
(30)

where $Y_i^k = Y_i^k + \Delta Y_i^k$ is the observation with disturbance, $\Delta Y_i^k$ is a disturbance term, $\Delta Y_i^k \sim N(0, R)$, and $R$ is the observation error covariance. Note that the corrected water level is used as the current moment value, which is then used as the next moment value of the water level prediction. The mean value of the corrected water level is set as the best estimate of the current moment of data assimilation (optimal solution).

The correction error variance matrix is also calculated as an approximation of the error covariance matrix of the corrected values:

$$
P_k = \frac{1}{m-1} \Delta \hat{X}_k (\Delta \hat{X}_k)^T,
$$
(31)

With recurrence in time, the above process can be cycled continuously to obtain the data assimilation results for each calculation moment. In addition, the filtering process can be ended when the correction error variance matrix is less than a certain value in order to improve the efficiency of the river network hydrodynamic model calculation; similarly, the filtering process can be started when the error value between the simulated and observed values exceeds a certain set value in order to improve the simulation accuracy of the river network hydrodynamic model.

5. Parameter Analysis

In this section, parameters, including the standard deviation $\sigma$ of the initial sample ensemble, the observation error/noise level $\delta$, and the size of the ensemble $m$, are analyzed in terms of their influence on the data assimilation performance for a simulated river network with unsteady-state flows. Note that this river network was adopted from the literature [22]. Figure 2 illustrates a looped canal river network with 14 branches connected by nodes depicted by number. This network was simulated as a convergent type, representing drainage systems in practice. Water level measurement stations were set at nodes 9 and 13, named Station 1 and Station 2. The detailed characteristics of this looped canal network can be found in the literature [22].

![Figure 2. Illustration of the simulated water river network.](image)

The analysis was conducted on a computer with an Intel Core i7-8700K CPU, 3.2 GHz CPU clock speed and 16.0 GB DDR4 RAM. The system was run on Windows 10 Enterprise, version 1909.

5.1. Observation Noise Level and Standard Deviation of the Initial Ensemble

Table 1 shows the test scheme. The results calculated based on the true value of the roughness were regarded as the precise water level values, and the observation noise (i.e., error) was added to the precise water level values to produce the observation values. The observation noise levels $\delta$ (0.0, 0.0001, 0.0005, 0.001, 0.005, 0.01, and 0.05 m) were analyzed.
The standard deviation of the initial ensemble was 0.0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, and 0.1 m. The average water depth value was 2.5 m; correspondingly, the relative observation noise level values were 0%, 0.004%, 0.02%, 0.04%, 0.2%, 0.4%, and 2.0%. The relative standard deviation of the initial ensemble was 0%, 0.004%, 0.02%, 0.04%, 0.2%, 0.4%, 2.0%, and 4.0%. A constant ensemble size of 100 was tested.

Table 1. Combination of standard deviation and observation noise levels for parameter analysis.

<table>
<thead>
<tr>
<th>Observation Noise Levels / m</th>
<th>0.0</th>
<th>0.0001</th>
<th>0.0005</th>
<th>0.001</th>
<th>0.005</th>
<th>0.01</th>
<th>0.05</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>√</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>0.0001</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>0.0005</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>0.001</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>0.005</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>0.01</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>0.05</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Note: √ means the calculation was successful; × means the calculation was unsuccessful due to the failure of convergence and data assimilation.

The performance of the EnKF algorithm with different parameters was assessed on the basis of the root-mean-square-error (RMSE) of the water level, which was calculated based on the simulated and precise values. When the observation noise level δ equaled 0.0, except for standard deviation = 0.0 (i.e., equivalent to no data assimilation), the data assimilation failed. This is because the gain matrix K decreased continuously during the data assimilation process, implying that the weight of using new information to correct the predicted values decreased, i.e., the tracking performance of the system decreased. It was thus necessary to use an observation noise level of more than zero to ensure a certain disturbance level and prevent data assimilation failure.

Figure 3 shows that the RMSE values changed greatly with different settings for the parameters of standard deviation, σ, and observation noise level, δ, at Stations 1 and 2. When σ = 0.0, the sample data in the initial ensemble were constant numbers, which were the initial conditions of the water level at each node. In this case, the simulation results were not affected by the observation noise or the ensemble size, and the results were equivalent to the results without data assimilation. The RMSE values decreased significantly with increasing standard deviation, particularly at the low level of observation noise. This indicates that a relatively high standard deviation and low observation noise level could lead to a good data assimilation result. For example, at Station 1, a negligible RMSE value of 0.0011 m was produced when σ = 0.1, δ = 0.0001, whereas a high RMSE value of 0.1563 m was obtained when σ = 0.0001, δ = 0.05.

Figure 3. RMSE values with various standard deviation and observation noise level values at (a) Station 1 and (b) Station 2.
5.2. Ensemble Size

As the computational resources are limited for realistic and large-scale hydraulic applications, the ensemble size should be small. We thus tested four ensemble sizes, \( n = 50, 100, 150, \) and 200. The observation noise level \( \delta \) was taken to be 0.0001 m and 0.05 m, and the standard deviation \( \sigma \) was taken to be 0.0001 m and 0.1 m. Table 2 presents the parameters and their values.

Table 2. Combination of ensemble size, standard deviation, and observation noise level.

<table>
<thead>
<tr>
<th>Ensemble Size m</th>
<th>( \delta = 0.0001, \sigma = 0.0001 )</th>
<th>( \delta = 0.0001, \sigma = 0.1 )</th>
<th>( \delta = 0.05, \sigma = 0.0001 )</th>
<th>( \delta = 0.05, \sigma = 0.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>√</td>
<td>√</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>100</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>150</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>/</td>
</tr>
<tr>
<td>200</td>
<td>√</td>
<td>√</td>
<td>/</td>
<td>/</td>
</tr>
</tbody>
</table>

Note: √ means the calculation was successful; × means the calculation was unsuccessful due to the failure of convergence and data assimilation.

Tables 3 and 4 show the RMSE values with different ensemble sizes at certain observation noise levels and standard deviation values. Generally, the RMSE values decreased with increasing ensemble size for the low observation noise level (i.e., \( \delta = 0.001, \sigma = 0.0001 \) or 0.1), whereas for large observation noise level of 0.05, either the RMSE values were constant or the calculation failed, indicating that the data assimilation process was not improved by increasing the ensemble size. As shown in Table 5, computational cost increased significantly with ensemble size, and a large ensemble size could result in an extensive cost in terms of computational resources. Thus, we determined that the ensemble sample size could be set at a medium level (i.e., 100 or 150). Based on the sensitivity analysis of the observation noise level and the standard deviation of the initial ensemble, we suggest that a relatively low observation noise level (e.g., from 0.0001 to 0.01 m) and a relatively high standard deviation (e.g., from 0.01 to 0.1 m), along with a medium ensemble sample size (e.g., from 100 to 150), should be used to produce a good data assimilation with low RMSE values.

Table 3. RMSE values with different ensemble sizes at fixed \( \delta \) and \( \sigma \) values for Station 1.

<table>
<thead>
<tr>
<th>Ensemble Size m</th>
<th>( \delta = 0.0001, \sigma = 0.0001 )</th>
<th>( \delta = 0.0001, \sigma = 0.1 )</th>
<th>( \delta = 0.05, \sigma = 0.0001 )</th>
<th>( \delta = 0.05, \sigma = 0.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.00877</td>
<td>0.00036</td>
<td>0.15169</td>
<td>0.01161</td>
</tr>
<tr>
<td>100</td>
<td>0.00573</td>
<td>0.00049</td>
<td>0.15169</td>
<td>0.00697</td>
</tr>
<tr>
<td>150</td>
<td>0.00304</td>
<td>0.00024</td>
<td>0.15169</td>
<td>/</td>
</tr>
<tr>
<td>200</td>
<td>0.00240</td>
<td>0.00025</td>
<td>0.15169</td>
<td>/</td>
</tr>
</tbody>
</table>

Table 4. RMSE values with different ensemble sizes at fixed \( \delta \) and \( \sigma \) values for Station 2.

<table>
<thead>
<tr>
<th>Ensemble Size m</th>
<th>( \delta = 0.0001, \sigma = 0.0001 )</th>
<th>( \delta = 0.0001, \sigma = 0.1 )</th>
<th>( \delta = 0.05, \sigma = 0.0001 )</th>
<th>( \delta = 0.05, \sigma = 0.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.01015</td>
<td>0.00038</td>
<td>0.05223</td>
<td>0.01627</td>
</tr>
<tr>
<td>100</td>
<td>0.00643</td>
<td>0.00051</td>
<td>0.05223</td>
<td>0.00807</td>
</tr>
<tr>
<td>150</td>
<td>0.00305</td>
<td>0.00023</td>
<td>0.05223</td>
<td>/</td>
</tr>
<tr>
<td>200</td>
<td>0.00233</td>
<td>0.00029</td>
<td>0.05223</td>
<td>/</td>
</tr>
</tbody>
</table>
Table 5. Computational cost in seconds with different ensemble sizes at fixed $\delta$ and $\sigma$ values.

<table>
<thead>
<tr>
<th>Ensemble Size m</th>
<th>$\delta = 0.0001, \sigma = 0.0001$</th>
<th>$\delta = 0.0001, \sigma = 0.1$</th>
<th>$\delta = 0.05, \sigma = 0.0001$</th>
<th>$\delta = 0.05, \sigma = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>43.42</td>
<td>44.05</td>
<td>44.61</td>
<td>43.73</td>
</tr>
<tr>
<td>100</td>
<td>95.19</td>
<td>96.34</td>
<td>98.21</td>
<td>98.77</td>
</tr>
<tr>
<td>150</td>
<td>151.92</td>
<td>153.06</td>
<td>153.84</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>219.14</td>
<td>222.66</td>
<td>222.48</td>
<td></td>
</tr>
</tbody>
</table>

6. Experimental Evaluation

6.1. Data Description

To demonstrate the effectiveness of the proposed algorithm with the optimized parameter values, obtained as described in the parameter analysis section, we used two sets of data from a real single open-channel river in Canada and a real river network in China.

Case 1: A Real Single Open-Channel River.

The Kootenai River originates in British Columbia, Canada, and flows through Montana, Idaho, USA. A 24-km-long section of the river was used for this study, as shown in Figure 4. There are three hydrological stations on the river: Bonners Ferry, Tribal Hatchery, and Klockmann Ranch, which are located upstream, midstream, and downstream of the river, respectively. The measured water volume flow at Tribal Hatchery was used as the boundary condition for the upstream entrance (as shown in Figure 5a), and the measured water level at Klockmann Ranch (as shown in Figure 5b) was used as the boundary condition for the downstream exit, whereas Bonners Ferry was used as the test site. The cross-sectional information and water level and flow information used in the tests were obtained from the official website of the American Geographic Society. Table 6 shows the basic information of the selected 24-km-long section of the Kootenai river.

Case 2: A Real River Network.

Located in the southeast of Zhejiang Province, China, the Le Liu Hong plain river network covers an area of 325 km$^2$. It is divided into three river network systems—Hongqiao, Lecheng, and Liushi—according to its topography and geomorphology, which are controlled by control gates and are independent of each other. Each of the three river network water systems has a water level station, Hongqiao station, Lecheng station, and Liushi station, as shown in Figure 6. The hydrodynamic model of the river network of Le Liu Hong plain includes 398 rivers, 304 nodes, and a total of 1962 cross-sections. In this study, we used the 2019 Lekima typhoon storm flood data for data assimilation testing.

6.2. Parameter Settings and Evaluation Metrics

The Box–Muller method was used to generate a random number set for the initial water level values at Bonners Ferry station in Case 1 and Hongqiao, Lecheng, and Liushi stations in Case 2. The set mean value was given as the state value at the beginning moment of data assimilation. Based on the parameter analysis and optimization described in the parameter analysis section, the ensemble size $m$, the standard deviation $\sigma$, and the observation noise level $\delta$ were taken to be 100, 0.1 m, and 0.005 m, respectively. As the water depth values were between 6 and 9 m for Case 1 and between 3 and 4.5 m for Case 2, the corresponding relative observation noise level values and the relative standard deviation values were approximately less than 5% of the water depth values.

The RMSE, calculated based on the water level difference between the simulation and measurements from the start of the data assimilation process to the end of the water level simulation, was used as an evaluation metric. Furthermore, the algorithm calculation cost in seconds and the time for the algorithm to reach stability were used as the other two evaluation metrics. For Case 1, the simulation period covered 27 days, and the data assimilation process started on day 5 (hour 96). For Case 2, the simulation period was
from 8:00 on 7 August 2019 to 8:00 on 12 August 2019 (5 days in total), assuming that data assimilation took place from 0:00 on 9 August 2019. The filtering ended when the elements on the diagonal of the water level correction error variance matrix were less than 0.001 m$^2$.

![Schematic diagram of the Kootenai river](image)

**Figure 4.** Schematic diagram of the Kootenai river.

![Boundary conditions for the upstream entrance](image)

**Figure 5.** Boundary conditions for the upstream entrance: (a) water flow at Tribal Hatchery and the downstream exit, (b) water level at Klockmann Ranch.

**Table 6.** Basic information of the Kootenai river.

<table>
<thead>
<tr>
<th>Section Number</th>
<th>River Length/m</th>
<th>River Width/m</th>
<th>Average Slope</th>
<th>Manning’s Roughness Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1595</td>
<td>265–432</td>
<td></td>
<td>0.035</td>
</tr>
<tr>
<td>2</td>
<td>4634</td>
<td>173–319</td>
<td></td>
<td>0.030</td>
</tr>
<tr>
<td>3</td>
<td>4627</td>
<td>155–299</td>
<td>0.0004</td>
<td>0.035</td>
</tr>
<tr>
<td>4</td>
<td>10,327</td>
<td>188–343</td>
<td></td>
<td>0.035</td>
</tr>
<tr>
<td>5</td>
<td>1567</td>
<td>151–289</td>
<td></td>
<td>0.035</td>
</tr>
<tr>
<td>6</td>
<td>1543</td>
<td>106–197</td>
<td></td>
<td>0.035</td>
</tr>
</tbody>
</table>
was much lower than the water level error without data assimilation. The RMSE values at Liushi station, as shown in Figure 6. The hydrodynamic model of the river network of Le Liu Hong plain includes 398 rivers, 304 nodes, and a total of 1962 cross-sections. In this study, we used the 2019 Lekima typhoon storm flood data for data assimilation testing. The network covers an area of 325 km². It is divided into three river network systems—Hongqiao, Lecheng, and Liushi—according to its topography and geomorphology, which are controlled by control gates and are independent of each other. Each of the three river stations in Case 2.

### 6.3. Algorithms for Comparison

Our data assimilation scheme based on the EnKF algorithm was compared to the EKF method, which is commonly used for hydraulic predictions for river networks. The corresponding observed noise level and filtering time period were consistent with those calculated using the ensemble Kalman filter for both testing cases. As both algorithms contain an observation noise level parameter, this parameter was varied, at 0.0005, 0.005, and 0.05 m, to examine whether it affected the difference in performance between the EnKF algorithm and the EKF algorithm. Note that the observation noise value of 0.05 m was so high that it was impractical in real river networks; thus, this setting was only used for research purposes.

### 6.4. Results and Discussion

#### 6.4.1. Comparison of the EnKF and EKF Algorithm

Figure 7 shows the water level error values between the simulation and the measurements among different data treatment schemes for Case 1. The evaluation metrics used for evaluating data assimilation performance are tabulated in Table 7. The data assimilation started in hour 96, and after 23 h of filtering, the data assimilation end condition had been reached for both the EnKF and EKF algorithms. The data assimilation process, whether using either EnKF or EKF, produced an instant and efficient reduction in the simulation errors. The time required for both EnKF and EKF algorithms to reach stability was minimal, indicating that both algorithms could correct the prediction results instantly for this single-river case. With the end of the assimilation phase at hour 119, the filtered water level error increased slightly but still fluctuated, mainly between −0.05 m and 0.05 m, which was much lower than the water level error without data assimilation. The RMSE values decreased significantly by around two times after the data assimilation process was carried out using EnKF or the EKF algorithm. Although the EnKF algorithm produced a slightly lower RMSE, the difference was negligible for Case 1. However, the algorithm calculation cost for EKF was 16.7 s, which was 19.2% higher than that of EnKF, indicating that EKF was more time-consuming than EnKF. This result is similar to those found in the literature [23], namely, that the computational time of EnKF was 0.8 of that of EKF. This is because EKF requires the computation of the tangent linear state operator, whereas EnKF avoids this computation step.
Figure 7. Water level simulation error values for different data treatment schemes. The black line represents the data that were not treated via data assimilation; the green line represents the data treated via EKF; the red line represents the data treated via EnKF.

Table 7. Comparison of evaluation metrics for EKF and EnKF in Case 1.

<table>
<thead>
<tr>
<th>Treatment Scheme</th>
<th>Calculation Cost (s)</th>
<th>RMSE (m)</th>
<th>Filtering Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without data assimilation</td>
<td>2.8</td>
<td>0.0841</td>
<td>/</td>
</tr>
<tr>
<td>EKF</td>
<td>16.7</td>
<td>0.0276</td>
<td>Instant</td>
</tr>
<tr>
<td>EnKF</td>
<td>13.5</td>
<td>0.0266</td>
<td>Instant</td>
</tr>
</tbody>
</table>

Note that the river in Case 1 was a mountainous single-channel river with a relatively simple water flow. This enabled a significant reduction in the algorithm calculation time and thus in the difference in calculation cost between EKF and EnKF. Indeed, for Case 2, which was a complex river network, we observed greater differences in the evaluation metrics.

Figure 8 shows the simulated water level without data assimilation at the beginning period of the simulation (before hour 40) and with subsequent data assimilation using EKF and EnKF. Generally, the simulation error was substantially reduced and approximated to zero at the end of data assimilation using EKF or EnKF. This simulation error would spread out gradually over the long term after the end of data assimilation, indicating that the data assimilation effect could still be maintained for a relatively long period (up to 20 to 30 h) with negligible errors after the end of the data assimilation process.
Table 8 presents a comparison of the evaluation metric results of EKF and EnKF for different stations. The calculation time required by EnKF for this river network was 4243.6 s, which was only 45% of that required by EKF (9335.4 s). Compared to the mountainous single-channel river, as expected, the calculation time increased hugely; furthermore, EnKF exhibited a much lower calculation cost value relative to EKF, i.e., the ratio of the calculation cost between EnKF and EKF decreased greatly, from 81% to 45%, indicating that the calculation efficiency of EnKF was much higher than that of EKF for a complex river network. We also found that the RMSE values of EnKF were much lower than those of EKF, regardless of the water level measuring stations selected. Particularly, Lecheng station had an RMSE value of only 9% when using EnKF compared to EKF. Interestingly, EnKF was more stable than EKF during the data assimilation period at the three stations. The high fluctuations of EKF may reduce its convergence. The filtering time (i.e., the time taken by the algorithm to reach stability after the beginning of filtering) was also much shorter for EnKF than for EKF. For example, for the Hongqiao station, EnKF took 5 h to reach stability whereas EKF took 20 h.

Table 8. Comparison of evaluation metric results for EKF and EnKF in Case 2.

<table>
<thead>
<tr>
<th>Treatment Scheme</th>
<th>Calculation Cost (s)</th>
<th>RMSE (m)</th>
<th>Filtering Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Hongqiao</td>
<td>Lecheng</td>
</tr>
<tr>
<td>Without data assimilation</td>
<td>204.3</td>
<td>0.1735</td>
<td>0.0655</td>
</tr>
<tr>
<td>EKF</td>
<td>9335.4</td>
<td>0.0113</td>
<td>0.0187</td>
</tr>
<tr>
<td>EnKF</td>
<td>4243.6</td>
<td>0.0026</td>
<td>0.0016</td>
</tr>
<tr>
<td>Ratio of EnKF to EKF</td>
<td>45%</td>
<td>23%</td>
<td>9%</td>
</tr>
</tbody>
</table>

6.4.2. Parameter Sensitivity

To verify the effectiveness of EnKF with different key parameter values, Figure 9 shows the RMSE values of EnKF compared to EKF at different observation noise levels. The EnKF algorithm exhibited much lower simulation errors at observation noise level values of 0.0005 and 0.005, regardless of the observation stations. The EKF algorithm had a slightly lower simulation error at Hongqiao station and Liushi station, which may be due to the impractically high value of observation noise in practice.
Treatment Scheme | Filtering Time (h) | Calculation Cost (s) | Hongqiao | Lecheng | Liushi | Hongqiao | Lecheng | Liushi | Ratio of EnKF to EKF |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\delta = 0.005$ m</td>
<td>$\delta = 0.005$ m</td>
<td>$\delta = 0.05$ m</td>
<td>$\delta = 0.005$ m</td>
<td>$\delta = 0.005$ m</td>
<td>$\delta = 0.05$ m</td>
<td>$\delta = 0.05$ m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EKF</td>
<td>26</td>
<td>20</td>
<td>37</td>
<td>11,378</td>
<td>9335</td>
<td>15,463</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EnKF</td>
<td>7</td>
<td>7</td>
<td>30</td>
<td>4248</td>
<td>4244</td>
<td>5415</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio of EnKF to EKF</td>
<td>27%</td>
<td>35%</td>
<td>81%</td>
<td>37%</td>
<td>45%</td>
<td>35%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 9. Comparison of RMSE values of EnKF and EKF at (a) Hongqiao station, (b) Lecheng station, and (c) Liushi station.

Similarly, EnKF’s calculation costs and the time required for EnKF to reach stability after the beginning of filtering at different observation noise levels were also consistently much lower than those of EKF, particularly at practical observation noise levels (0.0005 and 0.005), as shown in Table 9. The time for the EnKF to reach stability after the beginning of filtering was 27% to 81% of that for EKF, and the calculation cost for EnKF was 27% to 81% to that for EKF.

Table 9. Impact of observation noise level on data assimilation results based on EKF and EnKF.

The above testing comparison between EnKF and EKF showed the great advantages of the EnKF-based data assimilation model when it was applied to the complex river network. Indeed, both EnKF and EKF have their advantages in predicting water levels. However, EnKF was more effective, especially in terms of computational effort and the time to reach stable prediction.

The computational characteristics of the Kalman filter (e.g., a short online estimation time, small storage size, and convenient error analysis) are assessed to determine its suitability for real-time correction processing of dynamic systems. For a nonlinear dynamic system of a complex river network, the EKF algorithm is often used, but there are still some problems with this approach, mainly: (1) it is difficult to calculate the prediction error variance matrix, and its accuracy cannot be guaranteed, as verified by our experiments in which data assimilation based on EKF produced much higher simulation errors than EnKF-based data assimilation. (2) It is necessary to calculate both the tangent state operator and the covariance of dynamic noise [12], which makes the algorithm structure complex and causes a large amount of calculations. The calculation cost of EKF is almost twice as high as that of EnKF. (3) The hydraulic calculation of river networks is a strongly nonlinear problem. The use of the EKF algorithm requires one to omit the terms above the first-order derivative; however, the omitted terms could precisely reflect the nonlinear effect of the strong nonlinear problem. (4) The EKF algorithm contains an assumption in the application, that is, the error obeys a normal distribution; therefore, it is not suitable to be applied to a system with a non-normal distribution of errors. (5) EKF is not capable of addressing horizontal error correlations in the model or measurement errors in large systems due to computational limitations [24].

For water level prediction in practice, the accuracy and timing of water level predictions are also important. Our experiments on the Le Liu Hong plain river network showed that the EnKF-based assimilation model was able to forecast the water level after
5 to 8 h and make predictions up to 20 to 30 h after the data assimilation process was completed. These timings are similar to those found in other research results. For example, Karri et al. [25] conducted data assimilation research on the regional waters of Singapore, revealing that EnKF is efficient in short-term forecasting of water levels and currents, and the model could make better predictions up to 24 h, with the highest forecast quality within 6 h. Although increasing the initial ensemble size could lead to increased computation costs, as shown in Table 5, the overall computational cost was still relatively low. For example, the computational cost of the EnKF-based algorithm, as tested on the simulated river network, was only 223 s when using a personal computer when the ensemble size was 200, observation noise was 0.0001, and standard deviation was 0.1.

Given the complex nature of a river network, more parameters, such as the water-level measurement station number and the river roughness coefficient, can be simultaneously corrected during the simulation time with hydraulic state variables. This further increases the forecasting accuracy and efficiency of the EnKF-based data assimilation model developed in this paper. Sensitivity to the type of river network and connection complexity could also be essential to testing the model’s application capacity in practice. In the present study, we focused on comparing the data assimilation strategy based on the hydraulic model of a river network coupled with EKF and that coupled with EnKF. Our main objective was to understand the advantages of the data assimilation method based on EnKF. Thus, the 1D hydraulic model based on the Saint-Venant equations was used to solve the state variables of river networks. Indeed, 2D models are being developed by researchers [26–28], and future work can be carried out to develop a data assimilation strategy based on a 2D hydraulic model.

7. Conclusions

Given the complex nature of a river network, it is challenging to perform good water level predictions with high accuracy and low computational cost when using the traditional EKF-based data assimilation approach. In this study, we developed a new data assimilation scheme based on the EnKF algorithm to optimize real-time water level simulation in river networks. The parameter analysis results showed that the initial sample ensemble size, its standard deviation, and the observation noise level could significantly impact the data assimilation performance. Optimized parameter settings, with a medium ensemble sample size (e.g., from 100 to 150), a relatively low observation noise level (e.g., from 0.0001 to 0.01 m), and a relatively high standard deviation (e.g., from 0.01 to 0.1 m), resulted in good data assimilation performance.

The experimental testing conducted on a mountainous single-channel river showed that the data assimilation method based on EnKF had a similar capacity to correct simulation errors to that of the EKF-based model, but for a river network with complex branches and connecting nodes, compared to the EKF-based model, the EnKF-based data assimilation method was able to decrease simulation errors greatly by up to 49%, to reduce calculation costs by 45%, and to reduce the filtering time by up to 43%. In addition, this performance was not impacted by the key model parameter, i.e., the observation noise level.

The experiments conducted on a real water river network showed that the EnKF-based data assimilation model forecast water levels with a high accuracy after 5 to 8 h and could make predictions up to 20 to 30 h after the data assimilation process was completed, indicating the potential application capacity of the EnKF-based assimilation model developed in this paper.

Author Contributions: Conceptualization, Y.C. and F.C.; methodology, Y.C.; validation, X.M. and W.C.; formal analysis, Y.C.; writing—original draft preparation, Y.C.; writing—review and editing, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 12002310 and 51709235, General Scientific Research Projects of Education of Zhejiang Province, grant number Y201941076, and Zhejiang Water Science and Technology Research Project, grant number
References


19. Zhang, Z.; Fu, K.; Sun, X.; Ren, W. Multiple target tracking based on multiple hypotheses tracking and modified ensemble Kalman filter in multi-sensor fusion. *Sensors* 2019, 19, 3118. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.