

**Particle filter–based data assimilation in dynamic data-driven simulation
sensitivity analysis of three critical experimental conditions**

Huang, Yilin; Xie, Xu; Cho, Yubin; Verbraeck, Alexander

DOI

[10.1177/00375497221143988](https://doi.org/10.1177/00375497221143988)

Publication date

2022

Document Version

Final published version

Published in

Simulation

Citation (APA)

Huang, Y., Xie, X., Cho, Y., & Verbraeck, A. (2022). Particle filter–based data assimilation in dynamic data-driven simulation: sensitivity analysis of three critical experimental conditions. *Simulation*, 99(4), 403-415. <https://doi.org/10.1177/00375497221143988>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Particle filter–based data assimilation in dynamic data-driven simulation: sensitivity analysis of three critical experimental conditions

Yilin Huang¹ , Xu Xie², Yubin Cho³ and Alexander Verbraeck¹

Simulation: Transactions of the Society for Modeling and Simulation International
1–13

© The Author(s) 2022



DOI: 10.1177/00375497221143988

journals.sagepub.com/home/sim



Abstract

Data assimilation (DA) is a methodology widely used by different disciplines of science and engineering. It is typically applied to continuous systems with numerical models. The application of DA to discrete-event and discrete-time systems including agent-based models is relatively new. Because of its non-linearity and non-Gaussianity, the particle filter (PF) method is often a good option for stochastic simulation models of discrete systems. The probability distributions of model runs, however, make it computationally intensive. The experimental conditions therein are understudied. This paper studied three critical conditions of PF-based DA in a discrete event model: (1) the time interval between two consecutive DA iterations, (2) the number of particles, and (3) the actual level and perceived level of measurement errors (or noises). The study conducted identical-twin experiments of an M/M/1 single server queuing system. The ground truth is imitated in a stand-alone simulation model. The measurement errors are superimposed so that the effect of the three conditions can be quantitatively evaluated in a controlled manner. The results show that the estimation accuracy of such a system using PF is more constrained by the choice of time intervals than the number of particles. An under estimation of measurement errors produces worse state estimates than an over estimation of errors. A correct perception of the measurement errors does not guarantee better state estimates. Moreover, a slight over estimation of errors results in better state estimates, and it is more responsive to abrupt system changes than an accurate perception of measurement errors.

Keywords

Dynamic data-driven simulation, data assimilation, particle filter, measurement errors, time interval, sensitivity analysis

1. Introduction

Data assimilation (DA) is a methodology that combines observational data and the underlying dynamical principles that govern a system to produce the best estimate (according to some criteria) of the evolving state of that system.^{1–3} It is widely used by different disciplines of science and engineering (e.g., hydrology, meteorology, geophysics, and petroleum engineering) for state estimation and optimal control.¹ DA in its various forms has typically been applied to continuous systems with numerical models.^{4–7} The application of DA to discrete event systems (DESSs) and discrete time systems (DTSSs) including agent-based models (ABMs) is a recent development.

For example, Lloyd et al.⁸ used an urban crime ABM to generate ground truth data which was then assimilated into a discretized partial differential equation (PDE) model. The PDE was converted from the original ABM⁸

to overcome the computational demand of ABM. One of the first publications that explains how to use ensemble Kalman filter (EnKF) to calibrate simple ABM for social simulation is by Ward et al.⁹ They aim to present the

¹Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands

²College of Systems Engineering, National University of Defense Technology, China

³Samsung Electronics Benelux B.V., The Netherlands

Corresponding authors:

Yilin Huang, Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands.
Email: y.huang@tudelft.nl

Xu Xie, College of Systems Engineering, National University of Defense Technology, Changsha, China.
Email: x.xie@hotmail.com

method to ABM practitioners who are unfamiliar with DA. They also illustrate to DA experts the value of using DA (particularly sequential DA) in ABMs of complex social systems, and the new challenges these types of models present. Similarly, a DA framework for DES was published by Hu and Wu¹⁰ who applied particle filters (PFs) to a roadway modeled as a one-dimensional cellular space.

Different DA methods have been reported extensively. For example, Kalman filter (KF),¹¹ extended KF (EKF),¹² and EnKF¹³ are known for their efficiency. But they often are not applicable to DES, DTS, and ABM because of the requirement for linearity of model state and Gaussian errors.¹⁴ In addition, DES and ABM of real-world applications are typically stochastic and high dimensional. For these reasons, the PF method,^{11,15} which approximates the posterior distribution by Monte Carlo samples (also called particles), is a good alternative (in place of classical DA methods) for non-linearity and non-Gaussianity. However, the probability distributions of the runs (also called replications) of the DES and ABM models make the PF-based sequential DA computationally highly expensive.⁸

1.1. Background

The performance of sequential DA is strongly influenced by conditions such as the time interval between two consecutive iterations (hereafter simply *time interval*), sample sizes, and measurement errors. Surprisingly, not many publications studied and quantified the effect of such important conditions so far.

In a survey of DA in surface water quality modeling, Cho et al.¹⁶ stated that their domain utilized mainly three DA methods: the variational DA, EKF, and EnKF. With EKF-based DA for algal bloom prediction,¹⁷ longer update time intervals resulted in lower accuracy. Contradictory results also exist. For example, the frequency of EnKF-based water content DA is investigated for soil hydraulic models.¹⁸ The results show that DA with high update frequencies does not provide better results than those obtained using low frequencies. An EnKF-based DA procedure for water quality forecasting is developed by Kim et al.¹⁹ The authors suggested that the time interval (they called it *window size*) should be chosen carefully: if the window size is too small, the procedure works largely as a filter rather than a smoother, which reduces performance; if the window size is too large, some of the observations being assimilated may be too old and/or redundant to be informative.

Adaptive DA methods are also reported. For example, a (frequency) adaptive EnKF-based method is developed for hydrodynamic simulation²⁰ which reduced computation and increased error reduction. Besides works that used KF and its variations, PF is also coupled with a hydrobiogeochemical model using high-frequency data.²¹

However, how the change of the intervals and the number of particles can affect the results are not studied.²¹

In general, the literature on sensitivity analysis of DA conditions is not as rich as the literature on various DA methods and their applications. When such results are reported, as those mentioned above, they are often in the context of continuous systems (and numerical models). It is unknown whether such results are also applicable to PF-based DA for discrete systems.

1.2. DA for discrete systems

The sequential simulation of a DES, DTS or ABM can be denoted as a discrete time process, given by:

$$s_k = f_k(s_{k-1}) + \nu_{k-1}, k = 1, 2, \dots$$

where f_k is a (possibly non-linear) function of the state vector s_{k-1} , and ν_{k-1} represents a system noise process. This equation is referred to as a *system model* or simulation model which describes the state evolution of a system under interest. The state predicted by the system model is related to the measurement data by a *measurement model*, given by:

$$m_k = g_k(s_k) + \varepsilon_k, k = 1, 2, \dots$$

where g_k is a (possibly non-linear) function that maps the state s_k to the measurement m_k , and ε_k represents a measurement noise process. Both the system model and measurement model are discrete time models that assume a stepwise mode of execution, and the length of a time interval (ΔT) is largely determined by how often the sensor data are collected.

The objective of sequential DA is to estimate the conditional distribution of all states up to time k given all available measurements up to k . That is, $p(s_{0:k}|m_{1:k})$, where $s_{0:k} = \{s_i, i = 0, 1, 2, \dots, k\}$ and $m_{1:k} = \{m_j, j = 1, 2, \dots, k\}$. PF approximates the posterior distribution $p(s_{0:k}|m_{1:k})$ by a set of particles $\{s_{0:k}^i\}_{i=1}^N$ and their associated weights $\{w_k^i\}_{i=1}^N$. If the number of particles (N) is sufficiently large, the posterior can be approximated to an arbitrary accuracy.^{11,15,22}

For DES, sensitivity analysis of PF-based DA conditions and related issues are particularly important since applying PF to DES can be very beneficial but also computationally intensive. More research is needed to understand better how to use PF for applications such as social simulation and socio-technical systems simulation. For example, people's location estimation in smart buildings,^{23,24} household energy consumption behavior,^{25,26} vehicle trajectory reconstruction,²⁷ and traffic density estimation.²⁸

Those simulation models often provide detailed information about system states. But they are not typically data driven in the sense that the models are generally developed and calibrated (by human modelers) using historical data

before simulation.^{9,29} The data availability of such systems had been poor but is becoming greater with the advance of cheaper sensing technologies and pervasive use of smart devices such as smart meters and smartphones. This gave rise to the so-called dynamic data-driven simulation in DES, DTS, and ABM communities. The more available data offer new opportunities to complement and empower traditional simulation modeling approaches which have limitations in situational awareness and adaptation in a highly evolving socio-technical environment.^{9,10,30,31}

In this context, dynamic data-driven simulation can be explored in several directions, e.g., to automatically generate model structure aggregating predefined model components;^{32,33} to discover simulation models and their generative behaviors in an automated or semi automated way;³⁴ to assimilate real-time data into (online) simulation to support real-time decision-making.^{10,35} There is a rich body of knowledge with which experts from different domains using diverse simulation modeling paradigms and methods can collaborate and learn from one another. However, reported cases and synergies are rare in the literature.⁹

Some examples can be found with regard to the use of PF for discrete systems.^{9,10,36,37} The effect of conditions in PF such as modeling errors, measurement errors, and number of particles are studied with DES transportation models.^{27,38} The results show that the estimation accuracy of PF is robust to error assumptions of both the model and measurement data in the application. The accuracy increases as the error magnitude decreases, but it is far from being proportional. The estimation accuracy also improves as the number of particles increases due to increased state-space coverage. Similar findings are reported in a few other studies.^{24,28,35}

In this paper, we aim to further study the experimental conditions of PF applied to DES. The focus is on three common and critical conditions: the time interval, the number of particles, and the measurement errors. We study their influences on estimation accuracy with respect to computational demand. Sensitivity analysis of the time interval and number of particles in the PF method commonly used a one-factor-at-a-time approach.^{27,38} With this approach as a starting point, we explore the mutual influence of the two conditions. In addition, we experiment with the actual level and perceived level of measurement errors. The actual level of measurement errors refers to the level of noises added to the “ground truth” in order to imitate noisy measurement data. Because the actual measurement noises are often unknown in real-world situations, we distinguish the concept of perceived level of measurement errors in our experiments. It is the (assumed) level of measurement noises used for posterior computation in the DA sampling step.

The main contribution of this paper can be summarized as follows. First, we quantitatively analyze the effect of

time intervals and the number of particles in different experimental settings of PF-based sequential DA for DES. Second, the joint influence of the time interval and the number of particles is experimented and analyzed. These two conditions are often mutually restrictive because of limited computational time between two consecutive DA iterations. Third, the actual level of measurement errors is imitated such that it allows an investigation of the differences between the actual and perceived measurement errors. We analyze their effects on the estimation accuracy, discuss the implications, and give recommendations on future research directions. This paper is an extended version of Cho et al.³⁹

2. Methodology

This study conducted identical-twin experiments of a discrete event $M/M/1$ single server queuing system with balking. The goal is to study three experimental conditions of PF-based DA in a controlled manner: (1) the time interval between two consecutive DA iterations, (2) the number of particles, and (3) the actual level and perceived level of measurement errors (or noises). The true state of a real system is often uncertain for DA applications in practice due to uncertain measurement errors. In the experiments, a simulation model is used to imitate the real system so that perfect “ground truth” and measurement data can be obtained. Noises are then added to superimpose errors to the measurements used for DA. The true state of “the real system” is not revealed to the DA model. This way, we can accurately quantify the difference between the true system state, the noisy measurements, and the effect of perceived errors thereof. We evaluate the three experimental conditions with respect to the estimation accuracy of PF-based DA.

2.1. Scenario description

In the $M/M/1$ single server queuing system (Figure 1), the job arrival obeys a Poisson process with mean arrival rate λ . When the server is busy, the jobs will wait in a queue that has a limited size L for balking.⁴⁰ This means no new job is appended to the queue when the queue is full; in such cases, the generated job is discarded and will not enter the queue again at a later stage. The server can process jobs with processing times following an exponential distribution which has the mean service rate μ .

2.2. Modeling the scenario with Discrete Event System Specification formalism

The scenario described is a typical DES, which can be modeled using the Discrete Event System Specification (DEVS).⁴¹ As shown in Figure 2, our DEVS model of the

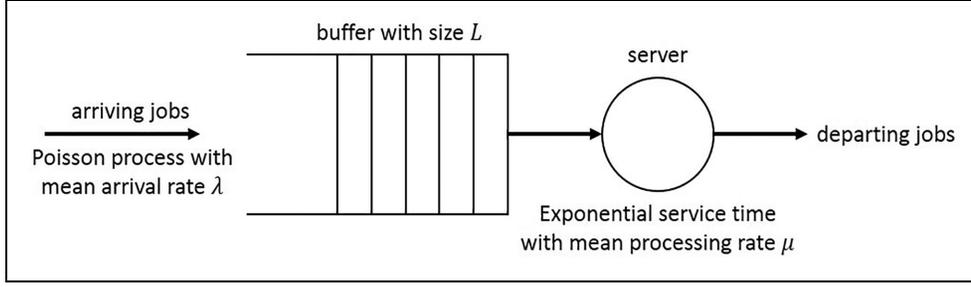


Figure 1. $M/M/1$ single server queuing system with balking.

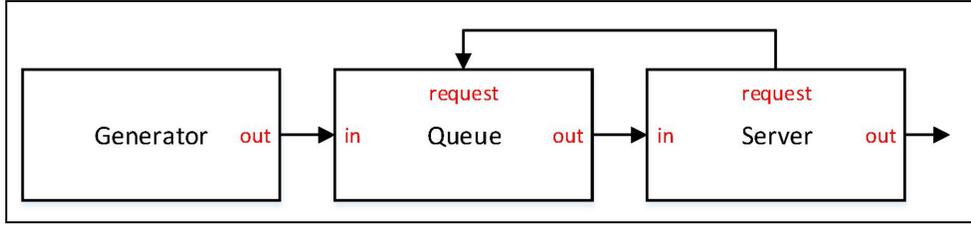


Figure 2. DEVS model of $M/M/1$ single server queuing system.

$M/M/1$ single server queuing system consists of three atomic components: *Generator*, *Queue*, and *Server*. The *Generator* model generates jobs, whose inter arrival times are exponentially distributed with mean $1/\lambda$, where λ is the job arrival rate. The jobs may have to wait in the queue to be processed by the server. The *Queue* model (with capacity L) is in a passive state unless it receives job requests from the *Server* model. Once the *Queue* received a request, it enters a transient phase (with zero length of lifetime) that sends the first job in the queue (if any) to the *Server* model. The *Server* processes the jobs one by one and transits between *BUSY* and *IDLE* phases. The lifetimes of *BUSY* follow an exponential distribution with mean $1/\mu$, where μ is the service rate. The *IDLE* phase has an infinite lifetime, which is interrupted once a job arrives. When finishing a job, the *Server* makes an internal state transition from *BUSY* to *IDLE* and requests a next job from the *Queue*.

To imitate the second-order dynamics³⁰ in the queuing system, at each time step during the simulation, the values of λ and μ are sampled from two uniform distributions: $\lambda \sim \mathcal{U}(0, 20)$ and $\mu \sim \mathcal{U}(0, 20)$. These are the two stochastic internal (non-observable) variables that the DA process needs to estimate for the simulation model. The state of the $M/M/1$ single Server queuing system at time step k is defined as

$$S_k = \{\lambda_k, \mu_k, n_k^q\}, k = 0, 1, 2, \dots \quad (1)$$

where λ_k is the mean arrival rate of jobs, μ_k is the mean service rate during the k th time step, and $n_k^q \in [0, L]$ is the

queue length at time step k . The state evolution of the $M/M/1$ single server queuing system (without particle filtering) is described as a discrete time process, i.e.:

$$S_{k+1} = \text{QueuingModel}(S_k) + \nu_k, k = 0, 1, 2, \dots \quad (2)$$

where *QueuingModel* is the (discrete event) simulation model of the $M/M/1$ single server queuing system, and ν_k is the system noise.

The *QueuingModel* is used as the base model for the simulation of ground truth, where ν_k is set to zero ($\nu_k = 0$). In the ground truth situation, the arrival rate λ_k and service rate μ_k are randomly generated from two uniform distributions $\lambda_k \sim \mathcal{U}(0, 20)$ and $\mu_k \sim \mathcal{U}(0, 20)$. They are “non observables.” The “observables” are the number of job arrivals n_k^a (after balking) and the number of job departures n_k^d during the k th time step. They are resulted from λ_k and μ_k at time step k . Measurement noises are added to n_k^a and n_k^d (Equation (3)) which are received by the DA process.

The *QueuingModel* is also used for DA (assimilating the noisy n_k^a and n_k^d data) that estimates the state of ground truth, namely, the state variables λ_k , μ_k , and n_k^q . Since a perfect model of the real system can hardly be obtained, we choose to add a Gaussian noise to each state variable. This acts as the system noise ν_k to imitate errors in the model. Each Gaussian noise has a mean of zero and a variance of 10% of the state value (see section 2.4). The initialization conditions of both models are listed in Table 1. In the experiments, the true state of the ground truth model is unknown to the DA model.

Table I. *QueuingModel* initialization conditions.

		Ground truth <i>QueuingModel</i>	Data assimilation <i>QueuingModel</i>
Queue capacity	L	100	100
Queue length	n_0^q	50	Sampled
Arrival rate	λ_0	4	Sampled
Service rate	μ_0	4	Sampled

2.3. Available data and measurement model

The ground truth *QueuingModel* is run for a certain length during which the state evolution of the model is recorded and regarded as the ground truth system state. In addition, the number of job arrivals (after balking) n_k^a and the number of job departures n_k^d during the k th time step are recorded. The job arrival and departure data are then processed to form the noisy measurement (m_k^o) which is used by the DA *QueuingModel*:

$$m_k^o = \begin{bmatrix} n_k^{a,o} \\ n_k^{d,o} \end{bmatrix} = \begin{bmatrix} n_k^a \\ n_k^d \end{bmatrix} + \begin{bmatrix} \varepsilon_k^a \\ \varepsilon_k^d \end{bmatrix} \quad (3)$$

where $\varepsilon_k^a \sim \mathcal{N}(0, \sigma_a^2)$ and $\varepsilon_k^d \sim \mathcal{N}(0, \sigma_d^2)$ are the imitated (actual) measurement errors (or noises). We used a binomial distribution to approximate the normal distribution of discrete values. The standard deviation σ_a (similarly, σ_d) is designed to take one of the four values denoted by $\varepsilon \cdot \Delta T$, where $\varepsilon \in [0, 3]$ represents the level of measurement errors from *zero* (0) to *low* (1), *medium* (2) till *high* (3), and ΔT is the time interval of assimilating measurement data. The unit of $\varepsilon \cdot \Delta T$ should not be understood as in the time domain but in the number domain, as $\Delta T \in \{1, 2, 3, 4, 5\}$ is used here as a proxy to indicate the magnitude of noises, which we assume is proportional to the data update time. For example, if $\Delta T = 5$, then σ_a (similarly, σ_d) is set to be in $\{0, 5, 10, 15\}$ depending on the corresponding level of errors. In addition, σ_a and σ_d are independent from each other. As a result, their joint probability can be obtained by the product of the two probabilities. The measurement model can then be formalized as:

$$m_k^o \sim p(m_k^o | S_k) \quad (4)$$

Note that in our experiments, the DA process uses the perceived level of measurement errors, denoted as ε' , which is not necessarily equal to the actual level of measurement errors, denoted as ε . The latter is rarely known in real-world situations. In previous works, ε' and ε were always deemed as the same.

2.4. DA process

In our experiments, PF (Algorithm 1) are employed to assimilate the noisy measurements m_k^o into the simulation

(of the DA model) to estimate the system state (Equation (1)). The main steps of the DA procedure are as follows:

- Initialization: in the initialization step (lines 2–5 in Algorithm 1), N particles are generated according to the given distributions. The i th particle $S_0^i = \{\lambda_0^i, \mu_0^i, n_0^{q,i}\}$ is a guess (i.e., Monte Carlo sample) of possible initial states of the *QueuingModel*. The weights of all particles are set to $1/N$.
- Sampling: after initialization, N model replications are run, each for one time step ΔT , to obtain N new particles (line 8 in Algorithm 1). Once new particles are generated, Gaussian noises are added to the states (lines 9–10 in Algorithm 1). After that, the weight of the i th particle is updated based on the newly available measurements (line 11 in Algorithm 1):

$$w_k^i = p(m_k^o | S_k^i) \times w_{k-1}^i$$

Note that the actual error level of the noisy measurement $m_k^o = [n_k^{a,o} \ n_k^{d,o}]^T$ (Equation (3)) is ε , while in the weight computation of our experiments, the perceived level of measurement errors ε' is used instead; i.e.:

$$p(m_k^o | S_k^i) = \frac{1}{2\pi\sigma_a'\sigma_d'} \cdot e^{-\frac{(n_k^{a,o} - n_k^{a,i})^2}{2(\sigma_a')^2} - \frac{(n_k^{d,o} - n_k^{d,i})^2}{2(\sigma_d')^2}}$$

where $\sigma_a' = \sigma_d' = \varepsilon' \cdot \Delta T$, and $[n_k^{a,i} \ n_k^{d,i}]^T$ is the corresponding value predicted by the i th particle.

- Resampling: to solve the degeneracy problem,^{11,15,22} we resample the particles using the standard resampling scheme (line 15 in Algorithm 1), which samples particles in proportion to their weights. Thereafter, all resampled particles are equally weighted, i.e., $w_k^i = 1/N$.
- Estimation: we estimate the system state at time step k by the following average (line 20 in Algorithm 1):

$$\hat{S}_k = \sum_{i=1}^N S_k^i \cdot w_k^i = \frac{1}{N} \sum_{i=1}^N S_k^i$$

2.5. Evaluation criteria

In the experiments discussed in the next section, three conditions in DA were investigated to study their effects on the estimation accuracy, i.e., the time interval ΔT , the number of particles N , and the level of measurement errors ε (as well as the perceived level of measurement errors ε'). To compare the estimation accuracy of different

Algorithm 1. The data assimilation procedure based on particle filters.

```

1 % initialization of particles at  $k = 0$ 
2 for  $i = 1 : N$  do
3   generate the  $i$  th particle  $S_0^i = \{\lambda_0^i, \mu_0^i, n_0^{q,i}\}$ , where
    $\lambda_0^i \sim \mathcal{U}(0,20), \mu_0^i \sim \mathcal{U}(0,20), n_0^{q,i} \sim \mathcal{U}(0,L)$ 
4   set weight  $w_0^i = \frac{1}{N}$ 
5 end
6 % the sampling step for any time  $k \geq 1$ 
7 for  $i = 1 : N$  do
8   generate the  $i$  th particle  $S_k^i$  at time step  $k$ , by running
   the  $M/M/1$  single server queuing simulation for one
   time step (i.e.,  $\Delta T$ ) with initial state  $S_{k-1}^i$ 
9   add system noises to the newly generated state
    $S_k^i = \{\lambda_k^i, \mu_k^i, n_k^{q,i}\}$ :
    $\lambda_k^i = \lambda_{k-1}^i + v_\lambda, v_\lambda \sim \mathcal{N}(0, \lambda_{k-1}^i/10)$ 
    $\mu_k^i = \mu_{k-1}^i + v_\mu, v_\mu \sim \mathcal{N}(0, \mu_{k-1}^i/10)$ 
    $n_k^{q,i} = n_{k-1}^{q,i} + v_n, v_n \sim \mathcal{N}(0, n_{k-1}^{q,i}/10)$ 
11  compute weight:  $w_k^i = p(m_k^o | S_k^i) \times w_{k-1}^i$ 
12 end
13 normalize the weights, and denote them as  $\{S_k^i, w_k^i\}_{i=1}^N$ 
14 % the resampling step
15 resample  $\{S_k^i, w_k^i\}_{i=1}^N$  using the standard resampling method,
   which samples particles in proportion to their weights, and
   the resampled results are again denoted as  $\{S_k^i, w_k^i\}_{i=1}^N$ 
16 for  $i = 1 : N$  do
17    $w_k^i = \frac{1}{N}$ 
18 end
19 % the estimation step
20 estimate the system state at time step  $k$ :
    $\hat{S}_k = \sum_{i=1}^N S_k^i \cdot w_k^i = \frac{1}{N} \sum_{i=1}^N S_k^i$ 

```

experimental settings, the distance correlation $dCor^{42,43}$ is used to measure the association between the ground truth state S and the estimated state \hat{S} :

$$dCor(S, \hat{S}) = \frac{dCov(S, \hat{S})}{\sqrt{dVar(S) \cdot dVar(\hat{S})}} \in [0, 1]$$

where S is the state vector recorded for all DA steps during the simulation of the ground truth *QueuingModel*, and \hat{S} is the estimated state vector by the DA *QueuingModel* for all DA steps. $dCor$ is measured for each state variable (i.e., we calculate $dCor$ for three state variables: λ , μ , and n^q). The overall distance correlation of the estimation is the mean of individual distance correlations.

3. Scenarios, sensitivity analysis, and discussions

The $M/M/1$ single server queue is a simple DES. When the simulation replications are run in parallel, the total execution time including (re)sampling, model initialization, estimation, and data logging can be completed faster than

real-time, but the minimal execution time rounded up (for an iteration) is 0.5 s. The time interval (ΔT) of DA thus started from 0.5 s, and varied to 1, 1.5 s, ... to explore the response (Note that ΔT in principle can be at an arbitrary length if it is not shorter than the execution time.). The number of particles (N) ranged from 10 to 2000, and the levels of measurement errors, i.e., ε and ε' , from zero to low till high. To imitate the second-order dynamics³⁰ in the system, we created two events of sudden stochastic changes of λ and μ in the ground truth queuing system. They happened at the time steps of 15 and 30 s. The DA experiments each lasted 50 s.

In the following, the results regarding ΔT and N are first presented as they produce related effects on computational cost and estimation accuracy. Since computational resource is often limited in practice, experiments are also made to show the trade-off between the two. The second part of this section compares the effect of actual measurement errors (ε) with perceived measurement errors (ε') and their differences.

3.1. Time interval and number of particles

In this set of experiments, the time interval ΔT of assimilating measurement data varied from 1 to 5 s, while the other conditions remained constant at $N = 1000$ and $\varepsilon = \varepsilon' = 1$. The interval started from 1 s which is rounded up from the minimum model execution time. The interval of 5 s is where the estimation accuracy is still over 0.5; above 5 s, the accuracy becomes lower than 0.5. Figure 3 shows the outcomes with boxplots of the estimation accuracy $dCor$ (y -axis) ordered along different ΔT (x -axis) in a 0.5-s interval. Note that $dCor$ is calculated using the state vectors recorded at all DA steps. This is also the case for Figures 4–8.

The three horizontal bars (from bottom to top) of each box represent the first, second and third quartiles of a data set (The second bar indicates the median.). The dot indicates the mean. The whiskers (i.e., the two vertical lines outside the box) extend to 1.5 times the inter quartile range (IQR). The upper whisker stops at the largest value smaller than 1.5 IQR above the third quartile; the lower whisker at the smallest value greater than 1.5 IQR below the first quartile. The data points beyond the whiskers are considered outliers and are plotted as individual circles (This applies to all boxplots in this paper.).

Figure 3 shows a convex decreasing trend between ΔT and $dCor$; the slope of the curve decreases. Note that we also performed runs with $\Delta T > 5$, the curve extends further and $dCor$ shows medium and weak correlation (< 0.5) with this experimental setup. It is expected that when DA is more frequent (i.e., ΔT is small), the estimation accuracy $dCor$ increases significantly with narrower IQR. The accuracy spread is heavily skewed, which is not surprising because PF use Monte Carlo samples. The samples with

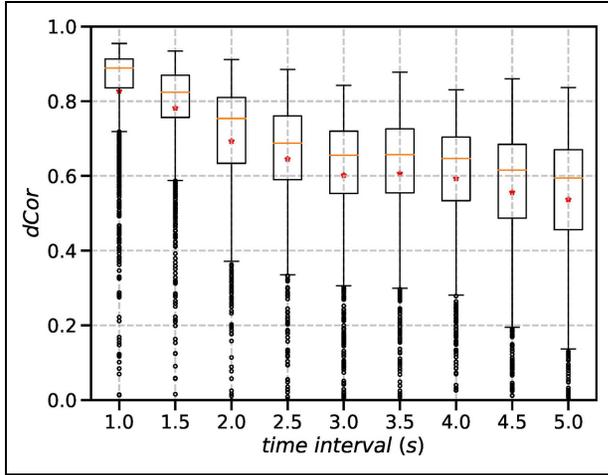


Figure 3. The effect of time interval $\Delta T \in [1, 5]$ on estimation accuracy $dCor$ ($N = 1000, \varepsilon = \varepsilon' = 1$). The construction of a boxplot: the three horizontal bars (from bottom to top) of each box represent the 1st, 2nd, and 3rd quartiles (Q1, Q2, Q3). The dot (in or right below the box) indicates the mean, and the Q2 bar is the median. The whiskers extend to $1.5 \times$ the inter quartile range ($IQR = Q3 - Q1$). The upper whisker stops at the largest value smaller than $1.5 \times IQR$ above Q3; the lower whisker stops at the smallest value greater than $1.5 \times IQR$ below Q1. Beyond the whiskers, the data points are considered as outliers and are plotted as individual circles.

negligible weights (i.e., low probabilities) would be replaced by resampling.

In the next set of experiments, the number of particles N varied from 10 to 2000, while the other conditions were set constant at $\Delta T = 1$ and $\varepsilon = \varepsilon' = 1$. Figure 4(a) shows the results of $N \in [10, 100]$ with increment of 10, and

Figure 4(b) shows $N \in [100, 2000]$ with increment of 100. With more particles used in DA, the estimation accuracy $dCor$ (mean and median) forms a concave increasing trend. This is also expected. When N exceeds around 50, the slope of increase gradually decreases till the accuracy stagnates. The Tukey test (confidence interval (CI) = 95%) was performed to compare the difference of $dCor$ with $N = 50, 100$ and larger numbers of particles. The results show that the increase of N above 400 in these experiments is no longer effective in improving estimation accuracy.

To understand the relation between ΔT and N with respect to $dCor$, additional experiments were performed where ΔT and N varied simultaneously. Each experiment used $(\Delta T, N)$ that satisfies $\{(\Delta T, N) | \Delta T \in \{0.5, 1, 1.5, \dots, 5\}\}$ and $N \in \{500, 1000, 1500, 2000\}$. The results are plotted in Figure 5, where each dot represents one DA experiment. The x-axis shows the total number of simulation runs (R) in an experiment; $R = T/\Delta T \times N$, where T is the length of each run (which is 50 s). For example, when an experiment had $\Delta T = 2$ and $N = 1000$, then $R = 50/2 \times 1000 = 25,000$.

Note that, in our experiments, when the number of simulation runs (R) is higher, the computational cost is proportionally higher. Figure 5(a) shows R on a linear scale and Figure 5(b) on a log scale. The y-axis shows $dCor$ of each experiment. The size of a dot corresponds to the number of particles N and the color of a dot (blue to red) corresponds to the time interval ΔT .

In Figure 5(a), different shades of large blue dots span a large horizontal area. Those are the experiments with short intervals and high numbers of particles as these conditions resulted in high numbers of runs. Consequently, they have high $dCor$ values. Since $\Delta T = 0.5$ is the shortest interval, low values of N (small dots) cannot reach the far end of the

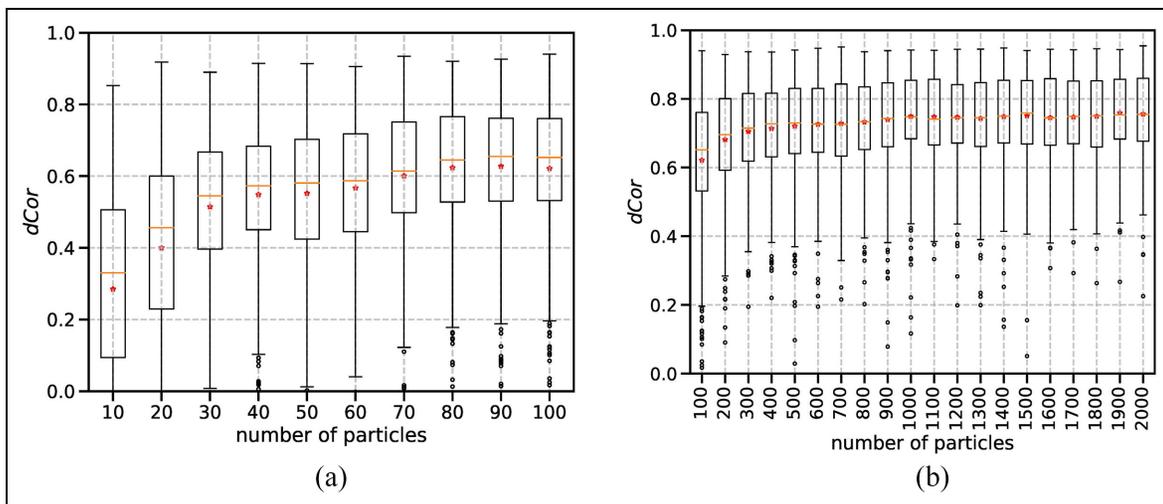


Figure 4. The effect of number of particles N on estimation accuracy $dCor$ ($\Delta T = 1, \varepsilon = \varepsilon' = 1$): (a) $N \in [10, 100]$ and (b) $N \in [100, 2000]$.

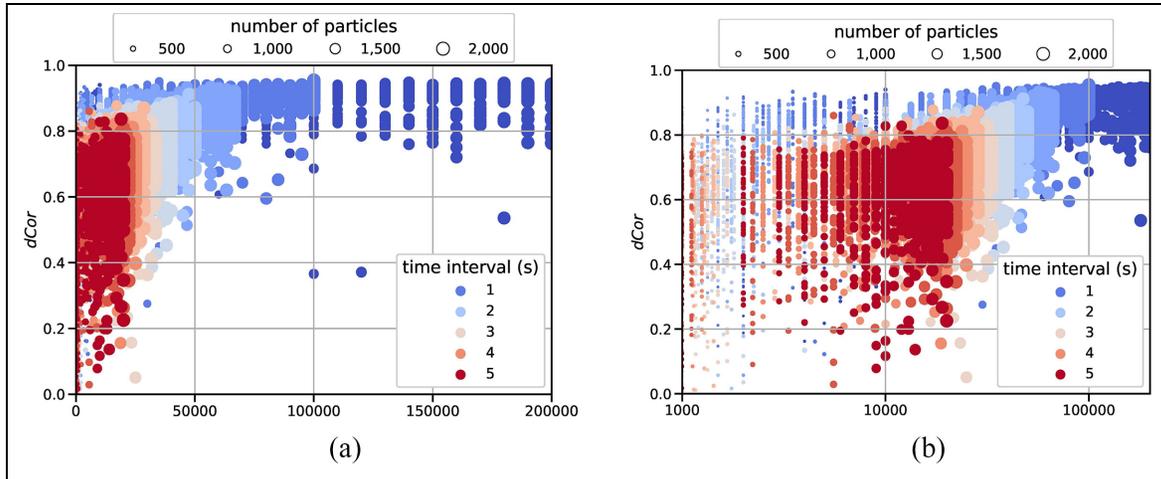


Figure 5. The effect of time interval $\Delta T \in \{0.5, 1, 1.5, \dots, 5\}$ and number of particles $N \in \{500, 1000, 1500, 2000\}$ on the estimation accuracy $dCor$ ($\varepsilon = \varepsilon' = 1$). A higher number of simulation runs (R) indicates higher computational cost: (a) number of simulation runs R on a linear scale and (b) number of simulation runs R on a log scale.

x-axis ($R = 50,000$ when $\Delta T = 0.5$ and $N = 500$). Small dots are “cramped” in the narrow left portion of Figure 5(a).

To make the small dots more visible, Figure 5(b) shows the x-axis on a log scale such that the part below $R = 50,000$ is stretched out. Given any vertical alignment (e.g., $R = 10,000$), small and large dots always have complementary colors. Different sizes of dots never have the same color along the same R value. More bluish smaller dots are at the back; more reddish larger dots are at the front. The dots located at the upper horizontal side of the plot are the experiments with higher $dCor$. The experiments with high numbers of particles can be better seen in Figure 5(a) while the ones with low numbers of particles can be better seen in Figure 5(b).

The results show that the estimation accuracy improves ($dCor \rightarrow 1$) when N increases (smaller to larger dots) and ΔT decreases (reddish to bluish dots) because more simulation runs are executed. The blue dots are closer to the top edge than the red dots, meaning that shorter time intervals do better than longer ones, which is expected. Also note that there are many large red dots in Figure 5(b), where $R > 10,000$. These dots did not perform much better than the smaller red dots at $R < 10,000$. In these experiments, more particles did not bring better performance. This confirms the result shown in Figure 4. In addition, those large red dots ($R > 10,000$) also performed worse than many smaller blue dots at $R < 10,000$. In these experiments, shorter ΔT brought better $dCor$ than more particles N even when the computational cost was higher in the latter case. This means that when ΔT is sufficiently short, good estimation accuracy can be achieved even though not many particles are used.

To summarize, while the number of particles is positively correlated and the time interval is negatively

correlated to the estimation accuracy in DA, the accuracy is more constrained by the choice of time interval than by the number of particles. This implies that given limited computational resources in DA applications, once the number of particles is sufficiently large, more computational resources can be allocated to shorten the time interval to improve the estimation accuracy.

3.2. Actual level and perceived level of measurement errors

The experiments presented in this section set the actual level of measurement errors ε from zero (0) to low (1), medium (2) till high (3). The perceived level of measurement errors ε' varied in a similar way and is explained in the corresponding experiments.

The first set of experiments varied the actual level of measurement errors $\varepsilon \in [0, 3]$, while the perceived level of measurement errors ε' remained constant ($\varepsilon' = 1$), and the other conditions were set at $\Delta T = 1$ and $N = 400$. Figure 6 shows that in this case, the estimation accuracy $dCor$ forms a concave downward decreasing trend with slight wider IQR. When the actual level of measurement errors is zero or low, the low perceived level of measurement errors yields similar results, but the estimation accuracy decreases when the actual error level is worse than the perceived level.

The second set of experiments changed the perceived level of measurement errors $\varepsilon' \in [1, 5]$, low (1), medium (2), high (3), higher (4), highest (5), while the actual level of the measurement errors remained constant ($\varepsilon = 1$), and the other conditions were again set at $\Delta T = 1$ and $N = 400$. Figure 7 shows the results with a slight concavity but reveals no clear pattern of the estimation accuracy

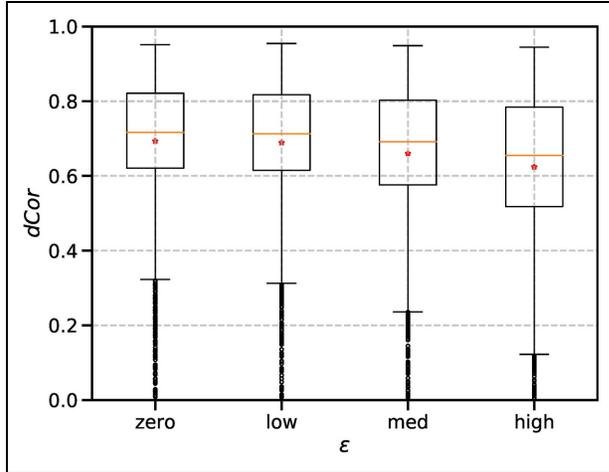


Figure 6. The effect of the actual level of measurement errors $\varepsilon \in \{\text{zero}, \text{low}, \text{medium}, \text{high}\}$ on the estimation accuracy $dCor$ when the perceived level of errors ε' is low ($\Delta T = 1, N = 400$).

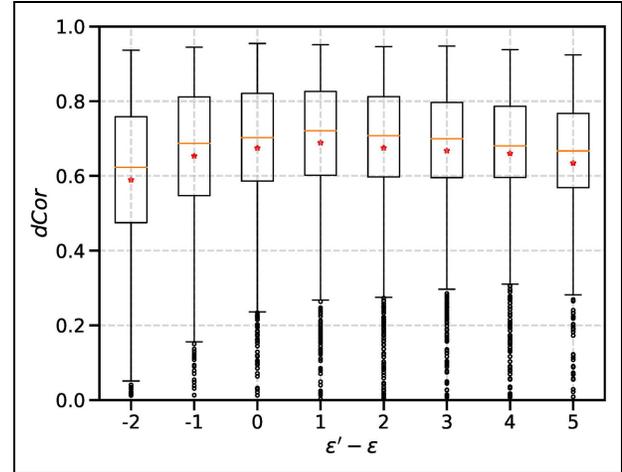


Figure 8. The effect of the difference between perceived level $\varepsilon' \in \{\text{low}, \text{medium}, \text{high}, \text{higher}, \text{highest}\}$ and actual level of measurement errors $\varepsilon \in \{\text{zero}, \text{low}, \text{medium}, \text{high}\}$ on the estimation accuracy $dCor$ ($\Delta T = 1, N = 400$).

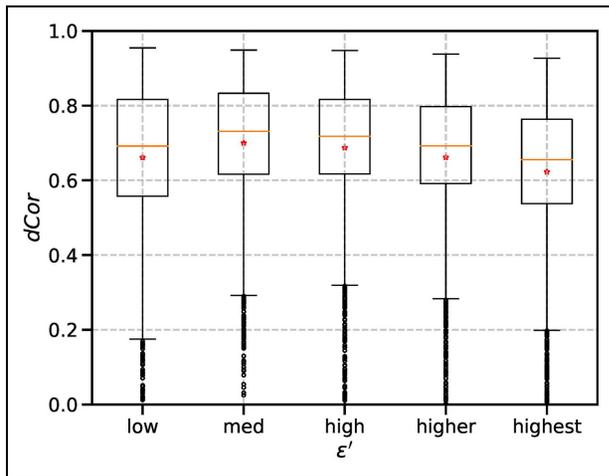


Figure 7. The effect of the perceived level of measurement errors $\varepsilon' \in \{\text{low}, \text{medium}, \text{high}, \text{higher}, \text{highest}\}$ on estimation accuracy $dCor$ when the actual level of errors ε is low ($\Delta T = 1, N = 400$).

$dCor$ in relation with the perceived level of measurement errors. However, the $dCor$ boxes of *medium* and *high* suggest better accuracy. Note that the actual level of errors is *low* in this case. Can this mean that the discrepancy between the actual level and perceived level of measurement errors matters for the estimation accuracy? We investigated this in the next set of experiments.

The difference between ε and ε' was experimented by sweeping $\varepsilon \in \{0, 1, 2, 3\}$ and $\varepsilon' \in \{1, 2, 3, 4, 5\}$, where the other conditions remain constant at $\Delta T = 1$ and $N = 400$. This resulted in 20 combinations of experimental setup, each of which used 400 particles. Figure 8 shows the results that are grouped by the difference between the

perceived level (ε') and the actual level (ε) of measurement errors such that $x = \varepsilon' - \varepsilon \in \{-2, -1, 0, 1, 2, 3, 4, 5\}$, plotted along the x -axis. A negative value of x thus indicates under estimation and a positive value of x indicates over estimation of measurement errors. The results show again a slight concave downward curve as in Figure 7. The curve is at its highest point where $x = 1$. The slope left to it is steeper than the one on the right. This suggests that under estimation of measurement errors ($x < 0$) leads to lower estimation accuracy $dCor$ than over estimation ($x > 0$). Perfect knowledge about measurement errors ($x = 0$) does not necessarily result in better estimation accuracy while slight over estimation ($x = 1$) leads to more accurate estimation results than that with perfect knowledge. In the experiments where $x > 1$, the estimation accuracy gradually decreases again, but it is no worse than the same levels of under estimation. In addition, the estimation accuracy $dCor$ has a narrower IQR when over estimating measurement errors than under estimating them, which is often a desired feature in DA.

To further illustrate the effect caused by misalignment of the perceived measurement errors with the actual level of errors, we present and discuss another experiment that compares two cases: (a) perfect knowledge about measurement errors ($x = 0$) and (b) slight over estimation of measurement errors ($x = 1$). In both cases, the actual level of measurement errors is *low* ($\varepsilon = 1$), the time interval $\Delta T = 2$, and the number of particles $N = 1300$. Case (a) has a *low* perceived level ($\varepsilon' = 1$) of measurement errors, which is the same as the actual level, while case (b) has a *medium* perceived level ($\varepsilon' = 2$) of measurement errors, which represents a slight over estimation of measurement errors. As shown in Figure 9, these two cases

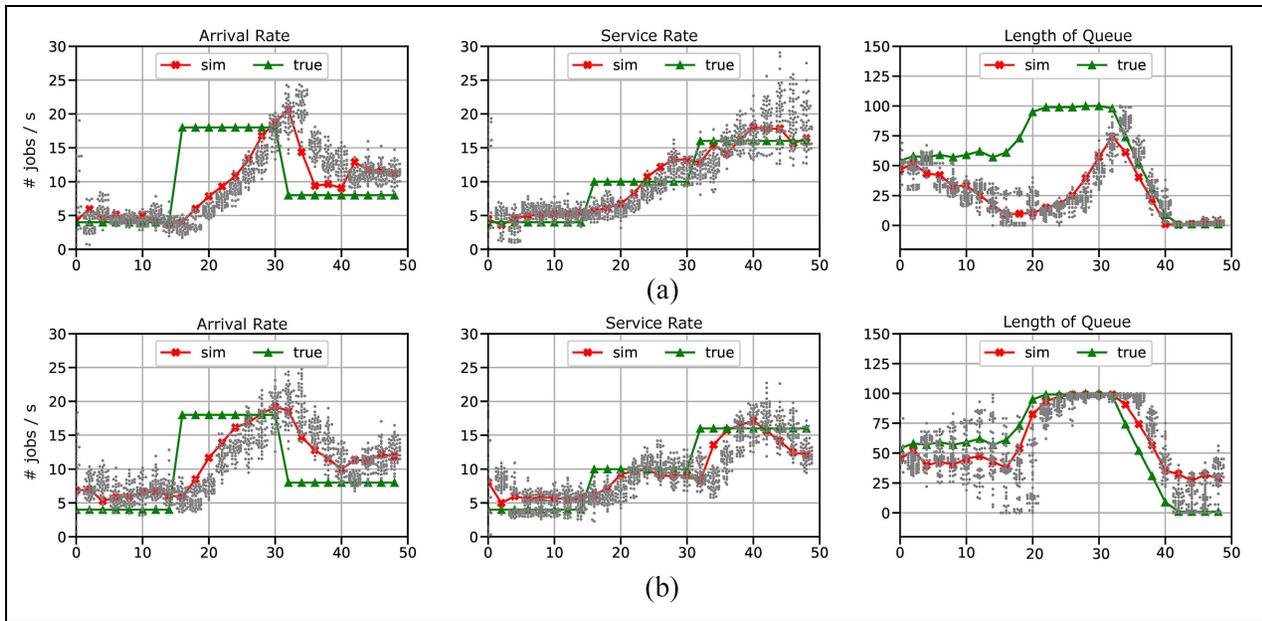


Figure 9. The effect of the different perceived levels of measurement errors on the estimation accuracy ($\Delta T = 2, N = 1300$): (a) accurate estimation of measurement errors ($\varepsilon = \varepsilon' = 1$) and (b) over estimation of measurement errors ($\varepsilon = 1, \varepsilon' = 2$).

performed distinctly in estimating the queue length n^q in the simulation after the sudden change of the arrival rate λ and the service rate μ at time $t = 15$ in the “real system.” In case (a), the simulation could not closely follow the trajectory of the queue n^q in the first 15 s. Once the change occurred at $t = 15$, n^q diverged more and could catch up with the system state again after 10 DA iterations. In case (b), the simulation can follow the change more responsively.

The difference in response time in the two cases can be explained by the spread of particles, which are depicted as gray dots in Figure 9. Note that the vertical spread of particles in case (a) is narrower than that in case (b). In case (a), only a few particles having a small deviation from the measurement can “survive” throughout the experiment, while the particles located apart are discarded. Consequently, abrupt changes in the system are not detected rapidly because of the restricted spread of particles. In case (b), as the particles spread wider, the aggregated result can quickly converge to the true value after sudden changes. Thus, more widespread particles are more tolerant and show more a responsive estimation in detecting capricious system changes.

Given these observations in the experiments, we conclude that a pessimistic view on measurement errors brings advantages over an optimistic view on measurement errors with respect to the accuracy of DA results. In addition, a slight pessimistic view on measurement errors leads to better estimation accuracy than an accurate view on

measurement errors in the experiments. This is rarely an intuitive choice in DA experimental setups.

4. Conclusion and future work

The experiments presented in this paper quantitatively studied the effect of three common and critical experimental conditions of PF-based sequential DA for DES: the time interval of assimilating measurement data, the number of particles, and the level of measurement errors (or noises). An identical-twin experimental scheme (of an $M/M/1$ single server queuing model) was adopted to model the ground truth and to predict the system states. This way, the true state of the system, the measurement errors, and the estimated system state can be quantified in a controlled manner. We evaluated the estimation accuracy of the DES states. The results of the sensitivity analysis can thus be interpreted in relative terms contrasting different experimental setups of the DA process. The main findings of our experiments are as follows.

The time interval of assimilating measurement data has a negative correlation with the estimation accuracy of system states. Although this is expected, it is not always true as reported by some studies (see section 1). More frequent assimilation of measurement data is effective to improve the estimation accuracy and the responsiveness of the estimation results. Although the number of particles has in general a positive correlation with the estimation accuracy, increasing the number of particles is ineffective in

improving accuracy beyond a certain level. Notably, good estimation accuracy can be achieved even though not many particles are used if the time interval is sufficiently short. Since both decreasing the time interval and increasing the number of particles require more computation, the former can be more cost effective when the number of particles is sufficiently large. With regard to measurement errors, in the experiments an over estimation of the level of measurement errors leads to more accurate estimation results than an under estimation. Under estimating the errors always produces worse state estimates. Interestingly, a correct perception of the measurement errors does not guarantee better state estimates. A slight over estimation of errors has better accuracy and more responsive model adaptation to system states than an accurate estimation of measurement errors. An exaggerated over estimation of errors, however, deteriorates the accuracy of state estimates.

Our work used a simple single server queuing model to explore conditions in PF-based DA applied to DES. The choice of a simple target system and simple scenarios has the advantage that thorough experiments can be performed with a high number of iterations and particles. The states of “real” and simulated systems can be easily compared. The work, nonetheless, demonstrates the usefulness and challenges of using PF-based DA for DES and points out a few interesting future directions.

The sensitivity quantification of the PF conditions investigated in this paper is specific to the target system and scenario setup of our experimental choices. In that regard, many uncertain (and/or stochastic) factors in DES models can be interesting to further investigate in order to understand better how to efficiently use PF for DES. For example, the level of system noises, the number of estimated state variables, and their relations to the level of measurement noises. We conjecture that the “regular” stochasticity captured by probabilistic distributions in DES models can be covered by the number of particles and their disperse in PF. The “irregular” uncertainty that falls out of the model descriptive power can be compensated by the power of sequential DA. With that in mind, it is possible to smartly balance out the number of particles (or even to calibrate the probabilistic distributions in DES models) and the time intervals to produce acceptable DA results for DES. Since the computational demand to apply PF-based DA to discrete systems will remain a challenge (in the foreseeable future), a better understanding of the interplay between the time interval and particles is key to promoting wider use of DA in social and socio-technical applications. Furthermore, an event-based DA method can be a novel approach unique for DA applications in DES. This means PF can be dynamically adjusted for, e.g., the time intervals, number of particles, and perception of measurement errors, according to certain event-triggers (which is relatively simple for a DES model to implement)

tailored to the application in order to obtain more cost effective state estimation performance.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ORCID iD

Yilin Huang  <https://orcid.org/0000-0002-4396-1990>

References

1. Mathieu PP and O’Neill A. Data assimilation: from photon counts to earth system forecasts. *Remote Sens Environ* 2008; 112: 1258–1267, <https://www.sciencedirect.com/science/article/pii/S0034425707003240>
2. Zhang Z and Moore JC (eds). Chapter 9—data assimilation. In: *Mathematical and physical fundamentals of climate change*. Amsterdam: Elsevier, 2015, pp. 291–311.
3. Routray A, Osuri KK, Pattanayak S, et al. Chapter 11. Introduction to data assimilation techniques and ensemble Kalman filter. In: Mohanty UC and Gopalakrishnan SG (eds) *Advanced numerical modeling and data assimilation techniques for tropical cyclone prediction*. Berlin: Springer, 2016, pp. 307–330.
4. Ma C, Wang T, Mizzi AP, et al. Multi-constituent data assimilation with WRF-Chem/DART: potential for adjusting anthropogenic emissions and improving air quality forecasts over eastern China. *J Geophys Res* 2019; 124: 7393–7412.
5. Ren L, Nash S and Hartnett M. Data assimilation with high-frequency (HF) radar surface currents at a marine renewable energy test site. In: Soares CG (ed.) *Renewable energies off-shore*. Boca Raton, FL: CRC Press, 2015, pp. 189–193.
6. Tran AP, Vanclooster M and Lambot S. Improving soil moisture profile reconstruction from ground-penetrating radar data: a maximum likelihood ensemble filter approach. *Hydrol Earth Syst Sci* 2013; 17: 2543–2556.
7. Shuwen Z, Haorui L, Weidong Z, et al. Estimating the soil moisture profile by assimilating near-surface observations with the ensemble Kalman filter (EnKF). *Adv Atmos Sci* 2005; 22: 936–945.
8. Lloyd DJB, Santitissadeekorn N and Short MB. Exploring data assimilation and forecasting issues for an urban crime model. *Eur J Appl Math* 2016; 27: 451–478.
9. Ward JA, Evans AJ and Malleson NS. Dynamic calibration of agent-based models using data assimilation. *R Soc Open Sci* 2016; 3: 150703, <https://royalsocietypublishing.org/doi/abs/10.1098/rsos.150703>
10. Hu X and Wu P. A data assimilation framework for discrete event simulations. *ACM Trans Model Comput Simul* 2019; 29: 171–1726.
11. Arulampalam MS, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process* 2002; 50: 174–188.
12. Gillijns S, Mendoza O, Chandrasekar J, et al. What is the ensemble Kalman filter and how well does it work? In: *Proceedings of the 2006 American control conference*.

- Minneapolis, MN, 14–16 June 2006, pp. 4448–4453. New York: IEEE.
13. Evensen G. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dyn* 2003; 53: 343–367.
 14. Yuan Y. *Lagrangian multi-class traffic state estimation*. PhD Thesis, Delft University of Technology, Delft, 2013.
 15. Djurić PM, Kotecha JH, Zhang J, et al. Particle filtering. *IEEE Signal Process Mag* 2003; 20: 19–38.
 16. Cho KH, Pachepsky Y, Ligaray M, et al. Data assimilation in surface water quality modeling: a review. *Water Res* 2020; 186: 116307, <https://www.sciencedirect.com/science/article/pii/S0043135420308435>
 17. Mao J, Lee JH and Choi K. The extended Kalman filter for forecast of algal bloom dynamics. *Water Res* 2009; 43: 4214–4224, <https://www.sciencedirect.com/science/article/pii/S0043135409003819>
 18. Valdes-Abellan J, Pachepsky Y, Martinez G, et al. How critical is the assimilation frequency of water content measurements for obtaining soil hydraulic parameters with data assimilation? *Vadose Zone J* 2019; 18: 180142, <https://acsess.onlinelibrary.wiley.com/doi/abs/10.2136/vzj2018.07.0142>
 19. Kim S, Seo DJ, Riazi H, et al. Improving water quality forecasting via data assimilation—application of maximum likelihood ensemble filter to HSPF. *J Hydrol* 2014; 519: 2797–2809, <https://www.sciencedirect.com/science/article/pii/S0022169414007331>
 20. Javaheri A, Babbar-Sebens M, Miller RN, et al. An adaptive ensemble Kalman filter for assimilation of multi-sensor, multi-modal water temperature observations into hydrodynamic model of shallow rivers. *J Hydrol* 2019; 572: 682–691, <https://www.sciencedirect.com/science/article/pii/S0022169419302318>
 21. Wang S, Flipo N and Romary T. Oxygen data assimilation for estimating micro-organism communities' parameters in river systems. *Water Res* 2019; 165: 115021, <https://www.sciencedirect.com/science/article/pii/S004313541930795X>
 22. van Leeuwen PJ. Particle filtering in geophysical systems. *Mon Weather Rev* 2009; 137: 4089–4114.
 23. Wang M and Hu X. Data assimilation in agent based simulation of smart environment. In: *Proceedings of the 1st ACM SIGSIM conference on principles of advanced discrete simulation*, Montreal, QC, Canada, 19–22 May 2013, pp. 379–384. New York: Association for Computing Machinery.
 24. Wang M and Hu X. Data assimilation in agent based simulation of smart environments using particle filters. *Simul Model Pract Theory* 2015; 56: 36–54.
 25. Huang Y, Warnier M, Brazier F, et al. Social networking for smart grid users—a preliminary modeling and simulation study. In: *Proceedings of 2015 IEEE 12th international conference on networking, sensing and control*, Taipei, 9–11 April 2015, pp. 438–443. New York: IEEE.
 26. Huang Y and Warnier M. Bridging the attitude-behaviour gap in household energy consumption. In: *Proceedings of IEEE PES innovative smart grid technologies Europe*, Bucharest, 29 September–2 October 2019, pp. 1–5. New York: IEEE.
 27. Xie X, van Lint H and Verbraeck A. A generic data assimilation framework for vehicle trajectory reconstruction on signalized urban arterials using particle filters. *Transp Res Part C Emerg Technol* 2018; 92: 364–391.
 28. Wang S, Xie X and Ju R. A mesoscopic traffic data assimilation framework for vehicle density estimation on urban traffic networks based on particle filters. *Entropy* 2019; 21: 1–20.
 29. Banks J. *Handbook of simulation: principles, methodology, advances, applications, and practice*. New York: John Wiley & Sons, 1998.
 30. Huang Y, Seck MD and Verbraeck A. Towards automated model calibration and validation in rail transit simulation. In: *Proceedings of the 2010 international conference on computational science*, Amsterdam, 31 May–2 June 2010, pp.1253–1259. Amsterdam: Elsevier.
 31. Hu X. Dynamic data driven simulation. *SCS M&S Mag* 2011; II: 16–22.
 32. Huang Y, Seck MD and Verbraeck A. Component based light-rail modeling in discrete event systems specification (DEVS). *Simulation* 2015; 91: 1027–1051.
 33. Huang Y, Verbraeck A and Seck MD. Graph transformation based simulation model generation. *J Simul* 2016; 10: 283–309.
 34. Keller N and Hu X. Towards data-driven simulation modeling for mobile agent-based systems. *ACM Trans Model Comput Simul* 2019; 29: 1–26.
 35. Xue H, Gu F and Hu X. Data assimilation using sequential Monte Carlo methods in wildfire spread simulation. *ACM Trans Model Comput Simul* 2012; 22: 231–2325.
 36. Xie X. *Data assimilation in discrete event simulations*. PhD Thesis, Delft University of Technology, Delft, 2018.
 37. Malleon N, Minors K, Kieu LM, et al. Simulating crowds in real time with agent-based modelling and a particle filter. *J Artif Soc Soc Simul* 2020; 23: 3.
 38. Xie X and Verbraeck A. A particle filter-based data assimilation framework for discrete event simulations. *Simulation* 2019; 95: 1027–1053.
 39. Cho Y, Huang Y and Verbraeck A. Strategic use of data assimilation for dynamic data-driven simulation. In: *Computational science—ICCS 2020—20th international conference proceedings*, Amsterdam, 3–5 June 2020, pp. 31–44. Berlin: Springer.
 40. Ancker CJ and Gafarian AV. Some queuing problems with balking and reneging—I. *Oper Res* 1963; 11: 88–100.
 41. Zeigler BP, Praehofer H and Kim TG. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. New York: Wiley, 2000.
 42. Bickel PJ and Xu Y. Discussion of Brownian distance covariance. *Ann Appl Stat* 2009; 3: 1266–1269.
 43. Székely GJ, Rizzo ML, Bakirov NK, et al. Measuring and testing dependence by correlation of distances. *Ann Stat* 2007; 35: 2769–2794.

Author biographies

Yilin Huang is Assistant Professor in Modeling and Simulation (M&S) at the Faculty of Technology, Policy and Management, Delft University of Technology (TU Delft), The Netherlands. She received her PhD from TU

Delft in 2013 (Automated Simulation Model Generation). From 2014 to 2016, she was a postdoc and researched on prosumer behavior modeling and demand-side management in the smart grids. Her research focuses on M&S methodologies such as data-driven modeling, multi-paradigm modeling, interoperability and composability of dynamic models, and their implications in complex socio-technical environments. The application of her research ranges from transportation and logistics, smart grid, sustainable consumption to humanitarian innovations.

Xu Xie is Associate Professor at the College of Systems Engineering, National University of Defense Technology, China. He received his PhD from TU Delft in 2018 (Data Assimilation in Discrete Event Simulations). His research interests cover dynamic data driven simulations, data assimilation, and distributed simulation.

Yubin Cho is a Data Scientist at Corporate Marketing Insight team of Samsung Electronics Benelux. He received a bachelor's degree in mechanical engineering from Seoul National University, South Korea and a

master's degree in system engineering from Delft University of Technology, The Netherlands. He has worked in aviation, transportation, and marketing field, mainly with modeling, simulation, and big data analysis. He is interested in modeling of complex system, behavior analysis, and real-time simulation.

Alexander Verbraeck (MSc in applied mathematics 1987; PhD in logistics 1991) is a full professor at Delft University of Technology, Faculty of Technology, Policy and Management, Multi Actor Systems Department. His research focuses on modeling and simulation, especially in heavily distributed environments and using real-time data. Examples of research on these types of simulations are real-time decision-making, interactive gaming using simulations, data-driven simulation, and digital twins. The major application domain for research is logistics and transportation, for which simulation models and serious games are developed in a number of government and industry-funded projects.