

# Model Order Reduction for Water Quality Dynamics

Shen Wang<sup>†</sup>, Ahmad F. Taha<sup>†,\*</sup>, Ankush Chakrabarty<sup>‡</sup>, Lina Sela<sup>♣</sup>, and Ahmed Abokifa<sup>◇</sup>

**Abstract**—A state-space representation of water quality dynamics describing disinfectant (e.g., chlorine) transport dynamics in drinking water distribution networks has been recently proposed. Such representation is a byproduct of space- and time-discretization of the PDE modeling transport dynamics. This results in a large state-space dimension even for small networks with tens of nodes. Although such a state-space model provides a model-driven approach to predict water quality dynamics, incorporating it into model-based control algorithms or state estimators for large networks is challenging and at times intractable. To that end, this paper investigates model order reduction (MOR) methods for water quality dynamics. The presented investigation focuses on reducing state-dimension by orders of magnitude, the stability of the MOR methods, and the application of these methods to model predictive control.

**Index Terms**—Water distribution network, water quality model, state-space representation, model order reduction, model predictive control.

## I. INTRODUCTION AND PAPER CONTRIBUTIONS

Water quality models can be expressed in the form of a state-space representation [1] representing the numerical solution of the PDEs describing the spatiotemporal evolution of disinfectant concentration (e.g., chlorine) in various network elements in drinking water distribution networks (WDN). Unfortunately, even for small-to-midsize networks, the dimension (or order) of the state-space models can reach  $10^4$  or  $10^6$ , due to ensuring high fidelity in model predictions.

This is due to discretization methods used for solving the partial differential equations (PDE) that describe the chlorine transport in pipes. Very high resolution in discretization yields improved accuracy in the control-oriented model but results in high state-space dimension, leading to heavy computational burden during simulation and difficulty for use in controller design [2]. This implies that high-accuracy models, though effective for predicting system dynamics, are not amenable to controller or estimator design for large-scale networks—especially in the presence of state or input constraints. To that end, model order reduction (MOR) is necessary to derive a compact model for fast simulation and efficient synthesis of controllers and state estimators.

A *full-order model* is referred to as an exact or near-exact model to describe water quality dynamics and usually is high in dimension. The motivation of MOR is to reduce the full-order model to a *reduced-order model* that has a much smaller number of states or order without significantly decreasing model accuracy while maintaining input-output relationships and retaining certain properties of the system such as controllability and observability. To this end, the high-dimensional state vector from the full-order model is projected onto a lower-dimensional subspace within which the model accuracy and certain properties are not significantly compromised. The MOR method has been widely applied in computational fluid dynamics [3]–[5], and other fields such as computational electromagnetics, as well as in micro- and nano-electro-mechanical systems design [6]. However, reduced-order models have never been explored for modeling water quality dynamics—a gap that is filled in this paper.

### A. Literature review of MOR algorithms

Various MOR algorithms have been developed in the rich literature of dynamic systems. The majority of these algorithms can be divided into two categories: singular value decomposition (SVD) based methods [3], [5], [7]–[10] and Krylov subspace methods or moment matching based methods [11]–[15]. There are also some studies combining SVD and Krylov methods [16]. In general, methods based on Krylov subspaces do not preserve important properties of the original system such as stability and passivity [4]. Hence, we are interested in SVD-based methods in which several methods can guarantee stability or controller designed related properties. Moreover, SVD-based approaches can be further divided into subcategories, for example, the methods using Hankel operators concepts [7], [8], the balanced realization theory [3], [9], [17], and proper orthogonal decomposition (POD) based methods [3], [5], [10], [18], [19]. This paper focuses on seeking proper SVD-based model reduction methods for the water quality dynamics represented linear discrete-time systems with a state-space presentation.

Moore [9] proposes the balanced truncation (BT) method which considers or balances controllability and observability Gramians. Unfortunately, the BT method becomes computationally intractable for large-scale systems (e.g.,  $10^4$  states or more). The POD method proposed by Sirovich [10] is tractable, but not as accurate as of the BT. To compute balancing transformations for high-order systems, Willcox [5] proposes a technique that combines the POD and concepts from balanced realization theory. However, when the number of outputs of a system is large, this method is intractable, and it has the risk of truncating states that are poorly observable yet very strongly controllable. As a result, the balanced POD (BPOD) [3] is proposed to address the same issues, and it combines the advantages

<sup>†</sup>Department of Electrical and Computer Engineering, The University of Texas at San Antonio, TX 78249. Emails: {shen.wang,ahmad.taha}@utsa.edu.

<sup>‡</sup>Research Scientist, Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. Email: achakrabarty@ieee.org

<sup>♣</sup>Department of Civil, Architecture, and Environmental Engineering, The University of Texas at Austin. Email: linasela@utexas.edu

<sup>◇</sup>Department of Civil, Materials, and Environmental Engineering, The University of Illinois Chicago. Email: abokifa@uic.edu

\*Corresponding author.

This work is partially supported by National Science Foundation under grants 1728629, 2015603, and 2015671.

of BT and POD methods successfully. That is, BPOD balances both controllability and observability and is tractable for a large-scale system, which results in the computational cost similar to POD and the accuracy similar to BT.

In the water systems research community, the reduced models are applied in simplifying networks in the context of hydraulics and the techniques adopted are different from the ones aforementioned in dynamic systems. That is, the reduced model achieves a high-fidelity representation for the complete network hydraulic, but greatly simplifies the computation. For example, a skeletonization method [20] removes nodes from a linearized network model through Gauss elimination, and reduces the network model while preserving the nonlinear characteristics of the original network model. Salomons [21] obtain a reduced model by the genetic algorithm. To meet the computational efficiency requirements of real-time hydraulic state estimation, Preis et al. [22] introduce a reduced model using a water system-aggregation technique. A methodology and application of a conjunctive hydraulic and water quality model for water distribution systems aggregation are presented in study [23]. In contrast, this paper focuses on a different research problem that reduces the order of water quality dynamics model, which has not been studied in the literature.

## B. Paper contributions

The major objective of the paper is to investigate the performance of water quality MOR algorithms, address their theoretical limitations, assess their computational performance, and evaluate their potential when applied within a model-based predictive control framework. The below list outlines in detail the contributions of the paper.

- 1) We present the first attempt to identify reduced-order models for water quality simulation with theoretical analysis: our case studies show that the reduced-order model produces nearly identical water quality simulations in comparison with the full-order models. The MOR algorithms are tested for different scales of networks in sense of accuracy, low computational burden, robustness to initial conditions, and potential to be obtained as a stable reduced-order model.
- 2) Classic MOR algorithms cannot ensure the stability of a reduced-order model, despite the stability of the underlying water quality dynamics. Thus, we propose two methods that guarantee that the reduced-order model is stable. The first method performs a stabilization process based on the standard MOR procedure while trying to maintain accuracy, and the second method simply adjusts a parameter in the standard MOR (i.e., BPOD) procedure. Moreover, the parameter adjustment has a clear physical interpretation for water quality dynamics, which is the largest travel time-step from booster stations to sensors.
- 3) The obtained stable reduced-order models are then utilized as predictive models in MPC for a WDN. Our case study shows that the application of MPC to the reduced-order model produces similar control effects compared with the application of MPC to the full-order model, while the computational burden is significantly reduced.

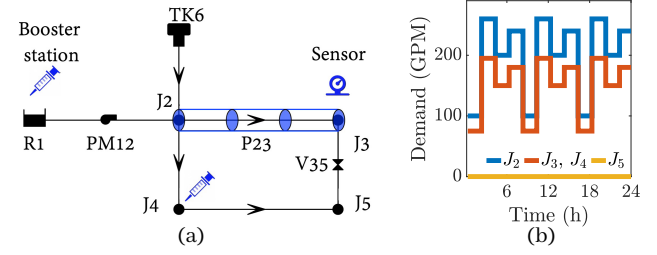


Fig. 1. (a) exemplar topology of a WDN (multi-input and multi-output system with two boosters (inputs) and one sensor (output) to inject and detect chlorine), and  $P23$  is split into three segments according to L-W scheme for illustration purpose; (b) demand profiles for all junctions during 24 hours.

The rest of the paper is organized as follows. Section II briefly describes control-oriented water quality modeling, then presents the full-order model in state-space form. The principle of MOR for the specific water quality modeling is given at the beginning of Section III, followed by the fundamental procedures of various MOR methods and the corresponding discussions in Section IV. Section V presents two methods to obtain a stable reduced-order model by adjusting the standard MOR procedures. Section VI presents case studies that corroborate our proposed approach. Limitations of our methods and future research directions are given in Section VII.

**Paper's Notation.** Boldface characters represent matrices and vectors:  $a$  is a scalar,  $\mathbf{a}$  is a vector, and  $\mathbf{A}$  is a matrix. Matrix  $\mathbf{I}$  denotes an identity matrix, whereas  $\mathbf{0}_{m \times n}$  denotes a zero matrix with size  $m$ -by- $n$ . The notation  $\mathbb{R}$  denotes the set of real numbers, and the notations  $\mathbb{R}^n$  and  $\mathbb{R}^{m \times n}$  denote a column vector with  $n$  elements and an  $m$ -by- $n$  matrix in  $\mathbb{R}$ . For any vector  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x}^\top$  is its transpose. For any two matrices  $\mathbf{A}$  and  $\mathbf{B}$  with same number of columns, the notation  $\{\mathbf{A}, \mathbf{B}\}$  denotes  $[\mathbf{A}^\top \mathbf{B}^\top]^\top$ . The norm of  $\mathbf{A}$  is  $\|\mathbf{A}\|_2$ . The  $i$ -th singular value of  $\mathbf{A}$  is denoted by  $\sigma_i$ ; a vector  $\boldsymbol{\sigma}(\mathbf{A})$  stands for all the singular values of  $\mathbf{A}$ ; the  $k$ -largest singular values of  $\mathbf{A}$  is denoted by vector  $\boldsymbol{\sigma}_k^\downarrow(\mathbf{A})$ . The  $i$ -th eigenvalue of  $\mathbf{A}$  is denoted by  $\lambda_i$ ;  $\boldsymbol{\sigma}(\mathbf{A})$  stands for all the eigenvalues of  $\mathbf{A}$ ; the  $k$ -largest eigenvalues of  $\mathbf{A}$  is expressed by vector  $\boldsymbol{\lambda}_k^\downarrow(\mathbf{A})$ .

## II. STATE-SPACE WATER QUALITY MODEL

In this section, we present the necessary background for the state-space model of water quality dynamics.

### A. Water quality dynamics

We model WDN by a directed graph  $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ . The set  $\mathcal{N}$  defines the nodes and is partitioned as  $\mathcal{N} = \mathcal{J} \cup \mathcal{T} \cup \mathcal{R}$  where  $\mathcal{J}$ ,  $\mathcal{T}$ , and  $\mathcal{R}$  are collection of junctions, tanks, and reservoirs. Let  $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$  be the set of links, and define the partition  $\mathcal{L} = \mathcal{P} \cup \mathcal{M} \cup \mathcal{V}$ , where  $\mathcal{P}$ ,  $\mathcal{M}$ , and  $\mathcal{V}$  represent the collection of pipes, pumps, and valves.

An example of a WDN graph is shown in Fig. 1a, and a disinfectant (such as chlorine) is added into the network with a proper mass and concentration by booster stations installed at the reservoir source  $R_1$  and the junction  $J_4$  to disinfect the WDN. Each junction in a WDN is a type of node that connects two links. They may consume water, that is, they have water

demands. During a specific period, demands from all junctions form a demand profile (see Fig. 1b) which is assumed as a priori for water quality simulations in water research. Moreover, all components are designed to meet a certain threshold of water quantity and quality. For example, the pump  $PM_{12}$  is designed to maintain the water pressure for all users (junctions). The concentration sensors are installed at specific junctions for measurement purposes (see the sensor installed at  $J_3$  in Fig. 1a), in this way the water system operators can obtain the concentrations around the network.

In the end, the water quality model can be used to represent the movement of all chemical and/or microbial species (contaminant, disinfectants, DBPs, metals, etc.) within a WDN as they traverse various components of the network. Specifically, the single-species interaction and dynamics of chlorine are considered in our paper, and it can be expressed using a state-space model. We first introduce the concepts and the corresponding methods that form the state-space model next.

In a WDN, the pipe model is represented by chlorine transport in differential pipe lengths by advection in addition to its decay due to reactions. Mathematically, for any pipe, the 1-D advection-reaction equation is given by a PDE. Although the PDE has no closed-form solution, the Lax-Wendroff scheme [24] can be adopted to discretize and solve it numerically. For example, pipe  $P_{23}$  in Fig. 1a is split into three segments for illustrative purpose according to the scheme. In practice, the number of segments for a pipe (denoted by  $s_L$ ) is decided by the flow velocity and the length of that pipe. After the discretization for all pipes, we define a vector  $\mathbf{c}^P(t)$  to collect all segments in all pipes at time  $t$ . Similarly, the concentrations in pumps and valves at time  $t$  are denoted by  $\mathbf{c}^M(t)$  and  $\mathbf{c}^V(t)$ . The  $\mathbf{c}^L(t) \triangleq \{\mathbf{c}^P(t), \mathbf{c}^M(t), \mathbf{c}^V(t)\}$  is a defined vector collecting all concentrations from links (i.e., pipes, pumps, and valves). For all nodes, we define  $\mathbf{c}^N(t) = \{\mathbf{c}^J(t), \mathbf{c}^R(t), \mathbf{c}^T(t)\}$  to collect the concentration from all junctions, reservoirs, and tanks.

### B. State-space representation

In general, the number of junctions, reservoirs, tanks, pipes, pumps, and valves is denoted by  $n_J, n_R, n_{TK}, n_P, n_M,$  and  $n_V$ . Hence, the number of states in nodes and links are  $n_N = n_J + n_R + n_{TK}$  and  $n_L = n_P \times s_L + n_M + n_V$ . Therefore,  $\mathbf{c}^N(t) \in \mathbb{R}^{n_N}$  and  $\mathbf{c}^L(t) \in \mathbb{R}^{n_L}$ .

Summarily, a water quality state-vector  $\mathbf{x}$  defining the concentrations of the disinfectant (chlorine) in the network at time  $t$  is denoted as  $\mathbf{x}(t) \triangleq \{\mathbf{c}^N(t), \mathbf{c}^L(t)\} \in \mathbb{R}^{n_x}$  and  $n_x = n_N + n_L$ . After applying the principle of conservation of mass, we can rewrite all component models (i.e., water quality model) using a discrete-time state-space model of the form

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (1a)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k), \quad (1b)$$

where  $\mathbf{x}(k) \in \mathbb{R}^{n_x}$  is the state vector, the vectors  $\mathbf{u}(k) \in \mathbb{R}^{n_u}$  and  $\mathbf{y}(k) \in \mathbb{R}^{n_y}$  are the system inputs and outputs (i.e.,  $n_u$  booster stations and  $n_y$  concentration sensors), and  $\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k),$  and  $\mathbf{D}(k)$  are time-varying system matrices. Note that state matrix  $\mathbf{A}(k)$  stands for the pure system dynamic (without any boosters); the element in input matrix  $\mathbf{B}(k)$  represents locations of booster stations and corresponding injected

flow rates that are both assumed as known; output matrix  $\mathbf{C}(k)$  indicates where the sensors are installed; feed-forward matrix  $\mathbf{D}(k)$  describes the relationship or path directly from inputs to outputs, for example, some booster stations are equipped with a built-in concentration sensor (i.e., both the booster station and the sensor are at the same location) to calculate the injection of chlorine accurately.

Note that these system matrices are usually updated from half hour to several hours according to the changing demands in demand profiles (e.g., demand changes every 2 hours in Fig. 1b). That is, during a specific period in the demand profile (demands at all junctions do not change), the system matrices remain the same, and the discrete-time linear time-varying (DT-LTV) system (1) degenerates into a discrete-time linear time-invariant (DT-LTI) system, that is

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2a)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k). \quad (2b)$$

Let us denote a generic  $n_x$ -th order DT-LTI system (2) as the tuple  $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$  for brevity. In this way, the DT-LTV system (1) comprises different tuples (DT-LTI systems). For example, DT-LTV models for 24 hours of the WDN in Fig. 1a can be represented by 12 different DT-LTI systems since the demand changes each two hours.

It is worthwhile to note that (i) the water quality model (1) is available in our recent published work [1], and the model is stable; (ii) after testing, each pipe is required to be split into at least 100 segments on average to guarantee the accuracy of discretizing PDEs. That is, the order  $n_x$  can easily reach  $10^4$  level for a mid-sized network with hundreds of pipes; (iii) all system matrices in this model are sparse, and the sparsity rate could be from 99% to 99.999% or higher. Moreover, the challenging MOR problem for a DT-LTV system is equivalent to solving multiple relatively easy MOR problems for DT-LTI systems, and the detail is presented next.

## III. MODEL ORDER REDUCTION

This section provides an overview of MOR principles for DT-LTI systems. For each specific MOR method, the corresponding procedure is summarized.

### A. Principle of MOR for DT-LTI system

The objective of MOR is to find a reduced-order model of order  $n_r \ll n_x$ , given by

$$\mathbf{x}_r(k+1) = \mathbf{A}_r\mathbf{x}_r(k) + \mathbf{B}_r\mathbf{u}(k) \quad (3a)$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}_r\mathbf{x}_r(k) + \mathbf{D}_r\mathbf{u}(k), \quad (3b)$$

where the new state vector  $\mathbf{x}_r \in \mathbb{R}^{n_r}$ ; the system input and output vectors  $\mathbf{u}$  and  $\hat{\mathbf{y}}$  retain the same dimension. From a simulation perspective, the estimated output  $\hat{\mathbf{y}}(k)$  from the reduced-order model is compared to the output from the full-order system  $\mathbf{y}(k)$  to test the performance of the MOR algorithms.

The general idea of the aforementioned reduction process is to find an invertible transformation matrix  $\mathbf{T}$  that maps state  $\mathbf{x}$  into another space state  $\mathbf{z}$ . Specifically,  $\mathbf{x} = \mathbf{T}\mathbf{z}$ , where the reduced-order model state  $\mathbf{z} \in \mathbb{R}^{n_x}$  and transformation matrix  $\mathbf{T} \in \mathbb{R}^{n_x \times n_x}$ . Moreover, matrix  $\mathbf{T}$  is constructed based on



some specific principles so that the elements in  $\mathbf{z}$  are ordered according to their *importance*; more on that in Section III-B. After substituting  $\mathbf{x} = \mathbf{T}\mathbf{z}$  into (2), we obtain

$$\mathbf{z}(k+1) = \mathbf{A}_z \mathbf{z}(k) + \mathbf{B}_z \mathbf{u}(k) \quad (4a)$$

$$\mathbf{y}_z(k) = \mathbf{C}_z \mathbf{z}(k) + \mathbf{D}_z \mathbf{u}(k), \quad (4b)$$

where system matrices  $\mathbf{A}_z = \mathbf{T}^{-1} \mathbf{A} \mathbf{T}$ ,  $\mathbf{B}_z = \mathbf{T}^{-1} \mathbf{B}$ ,  $\mathbf{C}_z = \mathbf{C} \mathbf{T}$ , and  $\mathbf{D}_z = \mathbf{D}$ .

However, the order of transformed system ( $\mathbf{A}_z, \mathbf{B}_z, \mathbf{C}_z, \mathbf{D}_z$ ) is still  $n_x$ . To obtain the reduced-order model (3), only the first  $n_r$  important elements in  $\mathbf{z}$  are kept and denoted by  $\mathbf{x}_r$ . That is,  $\mathbf{x}$  can be replaced by  $\mathbf{T}_r \mathbf{x}_r$  while the system properties remain the same, where  $\mathbf{T}_r \in \mathbb{R}^{n_x \times n_r}$  is the first  $n_r$  columns of  $\mathbf{T}$ . Finally, Equation (4) turns into reduced-order model (3) where

$$\mathbf{A}_r = \mathbf{S}_r \mathbf{A} \mathbf{T}_r, \mathbf{B}_r = \mathbf{S}_r \mathbf{B}, \mathbf{C}_r = \mathbf{C} \mathbf{T}_r, \mathbf{D}_r = \mathbf{D}. \quad (5)$$

Note that  $\mathbf{S}_r$  in (5) is neither the inverse or pseudo inverse of  $\mathbf{T}_r$ , it is simply the first  $n_r$  row of  $\mathbf{T}^{-1}$ .

In short, MOR is achieved by defining a certain subspace within the original space and transforming the system dynamics. A question that remains unanswered is which reduced-order models are most beneficial for achieving our goal: to reduce the difficulty of designing control/estimation algorithms such as Kalman filter, model predictive control (MPC), or linear-quadratic regulator (LQR) for water quality regulation. Since these control algorithms are related to the concepts of controllability and observability, one could argue that furthering our understanding of the controllable and observable subspaces could prove most useful for controller design.

Controllability describes the ability of an external input  $\mathbf{u}$  to drive system state  $\mathbf{x}(k)$  from any initial state  $\mathbf{x}(0)$  to any other final state  $\mathbf{x}(k_f)$  in a finite time interval  $[0, k_f]$  [25]. A quantitative metric for controllability in a DT-LTI system is the *controllability Gramian* defined as

$$\mathbf{W}_C = \sum_{m=0}^{\infty} \mathbf{A}^m \mathbf{B} \mathbf{B}^\top (\mathbf{A}^\top)^m, \quad (6)$$

which is the solution of Lyapunov equation  $\mathbf{W}_C - \mathbf{A} \mathbf{W}_C \mathbf{A}^\top = \mathbf{B} \mathbf{B}^\top$ . Similarly, observability describes the ability of reconstructing the initial unknown state  $\mathbf{x}(0)$  in finite  $k_f$  steps from the knowledge of output  $\mathbf{y}(k)$  [25]. The corresponding metric for observability is the *observability Gramian* defined as

$$\mathbf{W}_O = \sum_{m=0}^{\infty} (\mathbf{A}^\top)^m \mathbf{C}^\top \mathbf{C} \mathbf{A}^m, \quad (7)$$

and it is the solution of Lyapunov equation  $\mathbf{W}_O - \mathbf{A}^\top \mathbf{W}_O \mathbf{A} = \mathbf{C}^\top \mathbf{C}$ .

The physical meaning of MOR in terms of using controllable and observable subspaces are that the uncontrollable (not affected by the input  $\mathbf{u}$ ) and unobservable (does not affect the output  $\mathbf{y}$ ) parts of the system without significantly affecting input-output relations are discarded. The methods considering controllability or observability Gramian are also referred to as *Gramian based MOR*, and when both Gramians are considered in a method, it is then named as *cross-Gramian based MOR* [4], [26]. Next, we present several specific MOR algorithms that consider the Gramians which are suitable for water quality application.

## B. Description of common MOR algorithms

Herein, we describe three reduced-order modeling algorithms: balanced truncation (BT), proper orthogonal decomposition (POD), and balanced POD (BPOD).

1) *Balanced Truncation*: Balanced truncation computes a specific  $\mathbf{T}$  that transforms the coordinate system  $\mathbf{x}$  into a new coordinate system  $\mathbf{z}$  (see (4)) in which the controllability and observability Gramians (denoted  $\mathbf{W}_{Cz}$  and  $\mathbf{W}_{Oz}$ , respectively) are diagonal and identical [9]. We denote this particular Gramian by  $\Sigma_z$ . Mathematically,  $\Sigma_z = \mathbf{W}_{Cz} = \mathbf{W}_{Oz}$ , which implies that  $\Sigma_z^2 = \mathbf{W}_{Oz} \mathbf{W}_{Cz}$ , where

$$\mathbf{W}_{Cz} = \sum_{m=0}^{\infty} \mathbf{A}_z^m \mathbf{B}_z \mathbf{B}_z^\top (\mathbf{A}_z^\top)^m = \mathbf{T}^{-1} \mathbf{W}_C \mathbf{T}^{-1^\top},$$

and  $\mathbf{W}_{Oz} = \mathbf{T}^\top \mathbf{W}_O \mathbf{T}$ .

Note that  $\mathbf{W}_C$  and  $\mathbf{W}_O$  are the Gramians of the full-order model (2) before transformation. Hence,

$$\Sigma_z^2 = \mathbf{W}_{Oz} \mathbf{W}_{Cz} = \mathbf{T}^{-1} \mathbf{W}_O \mathbf{W}_C \mathbf{T} \Leftrightarrow \mathbf{W}_O \mathbf{W}_C \mathbf{T} = \mathbf{T} \Sigma_z^2,$$

which indicates the transformation  $\mathbf{T}$  can be obtained by finding the eigenvector matrix of  $\mathbf{W}_O \mathbf{W}_C$  (also known as *cross Gramian matrix*  $\mathbf{W}_X$  [27]), and the  $\Sigma_z^2$  is just the corresponding diagonal eigenvalue matrix. The final  $\mathbf{T}_r$  can be obtained simply by selecting the first  $n_r$  columns of  $\mathbf{T}$ .

Furthermore, the diagonal element  $\sigma_i$ ,  $i = 1, \dots, n_x$  in  $\Sigma_z$  measuring controllability and observability simultaneously are the *Hankel singular values* (HSVs) [28] of  $\mathbf{W}_X$ . Each single HSV  $\sigma_i = \sqrt{\lambda_i}$  provides a measure of “energy” for each state in a system. The model order reduction procedure provided by BT can also be viewed as retaining the  $n_r$ -largest HSVs (i.e.,  $\sigma_{n_r}^\downarrow(\mathbf{W}_X)$ ) or eigenvalues (i.e.,  $\lambda_{n_r}^\downarrow(\mathbf{W}_X)$ ) with higher energy and discarding the least important ones with lower energy. The  $n_r$  can be either specified as a fixed number directly or solved by specified energy level defined by  $\text{energy} = \sum_{j=1}^{n_r} \sigma_j / \sum_{i=1}^{n_x} \sigma_i$ . The simplified procedure of BT approach is summarized as Procedure 1.

---

### Procedure 1: Classical BT

---

- 1 Form cross Gramian matrix  $\mathbf{W}_X = \mathbf{W}_O \mathbf{W}_C$  after solving two Lyapunov equations
  - 2 Find the eigenvector matrix  $\mathbf{T}$  of  $\mathbf{W}_X$
  - 3 Specify  $n_r$  as needed or obtain  $n_r$  via setting an energy level
  - 4 Extract  $\mathbf{T}_r$  from the first  $n_r$  columns of  $\mathbf{T}$
  - 5 Extract  $\mathbf{S}_r$  from the first  $n_r$  rows of  $\mathbf{T}^{-1}$
  - 6 Obtain reduced-order model (3) by (5)
- 

This BT method considers and balances the controllability and observability simultaneously with high accuracy. Besides that, it preserves stability, which is a necessary property for the reduced-order system. However, it requires the computation of  $\mathbf{W}_O$  and  $\mathbf{W}_C$  or equally solving two Lyapunov equations and eigenvalue decomposition that are intractable for a large-scale network.

2) *Proper Orthogonal Decomposition*: The POD algorithm [10], [18], which is also known as principal component analysis or the Karhunen-Loève expansion, focuses on seeking a transformation  $\mathbf{T}_r$  such that a set of given data  $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ , with time  $t \in [0, m]$  is projected into the subspace of fixed dimension  $n_r$  as  $\mathbf{x}_r = \mathbf{T}_r \mathbf{x} \in \mathbb{R}^{n_r}$  while minimizing the total

error  $\sum_{t=0}^{m-1} \|\mathbf{x}(t) - \mathbf{x}_r(t)\|_2$ . To solve this problem, the data is assembled into an  $n_x \times m$  matrix

$$\mathbf{X}_m = [\mathbf{x}(0) \quad \mathbf{x}(1) \quad \dots \quad \mathbf{x}(m-1)], \quad (8)$$

which is called *snapshot*. Define matrix  $\mathbf{W}_{Cm}$  as  $\mathbf{X}_m \mathbf{X}_m^\top \in \mathbb{R}^{n_x \times n_x}$ . Matrix  $\mathbf{W}_{Cm}$  is controllability Gramian-like matrix containing the first  $m$  items when calculating  $\mathbf{W}_C$ , and it is an approximation of  $\mathbf{W}_C$ . We refer to  $\mathbf{W}_{Cm}$  as the *m-step controllability Gramian*. The optimal transformation  $\mathbf{T}_r$  in POD method can be obtained by eigenvalue decomposition  $\mathbf{W}_{Cm} \mathbf{T}_r = \mathbf{T}_r \mathbf{\Lambda}$ . Each eigenvector or the column vector in  $\mathbf{T}_r$  is named as *POD mode* [3], and the eigenvalues of  $\mathbf{W}_{Cm}$  are the diagonal elements of  $\mathbf{\Lambda}$ .

Note that a snapshot is given by (8) which, if  $\mathbf{A}$  is sparse, can be easily obtained even with  $n_x \geq 10^6$  since the computational task involves sparse matrix multiplication. However, finding the eigenvectors of an  $n_x \times n_x$  matrix  $\mathbf{W}_{Cm}$  is still challenging. Sirovich [10] solves this problem by defining  $\widetilde{\mathbf{W}}_{Cm} = \mathbf{X}_m^\top \mathbf{X}_m$  which is in  $\mathbb{R}^{m \times m}$  instead of  $\mathbb{R}^{n_x \times n_x}$  with  $m \ll n_x$ . The matrix form of eigenvector and eigenvalue of this new  $\widetilde{\mathbf{W}}_{Cm}$  are denoted as  $\mathbf{U}$  and  $\mathbf{\Lambda}$ . That is, the  $n_x \times n_x$  eigenvalue problem is solved thorough constructing an  $m \times m$  eigenvalue problem. Finally, the transformation matrices  $\mathbf{T}_r$  and  $\mathbf{S}_r$  can be obtained, and the detail is in Procedure 2. Note that when  $n_x \ll m$ , the above step is not necessary, and computing  $\mathbf{W}_{Cm}$  directly is easier and preferred.

---

#### Procedure 2: Classical POD

---

- 1 Construct snapshot  $\mathbf{X}_m$  using (8)
  - 2 Perform eigenvalue decomposition for  $\widetilde{\mathbf{W}}_{Cm} = \mathbf{X}_m^\top \mathbf{X}_m$  by finding  $\mathbf{U}$  and  $\mathbf{\Lambda}$  via  $\widetilde{\mathbf{W}}_{Cm} \mathbf{U} = \mathbf{U} \mathbf{\Lambda}$
  - 3 Specify  $n_r$  as needed or obtain  $n_r$  via setting an energy level
  - 4 Extract transformation  $\mathbf{T}_r$  from the first  $n_r$  column of  $\mathbf{X}_m \mathbf{U} \mathbf{\Lambda}^{-\frac{1}{2}}$
  - 5 Extract transformation  $\mathbf{S}_r$  from the first  $n_r$  rows of  $\mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U} \mathbf{X}_m^\top$
  - 6 Obtain reduced-order model (3) by (5)
- 

Since the eigenvalues of  $\mathbf{AB}$  are the same as those of  $\mathbf{BA}$  (see [29, Theorem 1.3.22]), i.e.,  $\lambda(\mathbf{AB}) = \lambda(\mathbf{BA})$ , the  $\widetilde{\mathbf{W}}_{Cm}$  shares the same eigenvalues with  $m$ -step controllability Gramian  $\mathbf{W}_{Cm}$ . That is, POD, not as the BT method that exploits the controllability Gramian (summations of infinity items), uses only the first  $m$  items of the controllability Gramian. This makes POD tractable for even large-scale systems but less accurate compared to BT. Thus, POD only considers controllability Gramian and selects the  $n_r$  modes with high-energy states (i.e.,  $\lambda_{n_r}^\downarrow(\mathbf{W}_{Cm})$ ), and discards the low-energy states.

3) *Balanced Proper Orthogonal Decomposition*: BPOD aims to combine the advantages of BT and POD together, that is, to obtain an approximation to BT that is computationally tractable for large-scale systems [3]. Different from the snapshot  $\mathbf{X}_m$  (8) defined in POD method, BPOD defines  $\mathbf{X}_m$  as

$$\mathbf{X}_m = [\mathbf{x}_1(0) \dots \mathbf{x}_1(m-1) \dots \mathbf{x}_{n_u}(0) \dots \mathbf{x}_{n_u}(m-1)], \quad (9)$$

where the  $\mathbf{x}_i(m)$  is the impulse response (when  $\mathbf{u}_i(k) = \delta(k)$  is an impulse signal). That is,  $\mathbf{x}_i(m) = \mathbf{A}^m \mathbf{b}_i$  and  $\mathbf{b}_i$  is the  $i$ -th column of  $\mathbf{B}$ .

Moreover, BPOD denotes  $\mathbf{Y}_m$  as the snapshot of the dual system of system  $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$  or equally (2). That is,  $\mathbf{Y}_m$  is the snapshot of system  $(\mathbf{A}^\top, \mathbf{C}^\top, \mathbf{B}^\top, \mathbf{D}^\top)$  with system state denoted as  $\mathbf{z}$ , and  $\mathbf{Y}_m$  can be expressed as

$$\mathbf{Y}_m = [\mathbf{z}_1(0) \dots \mathbf{z}_1(m-1) \dots \mathbf{z}_{n_y}(0) \dots \mathbf{z}_{n_y}(m-1)], \quad (10)$$

where the  $\mathbf{z}_i(m)$  is the impulse response, that is,  $\mathbf{z}_i(m) = (\mathbf{A}^\top)^m \mathbf{c}_i^\top$  and  $\mathbf{c}_i^\top$  is the  $i$ -th column of  $\mathbf{C}^\top$ .

Then, the balancing modes are computed by forming the singular value decomposition of the *block Hankel matrix*  $\mathbf{H}_m$ :

$$\mathbf{H}_m = \mathbf{Y}_m^\top \mathbf{X}_m = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top = [\mathbf{U}_r \quad \mathbf{0}] \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^\top \\ \mathbf{0} \end{bmatrix}, \quad (11)$$

where the diagonal element in  $\mathbf{\Sigma}_r \in \mathbb{R}^{n_r \times n_r}$  collects the largest  $n_r$  singular values of  $\mathbf{H}_m$ , and  $\mathbf{U}_r$  and  $\mathbf{V}_r^\top$  are the corresponding left- and right-singular vectors. Hence, the final transformation matrices are  $\mathbf{T}_r = \mathbf{X}_m \mathbf{V}_r \mathbf{\Sigma}_r^{-\frac{1}{2}}$  and  $\mathbf{S}_r = \mathbf{\Sigma}_r^{-\frac{1}{2}} \mathbf{U}_r^\top \mathbf{Y}_m^\top$ ; see Procedure 3 for details.

---

#### Procedure 3: Classical BPOD

---

- 1 Form snapshots  $\mathbf{X}_m$  (9) and  $\mathbf{Y}_m$  (10) of BPOD
  - 2 Perform singular value decomposition of  $\mathbf{H}_m = \mathbf{Y}_m^\top \mathbf{X}_m$  (11), and obtain  $\mathbf{U}_r$ ,  $\mathbf{\Sigma}_r$ ,  $\mathbf{V}_r$
  - 3 Specify  $n_r$  as needed or obtain  $n_r$  via setting an energy level
  - 4 Calculate  $\mathbf{T}_r = \mathbf{X}_m \mathbf{V}_r \mathbf{\Sigma}_r^{-\frac{1}{2}}$  and  $\mathbf{S}_r = \mathbf{\Sigma}_r^{-\frac{1}{2}} \mathbf{U}_r^\top \mathbf{Y}_m^\top$
  - 5 Obtain reduced-order model (3) by (5)
- 

From the above procedures, it is clear that BPOD considers both controllability and observability Gramians like BT method, successfully avoids solving the two Lyapunov equations, forming cross Gramian matrix  $\mathbf{W}_X$ , and finding the eigenvalues of  $\mathbf{W}_X$  to form a traceable algorithm simply by adopting the concept of the snapshot.

## IV. THEORETICAL COMPARISONS OF MOR ALGORITHMS FOR WATER QUALITY DYNAMICS

The advantages and disadvantages from the aspects of tractability for large-scale systems, relative accuracy, and stability preserving properties are summarized in this section. Subsequently, we present a comparison of these MOR algorithms to determine which properties of these algorithms fit best for our application—the water quality model.

First, the complexity of solving Lyapunov equations [31], eigenvalue decomposition [32], and SVD [33] are roughly estimated by  $\mathcal{O}(n^3)$  for an  $n \times n$  matrix. Although all BT, POD, and BPOD procedures include at least one of the above time-consuming computation, the matrix size appeared in these procedures and the number of executions of the procedures are different; see Tab. I. For example, BT solves two Lyapunov equations ( $n_x \times n_x$ ) and performs eigenvalue decomposition for  $\mathbf{W}_X$  once, that is, it performs algorithm with  $\mathcal{O}(m^3)$  three times; POD performs eigenvalue decomposition for  $\widetilde{\mathbf{W}}_{Cm}$ , the complexity is roughly  $\mathcal{O}(m^3)$ ; BPOD performs truncated SVD (rank  $n_r$ ; see (11)) for  $\mathbf{H}_m$ , the complexity is roughly  $\mathcal{O}(m^2 n_y n_u n_r)$ . Due to  $m \ll n_x$ , it is clear that BT is the slowest algorithm. The computational load of POD and BPOD

Tab. I  
COMPARISONS AMONG THREE CLASSICAL MOR METHODS.

	Matrix involved	Dimension	Decomposition method <sup>†</sup>	Mode retained	Complexity (Computational load)	Tractability for large-scale systems	Relative accuracy <sup>◊</sup>	Stability* preserving
BT	$\mathbf{W}_X$	$\mathbb{R}^{n_x \times n_x}$	ED	$\lambda_{n_r}^\downarrow(\mathbf{W}_X)$	$\mathcal{O}(n_x^3)$ (High)	No	High	Yes
POD <sup>‡</sup>	$\widetilde{\mathbf{W}}_{Cm}$	$\mathbb{R}^{m \times m}$	ED	$\lambda_{n_r}^\downarrow(\widetilde{\mathbf{W}}_{Cm})$	$\mathcal{O}(m^3)$ (Low)	Yes	Low	No
BPOD	$\mathbf{H}_m$	$\mathbb{R}^{mn_y \times mn_u}$	SVD	$\sigma_{n_r}^\downarrow(\mathbf{H}_m)$	$\mathcal{O}(m^2 n_y n_u n_r)$ (Low)	Yes	Medium	Depends <sup>♣</sup>

<sup>†</sup>Eigenvalue decomposition (ED); singular value decomposition (SVD); <sup>‡</sup>use  $\widetilde{\mathbf{W}}_{Cm}$  when  $m \ll n_x$ , otherwise, use  $\mathbf{W}_{Cm}$  instead.

<sup>◊</sup>Under the same number of modes; \*asymptotically stable; <sup>♣</sup>Rowley [3] conjectures that BPOD can preserve stability via theoretical analysis while other studies [6], [30] denied this assertion based on test results.

algorithms depends on the value of  $m$  and  $n_y n_u n_r$ . When these values are of the same order of magnitude, the complexity of POD and BPOD are similar and also tractable for large-scale systems; see Tab. I for the final computational load.

Second, a useful property of the BT method is that the error bounds provided are close to the lower bound achieved by any reduced-order model, and the upper bound of the error is guaranteed less than  $2 \sum_{i=n_r+1}^{n_x} \sigma_i$ , where  $\sigma_i$  is the  $i$ -th HSV of  $\mathbf{W}_X$  in Section III-B1 [3]. Furthermore, BT considers  $\mathbf{W}_X$  that balances controllability and observability. Hence, the relative error in terms of preserving input-output relations of the BT method is theoretically the best among these three methods. As for POD, it only takes advantage of  $m$ -step controllability Gramian  $\mathbf{W}_{Cm}$ , and has the risk of discarding the states that are highly observable but less controllable. This defect leads to the inaccuracy of POD or even the failure to reduce system order.

In fact, BPOD provides a good approximation of BT since they retain the same states/modes when performing MOR. The corresponding  $n_r$ -largest eigenvalues (singular values) of all modes/states retained from three procedures are summarized in Tab. I. That is, the states retained are those with  $\lambda_{n_r}^\downarrow(\mathbf{W}_X)$  ( $\sigma_{n_r}^\downarrow(\mathbf{H}_m)$ ) for BT (BPOD). Suppose  $\lambda_i$  ( $\sigma_i$ ) is the  $i$ -th element of  $\lambda_{n_r}^\downarrow(\mathbf{W}_X)$  ( $\sigma_{n_r}^\downarrow(\mathbf{H}_m)$ ), and  $\lambda_i \approx \sigma_i^2$  always holds true when  $m$  is large enough. This implies that BPOD, like BT, uses the same information (i.e., cross Gramians  $\mathbf{W}_X$ ) to obtain the transformation  $\mathbf{T}_r$  to reduce system order. The difference lies in the way each algorithm deals with such information. For example, BT requires an eigenvalue decomposition of  $\mathbf{W}_X$  while BPOD involves singular value decomposition to  $\mathbf{H}_m$  embedded in  $\mathbf{W}_X$ . From this perspective, the transformation  $\mathbf{T}$  obtained from the BPOD method is the balanced one that simultaneously diagonalizes  $\mathbf{W}_C$  and  $\mathbf{W}_O$ . Hence, BPOD has the same ability (when  $m$  is large enough) ensure the accuracy as BT does.

The BT method guarantees stability theoretically after MOR [9], while POD fails [4], [34] due to the projection it uses. As for the BPOD method, different studies report various results. Rowley [3] claims that the stability-preserving from BPOD is guaranteed for linear systems when snapshots are large enough. However, there is no guideline to find what is the proper size of snapshots. We suspect that this might be why the results from several studies [6], [30] do not support the stability-preserving property of BPOD. In other words, stability-preserving of BPOD depends on the parameter (i.e., length of snapshots) set. We would present an approach to find

the proper parameter that ensures stability in Section V.

In short, POD and/or BPOD methods are still preferred since they are tractable for a large-scale system, even if their stability-preserving properties are not sufficient for water quality dynamics MOR. Toward that goal, the stability-preserving model order reduction approaches for water quality modeling are explored and proposed next.

## V. STABILIZING MOR ALGORITHMS

In this section, the brief literature of stability-preserving MOR is presented first followed by proposing two methods suitable for the stable and discrete-time system.

The approaches developing stability-preserving reduced-order model involve *a priori* frameworks [35]–[39], and *a posteriori* frameworks [40]–[43]. *A priori* stability-preserving framework can be difficult to implement, and/or requires the solution of PDEs or special system structure. For example, the Galerkin projection (i.e. a congruence transform) needs sign-definite matrices in the full-order system. Conversely, *a posteriori* methods minimally modify the reduced-order system matrix so that the accuracy is not significantly changed. The more detailed literature can be found in [41]. Herein, we discuss stability-preservation with both these frameworks.

In short, we propose a novel posteriori method to stabilize the reduced-order model after performing the POD or BPOD procedure. Comparing to existing methods such as [42], the proposed posteriori method adjusts the final transformation  $\mathbf{A}_r$  directly instead of optimizing/adjusting the projection  $\mathbf{U}_r$  slightly while fixing the projection  $\mathbf{V}_r^\top$  in (11). Furthermore, published studies focus more on continuous systems while the proposed one is for discrete systems and uses a special trick to relax the non-convex optimization problem.

Moreover, *a priori* stabilization is proposed specifically for the BPOD procedure. To the best of our knowledge, this is the first *a priori* stabilization method for BPOD. Comparing to existing studies such as [35] focusing on finding a special and complex projection for POD, the proposed method is simple and intuitive since it only requires modifying a parameter in BPOD.

### A. A posteriori stabilization for POD-based approach

After performing the POD procedure, the stability of the reduced-order model can not be guaranteed. That is,  $\mathbf{A}_r$  is unstable. As for the stability-preserving from BPOD, it depends on the length of snapshots  $m$ . When  $m$  is not proper,  $\mathbf{A}_r$  from

BPOD tends to be unstable. The motivation behind this posterior method is to adjust  $\mathbf{A}_r$  by  $\Delta\mathbf{A}_r$  such that the adjusted system is stable while ensuring that the norm  $\|\Delta\mathbf{A}_r\|_2$  is as small as possible to maintain the accuracy of the full-order system.

From Lyapunov stability conditions [44], we know that a DT-LTI system  $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$  is stable when there exist a positive definite matrix  $\mathbf{P}$  such that  $\mathbf{A}^\top \mathbf{P} \mathbf{A} - \mathbf{P}$  is negative definite. Thus, in our case the stabilization problem can be expressed as an optimization problem below.

$$\min_{\mathbf{P}, \Delta\mathbf{A}_r, \alpha} \quad \alpha \quad (12a)$$

$$\text{subject to} \quad \mathbf{P} \succ 0, \alpha > 0 \quad (12b)$$

$$\|\Delta\mathbf{A}_r\|_2 \leq \alpha \quad (12c)$$

$$(\mathbf{A}_r + \Delta\mathbf{A}_r)^\top \mathbf{P} (\mathbf{A}_r + \Delta\mathbf{A}_r) - \mathbf{P} \prec 0, \quad (12d)$$

where  $\mathbf{P}$ ,  $\Delta\mathbf{A}_r$  in  $\mathbb{R}^{n_r \times n_r}$  and  $\alpha \in \mathbb{R}$  are the optimization variables. However, this problem (12) is nonconvex. We can take the following steps to transform it into an semidefinite programming (SDP) that is convex.

Since  $\mathbf{P} \succ 0$  is invertible, let  $\mathbf{P} = \mathbf{P}\mathbf{P}^{-1}\mathbf{P}$ , and taking Schur complements, we have

$$\begin{aligned} (12d) &\Leftrightarrow (\mathbf{A}_r + \Delta\mathbf{A}_r)^\top \mathbf{P}\mathbf{P}^{-1}\mathbf{P} (\mathbf{A}_r + \Delta\mathbf{A}_r) - \mathbf{P} \prec 0 \\ &\Leftrightarrow \begin{bmatrix} -\mathbf{P} & (\mathbf{P}\mathbf{A}_r + \Delta\mathbf{Y})^\top \\ \mathbf{P}\mathbf{A}_r + \Delta\mathbf{Y} & -\mathbf{P} \end{bmatrix} \succ 0, \end{aligned} \quad (13)$$

where  $\Delta\mathbf{Y} = \mathbf{P}\Delta\mathbf{A}_r$ . Furthermore, the Cauchy-Schwarz inequality yields

$$\|\Delta\mathbf{Y}\|_2 = \|\mathbf{P}\Delta\mathbf{A}_r\|_2 \leq \|\mathbf{P}\|_2 \|\Delta\mathbf{A}_r\|_2.$$

Invoking (12c) yields  $\|\Delta\mathbf{Y}\|_2 \leq \alpha \|\mathbf{P}\|_2$ . Taking Schur complements yields

$$(12c) \Leftrightarrow \|\Delta\mathbf{Y}\|_2 \leq \alpha_y \Leftrightarrow \begin{bmatrix} \mathbf{I} & * \\ \Delta\mathbf{Y} & \alpha_y^2 \mathbf{I} \end{bmatrix} \succeq 0,$$

where  $\alpha_y = \alpha \|\mathbf{P}\|_2$ . However, the  $\alpha_y^2$  is still nonlinear and can be relaxed as

$$\begin{bmatrix} \mathbf{I} & * \\ \Delta\mathbf{Y} & \alpha_y \mathbf{I} \end{bmatrix} \succeq 0, \quad \alpha_y > 0. \quad (14)$$

Finally, a convex SDP stabilizing the reduced-order model is formed as

$$\min_{\mathbf{P}, \Delta\mathbf{Y}, \alpha_y} \quad \alpha_y \quad (15a)$$

$$\text{subject to} \quad \mathbf{P} \succ 0, (13), (14). \quad (15b)$$

Note that the dimension of optimization variables such as  $\mathbf{P}$  and  $\Delta\mathbf{Y}$  depends on the size of  $\mathbf{A}_r$  in reduced-order model (3) instead of the size of  $\mathbf{A}$  in full-order model (2). This indicates the formulation (15) can be applied to solve the stabilization problem even for a large-scale system, provided that  $n_r$  is not large after reduction procedures.

### B. A priori stabilization for BPOD approach

The a priori stabilization method introduced in this section is different from the posterior one since it directly modifies the length of snapshots  $m$  in the standard BPOD Procedure 3 while generating a stable reduced-order model. Ideally, when parameter  $m$  is set as  $\infty$ , the procedure then has all system information as the cross-Gramian does in the BT method, and consequently BPOD reaches its best performance. However,

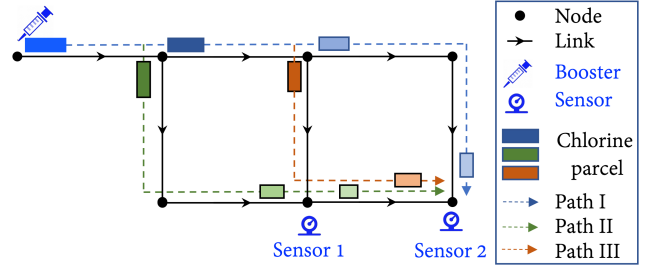


Fig. 2. Chlorine travel paths from the input location (booster station) to different output locations (Sensors 1 and 2).

parameter  $m$  cannot be too large (or equal to infinity) due to the prohibitive computational cost. Hence, selecting a lower bound for  $m$  denoted by  $\underline{m}$  is important. To the best of our knowledge, this is the first time attempt to produce such a value or condition that ensures the stability of a DT system using BPOD method. In particular, we propose two methods to find  $\underline{m}$ . The first method is generated from insights from transport dynamics in water networks. The second method is via a control theoretic approach. In the case studies section, we compare the performance of these two methods.

1) *First method:* The minimum number of steps  $\underline{m}$  is the equivalent time-step for a chlorine parcel traveling from its input location to the furthest output location. Mathematically, it can be expressed as

$$\underline{m} = \left\lceil \frac{T_{\text{travel}}}{\Delta t} \right\rceil = \left\lceil \sum_{ij \in \mathcal{L}_{\text{path}}} \frac{L_{ij}}{v_{ij} \Delta t} \right\rceil, \quad ij \in \mathcal{L}_{\text{path}}, \quad (16)$$

where  $\Delta t$  is determined by the L-W scheme stability condition; the total travel time is denoted by  $T_{\text{travel}}$ ;  $\mathcal{L}_{\text{path}}$  represents the links in the travel path, and the corresponding length and velocity of each pipe  $ij$  in that path are  $L_{ij}$  and  $v_{ij}$ .

To illustrate the application of this method, we use the network comprised of two output sensors and one booster station controller; see Fig. 2. After a chlorine parcel with a certain mass and concentration is injected at the booster station, it travels to the sensors with various velocities in different links. Note that both the mass and concentration of the chlorine parcel reduce gradually due to consumption by the nodes and decay of chlorine concentration. Furthermore, parcels (the rectangle in Fig. 2) with different masses and concentrations (the areas and colors of the rectangles) arrive at the sensors at various times. The number of parcels depends on the number of paths from an input location (booster) to output one (sensor). For example, there are three paths in Fig. 2 and the furthest location from the input is Sensor 2, there are three travel times (three paths) for it from the booster, and the longest travel time should be used when calculating (16).

2) *Second method:* From a control theory perspective, the lower bound  $\underline{m}$  can also be estimated by the settling time of water quality dynamic model (2). The settling time  $T_s$  of a dynamic system can be roughly computed by its dominant pole  $p$  which is equal to the eigenvalue of matrix  $\mathbf{A}$  with the largest magnitude. In particular,  $T_s$  can be computed as

$$T_s = \frac{-4\Delta t}{\ln|p|},$$



where  $\Delta t$  is the sampling time of DT-LTI system (2); the dominant pole  $p$  located within a unit disk indicates that  $\ln|p|$  is always negative. Consequently, the lower bound of  $m$  is

$$\underline{m} = \left\lceil \frac{T_s}{\Delta t} \right\rceil = \left\lceil \frac{-4}{\ln|p|} \right\rceil. \quad (17)$$

Note that the equivalent travel time-step in (16) has a physical meaning for water quality dynamics which can be computed easily. Meanwhile, the second method is estimated by the empirical formula (17). Section VI-B2 investigates the performance of these two methods.

Finally, the overall procedure of general, stability-preserving MOR procedure is summarized in Algorithm 1. In this algorithm, we use BPOD but roughly the same approach can be applied to POD.

---

**Algorithm 1:** Stability-Preserving BPOD (SBPOD).

---

**Input:** Full-order model  $(A, B, C, D)$  (2) and control input  $u(k)$   
**Output:** Stable reduced-order model  $(A_r, B_r, C_r, D_r)$  (3), outputs  $y(k)$  and  $\hat{y}(k)$

- 1 **if** a posterior stabilization method **then**
- 2     Standard BPOD approach given in Procedure 3
- 3     Solve a convex SDP problem (15) to stabilize the reduced-order model  $(A_r, B_r, C_r, D_r)$
- 4 **end if**
- 5 **if** a priori stabilization method **then**
- 6     Obtain parameter  $m$  via (16) or (17), and set length of snapshot as  $m = \underline{m}$
- 7     Execute BPOD procedure to obtain reduced-order models  $(A_r, B_r, C_r, D_r)$  (3)
- 8 **end if**
- 9 Generate  $y(k)$  and  $\hat{y}(k)$  from the full- and reduced-order models

---

## VI. CASE STUDIES

We present three examples (three-node, Net1, and Net3 networks [45]) to illustrate the performance of three different MOR methods in terms of accuracy, computational load/time, and stability-preserving properties. Then, we test the two stabilization methods proposed in Section V. Lastly, we showcase the application of MOR algorithms when MPC is used to control water quality dynamics for the full-order and reduced-order systems. All codes, parameters and tested networks simulated via EPANET Matlab Toolkit [46] are available on Github [47]. The testing environment is a Win 10 Precision 7920 Tower with Inter(R) Xeon(R) Gold 5218 CPU @2.3GHz and 64G memory.

### A. Network settings, practical conditions, and validation

1) *Settings:* The basic information of the tested networks are in Tab. II. The three-node network in Fig. 3a is a self-designed network for illustrative purpose that includes one junction J2, one pipe P23, one pump PM12, one tank TK3, and one reservoir R1. A booster station and a chlorine concentration sensor are installed at J2 and TK3, which indicates it is a single-input single-output (SISO) system ( $n_u = n_y = 1$ ). The pipe P23 connecting J2 and TK3 is split into  $s_L = 150$  segments according to L-W scheme in Fig. 1a. Hence the dimension of the full-order system  $n_x = 154$  ( $n_N = n_J + n_R + n_{TK} = 3$ ,

Tab. II  
BASIC INFORMATION OF THREE TESTED NETWORKS.

Networks	# of components*	Full order $n_x$	Booster station locations ( $n_u$ )	Sensors location ( $n_y$ )
three-node network	$\{1,1,1, 1,1,0\}$	154	J2 (1)	TK3 (1)
Net1	$\{9,1,1, 12,1,0\}$	1,293	J10 (1)	J22, J23 (2)
Net3	$\{92,2,3, 117,2,0\}$	29,374	J237, J247 (2)	J255, J241 J249 (3)

\*Number of each component in WDN:  $\{n_J, n_R, n_{TK}, n_P, n_M, n_V\}$ .

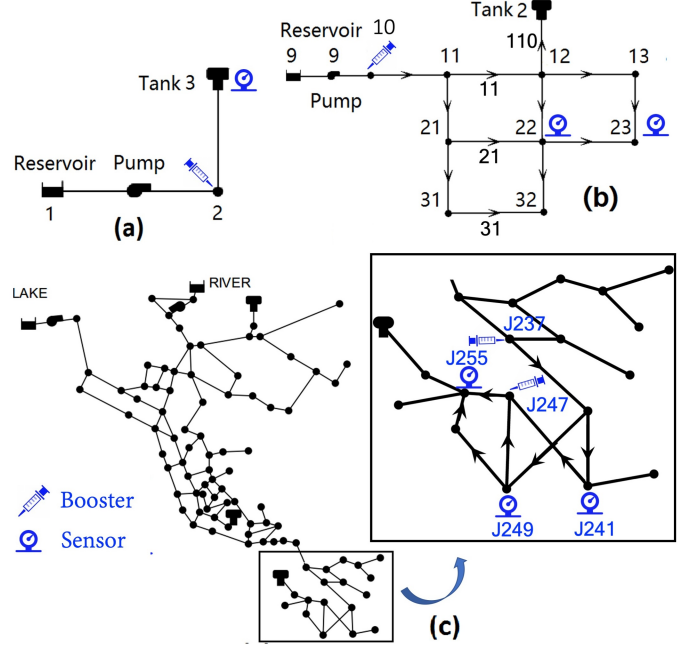


Fig. 3. Three tested networks: (a) three-node network; (b) Net1 network; (c) Net3 network (zoomed-in part is the controlled area).

and  $n_L = n_P \times s_L + n_M = 151$ ). The dimensions and other parameters of the other networks are listed in Tab. II.

2) *Non-zero initial conditions:* In a water quality simulation, initial chlorine concentrations are typically non-zero (i.e.,  $x(0) \neq 0$ ). With that in mind, some MOR algorithms are designed with zero initial conditions. To tailor the MOR algorithms for water quality dynamics with non-zero initial conditions, we adopt the method from [48] as follows. Suppose we have a full-order model  $(A, B, C, D)$  with  $x_o = x(0) \neq 0$ , let  $\tilde{x} = x - x_o$ , then

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}\tilde{u}(k) \quad (18a)$$

$$\tilde{y}(k) = \tilde{C}\tilde{x}(k) + \tilde{D}\tilde{u}(k), \quad (18b)$$

where  $\tilde{A} = A$ ,  $\tilde{B} = [B \quad Ax_o]$ ,  $\tilde{C} = C$ ,  $\tilde{D} = [D \quad Cx_o]$ , and  $\tilde{u} = \begin{bmatrix} u \\ 1 \end{bmatrix}$ . That is, the non-zero initial condition is viewed as an input for the new augmented full-order model (18) with  $\tilde{x}_o = 0$  such that the introduced MOR algorithms can be applied directly without any modification.

3) *Validation:* To quantify the performance of MOR methods such as accuracy, the comparisons of step responses for the



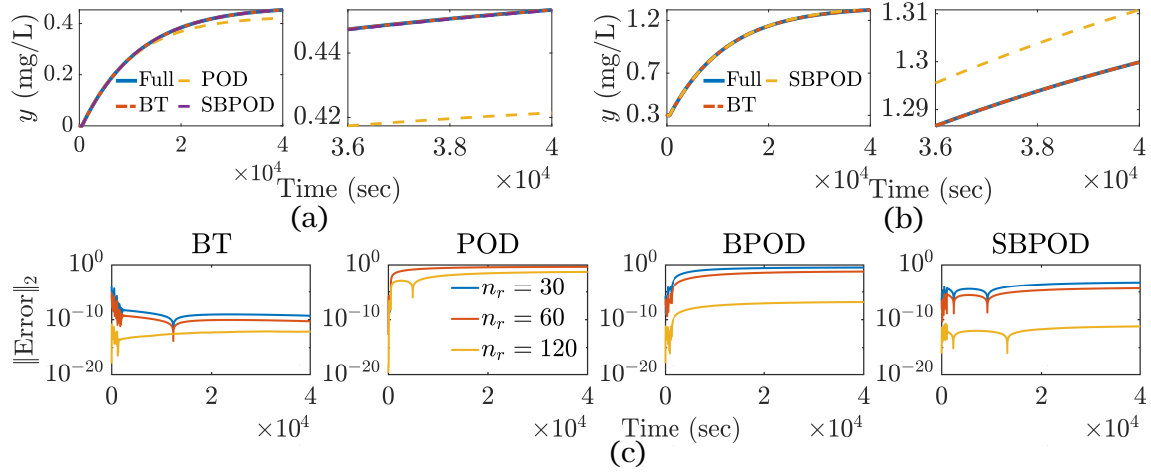


Fig. 4. Step responses (with amplitude of 50) of full-/reduced- order models for the three-node network under: (a) zero-initial conditions; (b) non-zero initial conditions. (c) The step response errors between full-/reduced- order models under zero-initial conditions as  $n_r$  increases from 30 to 120.

full-order model (2) and reduced-order model (3) are necessary. This is customary in MOR studies. The core idea of the designed experiments is applying the same input signal  $u(k)$  to both the full- and reduced-order systems and then comparing output differences (i.e., step response errors between  $y(k)$  and  $\hat{y}(k)$ ). To quantify the accuracy of the MOR algorithms, we use the root-mean-square error (RMSE) defined as

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{k=1}^m \|y(k) - \hat{y}(k)\|_2^2}$$

to quantify the error between full-order model output  $y(k)$  and reduced-order model  $\hat{y}(k)$  of a time horizon of  $m$  time-steps.

### B. Accuracy and stability of MOR methods

1) *The three-node network*: The accuracy of MOR methods of the three-node network under zero initial conditions (i.e.,  $x_o = 0$ ) is presented in Fig. 4a(left). The input is a step signal with amplitude of 50, that is, we inject 50 mg chlorine per sampling time ( $\Delta t = 20$  seconds) at J2 in Fig. 3a. The blue line is the step response of the full-order model (2) ( $n_x = 154$ ), and the dotted lines are the step responses of the reduced-order models (3) obtained from BT, POD, and SBPOD ( $n_r = 30, 118$ , and  $28$ ). The data is generated from Procedures 1, 2, 3, and Algorithm 1. Fig. 4a(right) depicts the magnified step responses when time is limited in  $[36000, 40000]$  seconds. It can be seen that (i) the reduced-order models from BT, POD, and SBPOD ( $n_r = 30, 118$ , and  $28$ ) successfully maintain the input-output relationship with smaller orders ( $n_r < n_x = 154$ ), and (ii) at the end of the test ( $[36000, 40000]$  sec), POD starts to show vast error while the results of BT and BPOD are still close to the full-order system.

The accuracy test under non-zero initial conditions (i.e.,  $x_o \neq 0$ ) is also evaluated for the three-node network. All settings including the input signal remain the same except that the initial chlorine concentrations at R1, J2, and TK3 are 1.0 mg/L, 0.5 mg/L, and 0.3 mg/L; the initial chlorine concentration in PM12 and P23 are 0.75 mg/L, and 0.3 mg/L. The corresponding results are shown in Fig. 4b. Note that the output of POD is not shown

due to it producing unreasonable results with huge RMSE = 23.51. This is because the augmented system (18) views non-zero initial conditions as inputs. In this way, the input matrix of (18)  $\tilde{B}$  includes two parts: the actual chlorine injection  $B$  and *virtual inputs* from the initial condition  $Ax_o$ . If the majority of elements in vector  $Ax_o$  are non-zeros, then this system has too many inputs making the POD hard to capture the input-output relationship. With that in mind, the non-zero initial conditions are not problematic for BT and BPOD since matrices  $W_X$  or  $H_m$  have the term  $CB$  that cancels such effect. It indicates POD is unsuitable for water quality dynamics with non-zero initial conditions.

To further test the accuracy of all MOR methods (i.e., classical BT, POD, and BPOD and proposed SPPOD) and the impact of parameter  $n_r$ , we show the norm of step response errors (denoted by  $\|\text{Error}\|_2 = \|y - \hat{y}\|_2$ ) between full-order model (2) and reduced-order models (3) under zero-initial conditions in Fig. 4c. As  $n_r$  increases from 30 to 120, the step response errors of all MOR methods decrease. We note the following. First, the amplitude of step response is in  $10^0$  level (see Fig. 4c) while the error is in  $10^{-5}$  or lower level for BT and SBPOD. This indicates that the error is tiny—it can be neglected and hence the reduced models are accurate. Second, POD does not perform well even  $n_r$  reaches to 120 ( $n_x = 154$ ). This is due to POD only considering the controllability Gramian in Section IV. Third, classical BPOD (with  $m = \underline{m} = 160$ ) performs worse than SBPOD ( $m = 400 > \underline{m} = 160$ ); this indicates classical BPOD can not ensure the stability or accuracy and is not suitable for water quality dynamics. The value of parameter  $\underline{m} = 160$  for this simple network can be obtained easily by Equation (16), that is, it equals the time-step from J2 to TK3. We show details of how to find  $\underline{m}$  or  $m$  using a more complex Net1 network in the next section. All these test results corroborate the analyses performed in Section IV and Tab. I.

Furthermore, the RMSEs of all MOR methods (classical BT and POD, proposed SBPOD) for the three-node network under zero or non-zero initial conditions are shown in Tab. III. The

Tab. III  
ACCURACY WITH (I) STEP SIGNAL AS INPUTS, AND (II) NON-ZERO (ZERO) INITIAL CONDITIONS AND CORRESPONDING COMPUTATIONAL TIME (IN SECOND) AMONG THREE CLASSICAL MOR METHODS FOR THREE TESTED NETWORKS.

	RMSEs when applying step signals with $\mathbf{x}_o = \mathbf{0}$ ( $\mathbf{x}_o \neq \mathbf{0}$ )			Computational time (in second)		
	three-node network	Net1	Net3	three-node network	Net1	Net3
BT	$1.75 \times 10^{-6}$ ( $1.71 \times 10^{-4}$ )	$1.63 \times 10^{-6}$ ( $6.7 \times 10^{-3}$ )	NA* (NA*)	0.245	17.2	Intractable*
POD	$1.87 \times 10^{-2}$ (23.51)	$9.24 \times 10^{-5}$ ( $1.0 \times 10^{30}$ )	$7.16 \times 10^{-5}$ (0.49)	0.219	6.4	245.8
SBPOD	$1.54 \times 10^{-4}$ ( $4.2 \times 10^{-3}$ )	$6.9 \times 10^{-4}$ ( $6.5 \times 10^{-4}$ )	$4.30 \times 10^{-6}$ ( $2.70 \times 10^{-3}$ )	0.241	6.6	150.4

\*NA is "Not Applicable" as BT fails to produce the reduced-order model after 12 hours of simulation.

classical BPOD cannot ensure stability under either zero or non-zero initial conditions (see the further test in Section VI-B2 for more details and discussions). Hence, the corresponding RMSEs are not included in the table. Based on these small RMSEs, we conclude that MOR methods perform well under zero initial conditions while BT and SBPOD perform the best under the realistic non-zero initial conditions of water quality dynamics.

2) *Net1 network*: The accuracy tests of classical BT, classical POD, and SBPOD methods for the Net1 network under zero/non-zero initial conditions are performed. Instead of presenting all overlapped lines in figures, we adopt RMSEs in Tab. III to compare accuracy among all MOR methods. The full order of Net1 is  $n_x = 1293$ , and BT and SBPOD reduce the order to  $n_r = 113$  and  $n_r = 117$  under non-zero initial conditions. Note that POD fails to provide accurate reduced-order models from the corresponding RMSEs shown in Tab. III. For zero initial conditions, MOR methods (i.e., classical BT and POD and SBPOD) successfully reduce the system orders, and  $n_r$  of BT, POD, and SBPOD are 145, 270, and 115. From these small RMSEs in Tab. III, it is evident that the reduced-order systems are accurate.

Next, we show how parameter  $\underline{m}$  is obtained for Net1, and test the performance of BPOD (with an  $m$  less than  $\underline{m}$ ), SBPOD with the posterior stabilization method (same  $m$  with BPOD), and SBPOD with the priori stabilization method ( $m$  larger than the  $\underline{m}$ ).

**Computing lower bounds for  $m$ .** The first step in generating stability-preserving reduced-order models is to compute the lower bound for  $m$  for the SBPOD method. When the length of snapshots  $m$  of BPOD is large enough for water quality dynamics, BPOD can preserve stability; otherwise, it cannot. This is the stability-preserving condition for BPOD as introduced in Section V-B. The two approaches to finding  $\underline{m}$  are tested for the Net1 network.

The first method given in (16) is used to obtain the proper  $\underline{m}$  which requires computing the travel time of the last chlorine parcel injected at the booster J10 to the furthest sensor J23. The first (second) sensor installed at J22 (J23) receives two (three) step responses from the booster installed at J10, and the corresponding time to reach the peak is marked in the third column of Fig. 5a. The peak time of the last step response at J22 (J23) is  $t_2 = 16550$  ( $t_3 = 45360$ ), and the arrival or travel time

of the last chlorine parcel (step response) is slightly less than the peak time. That is, 16000 for J22 and 39750 for J23 (we do not mark it in Fig. 5). Hence, parameter  $\underline{m}$  can be obtained by the longest time 39750 divided by the sampling time  $\Delta t = 15$  seconds. That is, parameter  $\underline{m}$  is set to 2650 via the first method.

We also choose Equation (17), that is the second method, to obtain the proper  $\underline{m}$  which requires computing the dominant pole  $p$ . For Net1,  $p = 0.9967 \pm 0.0563i$ , and  $T_s = -4/\ln|p| = 2350.9$  resulting in  $\underline{m} = 2351$  for the second method.

Although parameter  $\underline{m}$  obtained by two methods are different, both are reasonable. After testing, we notice that when  $m \in [2351, 2650]$ , the reduced-order model for Net1 via BPOD shows instability as an oscillation tail is observed at the end of the step response; see the first column of figures in Fig. 5a.

**Computing and simulating reduced-order models.** After finding the appropriate lower bounds for  $m$ , Algorithm 1 is implemented to compute a stable reduced-order model for Net1. The step responses of the posterior stabilization method (stabilizing unstable reduced-order model from classical BPOD via solving an SDP (15)) are shown in the second column of Fig. 5a. Comparing with the unstable step responses from classical BPOD with  $m = 2650$  (i.e., the first column of Fig. 5a), the yielding responses after posterior stabilization are not oscillatory but are unfortunately inaccurate. The step responses of the priori method ( $m$  is set to 4000 that is larger than 2650 found by stability-preserving condition) are shown in the third column of Fig. 5a. Compared with results of the posterior stabilization method, the responses are stable and the method results in accurate estimation of the step response (see the zoom-in areas).

3) *Net3 network*: We present the results for a larger network, namely Net3 shown in Fig. 3c, under zero ( $\mathbf{x}_o = 0$  mg/L) or non-zero initial chlorine concentrations ( $\mathbf{x}_o = 0.5$  mg/L). Two booster locations (J237 and J247) in Fig. 3c inject chlorine with a rate of 50 mg and 40 mg per sampling time ( $\Delta t = 0.25$  second), that are two step signals with amplitudes of 50 and 40. Fig. 5b presents step responses of full-order model and reduced-model from SBPOD with non-zero initial conditions at three sensors (J255, J241, and J249). The full order of Net3 is  $n_x = 29347$ , and the reduced-order  $n_r$  from SBPOD is 453. Note that the intractable BT method fails to produce a reduced-order model after 12 hours of simulation; thus, the results of BT are not included in Fig. 5b. Furthermore, POD method is not

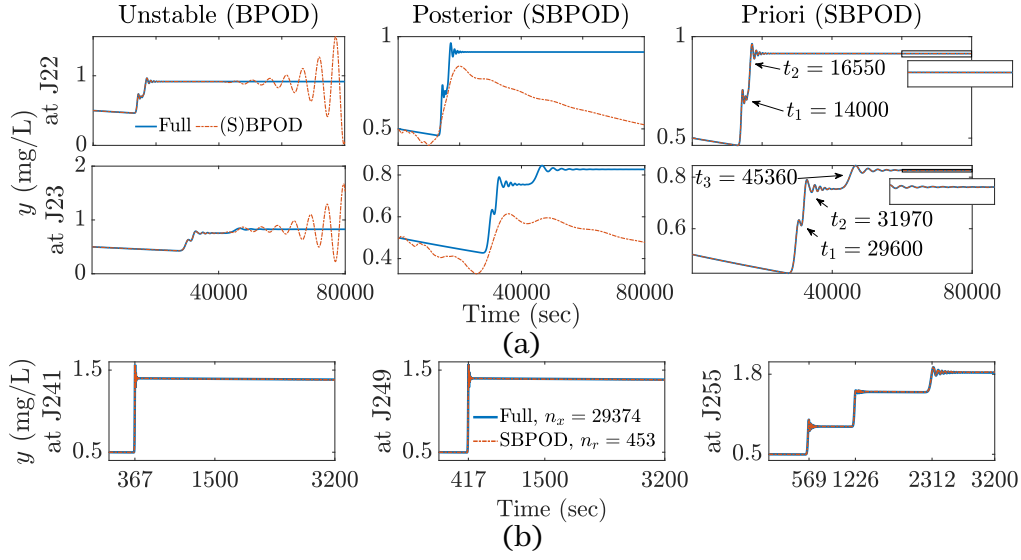


Fig. 5. (a) step responses from BPOD and SBPOD with two stabilization methods for Net1 under non-zero initial conditions; (b) step responses from different output locations of reduced-order models via BPOD methods for Net3 under on-zero initial chlorine concentrations.

suitable for non-zero initial condition case, and its results are not shown in Fig. 5b either. For zero initial condition case, both POD and SBPOD reduce system order successfully, and the reduced-orders  $n_r$  are 689 and 491. Instead of presenting step response figures, we showcase the RMSEs of MOR methods in Tab. III.

### C. Computational time

Tab. III presents the computational time for computing the reduced-order models for the three algorithms for different networks. We note that the computational time of classical BPOD and SBPOD are nearly identical (seeing that the computation of  $m$  requires little to no time), and only results of SBPOD are shown for simplicity. The full-order  $n_x$  and the length of snapshots  $m$  for the three-node network are set as 154 and 1000 when testing computational time. For Net1, they are 1293 and 7500; for Net3, these values are 29374 and 14000.

From Tab. III, it can be seen that BT is intractable when  $n_x$  is large since it fails to give a reduced-order model for Net3 after running the algorithm for 12 hours. The computational time for SBPOD is less than the one by POD for Net3 which interestingly contradicts the theoretical complexity from Tab. I. This happens because we implement a technique to avoid matrix-matrix multiplication by exploiting the structure of the block Hankel matrix in (11). Consequently, this results in a reduction of the computational time of the SBPOD method. The Github codes [47] also include a detailed description of how the problem structure is exploited to reduce the computational time. The details are omitted in this paper for brevity.

### D. SBPOD-Based MPC vs. Full-Order MPC

One of the main objectives of investigating MOR algorithms for water quality dynamics, besides the ability to perform forward state-simulations more efficiently, is to showcase the algorithms' performance in the context of real-time feedback

control (i.e., controlling chlorine concentrations  $x(k)$  through the installed booster station controllers  $u(k)$ ) or state observation and estimation (i.e., estimating the states  $x(k)$  via noisy sensor measurements  $y(k)$ ). To that end, the objective of this section is to test the applicability of the presented SBPOD-generated, reduced-order model within a model predictive control (MPC) framework that regulates chlorine concentrations through controlling dosages of injected chlorine at the booster stations. We specifically focus on SBPOD herein as it produces the best performance (in terms of accuracy under different initial conditions, computational time, and stability-preserving) among the other tested methods.

The motivation for using the reduced-order model is as follows: the regulation of chlorine concentrations with strict state constraints (e.g., as the ones mandated by the US EPA) in a water network can be posed as a constrained MPC optimization—as we have shown in our recent work [1]. We have also showcased that the control routine for the full-order system becomes computationally intractable when considering MPC with state- and input-constraints and a large control horizon (e.g., 30 minutes) even for a mid-size network (e.g., Net1). To solve this issue, we have relaxed the strict constraints and converted the intractable constrained MPC into an unconstrained and tractable one. However, this leads to another issue. That is, the unconstrained MPC cannot guarantee the state and input bounds, which is one of the limitations of our previous work [1]. In this paper, we investigate the following research question in this section:

*Can the water system operator utilize the SBPOD-generated, reduced-order model with constrained MPC to determine the desired dosages of injected chlorine? How would such an application compare to using a similar control routine for the full-order model? Would utilizing the reduced-order model result in a significant increase in the operation cost of controlling chlorine concentrations? How does*

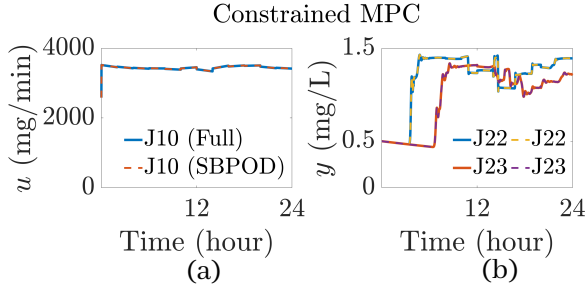


Fig. 6. The comparisons of (a) control actions and (b) subsequent chlorine concentrations solved by the full- and reduced-order model (i.e., SBPOD) for the constrained MPC problems.

*that impact the controlled chlorine concentrations?*

To answer these questions, we use the Net1 network as an illustration. Similar performance is exhibited for other networks. Note that we set a small control horizon parameter, that is 5 minutes, for Net1 to make optimization problems with full-order models tractable so that the control performance of full-/reduced- order models based constrained MPC can be tested and compared. First, the full-order models are computed for a time horizon of 24 hours. This is then followed by computing the SBPOD-generated reduced-order models via the priori stabilization method. Second, the optimal control signals of chlorine dosages for both the full- and reduced-order models are computed via the constrained MPC from study [1]. The reference signal for the MPC algorithm to track is set as 1.5 mg/L. That is, the goal of the controller is to maintain the chlorine concentration at around 1.5 mg/L at J22 and J23 (the sensor nodes in Net1) by controlling the injected mass rate of chlorine  $u(k)$  at J10 (the only control node in Net1). Finally, the control signals are then applied to the water quality simulation in the Net1 network. For brevity, we do not reproduce the details here; the interested reader can examine the provided Github codes for the parameters and settings [47].

To assess the performance of the SBPOD-based chlorine dosage constrained model predictive controller, in comparison with the similar MPC that utilizes the more computationally expensive full-order one, we test for three metrics: the computational time of solving optimization problems, the difference in the cost function of running the controllers (i.e., the operational cost of running the controller), and the evolution of the chlorine concentration. The three metrics are assessed for the two approaches: (i) the SBPOD-based constrained MPC and (ii) the standard constrained MPC for the full-order model. Fig. 6a and Fig. 6b show the injected mass rate  $u(t)$  and the corresponding chlorine concentrations at J22 and J23 for (i) and (ii) for the entire time horizon for both the reduced- and full-order models. The RMSEs for control input  $u$  and output  $y$  between the full-order model and reduced-order one are  $5.95 \times 10^{-2}$  and  $3.63 \times 10^{-6}$ . The computational time of solving constrained MPC problems with full- and reduced-order models are 50.7 seconds and 31.7 seconds. Both operational costs from these two different controller are  $4.49 \times 10^5$ .

By examining the results, we observe that the reduced-order model produces nearly identical performance (in terms of chlorine concentration dynamics and the optimal chlorine dosages)

when compared with the full-order one, while incurring smaller computational time. We note here that solving constrained MPC with full-order models for Net3, a larger network than Net1, is computationally intractable. After a running time of 12 hours, solutions are not obtained since this optimization problem has at least  $n_x = 29374$  optimization variables and millions of constraints. However, with the reduced-order model, the constrained optimization problem only has  $n_r = 453$  optimization variables and at around ten thousand constraints making it computationally tractable. In short, the number of optimization variables and constraints of constrained MPC with reduced-order models is small regardless of the size of the network/full-order model since parameter  $n_r$  is usually within a thousand after MOR. The reduced-order model enables water system operators to use constrained MPC that guarantees the input and output bounds while remaining tractable for larger networks.

## VII. WHAT SHOULD THE OPERATOR USE? CONCLUSIONS AND FUTURE WORK

The presented research in this paper explores the potential of model order reduction for water quality dynamics in drinking water networks. The presented stability-preserving BPOD (SBPOD) algorithm is more usable than its counterparts due to its accuracy, low computational burden, tolerance to initial conditions, and stability-insuring property. Compared with the classical BT method, SBPOD is computationally tractable while yielding accurate estimates of output measurements of chlorine concentrations. Furthermore, compared to POD or BPOD, SBPOD handles a variety of initial conditions while ensuring stability and accuracy of estimated outputs from the reduced-order models. Finally, we demonstrate that when using the proposed SBPOD within an MPC framework to control chlorine concentrations, the water system operator can use the SBPOD-generated reduced-order model instead of the full-order one without incurring any significant losses in the operational cost of controlling the network, while saving orders of magnitude in computational time. This makes control algorithms such as MPC amenable to real-time implementation—as a result of utilizing the reduced-order models. In short, and through the comprehensive testing and theoretical discussions, the water system operator can utilize the SBPOD to perform accurate simulations and constrained, feedback control for water quality dynamics.

Future research directions includes investigating model order reduction algorithms for water quality dynamics that involve multi-species interactions, in comparison with models of focusing on single-species ones. A state-space model for water quality dynamics in a multi-species framework would likely yield a nonlinear state-space description that then necessitate designing MOR algorithms for nonlinear dynamics. To that end, future work will focus on (i) thorough modeling of multi-species reaction dynamics and (ii) reducing such dynamics to lower order models.



## REFERENCES

- [1] S. Wang, A. F. Taha, and A. A. Abokifa, "How effective is model predictive control in real-time water quality regulation? state-space modeling and scalable control," *Water Resources Research*, 2020, in press.
- [2] P. Boulous, K. Lansey, and B. Karney, *Comprehensive Water Distribution Systems Analysis Handbook for Engineers and Planners*. MWH Soft, Incorporated, 2006.
- [3] C. W. Rowley, "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005.
- [4] U. Baur, P. Benner, and L. Feng, "Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective," *Archives of Computational Methods in Engineering*, vol. 21, no. 4, pp. 331–358, 2014.
- [5] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, vol. 40, no. 11, pp. 2323–2330, 2002.
- [6] L. Montier, T. Henneron, B. Goursaud, and S. Clenet, "Balanced Proper Orthogonal Decomposition Applied to Magnetoquasi-Static Problems Through a Stabilization Methodology," *IEEE Transactions on Magnetics*, vol. 53, no. 7, 2017.
- [7] V. M. Adamjan, D. Z. Arov, and M. Kreĭn, "Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem," *Mathematics of the USSR-Sbornik*, vol. 15, no. 1, p. 31, 1971.
- [8] K. Glover, "All optimal hankel-norm approximations of linear multivariable systems and their  $L^\infty$  error bounds," *International journal of control*, vol. 39, no. 6, pp. 1115–1193, 1984.
- [9] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE transactions on automatic control*, vol. 26, no. 1, pp. 17–32, 1981.
- [10] L. Sirovich, "Turbulence and the dynamics of coherent structures. i. coherent structures," *Quarterly of applied mathematics*, vol. 45, no. 3, pp. 561–571, 1987.
- [11] E. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, 1997.
- [12] A. Antoulas, J. Ball, J. Kang, and J. Willems, "On the solution of the minimal rational interpolation problem," *Linear Algebra and its Applications*, vol. 137, pp. 511–573, 1990.
- [13] C. A. Beattie and S. Gugercin, "Interpolation theory for structure-preserving model reduction," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4204–4208.
- [14] Z. Bai, "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems," *Applied Numerical Mathematics*, vol. 43, no. 1-2, pp. 9–44, 2002.
- [15] K. Gallivan, A. Vandendorpe, and P. Van Dooren, "Model reduction and the solution of sylvester equations," *MTNS, Kyoto*, vol. 50, 2006.
- [16] S. Gugercin, "An iterative svd-krylov based method for model reduction of large-scale dynamical systems," *Linear Algebra and its Applications*, vol. 428, no. 8-9, pp. 1964–1986, 2008.
- [17] S. Lall and C. Beck, "Error-bounds for balanced model-reduction of linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 946–956, 2003.
- [18] J. L. Lumley, *Stochastic tools in turbulence*. Courier Corporation, 2007.
- [19] P. Astrid, S. Weiland, K. Willcox, and T. Backx, "Missing point estimation in models described by proper orthogonal decomposition," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2237–2251, 2008.
- [20] B. Ulanicki, A. Zehnpfund, and F. Martinez, "Simplification of water distribution network models," in *Proc., 2nd Int. Conf. on Hydroinformatics*. Balkema Rotterdam, Netherlands, 1996, pp. 493–500.
- [21] U. Shamir and E. Salomons, "Optimal real-time operation of urban water distribution systems using reduced models," *Journal of Water Resources Planning and Management*, vol. 134, no. 2, pp. 181–185, 2008.
- [22] A. Preis, A. J. Whittle, A. Ostfeld, and L. Perelman, "Efficient hydraulic state estimation technique using reduced models of urban water networks," *Journal of Water Resources Planning and Management*, vol. 137, no. 4, pp. 343–351, 2011.
- [23] L. Perelman and A. Ostfeld, "Water distribution system aggregation for water quality analysis," *Journal of water resources planning and management*, vol. 134, no. 3, pp. 303–309, 2008.
- [24] P. D. Lax and B. Wendroff, "Difference schemes for hyperbolic equations with high order of accuracy," *Communications on pure and applied mathematics*, vol. 17, no. 3, pp. 381–398, 1964.
- [25] C. Chen, *Linear System Theory and Design*, ser. The Oxford Series in Electrical and Computer Engineering. Oxford University Press, Incorporated, 2013. [Online]. Available: <https://books.google.com/books?id=XyPAoAEACAAJ>
- [26] C. Himpe, "emgr-The empirical Gramian framework," *Algorithms*, vol. 11, no. 7, pp. 1–27, 2018.
- [27] —, "Comparing (Empirical-Gramian-Based) Model Order Reduction Algorithms," 2020. [Online]. Available: <http://arxiv.org/abs/2002.12226>
- [28] S. Lall, J. E. Marsden, and S. Glavaški, "Empirical model reduction of controlled nonlinear systems," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 2598–2603, 1999.
- [29] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [30] D. Amislem and C. Farhat, "Stabilization of projection-based reduced-order models," *International Journal for Numerical Methods in Engineering*, vol. 91, no. 4, pp. 358–377, 2012.
- [31] J.-R. Li and J. White, "Low rank solution of lyapunov equations," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, pp. 260–280, 2002.
- [32] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 1999, pp. 507–516.
- [33] X. Feng, W. Yu, and Y. Li, "Faster matrix completion using randomized svd," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 608–615.
- [34] S. Prajna, "Pod model reduction with stability guarantee," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 5. IEEE, 2003, pp. 5254–5258.
- [35] C. W. Rowley, T. Colonius, and R. M. Murray, "Model reduction for compressible flows using POD and Galerkin projection," *Physica D: Nonlinear Phenomena*, 2004.
- [36] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist, "Stable Galerkin reduced order models for linearized compressible flow," *Journal of Computational Physics*, 2009.
- [37] I. Kalashnikova and M. Barone, "On the stability and convergence of a galerkin reduced order model (rom) of compressible flow with solid wall and far-field boundary treatment," *International journal for numerical methods in engineering*, vol. 83, no. 10, pp. 1345–1375, 2010.
- [38] I. Kalashnikova and S. Arunajatesan, "A stable galerkin reduced order model for compressible flow," in *10th World Congress on Computational Mechanics (WCCM), WCCM-2012-19407, Sao Paulo, Brazil*, 2012.
- [39] S. Sirisup and G. E. Karniadakis, "A spectral viscosity method for correcting the long-term behavior of pod models," *Journal of Computational Physics*, vol. 194, no. 1, pp. 92–116, 2004.
- [40] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, "Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison," *Computer Methods in Applied Mechanics and Engineering*, vol. 237, pp. 10–26, 2012.
- [41] I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, and M. Barone, "Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment," *Computer Methods in Applied Mechanics and Engineering*, vol. 272, pp. 251–270, 2014.
- [42] B. N. Bond and L. Daniel, "Guaranteed stable projection-based model reduction for indefinite and unstable linear systems," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2008.
- [43] M. Benosman, J. Borggaard, and B. Kramer, "Robust pod model stabilization for the 3d boussinesq equations based on lyapunov theory and extremum seeking," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 1827–1832.
- [44] A. M. Lyapunov, "The general problem of the stability of motion," *International journal of control*, vol. 55, no. 3, pp. 531–534, 1992.
- [45] L. A. Rossman et al., "EPANET 2: users manual," 2000.
- [46] D. G. Eliades, M. Kyriakou, S. Vrachimis, and M. M. Polycarpou, "Epanet-matlab toolkit: An open-source software for interfacing epanet with matlab," in *Proc. 14th International Conference on Computing and Control for the Water Industry (CCWI)*, The Netherlands, Nov 2016, p. 8.
- [47] ShenWang9202, "ShenWang9202/MOR4WQM: MOR FOR Water quality dynamics," Feb. 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.4554174>
- [48] M. Heinkenschloss, T. Reis, and A. C. Antoulas, "Balanced truncation model reduction for systems with inhomogeneous initial conditions," *Automatica*, vol. 47, no. 3, pp. 559–564, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.automatica.2010.12.002>