DATA-DRIVEN PREDICTIVE CONTROL STRATEGIES OF WATER SYSTEMS

Saskia A. Putri[a], Faegheh Moazeni[b,*], and Javad Khazaei [c]

[a] *Civil & Environmental Engineering Department, Lehigh University, 1 West Packer Ave, Bethlehem, PA 18015. USA.*
sap322@lehigh.edu

[b] *Civil & Environmental Engineering Department, Lehigh University, 1 West Packer Ave, Bethlehem, PA 18015. USA.*
moazeni@lehigh.edu

[c] *Electrical & Computer Engineering Department, Lehigh University, 19 Memorial Drive West Bethlehem, PA 18015. USA*
khazaei@lehigh.edu

Abstract

Model predictive control (MPC) application has gained attention in water distribution systems (WDSs) due to its effectiveness in controlling multi-variable, highly nonlinear, and complex systems, all of which are inherent traits of WDSs, but rely on accurate dynamic models. Model misrepresentation can lead to nonoptimal solutions. Nevertheless, the emergence of utilizing high-resolution metering devices and historical data has enabled data-driven WDSs model identification, significantly reducing modeling challenges of MPC without losing the accuracy or robustness of the closed-loop control system. This paper contributes to developing a data-driven MPC framework for WDSs, eliminating the reliance on the WDSs' physical models. Sparse regression is utilized to identify the dynamics of WDSs from the available sensory data. The identified data-driven model is then embedded in MPC frameworks with multiple prediction model formulations: 1) linear-time in-

*Corresponding author. Tel: +1-610-758-3530. Email: moazeni@lehigh.edu (Faegheh Moazeni).

variant, 2) linear time-varying, 3) and nonlinear models. Four interconnected water tank systems representing the nonlinear and cross-coupling of WDSs are selected as the test system to evaluate the effectiveness of the proposed data-driven MPC frameworks. Contrary to the existing WDS control designs, which heavily rely on detailed WDS models and extensive historical data, the proposed framework accurately captures the nonlinear dynamics using minimum available measurements.

Keywords: Data-driven system identification; Sparse regression; Optimal scheduling; Nonlinear control; Water distribution systems;

1. Introduction

The resilience of the water supply system is imperative to adapt to hydro-climatic variations and socio-economic changes. Two main challenges underline this importance: 1) The exacerbation of climate change, which can lead to water infrastructure damage and disrupt the distribution, and 2) shifts in social and economic conditions, leading to an increase in water demand. These factors place significant pressure on the governments and involved parties to cope with these challenges [1]. Furthermore, ensuring safe water supply is a complex procedure in and of itself, particularly for water distribution systems (WDSs). WDSs comprise interconnected components such as pipes, valves, pumps, storage tanks, and other components [2] contributing to their operational complexity to ensure the safe and stable distribution of water. Furthermore, the prevalent use of pumps in WDSs contributes to elevated power consumption and operational costs [3]. Without suitable management, this could lead to unsustainable WDSs operations. To address these concerns, gaining adaptive control, optimal operational scheduling of WDSs, and a complete understanding of their behavior are vital to enhancing their resiliency and sustainability.

Various control techniques have been explored to design and provide optimal management of WDSs [1]. Four control actions can be implemented in the water system: 1) feedforward control, 2) feedback control, 3) optimal control, and 4) heuristic control, which is based on a rules-of-thumb approach or experience-based strategy to control the system [4]. However, to fully incorporate the complicated nature of WDSs, a control approach capable of handling multi-variable and multi-objective models, system constraints, and disturbances is required [5]. These aspects can be categorized under Model Predictive Control (MPC). In addition to that, MPC also integrates the aforementioned three control actions. First, MPC employs feedback control to minimize the deviation between desired setpoints and the controlled output. Furthermore, it utilizes feedforward control to adapt the control signal based on measured or predicted disturbances [4]. Additionally, MPC employs optimal control to formulate a control law that minimizes a quadratic objective function developed from the system's state space model [6]. Consequently, MPC emerges as an advanced and preferred control strategy.

As a model-based controller, MPC utilizes the dynamic and static models of the system being controlled as the prediction model to optimize future control actions [7, 8]. A sequence of future control actions is determined by solving a finite-time horizon of an online model-based optimization at each sampling instant, using the plant's current state as the initial state [9] subject to the objective functions (performance goals of the controlled system) and constraints at future time instants [7]. MPC follows the receding horizon framework by applying only the first value of the optimized control inputs and disregarding the rest of the trajectory [10]. The internal model, used in MPC implementation to obtain the predicted output, can be represented in the state space form, which can be linear or non-linear models [11].

Different strategies of MPC can be implemented that differ from the prediction models, such as 1) the linear time-invariant (LTI) model, where the structure of the model does not change over time [12]; 2) the linear time-varying (LTV) model, where the dynamic behavior of the system varies over time due to varying operating points [13], and 3) nonlinear model that accounts for the nonlinear features present in the controlled system [14]. Linear prediction model has been widely used [15] as it offers a tractable control problem via a convex optimization problem that guarantees a global minimum with low computational cost [13, 16]. However, it may suffer from a lack of representation of an actual system that exhibits nonlinear behavior [17]. This motivates the application of nonlinear MPC that enables the control of nonlinear systems and offers a more accurate system representation [15, 18]. Nonetheless, nonlinear MPC (NMPC) also has its own drawbacks. NMPC applies a non-convex optimization problem, where the solution may converge to a local optimum, and provides computation-intensive control algorithms [11]. Therefore, finding a suitable prediction model to control WDSs is critical to provide a trade-off between model accuracy, computation time, and tractability of the control problem [11].

In spite of MPC's benefits, one of its liabilities is its heavy reliance on the dynamic model of the controlled system [17]. Consequently, a mismatch between the plant model and the disturbances excluded from the model may lead to unfavorable control decisions [6, 19]. Furthermore, existing hydraulic models may not accurately reflect WDSs due to pipe deterioration (changing the roughness coefficient of the pipe), water loss, operational scheduling, and system topology, to name a few [20]. These divergences contribute to the difficulty in yielding optimal control solution for WDSs. This is because, in Considering the multi-variate, nonlinear, complex, and highly coupled dynamics of assets in WDSs [21], as well as the presence of varying disturbances, and time-evolving objectives, the control-oriented model formulation of MPC continues to be more challenging. Nevertheless, this can be solved by leveraging data availability from the plant measurement.

The drive to optimize the operation and management of WDSs, along with the goal of improving water infrastructure, has led to the development of advanced monitoring technologies. This has resulted in the deployment of smart sensors, including smart metering devices and wireless sensors [3]. Consequently, real-time and large data availability of WDSs can be obtained. This presents an opportunity to develop enhanced data-driven models that capture these systems' complex and nonlinear nature, including their response to disturbances. Previous studies have explored data-driven modeling approaches for various aspects of water systems. A wavelet-based artificial neural network technique to predict daily water levels in WDSs was investigated in [22]. In [23], model identification of a hysteresis-controlled pump using sparse identification of nonlinear dynamics was studied and validated on a single input and single output system (SISO). Multitask multi-view learning methods to forecast water quality were employed by [24] on an hourly basis by considering the spatial and spatiotemporal correlations between water treatment facilities. Other existing data-driven approaches include identifying water quality models via subspace identification method (SIM) [25], estimating drip tape irrigation discharge using temperature and pressure measurements [26], identifying reservoir operations using artificial neural networks and support vector regression [27], predicting water quality using various machine learning methods [28, 29], reduced network model to control WDSs using radial basis function neural models (RBFNN) [30], and forecasting reservoir water inflow using neural networks [31].

However, these studies have primarily focused on specific components or processes within the system and have not addressed the identification of nonlinear dynamics at a system level. Furthermore, many existing approaches rely on machine learning techniques that require extensive training data and lack interpretability, making them less suitable for system-level control in WDSs. In addition, given the nonlinearities and uncertainties of WDSs operation, it is not known whether a data-driven MPC can provide tracking guarantees and enhanced control performance to the optimal operation of WDSs. Therefore, there is a need for research that utilizes data-driven techniques to identify the nonlinear dynamics of WDSs at a system level, enabling more effective control and management of these complex systems.

1.1. Contributions of this work

To address the existing challenges, this work investigates the application of sparse identification theory for data-driven model identification in WDSs. In addition, to analyze whether a data-driven MPC can provide tracking guarantees and improved control performance in WDSs, we have designed three different techniques for formulating the optimal control problem in MPC to actuate the predicted data-driven dynamics and find suitable prediction models that comprehensively represent WDSs. Sparse identification theory is a viable method for accurately modeling the nonlinear dynamics of unknown systems [32–34]. It reduces training time and the need for neural networks for control and identification. To our best knowledge, no existing work has reported data-driven modeling of nonlinear dynamics of WDSs using sparsity-promoting techniques and implementing the predicted nonlinear dynamics to different approaches of MPC. Therefore, the main contributions of this paper can be summarized as follows:

1. A data-driven based on sparse regression technique is developed to identify the nonlinear dynamics of the benchmark quadruple water tank process that closely represent the nonlinear dynamics of WDSs.
2. Data-driven control of nonlinear WDSs dynamics using various MPC strategies, including linear time-invariant MPC (LMPC), linear time-varying MPC through successive linearization technique (SLMPC), and nonlinear MPC (NMPC) is investigated.
3. A comparative analysis involving computational burden, tracking performance, and robustness of data-driven MPC strategies with numerical results that further highlight the performance of different data-driven MPC approaches is provided.
4. Detailed descriptions of the MPC algorithms for each control law are provided, along with a thorough formulation of the MPC solution method utilizing sequential quadratic programming (SQP).

The rest of the paper is organized as follows: Section II describes the proposed model description in addition to the dynamics of the quadruple tank systems, sparse identification of dynamics, and optimal control problem formulation (OCP) of MPC. Several case studies are presented and discussed in Section III to validate the proposed model identification through the sparse regression technique with different strategies of control, while Section IV concludes the paper.

2. Methodology

2.1. Proposed Model Description

Fig. 1 exhibits an overview of the proposed data-driven model identification via the sparse regression-based nonlinear dynamics (SR-based) method combined with MPC to control a system that closely resembles WDSs. This work uses a quadruple tank process (QTP) from [35] that depicts simplified WDSs, a set of interconnected water tanks subject to external in and outflows. First, the sparse regression model identification technique will be utilized to identify the dynamic equations of the system solely from measurements. It involves taking measurements from the outputs of the plant (y_i), collecting the applied control inputs (u_i), constructing a library of functions that might represent the system's dynamics, and solving a sequentially thresholded least-square optimization problem using a sparse regression technique. Then, the predicted dynamics are conveyed to the MPC framework to generate optimal control problem formulation exploiting various MPC techniques, such as linear, successive linearization, and nonlinear MPCs. Different techniques are proposed to provide different trade-offs in utilizing MPC as the control strategy in WDSs. The process is described in details in Sections 2.3 and 2.4.

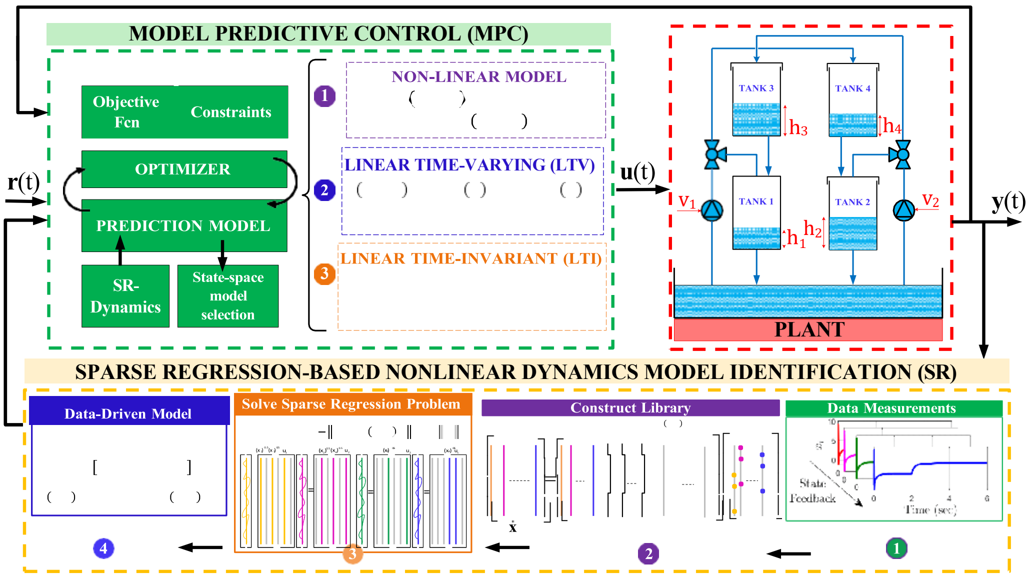


Fig. 1: Proposed data-driven control framework for identifying and controlling the dynamics of a quadruple tank process.

2.2. Application of SR-MPC to a benchmark system

2.2.1. Dynamic Model of Quadruple Tank Process (QTP)

The dynamics of each tank in the quadruple water tank process in Fig. 2 is derived from the combination of mass balance in the tank (assuming constant density) and Bernoulli's theorem with additional flow control valve (FCV) and pumps as the controllers. The governing equations can be represented by the following set of differential equations [35]:

$$\dot{h}_1(t) = \frac{-s_1}{S_1}\sqrt{2gh_1(t)} + \frac{-s_3}{S_1}\sqrt{2gh_3(t)} + \frac{\gamma_1 k_1}{S_1}v_1(t) \tag{1a}$$

$$\dot{h}_2(t) = \frac{-s_2}{S_2}\sqrt{2gh_2(t)} + \frac{-s_4}{S_2}\sqrt{2gh_4(t)} + \frac{\gamma_2 k_2}{S_2}v_2(t) \tag{1b}$$

$$\dot{h}_3(t) = \frac{-s_3}{S_3}\sqrt{2gh_3(t)} + \frac{(1-\gamma_2)k_2}{S_3}v_2(t) \tag{1c}$$

$$\dot{h}_4(t) = \frac{-s_4}{S_4}\sqrt{2gh_4(t)} + \frac{(1-\gamma_1)k_1}{S_4}v_1(t) \tag{1d}$$

where S_i is the cross-sectional area of tank i, s_i is a cross-sectional area of the tank's orifice in tank i, h_i is the water level of tank i, v_i is the voltage applied to pump i with a corresponding flow $k_i v_i$. Parameters $\gamma_i \in [0, 1]$ are determined from the settings of the valves. As illustrated in Fig. 2, $\gamma_1 k_1 v_1$ and $(1 - \gamma_1)k_1 v_1$ represent tanks 1 and 4 inflows, respectively. Similarly, the water flow to tank 2 is $\gamma_2 k_2 v_2$, and the water flow to tank 3 is $(1 - \gamma_2)k_2 v_2$. The system is uniquely designed to exhibit the effect of multivariable zero on the system behavior with zero location either on the left or right-hand plane by changing the valve positions [35]. The system is represented as minimum phase (left-hand plane zero) for $1 < \gamma_1 + \gamma_2 < 2$ and non-minimum phase for $0 < \gamma_1+\gamma_2 < 1$ (right-hand plane zero). The acceleration of gravity is denoted by g and output measurements can be computed with $k_c h_1$ and $k_c h_2$. Table 1 displays the constant parameter values used in this work adopted from [35].

Table 1: QTP Parameters from [35]

Parameters	Unit	Qty
$S_1,\ S_3$	cm^2	28
$S_2,\ S_4$	cm^2	32
$s_1,\ s_3$	cm^2	0.071
$s_2,\ s_4$	cm^2	0.057
h_i^{max}	cm	15
h_i^{min}	cm	0
kc	V/cm	1
g	cm/s^2	981

By re-arranging equations in Eq.(1), the state space model of the QTP can be expressed as follows:

$$\dot{\boldsymbol{x}} = \underbrace{\begin{bmatrix} -\frac{s_1\sqrt{2g}}{S_1}\sqrt{x_1} + \frac{s_3\sqrt{2g}}{S_1}\sqrt{x_3} \\ -\frac{s_2\sqrt{2g}}{S_2}\sqrt{x_2} + \frac{s_4\sqrt{2g}}{S_2}\sqrt{x_4} \\ -\frac{s_3\sqrt{2g}}{S_3}\sqrt{x_3} \\ -\frac{s_4\sqrt{2g}}{S_4}\sqrt{x_4} \end{bmatrix}}_{\mathbf{f}(\boldsymbol{x})} + \underbrace{\begin{bmatrix} \frac{\gamma_1 k_1}{S_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{S_2} \\ 0 & \frac{(1-\gamma_2)k_2}{S_3} \\ \frac{(1-\gamma_1)k_1}{S_4} & 0 \end{bmatrix}}_{\mathbf{g}(\boldsymbol{x})} \boldsymbol{u} \tag{2a}$$

$$\boldsymbol{y} = \underbrace{k_c \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T}_{\mathbf{h}(\boldsymbol{x})} \tag{2b}$$

which is equivalent to

$$\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}) + \mathbf{g}(\boldsymbol{x})\boldsymbol{u} \tag{2c}$$

$$\boldsymbol{y} = \mathbf{h}(\boldsymbol{x}) \tag{2d}$$

In Eq.(2), $\boldsymbol{x} = [h_1\ h_2\ h_3\ h_4]^T$ is the state vector, while $\boldsymbol{y} = k_c\boldsymbol{x}$ is the output vector, and $\boldsymbol{u} = [v_1\ v_2]^T$ is the input vector of the system, which includes the voltage applied to the pumps.

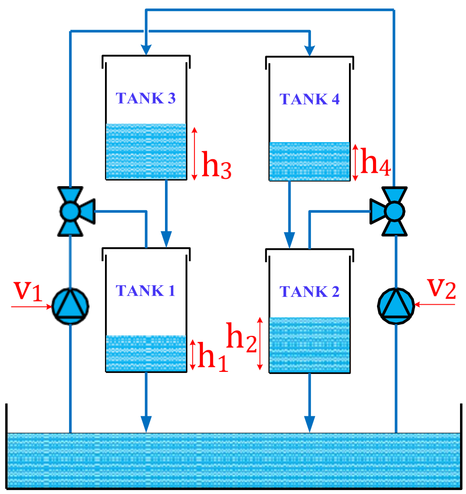


Fig. 2: Schematic of a quadruple tank process.

2.2.2. Operating Points

One of the objectives of this work is to identify the dynamic model presented in Eq. (2) solely from the available measurements of the states, assuming that detailed information regarding the QTP system and its parameters is unavailable. Two distinct operating points will be examined to assess the robustness of the proposed model identification framework. These operating points were intentionally chosen based on the system's behavior, with one exhibiting minimum phase (MP) characteristics and the other demonstrating nonminimum phase (NMP) behavior. This selection allows for a comprehensive validation of the proposed framework's effectiveness. The two operating points are listed in Table 2.

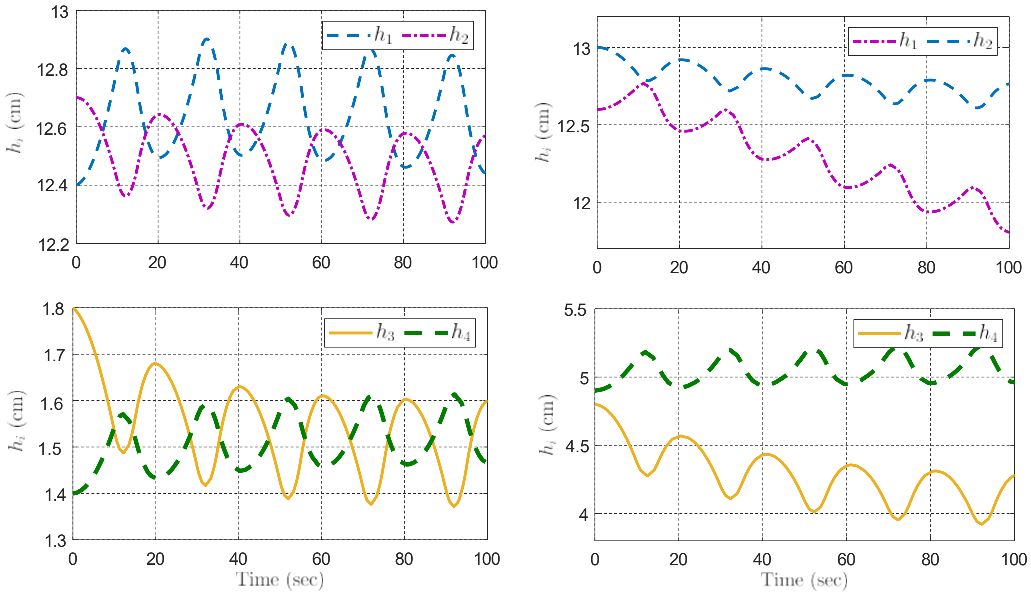
Table 2: Operating points of quadruple tank process.

Parameter	MP operating point	NMP operating point
h_1, h_2 [cm]	12.4, 12.7	12.6, 13
h_3, h_4 [cm]	1.8, 1.4	4.8, 4.9
v_1, v_2 [V]	3, 3	3.15, 3.15
k_1, k_2	3.33, 3.35	3.14, 3.29
γ_1, γ_2	0.7, 0.6	0.43, 0.34

Simulations were conducted to evaluate the system's operation in these two operating points by perturbing the control inputs (pump voltages, v_i) with a repetitive sequence waveform between 2 - 4 V (to activate the dynamic modes). Fig. 3 depicts the evolution of the states at two operating points over 100 s simulation in MATLAB.

2.3. Data-Driven Identification of Nonlinear Dynamics

To identify the nonlinear dynamical models of the studied QTP using measurements, the first step involves estimating the state derivatives ($\dot{\boldsymbol{x}}$). Subsequently, a library of candidate functions ($\psi_i(x)$) is constructed to describe the temporal changes of the state variables. In cases without prior knowledge about the system's dynamics, an extended basis of candidate functions can be chosen to accommodate all potential functions. Given that most dynamical systems exhibit a few nonlinear terms in their dynamics, techniques that promote sparsity can effectively identify the candidate functions that significantly contribute to the system's dynamics. Therefore, this paper utilizes a sparse regression-based nonlinear dynamics model identification (SR), initially proposed in [32], and is further explained step-by-step in



(a) MP operating point.

(b) NMP operating point.

Fig. 3: Trajectory of the states in different operating points.

the following sections.

2.3.1. Measurements

SR technique utilizes symbolic regression and sparse representations to determine the system's dynamics. This approach depends upon the fact that many dynamical systems which are represented by differential equations in the form of $\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x})$ have relatively few terms on the right-hand side [33]. In this work, the actual dynamics of the studied QTP is represented by $\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}) + \mathbf{g}(\boldsymbol{x})\boldsymbol{u}$, where $\boldsymbol{x}(\mathrm{t}) \in \mathrm{R}^{nx}$ is the state vector, $\boldsymbol{u}(\mathrm{t}) \in \mathrm{R}^{nu}$ is the control input vector, and $\mathbf{f}(\boldsymbol{x}(\mathrm{t}), \boldsymbol{u}(\mathrm{t})) := \mathrm{R}^{nx} \times \mathrm{R}^{nu} \rightarrow \mathrm{R}^{nx}$ maps a space of control inputs and states dimension to a space n_x dimension. Therefore, by collecting m measurement samples from the water tank levels and pump inputs, the

QTP dynamics can be identified by a library of candidate functions, $\mathbf{\Psi} \in \mathbb{R}^{m\times p}$, where p denotes the number of the library functions. To identify the governing equations of the system in Eq. (2), a time-history of the tank levels (state vector) $\boldsymbol{x}(t)$, pump inputs $\boldsymbol{u}(t)$, and derivatives of the states $\dot{\boldsymbol{x}}(t)$ are collected. Since only $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ might be available in most real-world systems, the derivative measurements $\dot{\boldsymbol{x}}(t)$ must be estimated first. This can be accomplished by numerically calculating the derivatives from the state measurements. To achieve this, the measurement data is first sampled at m intervals $t_1, t_2, \dots, t_m$ and arranged into [34]:

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{x}^T(t_1) \\ \boldsymbol{x}^T(t_2) \\ \vdots \\ \boldsymbol{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_{n_x}(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_{n_x}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \dots & x_{n_x}(t_m) \end{bmatrix} \quad (3)$$

and inputs for t_m samples are written into a matrix $\mathbf{U}$ such that

$$\mathbf{U} = \begin{bmatrix} \boldsymbol{u}^T(t_1) \\ \boldsymbol{u}^T(t_2) \\ \vdots \\ \boldsymbol{u}^T(t_m) \end{bmatrix} = \begin{bmatrix} u_1(t_1) & u_2(t_1) & \dots & u_{n_u}(t_1) \\ u_1(t_2) & u_2(t_2) & \dots & u_{n_u}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \dots & u_{n_u}(t_m) \end{bmatrix} \quad (4)$$

Then, the measurements for derivatives can be approximated numerically from $\mathbf{X}$ by following the procedure described in the subsequent section.

2.3.2. Estimating the Derivatives, $\dot{\mathbf{X}}$

Differential and partial differential equations can be solved numerically using methods such as difference approximation. It involves approximating the derivatives of a smooth function using Taylor series expansions at specific mesh points. This study employs the central difference approximation due to its higher accuracy when dealing with smooth functions. Accordingly, $\dot{\mathbf{X}}$ can be approximated by [36]:

$$\dot{\mathbf{X}} \approx \frac{\mathbf{X}(j+1) - \mathbf{X}(j-1)}{2s_t} \tag{5}$$

where $\mathbf{X}(j+1)$ is the measured data at sample $j+1$ and s_t is the sampling time of the simulation or data collection platform.

2.3.3. Sparse Identification of System Dynamics

The states derivative data, obtained from utilizing the measured data $\mathbf{X} \in \mathrm{R}^{m \times nx}$, is a linear combination of columns from the candidate function (e.g., polynomials, or sinusoids) library expressed by entries of the matrix $\mathbf{\Xi} \in \mathrm{R}^{p \times nx}$ such that [33]:

$$\dot{\mathbf{X}} = \mathbf{\Psi}(\mathbf{X}, \mathbf{U})\mathbf{\Xi}. \tag{6}$$

Having estimated $\dot{\mathbf{X}}$, $\mathbf{\Psi}(\mathbf{X}, \mathbf{U})$ can be constructed by linear and nonlinear functions of the columns of $\mathbf{X}$ and $\mathbf{U}$. Furthermore, monomials and trigonometric functions are typically considered candidate functions for nonlinear systems. An example of such functions is represented in Eq. (7):

$$
\boldsymbol{\Psi}^T(\mathbf{X},\mathbf{U}) = \begin{bmatrix} \mathbf{M}_{0.5}(\mathbf{X},\mathbf{U}) \\ \mathbf{X} \\ \mathbf{U} \\ \mathbf{M}_2(\mathbf{X},\mathbf{U}) \\ \vdots \\ \sin(\mathbf{X},\mathbf{U}) \\ \cos(\mathbf{X},\mathbf{U}) \\ \sin(2\mathbf{X},\mathbf{U}) \\ \vdots \end{bmatrix} \tag{7}
$$

where $\mathbf{M_i}(\mathbf{X},\mathbf{U})$ corresponds a nonlinear combination of i-order monomials of $\mathbf{X}$ and $\mathbf{U}$. For instance, $\mathbf{M}_{0.5}(\mathbf{X},\mathbf{U})$ includes square-root functions that exist in the QTP system, or $\mathbf{M}_2(\mathbf{X},\mathbf{U})$ involves polynomials up to the second order. Once the estimation of $\dot{\mathbf{X}}$ and the determination of $\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})$ based on the available $\mathbf{X}$ and $\mathbf{U}$ are performed, then $\dot{\mathbf{X}} = \boldsymbol{\Psi}(\mathbf{X},\mathbf{U})\boldsymbol{\Xi}$ can be acquired by solving for the sparse vectors of coefficients in $\boldsymbol{\Xi}$. These coefficients determine the active terms in the $\dot{\mathbf{X}}$ dynamics. This is achieved by solving an optimization of the form [18]:

$$
\boldsymbol{\xi}_i = \arg\min_{\hat{\boldsymbol{\xi}}_i} \frac{1}{2}||\dot{\mathbf{X}}_i - \boldsymbol{\Psi}(\mathbf{X},\mathbf{U})\hat{\boldsymbol{\xi}}_i||_2 + \eta||\hat{\boldsymbol{\xi}}_i||_0 \tag{8}
$$

where $\boldsymbol{\xi}_i$ is the i-th column of $\boldsymbol{\Xi}$ represented by $\boldsymbol{\Xi}_i = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \dots & \boldsymbol{\xi}_p \end{bmatrix}^T$ and $\dot{\mathbf{X}}_i$ represents the i-th column of $\dot{\mathbf{X}}$. The objective function in (8) utilizes the L2 norm $||.||_2$ to minimize the error between the derivatives $\dot{\mathbf{X}}$ and estimated derivatives using calculated $\boldsymbol{\xi}_i$ through a least-squares problem and the L0 norm, $||.||_0$ minimizes the number of nonzero elements in $\boldsymbol{\xi}_i$ to promote sparsity in the coefficients matrix $\boldsymbol{\Xi}$. In addition, η is the regularizing parameter critical in the SR technique to promote sparsity degree in the solution, which can be tuned using various hyperparameter tuning [37].

The minimization problem of (8) is solved by the sequentially thresholded least squares algorithm, which is an iterative algorithm defined by [38]:

$$C^{\varepsilon} = \{ j \in [p] : |\xi_j^{\varepsilon}| \geq \eta \}, \forall \varepsilon \geq 0 \tag{9}$$

$$\hat{\boldsymbol{\xi}}_i^{0} = \boldsymbol{\Psi}(\mathbf{X},\mathbf{U})^{\ddagger}\dot{\mathbf{X}}_i \tag{10}$$

$$\boldsymbol{\xi}^{\varepsilon+1} = \underset{\hat{\boldsymbol{\xi}}_i \in \mathbb{R}^p : \mathrm{supp}(\boldsymbol{\xi}_i) \subseteq C^{\varepsilon}}{\operatorname{argmin}} \| \dot{\mathbf{X}}_i - \boldsymbol{\Psi}(\mathbf{X},\mathbf{U})\hat{\boldsymbol{\xi}}_i \|_2, \tag{11}$$

where ε is the iteration number, $\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})^{\ddagger}$ is the pseudo inverse of $\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})$ defined as:

$$\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})^{\ddagger} := [\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})^{T}\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})]^{-1}\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})^{T} \tag{12}$$

and the support set of $\boldsymbol{\xi}_j$ is defined by $\mathrm{supp}(\boldsymbol{\xi}_j) := \{j \in [p] : \xi_j \neq 0\}$. The coefficients $\boldsymbol{\xi}_j$ can be computed using the sparse regression formulation exhibited in **Algorithm 1**. If the purpose is to identify the signal $\mathbf{U}$ for the feedback control, i.e., $\mathbf{U} = H(s)\mathbf{X}$, where $H(s)$ is the transfer function of the controller, the matrix of inputs can be identified using $\mathbf{U} = \boldsymbol{\Psi}(\mathbf{X})\boldsymbol{\Xi}_{\mathbf{u}}$, where $\boldsymbol{\Psi}(\mathbf{X})$ is the matrix of candidate functions with the terms corresponding to $\mathbf{U}$ have been removed from $\boldsymbol{\Psi}(\mathbf{X},\mathbf{U})$ and $\boldsymbol{\Xi}_{\mathbf{u}}$ can be found using the sparse regression algorithm similar to $\boldsymbol{\Xi}$.

Algorithm 1 Sparse Regression-based Model Identification Algorithm

Input: Measurements $\mathbf{X}, \mathbf{U}$

Input: Estimated derivatives $\dot{\mathbf{X}}$

procedure Sparsity Promoting Algorithm

$\quad \boldsymbol{\Xi} = \boldsymbol{\Psi} \backslash \dot{\mathbf{X}}$ ▷ least-square solution

$\quad$ **for** $\varepsilon = 1 : T$ **do** ▷ number of iterations

$\quad\quad$ Set η ▷ sparsification knob

$\quad\quad |\boldsymbol{\Xi}| < \eta \dashrightarrow ind_{small}$

$\quad\quad \boldsymbol{\Xi}(ind_{small}) \dashrightarrow 0$

$\quad\quad$ **for** $\varepsilon = 1 : n_x$ **do** ▷ n_x is state's dimension $\mathbf{X}$

$\quad\quad\quad ind_{big} \neq ind_{small}(:, \varepsilon)$

$\quad\quad\quad \boldsymbol{\Xi}(ind_{big}, \varepsilon) = \boldsymbol{\Psi}(:, ind_{big}) \backslash \dot{\mathbf{X}}(:, \varepsilon)$

$\quad\quad$ **end for**

$\quad$ **end for**

end procedure

Output: Sparse matrix $\boldsymbol{\Xi}$ and $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{u})\boldsymbol{\Xi}$

2.4. Model Predictive Control

MPC is a control strategy that solves multiple open-loop control problems over a receding time horizon, subject to constraints [1, 17], illustratively shown in Fig. 4. As shown from the figure, MPC is composed of four elements such as 1) a prediction model, 2) a set of constraints, 3) a cost function, and 4) an optimization algorithm [6]. The prediction model is developed using the controlled system model and the current value of the states (assuming full state measurement, $\boldsymbol{y} = \boldsymbol{x}$). The model is typically represented using a transfer function or state space model (in this study, the latter is used). The states can be written based on the actual states or the deviation between the desired and the actual states (error signal). A set of constraints is obtained by including the minimum and maximum values of the controlled system to limit the input and state variables, while the cost function is derived based on optimal control formulation over a finite horizon. Then the optimization algorithm, incorporating the above-mentioned three components, is utilized to yield a sequence of optimal control actions over a prediction horizon[6, 8].

MPC applies a receding horizon control (RHC) approach where the mathematical optimization is solved online and reiterated forward in time over a finite-time horizon (continually shifted forward the horizon in a receding manner) as depicted in Fig. 4. After the optimization problem is solved, only the first control action of the optimized control sequence is actuated to the controlled system [1].

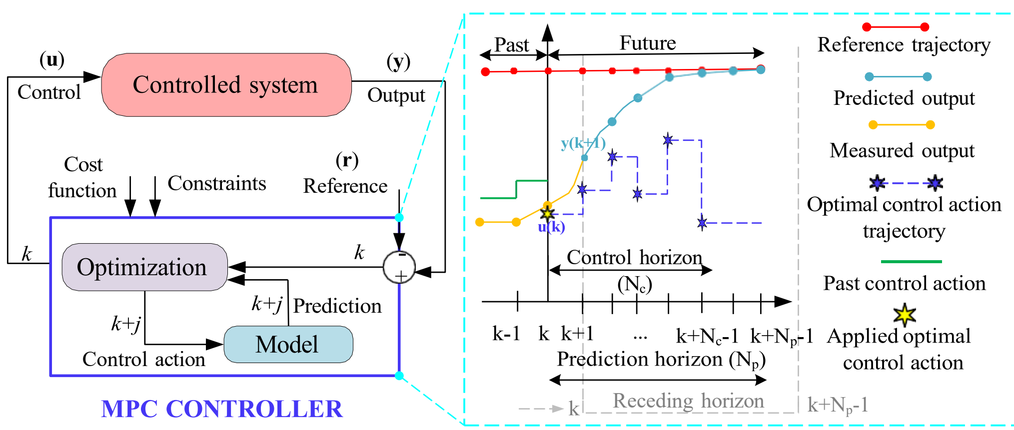


Fig. 4: Schematics of model predictive control

In this paper, to further validate the model derived using the sparse regression (SR) technique, MPC will be utilized to control the water levels of the four tanks such that the objectives of the QTP can be obtained, including minimizing the tracking error signal and rate of change of the controller, which will be further discussed in Section 2.4.1.

Remark. *Assuming full observability of the states ($\boldsymbol{x} = \boldsymbol{y}$), the prediction model and the optimal control problem formulation of MPC, henceforth, will be described solely based on the states and the control variables of the system.*

2.4.1. Optimal Control Problem Formulation

This work adopts the general optimal control problem (OCP) formulation, which has been thoroughly described in [8, 12]. The objective is to minimize the tracking error signal such that the states follow the desired set-point values and to minimize the rate of change of the controller to ensure a longer lifespan. To formulate the OCP, the system's dynamics generated from the SR method are utilized as follows:

$$\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}) + \mathbf{g}(\boldsymbol{x})\boldsymbol{u} \tag{13}$$

$$\dot{\hat{\boldsymbol{x}}} = \underbrace{\boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{u})\boldsymbol{\Xi}}_{\hat{\mathbf{f}}(\hat{x}, \boldsymbol{u})} \tag{14}$$

$$\hat{\boldsymbol{x}}(\mathrm{k}+1) = \tilde{\mathbf{f}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \tag{15}$$

where Eq. (13) describes the dynamic equations of the controlled system, detailed in Section 2.2, Eq. (14) corresponds to the predicted dynamics from the SR technique, and Eq. (15) describes the discretized predicted dynamics intended for MPC implementation. Fourth-order Runge-Kutta methods (RK4) are employed to discretize the function expressed in Eqs. (14)-(15) as follows [39, 40]:

$$k_1 = \hat{\mathbf{f}}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) \tag{16a}$$

$$k_2 = \hat{\mathbf{f}}\left(\hat{\boldsymbol{x}}_k + \frac{t_s}{2}k_1, \boldsymbol{u}_k\right) \tag{16b}$$

$$k_3 = \hat{\mathbf{f}}\left(\hat{\boldsymbol{x}}_k + \frac{t_s}{2}k_2, \boldsymbol{u}_k\right) \tag{16c}$$

$$k_4 = \hat{\mathbf{f}}(\hat{\boldsymbol{x}}_k + t_s k_3, \boldsymbol{u}_k) \tag{16d}$$

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \frac{t_s}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{16e}$$

where k_i is the i-th slope, $t_s \in \mathbb{R}^+$ is the sampling time (set as $t_s = 0.1$) with given initial conditions of $\boldsymbol{x}_0, \boldsymbol{u}_0$ at $t(0)$. In this work, the RK4 method is also interchangeably used in the successive linearization MPC (SR-SLMPC) framework to find numerical solutions from the differential equations.

Combining the system's dynamics in Eq. (15) and the physical constraints to the states and the inputs, depicted in Table 1, the final OCP formulation is displayed in Eq. (17).

$$\min_{u_k,\ldots,u_{k+N_p-1}} J(k) := \sum_{j=0}^{N_p-1} \|\hat{\boldsymbol{x}}(k+j) - \boldsymbol{x}^{ref}(k+j)\|^2_{\mathbf{Q}_j} + \|\Delta\boldsymbol{u}(k+j)\|^2_{\mathbf{R}_j} \tag{17a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{x}}(k+j+1) = \tilde{\mathbf{f}}(\hat{\boldsymbol{x}}(k+j), \boldsymbol{u}(k+j)), \tag{17b}$$

$$j = 0, 1, \ldots, N_p - 1$$

$$\underline{\boldsymbol{u}} \leq \boldsymbol{u}(k+j) \leq \overline{\boldsymbol{u}}, \quad j = 0, 1, \ldots, N_p - 1 \tag{17c}$$

$$\underline{\boldsymbol{x}} \leq \hat{\boldsymbol{x}}(k+j) \leq \overline{\boldsymbol{x}}, \quad j = 1, 2, \ldots, N_p \tag{17d}$$

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_{plant}(t_0) \tag{17e}$$

where $N_p \in \mathbb{N}^+$ is the prediction horizon, $k := kt_s \in \mathbb{R}^+$ is the current time step at sampling time t_s, $\mathbf{Q} \succ 0$ and $\mathbf{R} \succ 0$ are the penalty weights for the states and the control inputs, respectively, $\hat{\boldsymbol{x}}(k) \in \mathbb{R}^{n_x}$, $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$ and $\Delta\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$ denote the states, input and input's rate of change at time step k, respectively, $\boldsymbol{x}^{ref} \in \mathbb{R}^{n_x}$ is the desired set-points, $\underline{\boldsymbol{x}}$, $\underline{\boldsymbol{u}}$, $\overline{\boldsymbol{x}}$, $\overline{\boldsymbol{u}}$ express the minimum and maximum of the states and control inputs, respectively, and $\tilde{\mathbf{f}} : \mathbb{R}^{n_x+n_u} \rightarrow \mathbb{R}^{n_x}$ is the discretized prediction model with varying formulation, further described in Section 2.4.2.

Remark. *For clarity, the predicted states and dynamics from the SR technique ($\dot{\hat{\boldsymbol{x}}} = \hat{\mathbf{f}}(\hat{\boldsymbol{x}}, \boldsymbol{u})$) are expressed in the notation of the actual states and dynamics ($\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$).*

2.4.2. Prediction Model

Prior to formulating the optimal control problem formulation, MPC requires the formulation of the prediction model based on the plant model (dynamics predicted by SR-based model identification) that will be used to optimize a sequence of future control actions. Therefore, a suitable choice of prediction model is crucial to obtain optimal performance of the controlled system. The present study utilizes three prediction models: linear time-invariant MPCs, linear time-varying MPCs, and nonlinear MPCs, which will be discussed in the following sections. To quantify the performance of different MPC strategies, Mean Absolute Percentage Error (MAPE) will be used, which can be computed as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \times 100\% \tag{18}$$

where, n represents the total number of samples, x_i represents the desired value, and $\hat{x}_i$ represents the predicted value.

2.4.2.1. Linear Time-Invariant (LTI).

The first model utilizes a linear time-invariant approximation of the nonlinear model predicted from the SR technique in Eq. (14), subsequently referred to as SR-LMPC. The system's dynamics is linearized around the MP operating points ($\boldsymbol{x}^{op}, \boldsymbol{u}^{op}$) displayed in Table 2 with linearized predicted models as follows [35]:

$$\delta\dot{h}_1(t) = \frac{\xi_{11}}{2\sqrt{h_1^{op}}}\underbrace{\left(h_1(t) - h_1^{op}\right)}_{\delta h_1(t)} + \frac{\xi_{13}}{2\sqrt{h_3^{op}}}\underbrace{\left(h_3(t) - h_3^{op}\right)}_{\delta h_3(t)} + \xi_{15}\underbrace{\left(v_1(t) - v_1^{op}\right)}_{\delta v_1(t)} \tag{19a}$$

$$\delta\dot{h}_2(t) = \frac{\xi_{22}}{2\sqrt{h_2^{op}}}\underbrace{\left(h_2(t) - h_2^{op}\right)}_{\delta h_2(t)} + \frac{\xi_{24}}{2\sqrt{h_4^{op}}}\underbrace{\left(h_4(t) - h_4^{op}\right)}_{\delta h_4(t)} + \xi_{26}\underbrace{\left(v_2(t) - v_2^{op}\right)}_{\delta v_2(t)} \tag{19b}$$

$$\delta\dot{h}_3(t) = \frac{\xi_{33}}{2\sqrt{h_3^{op}}}\delta h_3(t) + \xi_{46}\,\delta v_2(t) \tag{19c}$$

$$\delta\dot{h}_4(t) = \frac{\xi_{44}}{2\sqrt{h_4^{op}}}\delta h_4(t) + \xi_{35}\,\delta v_1(t) \tag{19d}$$

where ξ_{ij} represents the active terms of the i-th row and j-th column of the sparse matrix, Ξ^T, obtained from SR-based model identification in **Algorithm 1**, h_m^{op} and v_n^{op} correspond to the operating points of the m-th tanks' water levels and the n-th pumps' voltage, used to linearize the system, respectively. Then the linearized state space model representation can be expressed as follows:

$$\dot{x} = \underbrace{\begin{bmatrix} -\frac{\xi_{11}}{2\sqrt{h_1^{op}}} & 0 & \frac{\xi_{13}}{2\sqrt{h_3^{op}}} & 0 \\ 0 & -\frac{\xi_{22}}{2\sqrt{h_2^{op}}} & 0 & \frac{\xi_{24}}{2\sqrt{h_4^{op}}} \\ 0 & 0 & -\frac{\xi_{33}}{2\sqrt{h_3^{op}}} & 0 \\ 0 & 0 & 0 & -\frac{\xi_{44}}{2\sqrt{h_4^{op}}} \end{bmatrix}}_{\mathbf{A}_c} \delta h^m(t) + \underbrace{\begin{bmatrix} \xi_{15} & 0 \\ 0 & \xi_{26} \\ 0 & \xi_{46} \\ \xi_{35} & 0 \end{bmatrix}}_{\mathbf{B}_c} \delta v_n(t) \qquad (20)$$

Next, the continuous state space model in Eq. (20) is discretized and rewritten as follows:

$$\boldsymbol{x}^{LTI}(k+1) = \mathbf{A}_d \boldsymbol{x}^{LTI}(k) + \mathbf{B}_d \boldsymbol{u}^{LTI}(k) \qquad (21)$$

where k is the time instant, $\mathbf{A}_d \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{B}_d \in \mathbb{R}^{n_x \times n_u}$ correspond to the time-invariant system's matrices with subscript d denoting the discrete time. Thereafter, the linear prediction model over the prediction horizon can be written as follows:

$$\boldsymbol{x}^{LTI}(k+j) = \mathbf{A}_d^j \boldsymbol{x}^{LTI}(k) + \sum_{i=0}^{j-1} \mathbf{A}_d^i \mathbf{B}_d \boldsymbol{u}^{LTI}(k+i) \quad \forall\ j \in \{1, \ldots N_p\} \qquad (22)$$

where $\boldsymbol{x}^{LTI}(0)$ is the current state measurement of the states at time instant $k = 0$. The LTI model formulated in Eq. (21) assumes that the system dynamics remain constant over time. In addition, the problem in Eq. (17) is a convex optimization formulated under linear constraints and quadratic objectives. Therefore, LMPC can be considered mathematically tractable and computationally efficient. To reduce the number of decision variables to be solved in Eq. (17), the problem is formulated in terms of the incremental form of the control inputs: $\boldsymbol{\Delta u}(k) := \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$ which has been comprehensively discussed in [8, 12]. Complete implementation of LMPC is exhibited in **Algorithm 2**.

Algorithm 2 Linear Time Invariant MPC (LMPC)

Input $N_p, \mathbf{Q}, \mathbf{R}, T, t_s, n_x, n_u, \boldsymbol{x}^{\boldsymbol{ref}}, \boldsymbol{x}^{op}, \boldsymbol{u}^{op}, \mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$

Initialize $\boldsymbol{x}_0 \in \mathbb{R}^{nx}, \boldsymbol{u}_0 \in \mathbb{R}^{nu}, k = 0$

Linearize & Discretize nonlinear dynamics from $\mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$ around $\boldsymbol{x}^{op}, \boldsymbol{u}^{op}$ at t_s, Eq. (20)

Return LTI model, Eq. (21)

for $k = 0 \rightarrow T - 1$ **do** ▷ Simulation time

    for $j = 0$ to $N_p - 1$ **do**

        Construct prediction model, Eq. (22)

        Formulate quadratic cost: $J(\boldsymbol{x}(k + j), \boldsymbol{u}(k + j))$, Eq. (17a)

    end for

    Solve $J(k)$ s.t. Eqs. (17b)-(17d)

    Return $[\boldsymbol{\Delta u}^*(0|k), \ldots, \boldsymbol{\Delta u}^*(N_p - 1|k)]$

    Extract $[\boldsymbol{u}^*(0|k), \ldots, \boldsymbol{u}^*(N_p - 1|k)]$

    Apply only $\boldsymbol{u}^*(0|k)$ ▷ Receding horizon control (RHC)

    Measure $\boldsymbol{x}(k + 1|k)$ from Eq. (21)

    Update for $k + 1$, $\boldsymbol{x}_0 = \boldsymbol{x}(k + 1|k)$ and $\boldsymbol{u}_0 = \boldsymbol{u}^*(0|k)$

end for

2.4.2.2. Linear Time-Varying via Successive Linearization MPC.

The second prediction model utilizes the linear time-varying (LTV) model via a successive linearization method, in this paper referred to as SR-SLMPC. SR-SLMPC differs from SR-LMPC mainly due to its approach to linearizing the nonlinear model. While SR-LMPC linearizes the nonlinear model around steady state conditions using constant systems' matrices for prediction, SR-SLMPC performs online linearization of the nonlinear model using the current operating points. Then the formulated prediction model over N_p is used in Eq. (17) to obtain a sequence of optimal control actions. SR-SLMPC is formulated as follows [13, 41, 42]:

1. Formulate a state space representation of the QTP using the predicted continuous nonlinear system dynamics in Eq. (14), which can be expressed as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \xi_{11}\sqrt{x_1} + \xi_{13}\sqrt{x_3} \\ \xi_{22}\sqrt{x_2} + \xi_{24}\sqrt{x_4} \\ \xi_{33}\sqrt{x_3} \\ \xi_{44}\sqrt{x_4} \end{bmatrix} + \begin{bmatrix} \xi_{15} & 0 \\ 0 & \xi_{26} \\ 0 & \xi_{46} \\ \xi_{35} & 0 \end{bmatrix} \boldsymbol{u} \tag{23a}$$

$$= \mathbf{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{23b}$$

2. Linearize the continuous nonlinear state space model in Eq. (23) at current operating points with detailed formulation in Eq. (19) and rewritten as follows:

$$\dot{\boldsymbol{x}}_l = \dot{\boldsymbol{x}}^{op} + \mathbf{A}_c(t)(\boldsymbol{x}_l - \boldsymbol{x}^{op}) + \mathbf{B}_c(t)(\boldsymbol{u}_l - \boldsymbol{u}^{op}) \tag{24a}$$

$$= \mathbf{A}_c(t)(\boldsymbol{x}_l) + \mathbf{B}_c(t)\boldsymbol{u}_L + \underbrace{\boldsymbol{\Gamma}_{\mathbf{c}}(t)}_{\dot{\boldsymbol{x}}^{op} - \mathbf{A}_c(t)\boldsymbol{x}^{op} - \mathbf{B}_c(t)\boldsymbol{u}^{op}} \tag{24b}$$

where $\mathbf{A}_c(i,j) = \frac{\partial \mathbf{f}_i}{\boldsymbol{x}_j} \in \mathsf{R}^{n_x \times n_x}$, $\mathbf{B}_c(i,j) = \frac{\partial \mathbf{f}_i}{\boldsymbol{u}_j} \in \mathsf{R}^{n_x \times n_u}$ and $\boldsymbol{\Gamma}_{\mathbf{c}} \in \mathsf{R}^{n_x}$ is the constant form of the linearization and subscript l corresponds to linearized model. Note that the system's matrices associated with time t are different than that in Eq. (20). This is to indicate that SR-SLMPC utilizes time-varying matrices. Initially, the model is linearized offline at the MP operating points defined in Table 2. Then, during SR-SLMPC implementation, the dynamics will be linearized around the operating points at time instant k.

3. Integrate Eq. (24) with RK4 numerical integration scheme, described in Eq. (16).

$$\boldsymbol{x}_l(t) = e^{\mathbf{A}_c(t)}\boldsymbol{x}_l(0) + \int_0^t e^{\mathbf{A}_c(t-\tau)}(\mathbf{B}_c\boldsymbol{u}_l(\tau) + \boldsymbol{\Gamma}_\mathbf{c})d\tau \tag{25}$$

4. Discretization of the continuous state space representation.

$$\begin{aligned}\boldsymbol{x}_l(k+1) = {} & \underbrace{e^{\mathbf{A}_c t_s}}_{\mathbf{A}_d(k)}\boldsymbol{x}_l(k) + \underbrace{\mathbf{A}_c^{-1}(e^{\mathbf{A}_c t_s} - \boldsymbol{I})\mathbf{B}_c}_{\mathbf{B}_d(k)}\boldsymbol{u}_l(k) \\ & + \underbrace{\mathbf{A}_c^{-1}(e^{\mathbf{A}_c t_s} - \boldsymbol{I})\boldsymbol{\Gamma}_\mathbf{c}}_{\boldsymbol{\Gamma}_\mathbf{d}(\mathbf{k})}\end{aligned} \tag{26}$$

where t_s is the discretization time step, equivalent to the sampling time employed in MPC.

5. The final discrete state space model in LTV-MPC that will be used for the formulation of the prediction model is described below:

$$\boldsymbol{x}^{LTV}(k+1) = \mathbf{A}_d(k)\boldsymbol{x}^{LTV}(k) + \mathbf{B}_d(k)\boldsymbol{u}^{LTV}(k) \tag{27}$$

During the SR-SLMPC implementation, steps 1-5 are repeatedly conducted by linearizing the system around the measurement of the current states and the previous control inputs $(\boldsymbol{x}_0|k, \boldsymbol{u}_0|k)$ for all $k \in \{0, \ldots, T-1\}$. Using Eq. (27), the prediction model is then expressed as follows:

$$\begin{aligned}\boldsymbol{x}^{LTV}(k+j) = {} & \mathbf{A}_d^j(k)\boldsymbol{x}^{LTV}(k) + \sum_{i=0}^{j-1}\mathbf{A}_d^i(k)\mathbf{B}_d(k)\boldsymbol{u}^{LTV}(k+i) \\ & + (\boldsymbol{I} + \mathbf{A}_d(k))^{j-1}\boldsymbol{\Gamma}_\mathbf{d}(\mathbf{k}) \quad \forall\, j \in \{1, \ldots N_p\}\end{aligned} \tag{28}$$

where $\mathbf{A}_d^j(k) \in \mathbb{R}^{N_p n_x \times n_x}$ is the time-varying prediction matrix associated with current states measurement, $\sum_{i=0}^{j-1} \mathbf{A}_d^i(k)\mathbf{B}_d(k) \in \mathbb{R}^{N_p n_x \times N_p(n_x+n_u)}$ corresponds to the time-varying prediction matrix associated to the control sequence, and $(\boldsymbol{I} + \mathbf{A}_d(k))^{j-1} \in \mathbb{R}^{N_p n_x \times n_x}$ represents the prediction matrix of the affine term from the linearization. All system matrices are formulated over the prediction horizon. Similar to SR-LMPC, OCP in Eq. (17) is formulated in the incremental form of control signals [41]. An overview of SR-SLMPC implementation can be found in **Algorithm 3**.

Algorithm 3 LTV Model Predictive Control (SLMPC)

Input $N_p, t_s, \mathbf{Q}, \mathbf{R}, T, n_x, n_u, \mathbf{x}^{ref}, \mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$
Initialize $\boldsymbol{x}_0 \in \mathsf{R}^{n_x}, \boldsymbol{u}_0 \in \mathsf{R}^{n_u}, k = 0$
for $k = 0 \to T - 1$ **do** ▷ Simulation time
    for $j = 0$ to $N_p - 1$ **do**
        Integrate & Linearize $\mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$ around $\boldsymbol{x}_0, \boldsymbol{u}_0$
        Return LTV model, Eq. (27)
        Discretize Eq. (27), **return** $A_d(k)$ and $B_d(k)$
        Construct prediction model, Eq. (28)
        Formulate quadratic cost: $J(\boldsymbol{x}(k + j), \boldsymbol{u}(k + j))$, Eq. (17a)
    end for
    Solve $J(k)$ s.t. Eqs. (17b)-(17d)
    Return $[\boldsymbol{\Delta u}^*(0|k), \dots, \boldsymbol{\Delta u}^*(N_p - 1|k)]$
    Extract $[\boldsymbol{u}^*(0|k), \dots, \boldsymbol{u}^*(N_p - 1|k)]$
    Apply only $\boldsymbol{u}^*(0|k)$ ▷ RHC
    Measure $\boldsymbol{x}(k + 1|k)$ from Eq. (27)
    Update for $k + 1$, $\boldsymbol{x}_0 = \boldsymbol{x}(k + 1|k)$ and $\boldsymbol{u}_0 = \boldsymbol{u}^*(0|k)$
end for

2.4.2.3. Nonlinear MPC.

The main difference between nonlinear MPC, herein referred to as SR-NMPC, compared to the above-mentioned MPCs is that the prediction model directly utilizes the discretized nonlinear dynamics of the system to represent the controlled system sufficiently. In this work, the discretized system's dynamic in Eq. (23) is expressed as follows:

$$\boldsymbol{x}(k+1) = \tilde{\mathbf{f}}(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{29}$$

Therefore, the prediction model over the prediction horizon can be expressed as follows:

$$\boldsymbol{x}(k+j+1) = \tilde{\mathbf{f}}(\boldsymbol{x}(k+j), \boldsymbol{u}(k+j)) \quad \forall j \in \{0, \ldots, N_p - 1\} \tag{30}$$

Given that the prediction model uses a nonlinear model, Eq. (17) is no longer convex. Thus, convergence to a global minimum may not be guaranteed. An overview of SR-NMPC implementation in this work can be found in **Algorithm 4**.

Algorithm 4 Nonlinear Model Predictive Control (NMPC)

```
Input: N_p, Q, R, T, n_x, n_u, x^ref, f(x, u)
Initialize x_0 ∈ R^nx, u_0 ∈ R^nu, k = 0
for k = 0 → T − 1 do                                      ▷ Simulation time
    for j = 0 to N_p − 1 do
        Define & Discretize state space model, from Eq. (14) to Eq. (29)
        Construct prediction model, Eq. (30)
        Formulate quadratic cost: J(x(k + j), u(k + j)), Eq. (17a)
    end for
    Solve J(k) s.t. Eqs. (17b)-(17d)
    Extract [u*(0|k), . . . , u*(N_p − 1|k)].
    Apply only u*(0|k)                                    ▷ RHC
    Measure x(k + 1|k) from Eq. (29)
    Update for k + 1, x_0 = x(k + 1|k) and u_0 = u*(0|k)
end for
```

2.4.3. Solution method

Considering the nonlinearity of the quadruple tanks' dynamics used as the equality constraint in the OCP formulation (when implementing SR-NMPC), sequential quadratic programming (SQP) is selected in this study to solve the OCP in Eq (17). SQP is an iterative method that solves nonlinear constrained optimization by solving a sequence of quadratic programming (QP) sub-problems given in Eq. (32) [43]. SQP starts with an initial guess for $\boldsymbol{x}^k$ for a given iterate k and continues iteratively by updating $\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \alpha \boldsymbol{p_k}$. A new iterate $\boldsymbol{x}^{k+1}$ is then used again to solve the QP subproblem to obtain $\boldsymbol{p}$ such that a sequence of x^k is created to converge to a local minimum x^* as $k \rightarrow \infty$. In this study, the rate of change of the control input ($\boldsymbol{\Delta U} \in \mathbb{R}^{nuNp}$) (for SR-SLMPC and SR-LMPC) as well as the states ($\mathbf{X} \in \mathbb{R}^{nxNp}$), additionally for SR-NMPC, are the decision variables ($\boldsymbol{x}^k$) for the formulated nonlinear optimization problem.

The associated Lagrangian function to the nonlinear problem in Eq (17) is expressed by [43]:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i h_i(\boldsymbol{x}) + \sum_{i=1}^{p} \mu_j g_j(\boldsymbol{x}) \tag{31}$$

where the functions $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ describe the equality and inequality constraints, respectively, concatenated from Eqs. (17b)-(17d). In addition, n, m, and p are the number of decision variables, equality constraints, and inequality constraints, respectively, with $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^n$ as the Lagrangian multipliers for the associated equality and inequality constraints. Then the QP subproblem is formulated by approximating the Lagrangian of Eq. (31) and linearizing the nonlinear constraints shown in

Eq. (32).

$$\min_{p \in \mathbb{R}^m} \quad \frac{1}{2}\boldsymbol{p}^T \mathbf{H}_k \boldsymbol{p} + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p} \tag{32a}$$

$$\text{s.t.} \quad \nabla g_i(\boldsymbol{x}_k)^T p + g_i(\boldsymbol{x}_k) = 0, \ \forall \ i \in \mathbf{I} \tag{32b}$$

$$\nabla g_j(\boldsymbol{x}_k)^T \boldsymbol{p} + g_j(\boldsymbol{x}_k) \leq 0, \ \forall \ j \in \mathbf{J} \tag{32c}$$

where $\boldsymbol{p}$ is the search direction from the QP subproblem and $\mathbf{H}_k$ and is the Hessian matrix of Eq. (31). However, to avoid the computational complexity of Hessian computation, an approximate Hessian Matrix $\mathbf{B}_k$ can be computed in place of $\mathbf{H}_k$ from Eq. (32) and updated for each iteration. In this study, $\mathbf{B}_k$ is computed and updated using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. For a detailed description of the BFGS method, readers are encouraged to refer to [43, 44]. Furthermore, the solution to the QP subproblem ($\boldsymbol{p}_k$) is used to form a new iterate, $\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \alpha_k \boldsymbol{p}_k$, where the step size α_k should be determined to ensure a sufficient decrease in a merit function.

Algorithm 5 Sequential quadratic programming

for $k \in \{1, 2..., n\}$ **do** ▷ number of iterations
  Input $\boldsymbol{x}_0, \mathbf{H}_0$
  Set $k \leftarrow 0$
  repeat until a convergence is satisfied
  Evaluate $f(x_k)$, $\nabla_x L(x_k, \lambda_k, \mu_k)$
  Solve QP subproblems in Eq. (32) to obtain $\boldsymbol{p}_k$
  Compute step size α_k such that $m(\boldsymbol{x}^k + +\alpha_k \boldsymbol{p}_k) \leq m(\boldsymbol{x}^k)$
  Set $x_{k+1} \leftarrow x_k + \alpha_k \boldsymbol{p}_k$
  Update $\mathbf{B}_{k+1}$ using the BFGS method
  Set $k \leftarrow k + 1$
  end (repeat)
end for

3. Case Studies

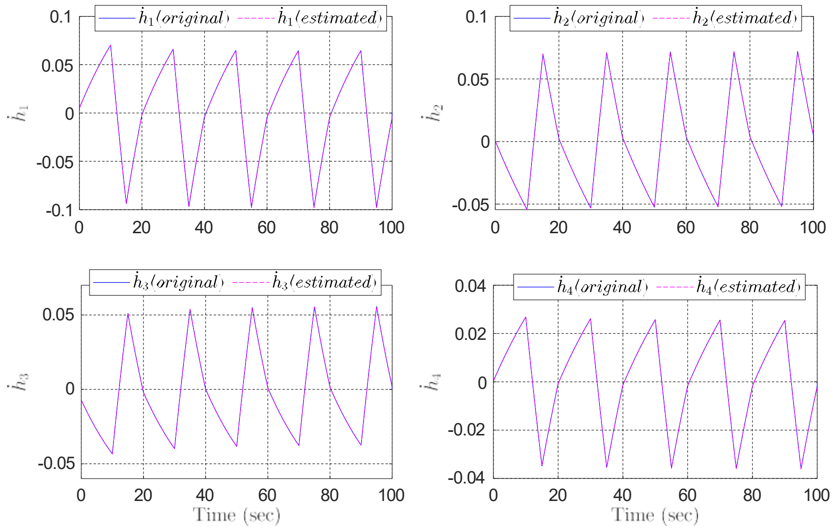
In this study, a bi-level process is presented where data-enabled model-free identification of the quadruple tank process dynamics is first implemented solely from available measurement data and then used to control the system using MPC. To validate the accuracy of the proposed data-driven identification of the QTP dynamics, time-domain simulations are carried out. Furthermore, the predicted dynamics are continued to be actuated by comparing different MPC strategies. All simulations are carried out in MATLAB R2022b on a processor of Intel Core CPU i7-6700 at 3.40 GHz and 32GB RAM.

3.1. Case study 1: Validation for SR-based Method Model Identification

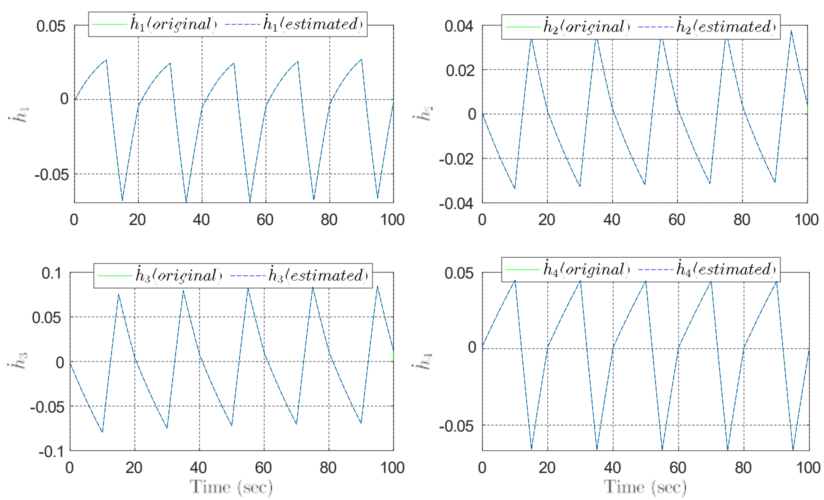
The first case study involves conducting parameter identification comparisons and time-domain simulations. The water levels of the tanks are observed by perturbing the inputs of the pumps. The objective is to validate the accuracy and effectiveness of the identified nonlinear dynamic models in capturing the actual system dynamics.

3.1.1. Model Identification

First, data was collected on states and inputs of a quadruple tank system simulated in MATLAB, using the parameters provided in [35] for training purposes. Then the estimated values of the derivative for each state were generated using central difference approximation, explained in Section 2.3.2. Utilizing the measurements from the states and perturbing the inputs (pump voltages, v_i) with a repetitive sequence waveform between 2 - 4 V (to activate the dynamic modes) over 100 seconds of simulation, the results of the estimated derivatives for MP and NMP operating points of the QTP are illustrated in Fig. 5. It is observed that the estimated derivative is capable of representing the measured derivative accurately.



(a) MP operation point.



(b) NMP operating point.

Fig. 5: Derivative estimation.

Accordingly, various candidate terms were considered for the function library $\Psi(\mathbf{X}, \mathbf{U})$, such as square-root functions, polynomials up to degree 2, and sinusoidal functions. In this work, 18 candidate terms are included with variables such as x_i, u_i, $\sqrt{x_i}$, x_iu_j, x_i^2, $\exp(x_i)$, $x_i\cos x_j$, $x_i\sin x_j$, $u_i\cos x_j$, and $u_i\sin x_j$. Following this, a sparse identification method was employed to determine the sparse matrix of coefficients, denoted as Ξ, with a sparsification hyperparameter set to $\eta = 0.01$ for both operating points. The identified coefficients were then utilized to construct a data-driven model and control of the quadruple tank system in MATLAB. A comparison was conducted between the parameters of the physical model (via simulation) and the identified model (via SR-based nonlinear dynamics model identification) for both MP and NMP operating points, as presented in Tables 3 and 4. The results show that all active terms from Ξ closely configure the actual parameters that reside in the system's dynamics. This proves that the identified data-driven model accurately captures the dynamics of the physical model.

Table 3: Parameter identification using SR-based technique at MP operating point.

Dynamics	**Term**	**Term**	**Term**	**Term**	**Term**	**Term**
$\dot{x}_1$	$-\frac{a_1\sqrt{2g}}{A_1}\sqrt{x_1}$	$0\sqrt{x_2}$	$\frac{a_3\sqrt{2g}}{A_1}\sqrt{x_3}$	$0\sqrt{x_4}$	$\frac{\gamma_1k_1}{A_1}v_1$	$0v_2$
Physical	−0.1123	0	0.1123	0	0.0833	0
Identified	−0.1125	0	0.1125	0	0.0834	0
$\dot{x}_2$	$0\sqrt{x_1}$	$-\frac{a_2\sqrt{2g}}{A_2}\sqrt{x_2}$	$0\sqrt{x_3}$	$\frac{a_4\sqrt{2g}}{A_2}\sqrt{x_4}$	$0v_1$	$\frac{\gamma_2k_2}{A_2}v_2$
Physical	0	−0.0789	0	0.0789	0	0.0628
Identified	0	-0.079	0	0.079	0	0.0629
$\dot{x}_3$	$0\sqrt{x_1}$	$0\sqrt{x_2}$	$-\frac{a_3\sqrt{2g}}{A_3}\sqrt{x_3}$	$0\sqrt{x_4}$	$0v_1$	$\frac{(1-\gamma_2)k_2}{A_3}v_2$
Physical	0	0	−0.1123	0	0	0.0479
Identified	0	0	−0.1125	0	0	0.0480
$\dot{x}_4$	$0\sqrt{x_1}$	$0\sqrt{x_2}$	$0\sqrt{x_3}$	$-\frac{a_4\sqrt{2g}}{A_4}\sqrt{x_4}$	$\frac{(1-\gamma_1)k_1}{A_4}v_1$	$0v_2$
Physical	0	0	0	-0.0789	0.0314	0
Identified	0	0	0	-0.0790	0.0312	0

Table 4: Parameter identification using SR-based technique at NMP operating point.

Dynamics	**Term**	**Term**	**Term**	**Term**	**Term**	**Term**
$\dot{x}_1$	$-\frac{a_1\sqrt{2g}}{A_1}\sqrt{x_1}$	$0\sqrt{x_2}$	$\frac{a_3\sqrt{2g}}{A_1}\sqrt{x_3}$	$0\sqrt{x_4}$	$\frac{\gamma_1 k_1}{A_1}v_1$	$0\,v_2$
Physical	−0.1123	0	0.1123	0	0.0482	0
Identified	−0.1129	0	0.1129	0	0.0485	0
$\dot{x}_2$	$0\sqrt{x_1}$	$-\frac{a_2\sqrt{2g}}{A_2}\sqrt{x_2}$	$0\sqrt{x_3}$	$\frac{a_4\sqrt{2g}}{A_2}\sqrt{x_4}$	$0v_1$	$\frac{\gamma_2 k_2}{A_2}v_2$
Physical	0	−0.0789	0	0.0789	0	0.0350
Identified	0	-0.0793	0	0.0793	0	0.0351
$\dot{x}_3$	$0\sqrt{x_1}$	$0\sqrt{x_2}$	$-\frac{a_3\sqrt{2g}}{A_3}\sqrt{x_3}$	$0\sqrt{x_4}$	$0v_1$	$\frac{(1-\gamma_2)k_2}{A_3}v_2$
Physical	0	0	−0.1123	0	0	0.0776
Identified	0	0	−0.1129	0	0	0.0779
$\dot{x}_4$	$0\sqrt{x_1}$	$0\sqrt{x_2}$	$0\sqrt{x_3}$	$-\frac{a_4\sqrt{2g}}{A_4}\sqrt{x_4}$	$\frac{(1-\gamma_1)k_1}{A_4}v_1$	$0v_2$
Physical	0	0	0	-0.0789	0.0586	0
Identified	0	0	0	-0.0793	0.0562	0

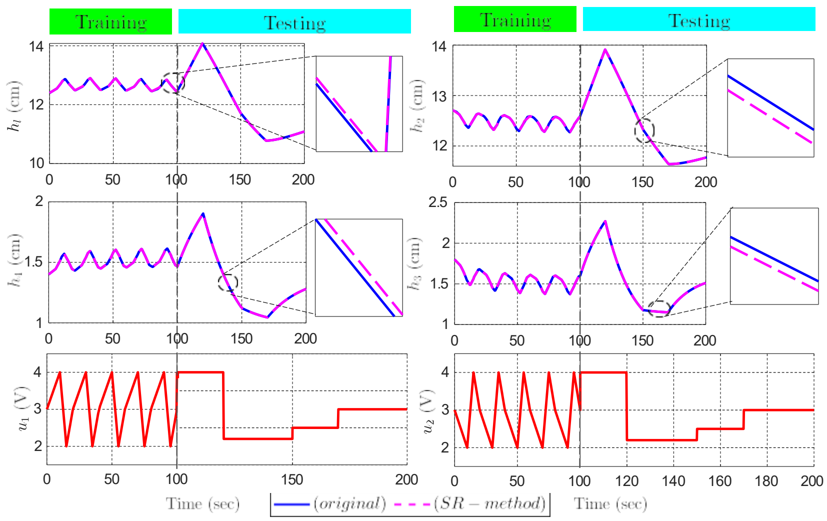
3.1.2. Time-domain Validation

In the second part (testing), the identified model obtained through the SR-based method was evaluated by comparing it with the physical model of the quadruple tank process in various operational scenarios. Specifically, the control signal ($\boldsymbol{u}$(t)) underwent several step changes at different time instances. At 100 seconds, a step change was introduced from 3V to 4V. Subsequently, at 120 seconds, another step change occurred to decrease $\boldsymbol{u}$(t) from 4V to 2.2V. At 150 seconds, a 0.5 step change was added for $\boldsymbol{u}$(t) to increase the pump voltage. Lastly, after 170 seconds, $\boldsymbol{u}$(t) is returned to the initial condition. Different control signals compared to the training period were implemented to validate the identified model.

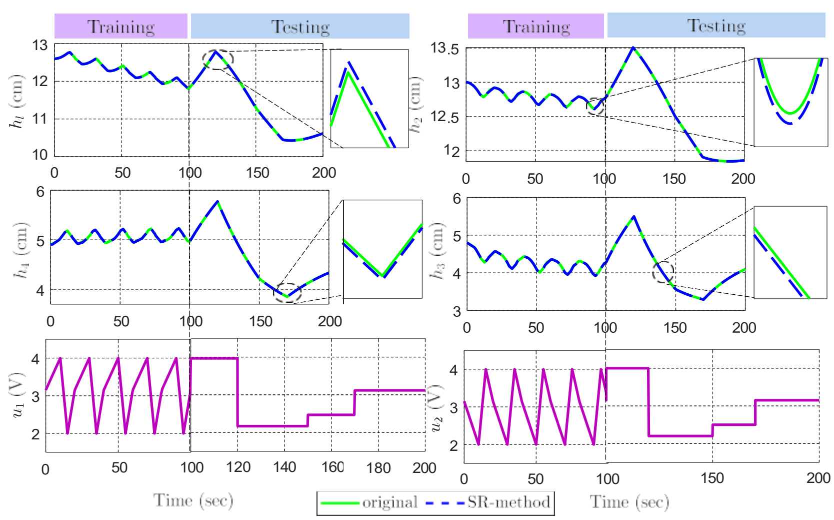
A comparison was made between the physical system and the identified model as depicted in Fig. 6 at MP and NMP operating points. Mean absolute percentage error (MAPE) using Eq. (18) is also used to quantify the error of the predicted dynamics during time series validation of the tanks' water levels. As illustrated, the system was forced with a control signal ($\boldsymbol{u}$(t)) showing a repetitive triangle waveform for the first 100 seconds to train the SR method while the following 100 seconds were implemented to validate the identified model using varying control signal values. It is observed from the figure and MAPE ranging from 0.01% - 0.02% that the proposed data-driven model identification approach (SR-based technique) exhibits a high level of accuracy in identifying the nonlinear dynamics of the QTP system. Furthermore, a slight deviation between the predicted and the original dynamics can be attributed to their inherent sparsity [18], which further confirms that the SR technique promotes the interpretability of the predicted models that prevents overfitting [32]. Consequently, the proposed data-driven model can serve as a reliable substitute for complex physics-based models, offering a simplified yet effective alternative for understanding and controlling large-scale water distribution systems.

3.2. Case study 2: Comparative Analysis of Three MPC Control Laws

Given the accuracy of using the SR-based method as described in the first case study (Section 3.1), this case study further utilized the predicted dynamic models of QTP in Eq. (14) using MPC with techniques described in Section 2.4. Taking into account that all states are assumed to be fully observable ($\boldsymbol{y} = \boldsymbol{x}$), four water levels were controlled by MPC to meet the desired water level values. Three references with initial conditions at MP operating points were used, as shown in Table 5 with model parameters in Table 6.



(a) MP operating point.



(b) NMP operating point.

Fig. 6: Case study 1: Time-domain validation of the identified model

Table 6: Model parameters of the QTP

Parameter	Values	Parameter	Values
$\mathbf{Q}$	diag[1 1 1 1]	$\mathbf{R}$	diag[0.001 0.001]
$\underline{\boldsymbol{x}}$	0.2 m	$\overline{\boldsymbol{x}}$	15 m
$\underline{\boldsymbol{u}}$	0 V	$\overline{\boldsymbol{u}}$	7 V
$\boldsymbol{x}_0\|k$	[12.4 12.7 1.8 1.4]	$\boldsymbol{u}_0\|k$	[3 3]

Table 5: Reference points over time

	Time (s)	h_1 (cm)	h_2 (cm)	h_3 (cm)	h_4 (cm)
Ref1	t = 0	12.4	12.7	1.80	1.40
Ref2	0 < t ≤ 400	10.0	12.0	1.77	0.956
Ref3	t > 400 and t ≤ 800	12.0	14.0	2.01	1.19
Ref4	800 < t ≤ 1200	8.00	10.0	1.53	0.721

From the OCP in Eq. (17), the pump voltage is required to control the water levels of the four tanks to meet the reference points in Table 5 while minimizing the voltage change rate between two consecutive time steps. The model was run for 1200 s with a 0.1 s sampling time (t_s) and a prediction horizon of $N_p = 10$, generating 12000 samples.

Fig. 7 illustrates the evolution of the water levels of the four tanks as a response to the reference point changes controlled by SR-LMPC, SR-SLMPC, and SR-NMPC. As depicted, a change in the reference value was introduced at t= 400 s, causing a jump in the reference signal. Similarly, at t= 800 s, the reference value was altered, resulting in a drop in the reference signal. As shown from the figure, the present findings obtained from using different MPC techniques, incorporating the predicted dynamics derived from the sparse regression-based nonlinear dynamics model identification (SR-based method), revealed similar behavior (i.e., similar time evolution of states) with prior studies investigating model-based control approaches for the quadruple tank system [11, 19, 45]. This further confirms the feasibility of the SR-based method to be applied to different control approaches.

According to the top sub-plots from Fig. 7, the water levels of Tanks 1 and

2 followed the reference points using all controllers. Furthermore, with varying reference points, Tanks 1 and 2 water levels remain within the constraints without any overshoots observed. As observed from the two subplots at the bottom of Fig. 7, all controllers managed to maintain the water levels of Tanks 3 and 4 within the bound, with overshoots (or undershoots) observed in both tanks. It is observed that SR-NMPC and SR-SLMPC demonstrate satisfactory tracking performance. In contrast, the SR-LMPC controller displayed limitations in reaching the reference points, as evidenced by notable deviations in Tank 3 and a minor error signal in Tank 4. This is expected as SR-LMPC relies on linearizing the model of the nonlinear dynamics at a constant operating point. Therefore, when the system's behavior differs from the operating points, the tracking performance of SR-LMPC can degrade. In addition, the QTP system is designed with cross-coupling effects in its dynamics, making it challenging to control Tank 3 without affecting the tracking performance in Tank 1.

The response of the pumps' voltages in all controllers aligns with the changing reference values shown in Fig. 8. The results indicate that all control strategies followed a similar pattern to achieve tracking performance, irrespective of the control performance objective. This is indicated by the sudden changes in the pumps' voltages when reference points were modified. These outcomes were expected since the simulation setup assigned higher penalty weights to minimize the error signal of the states. Despite the abrupt adjustments to the pump voltage, depicted by maximizing the voltage level during reference jumps and minimizing it during reference drops, all controllers successfully maintained the voltage level within the physical

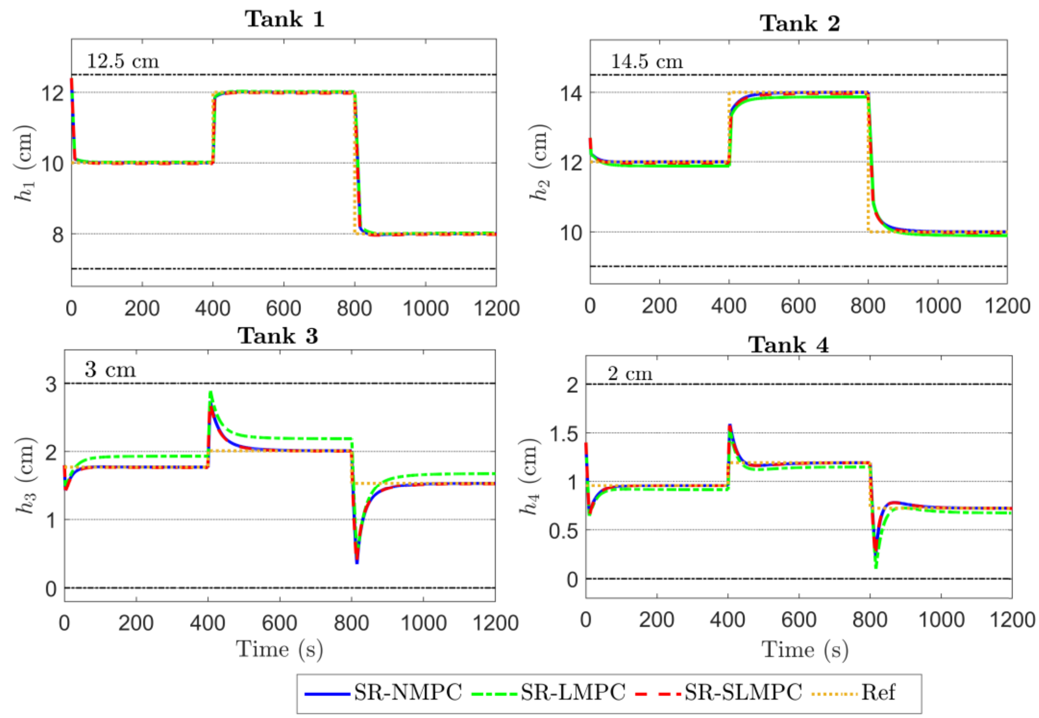


Fig. 7: Evolution of the four tanks over time: state bounds (dash-dotted lines)

limits.

Furthermore, it is observed that during reference changes, SR-SLMPC (represented by dashed red lines) exhibited faster adjustments (approx. 1 s) in the pump voltages compared to the SR-NMPC and SR-LMPC controllers. Additionally, both SR-NMPC and SR-LMPC controllers required a duration of 5 seconds to reach the maximum or minimum values of the pump voltages during the transition period. In contrast, SR-SLMPC required a shorter time at 4.46 s with slightly smoother control actions, similarly seen in [41].

In Fig. 9, the state trajectory of Tanks 1 and 2 is displayed for all control strategies, further highlighting the tracking performance. Four reference points were depicted with yellow markers. *Ref1* corresponds to the initial

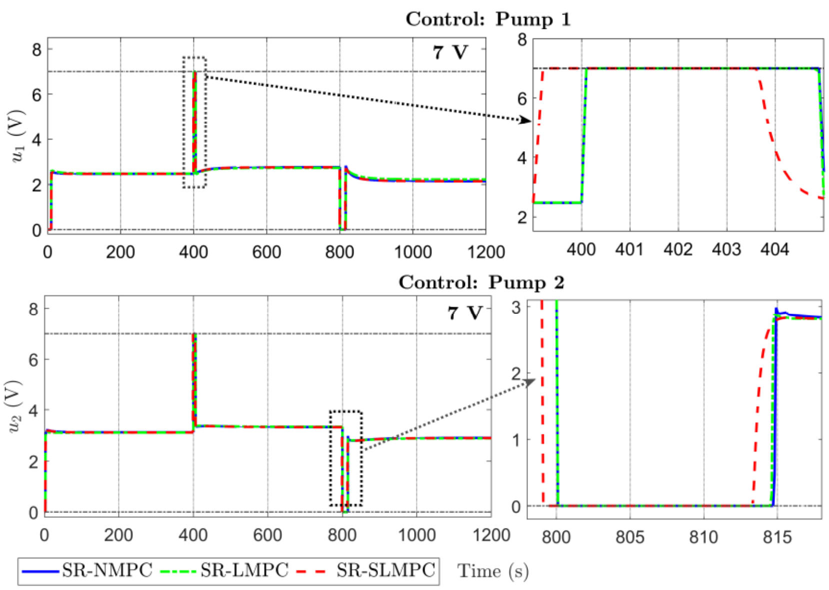


Fig. 8: Evolution of pumps' voltage over time: control bounds (dash-dotted lines)

conditions at MP operating points, while *Ref4* denotes the last change of the reference point. In terms of tracking the performance, SR-NMPC demonstrates superiority over SR-SLMPC and SR-LMPC, consistently reaching the reference points with identical final states. This highlights the capability of SR-NMPC to accurately represent and control systems with nonlinearities, making it a suitable choice for a controller. This finding aligns with the description of SR-NMPC's main functionality as stated in [14], which focuses on stabilization and tracking objectives. As depicted in Fig. 9, both SR-SLMPC and SR-LMPC controllers exhibit non-zero offsets and do not achieve perfect tracking. However, the trajectory of SR-SLMPC closely resembles that of SR-NMPC, indicating a better overall performance compared to SR-LMPC.

Similarly, SR-NMPC repeatedly reached the reference points for Tanks 3

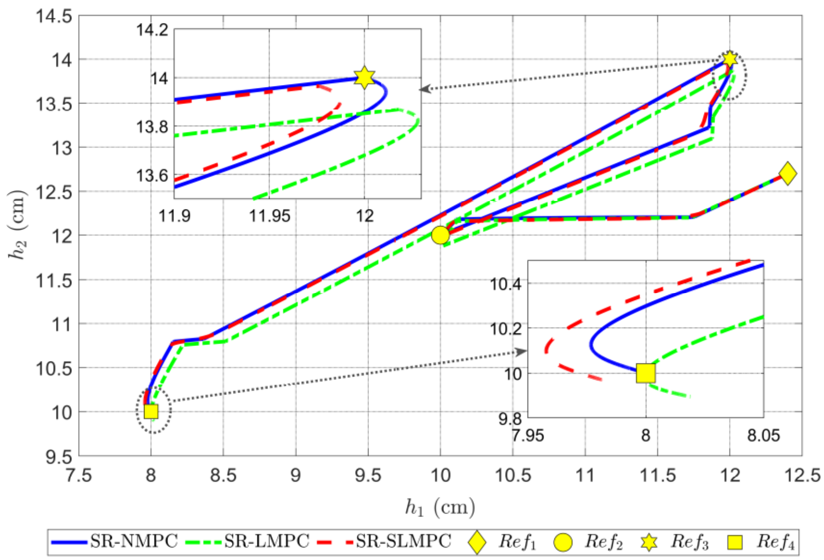


Fig. 9: State trajectory of Tanks 1 and 2

and 4, as depicted in Fig. 10. SR-SLMPC followed the trajectories of SR-NMPC to reach the reference points. This is because SR-SLMPC leverages the system's nonlinearities to compute the time-varying operating points for each time step to linearize the system. Thereby, the linearized dynamics follow the trajectory of the nonlinear model. In contrast, SR-LMPC showed a different trajectory than SR-SLMPC and SR-NMPC and performed relatively less satisfactory tracking purposes in Tanks 3 and 4 for each reference point.

Table 7 displays a mean absolute percentage error (MAPE) between the predicted and reference states, and the maximum elapsed time for executing the control to the water levels of the tanks. All controllers showed an error percentage below 5%, with SR-LMPC higher than the rest. Interestingly, SR-SLMPC showed lower MAPE than SR-NMPC when tracking the reference

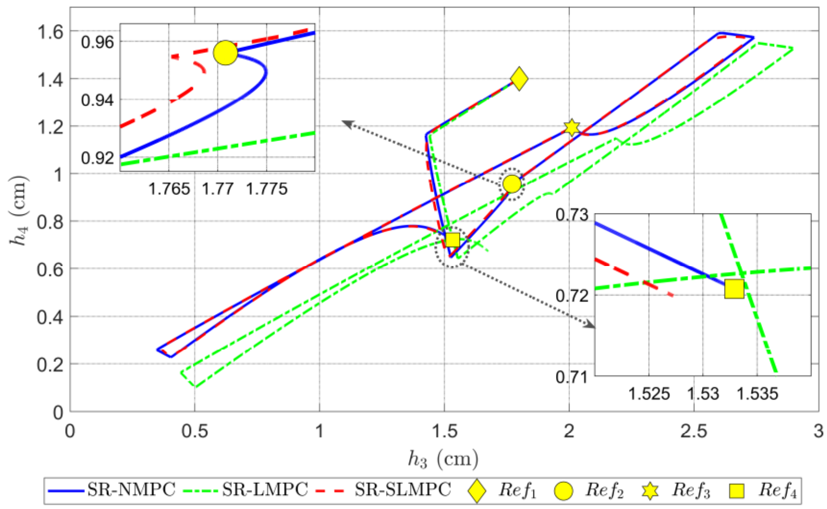


Fig. 10: State trajectory of Tanks 3 and 4

for h_1 and h_2. It is observed that during sudden changes of the reference points, SR-SLMPC decreased the error signal at each time step with greater magnitude than SR-NMPC to reach the steady state condition. This is because SR-SLMPC utilizes a series of convex optimization problems, which guarantees a global minimum solution [46]. Slightly higher MAPE is seen for SR-SLMPC compared to SR-NMPC, which may be due to the cross-coupling dynamics of the QTP system.

In contrast to the tracking analysis, Table 7 reveals that SR-LMPC outperforms SR-NMPC and SR-SLMPC in terms of execution time, as it provides feasible control actions to the QTP system more quickly, albeit with the trade-off of not precisely reaching zero offsets. It can be observed that SR-LMPC and SR-SLMPC executed their control actions 93% and 81.3% faster than SR-NMPC, respectively. The longer execution time of SLMPC compared to SR-LMPC is expected since SR-SLMPC involves additional steps, such as linearization for each time step, to solve the MPC optimization problem.

Table 7: Quantitative Performance of MPC with Varying Prediction Model

		NMPC	SLMPC	LMPC
MAPE (%)	h_1	0.31%	0.09%	0.51%
	h_2	0.30%	0.02%	0.61%
	h_3	3.34%	3.67%	4.69%
	h_4	1.34%	1.50%	8.39%
Max elapsed time (s)		0.545	0.102	0.040

4. Conclusion

In this paper, a data-driven identification of nonlinear dynamics of the four interconnected water tanks using the sparse regression technique (SR technique) was studied. The predicted dynamics were further actuated in the MPC framework by varying the prediction models utilized in the optimal control formulation, such as linear time-invariant MPC (SR-LMPC), linear time-varying via successive linearization MPC (SR-SLMPC), and nonlinear MPC (SR-NMPC).

The proposed model-free identification framework successfully delivers the nonlinear dynamics of the quadruple tank process, verified by the close tracking of system states with MAPE ranging from 0.01% - 0.02% when implementing time series validation and varying control signals to the system. Furthermore, the sparse regression-based MPC provides guaranteed tracking performance with tracking error below 5% and demonstrates the advancement of control strategy for easy performance tuning subject to constraints and varying desired state values. All studied controllers performed successfully in reference tracking, with SR-NMPC outperforming the rest, reaching zero offsets. Albeit with the tracking error up to 4.7%, SR-LMPC exhibits the fastest execution time compared to the rest of the controllers. The SR-SLMPC framework presents an opportunity to provide a trade-off, demonstrating similar outcomes to SR-NMPC while achieving an execution time that is 80% faster.

The proposed work will be extended in future work to identify the nonlinear dynamics and to control large-scale WDSs with multiple pumping stations and complex distribution systems from available measurements.

Funding

This research was in part under support from the National Science Foundation under Grant NSF-EPCN 2221784.

References

[1] A. Castelletti, A. Ficch`ı, A. Cominola, P. Segovia, M. Giuliani, W. Wu, S. Lucia, C. Ocampo-Martinez, B. De Schutter, J. M. Maestre, Model predictive control of water resources systems: A review and research agenda, Annual Reviews in Control (2023).

[2] US EPA, Six-year review of drinking water standards (2022). URL https://www.epa.gov/dwsixyearreview

[3] H. Oh, I. Guk, S. Chung, Y. Lee, Energy saving through modifications of the parallel pump schedule at a pumping station: A case study, Journal of Water Process Engineering 54 (2023) 104035.

[4] P.-J. Van Overloop, Model predictive control on open water systems, IOS Press, 2006.

[5] V. Fambrini, C. Ocampo-Martinez, Modelling and Decentralized Model Predictive Control of Drinking Water Networks, Institut de Robotica Informatica Industrial (IRI) (2009) 56.

[6] G. Conde, N. Quijano, C. Ocampo-Martinez, Modeling and control in open-channel irrigation systems: A review, Annual Reviews in Control 51 (2021) 153–171.

[7] R. K. Ringset, Efficient Optimization in Model Predictive Control, Ph.D. thesis, Norwegian University of Science and Technology (Jul. 2009).

[8] J. M. Maciejowski, Predictive Control with Constraints, Prentice Hall, 2002.

[9] D. Q. Mayne, Model predictive control: Recent developments and future promise, Automatica 50 (12) (2014) 2967–2986. doi:10.1016/j.automatica.2014.10.128.

[10] M. Schwenzer, M. Ay, T. Bergs, D. Abel, Review on model predictive control: an engineering perspective, The International Journal of Advanced Manufacturing Technology 117 (5-6) (2021) 1327–1349. doi:10.1007/s00170-021-07682-3.

[11] P. C. Blaud, P. Chevrel, F. Claveau, P. Haurant, A. Mouraud, Four mpc implementations compared on the quadruple tank process benchmark: pros and cons of neural mpc, IFAC-PapersOnLine 55 (16) (2022) 344–349.

[12] L. Wang, Model Predictive Control System Design and Implementation Using MATLAB®, 1st Edition, Advances in Industrial Control, Springer London, 2009. doi:10.1007/978-1-84882-331-0.
URL https://doi.org/10.1007/978-1-84882-331-0

[13] Á. Domina, V. Tihanyi, Ltv-mpc approach for automated vehicle path following at the limit of handling, Sensors 22 (15) (2022) 5807.

[14] L. Grüne, J. Pannek, L. Grüne, J. Pannek, Nonlinear model predictive control, Springer, 2017.

[15] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, M. Diehl, From linear to nonlinear mpc: bridging the gap via the real-time iteration, International Journal of Control 93 (1) (2020) 62–80.

[16] A. Bamimore, O. Taiwo, R. King, Comparison of two nonlinear model predictive control methods and implementation on a laboratory three tank system, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 5242–5247.

[17] E. Kaiser, J. N. Kutz, S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, Proceedings of the Royal Society A 474 (2219) (2018) 20180335.

[18] U. Fasel, E. Kaiser, J. N. Kutz, B. W. Brunton, S. L. Brunton, Sindy with control: A tutorial, in: 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 16–21.

[19] M. Askari, M. Moghavvemi, H. A. Almurib, K. M. Muttaqi, Multivariable offset-free model predictive control for quadruple tanks system, IEEE transactions on industry applications 52 (2) (2015) 1882–1890.

[20] J. Diaz-Ortiz, P. Alvarado-Medellin, A. M. Ramirez-Aguilera, H. Badillo-Almaraz, R. O. Gomez, C. B. Capetillo, Water distribution system calibration using the finite element method coupled to a genetic algorithm, Journal of Water Process Engineering 54 (2023) 104017.

[21] T. M. Tung, Z. M. Yaseen, et al., A survey on river water quality modelling using artificial intelligence models: 2000–2020, Journal of Hydrology 585 (2020) 124670.

[22] Y. Seo, S. Kim, O. Kisi, V. P. Singh, Daily water level forecasting using wavelet decomposition and artificial intelligence techniques, Journal of Hydrology 520 (2015) 224–243.

[23] G. Thiele, A. Fey, D. Sommer, J. Krüger, System identification of a hysteresis-controlled pump system using sindy, in: 2020 24th International Conference on System Theory, Control and Computing (ICSTCC), IEEE, 2020, pp. 457–464.

[24] Y. Liu, Y. Zheng, Y. Liang, S. Liu, D. S. Rosenblum, Urban water quality prediction based on multi-task multi-view learning, in: Proceedings of the 25th international joint conference on artificial intelligence, 2016.

[25] S. Wang, A. Chakrabarty, A. F. Taha, Data-driven identification of dynamic quality models in drinking water networks, Journal of Water Resources Planning and Management 149 (4) (2023) 04023008.

[26] A. Seyedzadeh, S. Maroufpoor, E. Maroufpoor, J. Shiri, O. Bozorg-Haddad, F. Gavazi, Artificial intelligence approach to estimate discharge of drip tape irrigation based on temperature and pressure, Agricultural Water Management 228 (2020) 105905.

[27] D. Zhang, J. Lin, Q. Peng, D. Wang, T. Yang, S. Sorooshian, X. Liu, J. Zhuang, Modeling and simulating of reservoir operation using the artificial neural network, support vector regression, deep learning algorithm, Journal of Hydrology 565 (2018) 720–736.

[28] A. N. Ahmed, F. B. Othman, H. A. Afan, R. K. Ibrahim, C. M. Fai, M. S. Hossain, M. Ehteram, A. Elshafie, Machine learning methods for better water quality prediction, Journal of Hydrology 578 (2019) 124084.

[29] N. Nasir, A. Kansal, O. Alshaltone, F. Barneih, M. Sameer, A. Shanableh, A. Al-Shamma'a, Water quality classification using machine learning algorithms, Journal of Water Process Engineering 48 (2022) 102920.

[30] K. M. Balla, T. N. Jensen, J. D. Bendtsen, C. S. Kallesøe, Model predictive control using linearized radial basis function neural models for water distribution networks, in: 2019 IEEE Conference on Control Technology and Applications (CCTA), IEEE, 2019, pp. 368–373.

[31] T. Yang, A. A. Asanjan, E. Welles, X. Gao, S. Sorooshian, X. Liu, Developing reservoir monthly inflow forecasts using artificial intelligence and climate phenomenon information, Water Resources Research 53 (4) (2017) 2786–2812.

[32] S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proceedings of the national academy of sciences 113 (15) (2016) 3932–3937.

[33] S. L. Brunton, J. L. Proctor, J. N. Kutz, Sparse identification of nonlinear dynamics with control (sindyc), IFAC-PapersOnLine 49 (18) (2016) 710–715.

[34] J. Khazaei, R. S. Blum, Model-free distributed control of dynamical systems, International Journal of Information and Communication Engineering 16 (10) (2022) 475–480.

[35] K. H. Johansson, The quadruple-tank process: A multivariable laboratory process with an adjustable zero, IEEE Transactions on control systems technology 8 (3) (2000) 456–465.

[36] S. Larsson, V. Thomée, Partial differential equations with numerical methods, Vol. 45, Springer, 2003.

[37] B. Bhadriraju, J. S.-I. Kwon, F. Khan, Risk-based fault prediction of chemical processes using operable adaptive sparse identification of systems (oasis), Computers & Chemical Engineering 152 (2021) 107378.

[38] L. Zhang, H. Schaeffer, On the convergence of the sindy algorithm, Multiscale Modeling & Simulation 17 (3) (2019) 948–972. doi:10.1137/18M1189828.

[39] S. C. Chapra, R. P. Canale, Numerical methods for engineers, Mcgraw-hill, 2020.

[40] G. Sánchez, M. Murillo, L. Genzelis, N. Deniz, L. Giovanini, Mpc for nonlinear systems: A comparative review of discretization methods, in: 2017 XVII Workshop on Information Processing and Control (RPIC), IEEE, 2017, pp. 1–6.

[41] A. Zhakatayev, B. Rakhim, O. Adiyatov, A. Baimyshev, H. A. Varol, Successive linearization based model predictive control of variable stiffness actuated robots, in: 2017 IEEE international conference on advanced intelligent mechatronics (AIM), IEEE, 2017, pp. 1774–1779.

[42] M. H. Murillo, A. C. Limache, P. S. Rojas Fredini, L. L. Giovanini, Generalized nonlinear optimal predictive control using iterative state-space trajectories: Applications to autonomous flight of uavs, International Journal of Control, Automation and Systems 13 (2015) 361–370.

[43] J. Nocedal, S. Wright, Numerical Optimization, 2nd Edition, Springer Series in Op. Res. and Fin. Eng., New York, NY, 2006.

[44] MathWorks, Constrained nonlinear optimization algorithms, https://www.mathworks.com/constrained-nonlinear-optimization-algorithms.html, accessed on May 5, 2023 (2023).

[45] I. Alvarado, D. Limon, D. M. De La Peña, J. M. Maestre, M. Ridao, H. Scheu, W. Marquardt, R. Negenborn, B. De Schutter, F. Valencia, et al., A comparative analysis of distributed mpc techniques applied to the hd-mpc four-tank benchmark, Journal of Process Control 21 (5) (2011) 800–815.

[46] P. Couchman, B. Kouvaritakis, M. Cannon, Ltv models in mpc for sustainable development, International Journal of Control 79 (1) (2006) 63–73.