

On the study of control effectiveness and computational efficiency of reduced Saint-Venant model in model predictive control of open channel flow

M. Xu*, P.J. van Overloop, N.C. van de Giesen

Section of Water Resources Management, Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA, Delft, The Netherlands

ARTICLE INFO

Article history:

Received 18 June 2010

Received in revised form 27 November 2010

Accepted 30 November 2010

Available online 7 December 2010

Keywords:

Integrator Delay model

Model predictive control

Model reduction

Open channel flow

Proper Orthogonal Decomposition

Saint-Venant equations

ABSTRACT

Model predictive control (MPC) of open channel flow is becoming an important tool in water management. The complexity of the prediction model has a large influence on the MPC application in terms of control effectiveness and computational efficiency. The Saint-Venant equations, called SV model in this paper, and the Integrator Delay (ID) model are either accurate but computationally costly, or simple but restricted to allowed flow changes. In this paper, a reduced Saint-Venant (RSV) model is developed through a model reduction technique, Proper Orthogonal Decomposition (POD), on the SV equations. The RSV model keeps the main flow dynamics and functions over a large flow range but is easier to implement in MPC. In the test case of a modeled canal reach, the number of states and disturbances in the RSV model is about 45 and 16 times less than the SV model, respectively. The computational time of MPC with the RSV model is significantly reduced, while the controller remains effective. Thus, the RSV model is a promising means to balance the control effectiveness and computational efficiency.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

More and more attention is being paid to increase the efficiency of water delivery and usage and decrease spilling of water. From an operational water management point of view, proper real-time control techniques can help achieve this goal. Most of the research and applications of diverse control techniques on open channel flow were originally designed for irrigation systems, for example [1–3]. Model predictive control (MPC) is one of the most advanced control techniques, as it can deal with setting an optimal trade-off between water level deviations from the target level and flow changes while taking their physical limitations (constraints) into account. The drawback of this methodology is the heavy computational demand. With the improvements of both hardware and software, the application of MPC became practically possible. Advances in hardware, in terms of computer capacities, are outside the scope of this research. Software improvements can be achieved through faster optimization algorithms or through the reduction of model complexity. Model reduction is the focus of this paper.

MPC requires a prediction model to estimate the dynamic system behavior over a prediction horizon. Different prediction models have different model accuracy and complexity. In general, it can be stated that the larger the model complexity, the higher the model accuracy. However, the model accuracy and complexity influence the control effectiveness in terms of control goal

achievement in the closed-loop implementation, and computational efficiency regarding the computational time in delivering an accurate solution of the constraint optimization problem in MPC. The trade-off between model accuracy and complexity is the central consideration of this paper.

The Integrator Delay (ID) model [4] is a commonly used prediction model for MPC in water management. The model is usually linearized around the average flow condition and only has a small number of states depending on the number of controlled water levels and delay steps. However, due to the linearization, it is limited to small flow changes. In contrast with the ID model, the Saint-Venant (SV) equations accurately calculate the system dynamics over the full range of flow conditions, but this mathematical model includes many states. It is extremely computationally costly when used in MPC. Therefore, we propose the reduced Saint-Venant (RSV) model developed in [5]. The model captures the main dynamics of the SV model, but the number of states and disturbances is significantly reduced. Additionally, the RSV model does not have the limitation of small flow change as long as the coherent flow structure is detected through snapshots of the full set of states in an off-line simulation.

This paper compares the RSV model and control performance with the SV model and ID model. It extends previous work by Xu and Van Overloop [5] with more realistic flow conditions. In addition, the number of terms in disturbance vector is reduced, which further reduces the computational time. The paper is structured as follows: After briefly introducing MPC for open channel flow, three different MPC prediction models are discussed with focus on the

* Corresponding author. Tel.: +31 (0) 152782345; fax: +31 (0) 152785559.
E-mail address: min.xu@tudelft.nl (M. Xu).

RSV model. Then the different model and control performances are elaborated in the results through a test case. Finally, based on the results, the conclusion that the RSV model is a promising means to balance the trade-off between control effectiveness and computational efficiency is drawn.

2. Model predictive control of open channel flow

When controlling open channel flow, a common goal is to keep a specific water level at a target level by smoothly adjusting the controllable structures, for example weirs, gates and pumps. Additionally, the control actions for the structures need to remain within the physically possible capacities, such as maximum gate opening or maximum pump capacity. MPC solves this goal as an optimization problem, formulated as an objective function subject to certain constraints. The objective function should capture the future dynamic behavior which can be described by a prediction model (internal model). Physical disturbances, such as rain inflows or off-take flows to water users, can also be included in the model. Finally, an optimization technique is required to calculate the optimal control actions. The whole process runs with a finite prediction horizon, and only the first optimal control action is applied in each closed loop step (receding horizon). Fig. 1 shows the diagram of the MPC structure.

2.1. State-space model formulation

From a control point of view, it is convenient to structure the prediction model into a state-space formulation. The model could be either linear time invariant or linear time variant depending on whether the state matrix, control matrix, and disturbance matrix are fixed or change over time. A linear model can make the controller design easier. A linear time variant state-space formulation looks as follows:

$$\begin{aligned} \dot{x}(k+1) &= A(k) \cdot x(k) + Bu(k) \cdot u(k) + Bd(k) \cdot d(k) \\ y(k) &= C \cdot x(k) \end{aligned} \quad (1)$$

where k is the discrete time step, A is the state matrix, Bu is the control input matrix, Bd is the disturbance matrix, C is the output matrix, x is the state vector, y is the output, u is the control input vector, d is the disturbance vector. In open channel flow control, x contains the states of water level deviation from the target level, u is the change of control flow, and d can include physical disturbances, e.g. rain inflow or lateral flow, and terms generated from discretization when constructing the state-space model. Since the states only contain water level deviations, when substituting flows at time step $k+1$ in the momentum equation into the continuity equation, all the flow related terms at time step k during the discretization of the Saint-Venant equations are constructed into the disturbance vector. Output y is the same as the state vector x in this case and the output matrix C is an identity matrix.

Note that a non-linear system can be approximated by a linear time variant model. As a prediction model of MPC, the Saint-Venant model or reduced SV model in this case are linearized at

specific flow conditions of each time step over the prediction horizon. The values of matrices A , Bu and Bd over the prediction horizon can be estimated by running the prediction model with the optimal control actions of the previous time step. This is referred to as ‘forward estimation’ in [6]. The ‘forward estimation’ is a pure Saint-Venant model simulation and there is no control involved.

The state-space model in Eq. (1) assumes that the complete system states $x(k)$ are known. In practice, it is not possible to measure all states. For example, water levels in an irrigation system are measured typically only at the most upstream and downstream end of each canal reach, which represent the first and last values in the state vector $x(k)$, while the intermediate water levels are unmeasured. The test case in this paper has the same set of measurements. Therefore, a proper estimation of unmeasured states is needed based on a limited number of measured values. The Kalman filter [7] is a commonly used estimator. It assumes a certain initial model error ε and independent white noise with a normal distribution on both measurements $p(w_{\text{meas}}) \sim N(0, \sigma_1^2)$ and the model $p(w_{\text{model}}) \sim N(0, \sigma_2^2)$. Because the model states are correlated, when updating the measured model states with the measurements, the unmeasured states can be updated as well [7]. The estimator equation is given as follows, driven by the error between measured output $y_m(k)$ and corresponding model state $x(k)$. And the model state $x(k)$ is replaced by the updated estimation $\hat{x}(k)$:

$$\hat{x}(k) = x(k) + L \cdot [y_m(k) - H_m \cdot x(k)] \quad (2)$$

where H_m is the measurement matrix, L is the optimal Kalman gain, which represents the relative importance of measurements against model calculation. Its calculation is based on the measurement noise covariance RN which equals $E(w_{\text{meas}} \cdot w_{\text{meas}}^T)$, measurement matrix H_m , and *a priori* model error covariance P_k^- . Here E means expectation.

Before measurements are available, the model runs to calculate the *a priori* P_k^- , which equals $AP_{k-1}A^T + QN$, where A is the state (system) matrix, P_{k-1} is *a posteriori* model error covariance which is calculated from the previous step, and QN is the model noise covariance which equals $E(w_{\text{model}} \cdot w_{\text{model}}^T)$. Before starting, P_0 is calculated by assuming a constant model error ε in this case, which equals $E(\varepsilon \cdot \varepsilon^T)$. After the measurements are available, L is calculated as $P_k^- H_m^T (H_m P_k^- H_m^T + RN)^{-1}$, then the states are updated with Eq. (2) and P_k is also updated for the next calculation of the *a priori* model error covariance, which equals to $(I - LH_m)P_k^-$ with I the identity matrix.

2.2. Optimization problem

According to the control goal, MPC formulates the objective function J to minimize the water level deviation from the target level and the change of structure flow along the prediction horizon n . A typical quadratic function is the result, which can be solved with the “quadprog” function in MATLAB [8]:

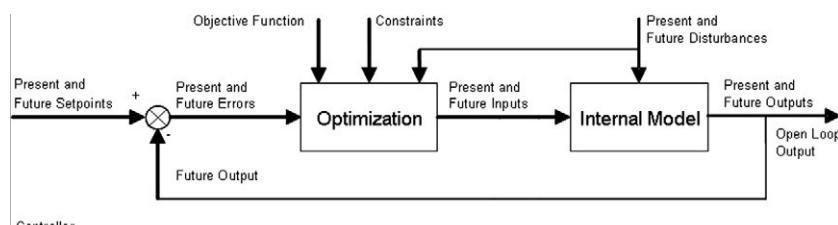


Fig. 1. MPC structure.

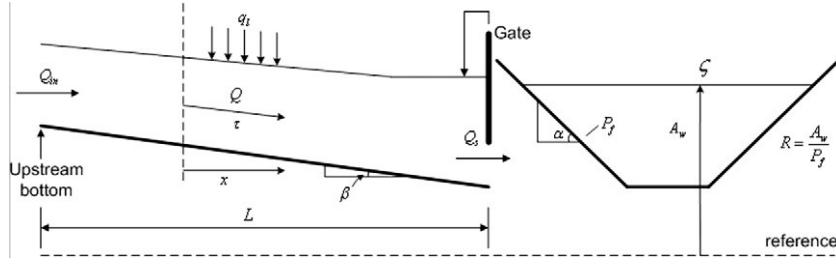


Fig. 2. Canal reach schematization.

$$J = \min \sum_{k=1}^n \left\{ \hat{x}^T(k) \cdot Q_x \cdot \hat{x}(k) + \Delta Q_s^T(k) \cdot R_u \cdot \Delta Q_s(k) \right\} \quad (3)$$

Subject to : $\dot{\hat{x}}(k+1) = A(k) \cdot \hat{x}(k) + Bu(k) \cdot u(k) + Bd(k) \cdot d(k)$

$$Q_{s,min} \leq Q_s \leq Q_{s,max}$$

where n is the number of prediction steps, Q_x and R_u are the weighting factors on \hat{x} and ΔQ_s , respectively. Note that $\Delta Q_s(k)$ is the change of control flow in open channel flow notation, which represents the control notation $u(k)$ in Eq. (1). Assuming the downstream water level of a canal reach is controlled, Q_x only penalizes on this controlled water level deviation from the target level. Q_s is the controlled flow, which is constricted between the minimum and maximum flow of $Q_{s,min}$ and $Q_{s,max}$.

When substituting the model (1) with the estimated state $\hat{x}(k)$ into the objective function (3) and minimizing J with respect to the control action U (or ΔQ_s) over the prediction horizon, the problem becomes:

$$\frac{\partial J}{\partial U} = \frac{1}{2} \cdot U^T \cdot H \cdot U + f \cdot U = 0 \quad (4)$$

where H is the Hessian matrix and f is the Lagrangian matrix. These matrices are calculated as the input of the optimization algorithm: (Assume the number of controls, e.g. the number of control gates, is nu)

$$\begin{cases} H = 2 \cdot (Bu_n^T \cdot Q_{x,n} \cdot Bu_n + R_{u,n}) & \text{dimension : } (n \cdot nu) \times (n \cdot nu) \\ f = 2 \cdot (\hat{x}(k)^T \cdot A_n^T + D_n^T \cdot Bd_n^T) \cdot Q_{x,n} \cdot Bu_n & \text{dimension : } 1 \times (n \cdot nu) \end{cases} \quad (5)$$

Note that all the matrices except $\hat{x}(k)$ are over the prediction horizon labeled by the subscript 'n'. The calculation of these large matrices is presented in Appendix A, when deleting the subscript 'r' in the variable notation. Although the appendix is formulated for the reduced model, the procedure fits the general MPC internal model generation.

In MPC, the total control process time contains the optimization time itself, the time to build up the input matrices of optimization (H and f) and their related matrices (A_n, Bu_n, Bd_n), the time to call prediction models which is determined by the model complexity and the rest of some process time related to the matrix size. The first two time consumptions are discussed below.

First of all, from the matrices dimension in Eq. (5), it is easily verified that matrices H and f in different models have the same dimensions, separately, as long as the number of control inputs nu and the number of time steps over the prediction horizon n are the same. Thus, H and f related matrix operations in MPC will not influence the optimization time. The optimization time with different models will only be affected by the search space and the initial search point.

The time to build up the large matrices over the prediction horizon (A_n, Bu_n, Bd_n, H and f) may differ significantly with different models. Because building up these matrices contains a lot of matrix

multiplication (see Appendix A, without subscript 'r') and the matrix size, mainly determined by the number of states and disturbances in this case, largely influences the matrix multiplication time, it is necessary to decrease the matrix size by means of model reduction, in order to reduce the computational time.

3. Process model formulation

Open channel flow is commonly described by the Saint-Venant equations. They consist of the mass and momentum conservation equations, which are expressed in Eqs. (6) and (7). An example of an open canal is schematized in Fig. 2 with major variables.

$$\frac{\partial A_w}{\partial t} + \frac{\partial Q}{\partial x} = q_l \quad (6)$$

$$\frac{\partial Q}{\partial t} + \frac{\partial (Qv)}{\partial x} + gA_w \frac{\partial \zeta}{\partial x} + g \frac{Q|Q|}{C_z^2 RA_w} = 0 \quad (7)$$

where A_w is the wetted area [m^2], Q is the flow [m^3/s], q_l is the lateral inflow per unit length [$\text{m}^3/\text{s}/\text{m}$], v is the average flow velocity [m/s], which equals Q/A_w , ζ is the water depth above the reference plane [m], C_z is the Chezy coefficient [$\text{m}^{1/2}/\text{s}$], R is the hydraulic radius [m], which equals A_w/P_f (P_f is the wetted perimeter [m]) and g is the gravity acceleration [m/s^2], t is time and x is horizontal length.

3.1. State-space model formulation with SV model

The SV model is usually not used as the prediction model in MPC because of the costly computation to achieve an accurate prediction, but it is the basis of the RSV model. The SV model presented here is used to test the model and control performance with the RSV model. Following to Stelling and Duinmeijer [9], the spatial discretization of the Saint-Venant equations uses the staggered grid scheme listed in Eqs. (8) and (9), and the integration scheme in time is based on the ' θ method', e.g. $\zeta^{n+\theta} = \theta\zeta^{n+1} + (1-\theta)\zeta^n$ in Eq. (9). The scheme is regarded as fully implicit when θ equals 1 and fully explicit when θ is 0. The fully implicit scheme normally has large wave damping especially when large time steps are used. Accurate results can be achieved when taking θ as 0.55. The advection term in Eq. (9) is calculated explicitly by first-order upwinding.

$$\frac{dA_{w,i}}{dt} = \frac{Q_{i-1/2} - Q_{i+1/2}}{\Delta x} + q_{l,i} \quad (8)$$

$$\begin{aligned} \frac{d\nu_{i+1/2}}{dt} + \frac{1}{\bar{A}_{w,i+1/2}} \left(\frac{\bar{Q}_{i+1} * v_{i+1} - \bar{Q}_i * v_i}{\Delta x} - v_{i+1/2} \frac{\bar{Q}_{i+1} - \bar{Q}_i}{\Delta x} \right) \\ + g \frac{\zeta_{i+1} - \zeta_{i-1}}{\Delta x} + g \frac{v_{i+1/2} |v_{i+1/2}|}{C_z^2 \cdot R} = 0 \end{aligned} \quad (9)$$

where $\bar{Q}_i = \frac{Q_{i-1/2} + Q_{i+1/2}}{2}$, $\bar{A}_{w,i+1/2} = \frac{A_{w,i} + A_{w,i+1}}{2}$ and $* v_i = \begin{cases} v_{i-1/2} & (\text{positive flow}) \\ v_{i+1/2} & (\text{negative flow}) \end{cases}$. Substituting Eq. (9) into (8) and writing them

into the state-space format considering that the control input is the change of gate flow (ΔQ_s), we obtain:

$$\begin{aligned} & \left[\begin{array}{cccccc} a_{1,1}(k) & a_{1,2}(k) & 0 & 0 & 0 & 0 \\ a_{2,1}(k) & a_{2,2}(k) & a_{2,3}(k) & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & a_{l-1,l-2}(k) & a_{l-1,l-1}(k) & a_{l-1,l}(k) & 0 \\ 0 & 0 & 0 & a_{l,l-1}(k) & a_{l,l}(k) & \frac{\Delta t}{T_{w,l}(k) \cdot \Delta x} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \\ & \cdot \begin{bmatrix} \hat{e}_1(k+1) \\ \hat{e}_2(k+1) \\ \vdots \\ \hat{e}_{l-1}(k+1) \\ \hat{e}_l(k+1) \\ Q_s(k+1) \end{bmatrix} \\ & = I_{l+1,l+1} \cdot \begin{bmatrix} \hat{e}_1(k) \\ \hat{e}_2(k) \\ \vdots \\ \hat{e}_{l-1}(k) \\ \hat{e}_l(k) \\ Q_s(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \cdot \Delta Q_s(k+1) \\ & + \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -b_1(k) \\ b_1(k) - b_2(k) \\ \vdots \\ b_{l-2}(k) - b_{l-1}(k) \\ b_{l-1}(k) \\ \frac{\Delta t}{T_{w,l}(k) \cdot \Delta x} Q_{in}(k+1) \end{bmatrix} \quad (10) \end{aligned}$$

where \hat{e} equals the water level from the Kalman estimator minus the target water level, Q_{in} is the inflow of the reach, Q_s is the outflow (structure flow) of the reach, ΔQ_s is the change of outflow, Δt is the control time, Δx is the spatial increment, l is the number of spatial discretization point, $T_{w,l}(k)$ is the top width at calculation points i at time step k , $a_{i,j}$ and b_i are functions of variables that change over time and are estimated by the ‘forward estimation’ [6]. By multiplying the inverse of the first matrix on both sides of Eq. (10), it gives the linear time variant state-space model format as Eq. (1). Note that the Kalman filter in Eq. (2) is usually required to estimate the unmeasured values along the canal reach.

3.2. State-space model formulation with RSV model

In order to cope with the computational burden in MPC with the SV model, a much simpler prediction model needs to be developed, containing less states and disturbances. Model reduction is an important tool to reduce model order, which can be formed as $z = \Phi \cdot z_r$. z and z_r are column vectors in original and reduced domain separately. The key process of generating the reduced vector is to calculate the basis function Φ . In linear algebra, this is formulated as an over-determination problem, which can be solved based on the least square error between the original and projected vectors. Proper Orthogonal Decomposition (POD) is a known model reduction technique, e.g. [10–12]. A snapshot method is usually incorporated into POD to capture the coherent flow structure. In addition, POD calculates an orthogonal matrix Ψ of the basis function: $\Psi^T \cdot \Phi = I$. Then the reduced vector becomes: $z_r = \Psi^T \cdot z$.

Sirovich [13] pointed out that the eigenvectors of the spatial correlation matrix (kernel matrix) are a linear combination of the snapshots and formulate the basis functions as:

$$\varphi_i = \sum_{j=1}^M \alpha_j^i \cdot z_j \quad (11)$$

where φ_i is the i th eigenvector of the kernel matrix, z_j is the j th snapshot, M is the number of snapshots, and α_j^i is a coefficient.

According to Sirovich [13], the coefficient α_j^i is selected from the eigenvector of the correlation matrix CR , assuming that M independent snapshots z_1, z_2, \dots, z_M are taken from an off-line simulation of a high-order model, the SV model in this case. Each snapshot is a column vector containing N states that are the water level deviations from a target level in the open channel flow model:

$$CR_{ij} = \frac{1}{M} (z_i^T \cdot z_j) \quad (12)$$

Finally, the number of the basis functions in use is selected based on the m largest eigenvalues of the correlation matrix CR , and the combinations of φ_i formulates the basis function matrix Φ with a dimension of $N \times m$.

The snapshots can be taken on both states and disturbances considering the state-space model formulation. Therefore, the full states $\hat{x}(k)$ from the Kalman estimator and the disturbances $d(k)$ become a function of the reduced states $\hat{x}_r(k)$ and reduced disturbances $d_r(k)$, respectively, with respect to the basis function matrix $\Phi_1(N \times m_1)$ and $\Phi_2(N \times m_2)$:

$$\begin{aligned} \hat{x}(k) &= \Phi_1 \cdot \hat{x}_r(k) \\ d(k) &= \Phi_2 \cdot d_r(k) \end{aligned} \quad (13)$$

Substituting Eq. (13) into state-space model (1), it becomes:

$$\begin{aligned} \hat{x}_r(k+1) &= A_r(k) \cdot \hat{x}_r(k) + B u_r(k) \cdot u(k) + B d_r(k) \cdot d_r(k) \\ \hat{y}_r(k) &= C_r \cdot \hat{x}_r(k) \end{aligned} \quad (14)$$

where $A_r(k) = \Psi_1^T \cdot A(k) \cdot \Phi_1$, $B u_r(k) = \Psi_1^T \cdot B u(k)$, $B d_r(k) = \Psi_1^T \cdot B d(k) \cdot \Phi_2$, $C_r = C \cdot \Phi_1$, Ψ_1^T is orthogonal with $\Phi_1 (\Psi_1^T \cdot \Phi_1 = I)$, Ψ_2^T is orthogonal with $\Phi_2 (\Psi_2^T \cdot \Phi_2 = I)$, and I is the identity matrix. $A_r(k)$ has a dimension of $m_1 \times m_1$, $B u_r(k)$ is $m_1 \times n_u$, and $B d_r(k)$ is $m_1 \times m_2$. If $m_1 \ll N$ and $m_2 \ll N$, the model order is significantly reduced.

Because of the connection between states and disturbances through the state-space model formulation, reducing the order of disturbance decreases not only the disturbance accuracy itself, but also the state accuracy. Therefore, we suggest reducing the number of disturbances to an acceptable tolerance and the number of disturbances is always higher than the number of states.

The overall state-space model over the prediction horizon is described in Appendix A. The same ‘forward estimation’ procedure described by Xu et al. [6] can be used to calculate the time varying matrices: $A_r(k) \dots A_r(k+n-1)$, $B u_r(k) \dots B u_r(k+n-1)$ and $B d_r(k) \dots B d_r(k+n-1)$, based on the optimal solution of the previous step.

For the application of the reduced model to MPC (see Fig. 1 for MPC procedure), the “internal model” calls the SV model first, then the basis functions are used to generate the RSV model over the prediction horizon by calculating time-varying matrices of A_r , $B u_r$ and $B d_r$. Although this introduces an extra procedure of generating the reduced model, the MPC is expected to take much less process time due to the time reduction in matrix multiplication with the reduced model. Note that the Kalman filter in Eq. (2) is usually required to estimate the unmeasured values along the canal reach.

Because of model reduction, the objective function used by the RSV model predictive control is changed when substituting the reduced states function in Eq. (13) into the objective function (3):

$$J = \min \sum_{k=0}^n \left\{ \hat{x}_r^T(k) \cdot Q_{x,r} \cdot \hat{x}_r(k) + \Delta Q_s^T(k) \cdot R_u \cdot \Delta Q_s(k) \right\} \quad (15)$$

$$\begin{aligned} \text{Subject to : } \hat{x}_r(k+1) &= A_r(k) \cdot \hat{x}_r(k) + B u_r(k) \cdot u(k) \\ &+ B d_r(k) \cdot d_r(k) \\ Q_{s,min} &\leq Q_s \leq Q_{s,max} \end{aligned}$$

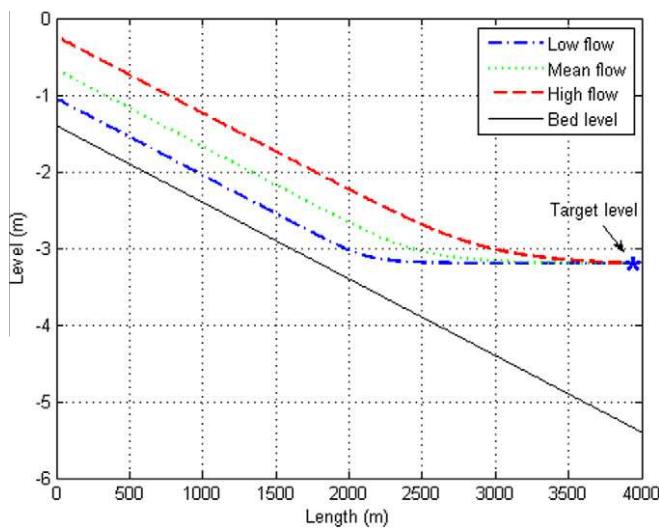


Fig. 3. Longitudinal profile.

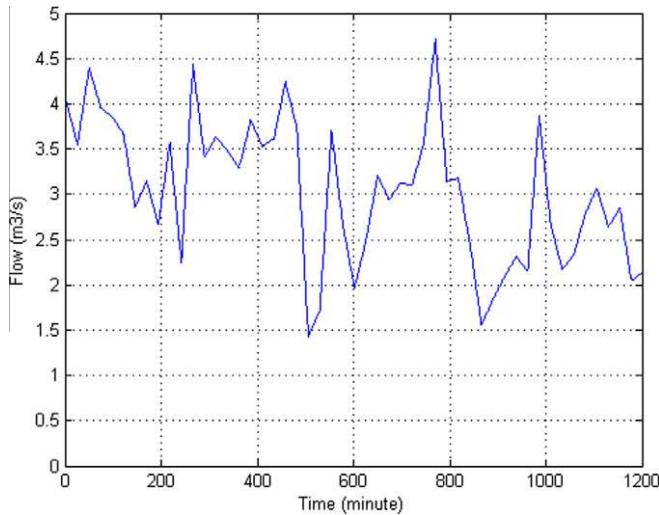


Fig. 4. Upstream flow condition for MPC test.

where $Q_{x,r} = \Phi^T \cdot Q_x \cdot \Phi$. The Hessian and Lagrangian matrices are calculated by substituting the reduced model in Eq. (14) into the objective function (15).

3.3. State-space model formulation with ID model

In order to compare the control effectiveness and computational efficiency with the RSV model, the commonly used ID model is presented here as the ultimate simplification of open channel flow. The mathematical expression of the ID model is a mass balance combined with a water travelling delay:

$$e(k+1) = e(k) + \frac{\Delta t}{A_s} [Q_{in}(k - k_d) - Q_s(k)] \quad (16)$$

where e is the water level deviation from the target water level at the downstream end of the canal reach [m], Δt is the control time [s], A_s is the storage area in the backwater part [m^2], Q_{in} is the upstream inflow [m^3/s], Q_s is the downstream structure flow [m^3/s], k is the time step [−], k_d is the number of delay steps [−], which equals the delay time τ [s] divided by the control time Δt and k_d is always rounded up to be conservative.

The ID model requires the determination of the two pool properties: delay time and storage area. They are pre-determined with a hydraulic model simulation through a standard procedure, which uses a small step flow change on the upstream side and a constant downstream flow, following [4,14]. Kalman filter is not necessary for the ID model, since the downstream water level in the model is directly measured in practice.

Due to the linearization, the ID model is inaccurate over the entire simulation period. Fig. 3 shows water level profiles with different flow conditions. The two pool properties, especially the storage area in the backwater part, are significantly different under the three flow conditions. This is a disadvantage of the ID model.

4. Test case

The test canal reach has a total length of 4000 m with a downstream gate flow controlling the upstream water level of the gate to -3.2 m (negative value means below mean sea level). The upstream flow trajectory shown in Fig. 4 was considered as a known physical disturbance and used to test the MPC controller with three models. The maximum downstream gate flow was assumed to be $4 m^3/s$. Note that a virtual gate is used here, which only forces a certain discharge instead of gate opening. It works as a pump. No lateral flow was considered. The canal geometric parameters are listed in Table 1. Each of the simulation continued for 20 h with a time step of 2 s. The θ coefficient in time integration of the simulation model is set to 0.55, in order to avoid strong wave damping and keep the model accuracy. But the θ is set to 1 for the SV and RSV model in MPC prediction model, in order to avoid model instability with large control time step of 240 seconds.

We assume water level measurements are only available at the upstream and downstream end of the reach. For the Kalman filter design, when using SV and RSV models, we assume that both the measurements and the model have normal distributed white noise and the model also has a certain initial error. In addition, the two measurements have the same white noise. The parameters are listed in Table 2.

The three implementations of MPC have the same control setup, for example the same weighting factors, the same control time step Δt of 240 s with a prediction horizon of 2 h. This gives a prediction

Table 1
Test canal geometric parameters.

Length (m)	Upstream bottom level (m)	Bottom slope $\tan(\beta)$	Bottom width (m)	Side slope $\tan(\alpha)$	Chezy friction ($s^{1/2}/m$)
4000	-1.4	1:1000	1	1:1.5	45

“-1.4” means the elevation is 1.4 m below mean sea level.

Table 2
Parameters for Kalman filter design.

Measurement noise $p(w_{meas})$ (m)	Model noise $p(w_{model})$ (m)	Initial model error ε (m)
$N(0, 0.001^2)$	$N(0, 0.005^2)$	0.01

* The unit of m on noise is for mean and standard deviation, and the noise variances have a unit of m^2 .

Table 3
Weighting factors.

Weighting factor	Q_x	R_u
400	400	4

length n of 30 steps. The weighting factors on water level deviation from the target level (Q_x) and change of gate flow (R_u) are selected according to the MAVE factor that represents the Maximum Allowed Value Estimate [14]. The maximum allowed water level deviation from the target level (assumed 10 cm) and the maximum gate flow ($4 \text{ m}^3/\text{s}$) are used as the reasonable initial guess of MAVE factors. Because the states and control input are in units of m and m^3/s , by taking the reciprocal of the squared MAEV factors, the objective function can be normalized. It is allowed to make some additional tuning on the penalties through trial-and-error. The tuned weighting factors are listed in Table 3.

4.1. SV model setup

The SV model is spatially discretized into 500 calculation points with a space step Δx equal to 8 meters. With the 30 prediction steps, the controller gives 15,531 (501×31) present and future states in total. In order to use the change of flow for the control input, an extra state Q_s is added to the state, (see Eq. (10)). The optimization problem was unsolvable on a 32-bit computer due to the memory limit. It was tractable on a 64-bit computer with an 8 Gb internal memory. From another perspective, this shows the heavy calculation burden of large matrix multiplication and the importance of model reduction.

4.2. RSV model setup

In order to generate the RSV model, the SV model was simulated with the upstream flow trajectory Q_m according to Fig. 5, and the downstream water level was kept to the target water level through feedback control of the downstream gate. During the simulation, 100 independent snapshots were taken on both states and disturbances. The state basis function was formulated by the 10 eigenvectors corresponding to the 10 dominant eigenvalues of the state correlation matrix, and the disturbance basis function used 30 corresponding eigenvectors of the disturbance correlation matrix. The number of eigenvectors in use was found by trial-and-error in this paper.

4.3. ID model

The model is linearized around the average flow condition of $3.04 \text{ m}^3/\text{s}$. With a $\pm 0.1 \text{ m}^3/\text{s}$ step change in upstream flow, the test system is estimated with 8 delay steps (k_d) with 240 s control time

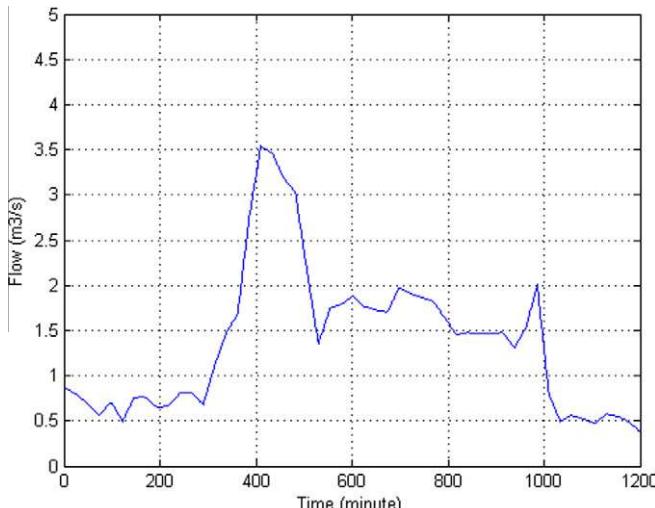


Fig. 5. Upstream flow condition for reduced model.

and 7600 m^2 storage area (A_s), through the downstream water level response, according to the description in Section 3.3.

4.4. MPC performance indicators

After the test of three implementations of MPC, the control performance needs to be analyzed. According to Clemmens et al. [15], several performance indicators can be used to examine the water level error and gate discharge, which represent the overall MPC behavior. Maximum Absolute Error (MAE) is one of the water level indicators, which concerns the percentage of maximum absolute water level deviation from the target level against the target level. The calculation is as follows:

$$\text{MAE} = \frac{\max(|y_t - y_{\text{target}}|)}{y_{\text{target}}} \quad (17)$$

where: y_t is controlled water level at time step t , y_{target} is the target water level.

Integrated Absolute Discharge Change (IAQ) is an indicator of the change of gate discharge which reflects the tear and wear of the gate along the whole simulation. IAQ is calculated as follows:

$$\text{IAQ} = \sum_{t=t_1}^{t_2} (|Q_t - Q_{t-1}|) - |Q_{t_1} - Q_{t_2}| \quad (18)$$

where: Q_t is gate discharge at time step t , t_1 and t_2 are the initial and final time step separately.

5. Results

This part is intended to demonstrate the results of model accuracy and complexity, control effectiveness, and computational efficiency. It shows the advantages of using the RSV model in MPC compared to the other two models. It is assumed that the SV model is the most accurate one in describing the open channel flow and the MPC is the most effective with the SV model under the same control parameter setup. It is noticed that there is always a tradeoff between the MPC control effectiveness and computational efficiency.

5.1. Results of RSV model accuracy and model complexity

As model complexity increase, the model accuracy is expected to increase. This principle is reflected in Fig. 6 which shows the influence of both state and disturbance reduction on model accuracy. The figure is produced by the flow condition in Fig. 5.

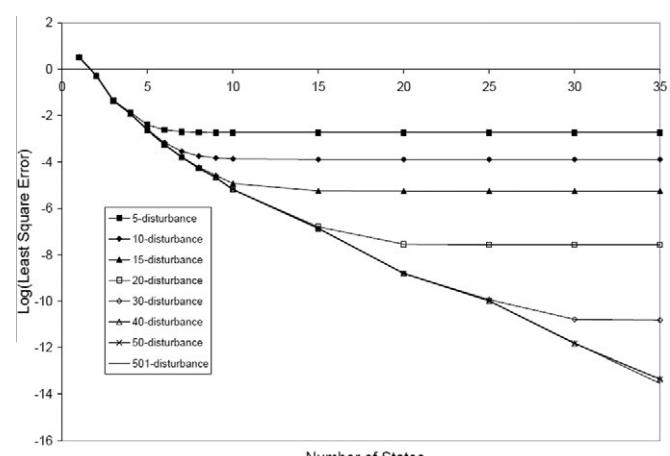


Fig. 6. Accuracy vs complexity.

Fig. 6 is a semi-logarithmic plot with the model accuracy on the y-axis, calculated by the sum of least square water level errors along the canal reach between the projected high-order model and the SV model over the 20-h simulation period. Here the projected high-order model means an SV model converted back from the RSV model. The model complexity is represented by the number of states and disturbances.

Fig. 6 can be used as a selection reference of the number of eigenvectors (number of states or disturbances) in use. In each disturbance scenario of **Fig. 6**, the model accuracy has an exponential change first with respect to the number of states and stays flat under a certain number-of-disturbance threshold. This implies that the number of states in selection should be on the skewed line. When the number of states remains unchanged, reducing the order of disturbance decreases the model accuracy.

The result of the RSV model (after projecting to the high-order model) is presented in **Fig. 7**. It shows the water level difference along the canal reach between the RSV model and the SV model. The accuracy of the RSV model is outstanding, with an insignificant water level difference of less than ± 3 mm from the SV model.

5.2. Results of control effectiveness and computational efficiency in MPC

5.2.1. Control effectiveness

Because of the change of objective function in the RSV model predictive control, it is unfair to compare the control effectiveness with different models through the objective function values. Instead, it is more interesting to compare the closed-loop optimization results, since it reflects how effective the controller acts exactly on the water system. The MPC results with different models are shown in **Figs. 8** and **9** for the controlled downstream gate flow (Q_s) and the controlled water level.

In **Fig. 8**, the upstream flow (Q_{in}) is also presented, which works as a known physical disturbance on the canal reach. **Fig. 8** clearly shows the downstream gate flow constraint of $4 \text{ m}^3/\text{s}$ and demonstrates the advantage of MPC in prediction. Because of the flow limit, the gate flow increases in advance to decrease the water level to create extra storage. For example, the fast water level drop at about 800 min simulation time (the 2.4×10^4 point on the x-axis)

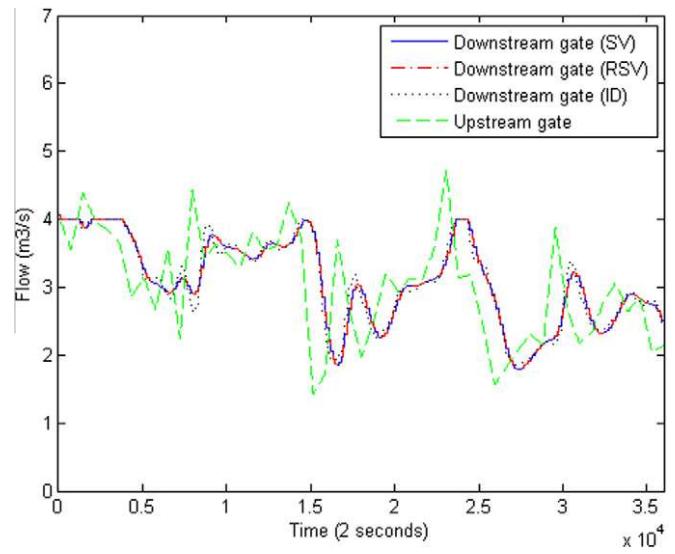


Fig. 8. Gate flow.

in **Fig. 9** is due to the prediction of the peak flow (dashed green line) in **Fig. 8**. This is a common feature of MPC regardless of the prediction model type.

According to **Fig. 9**, the overall MPC performance with the three models is good. The water level is controlled around the target level with a maximum deviation of 4 cm . But the controlled water level with the RSV model follows the SV model track better and is more accurate than the ID model. **Fig. 9** obviously shows that the controlled water level in the ID model (dotted black line) shifts towards the right, which means the overall delay steps in the ID model are overestimated. In addition to the fixed storage area, the water level fluctuates more in the ID model than in the other two models.

The performance indicators of Maximum Absolute Error (MAE) and Integrated Absolute Discharge Change (IAQ) with three different models are presented in **Table 4**. This table demonstrates the overall performance of MPC with three different models. The results are equivalent to the expectation that the SV model is the

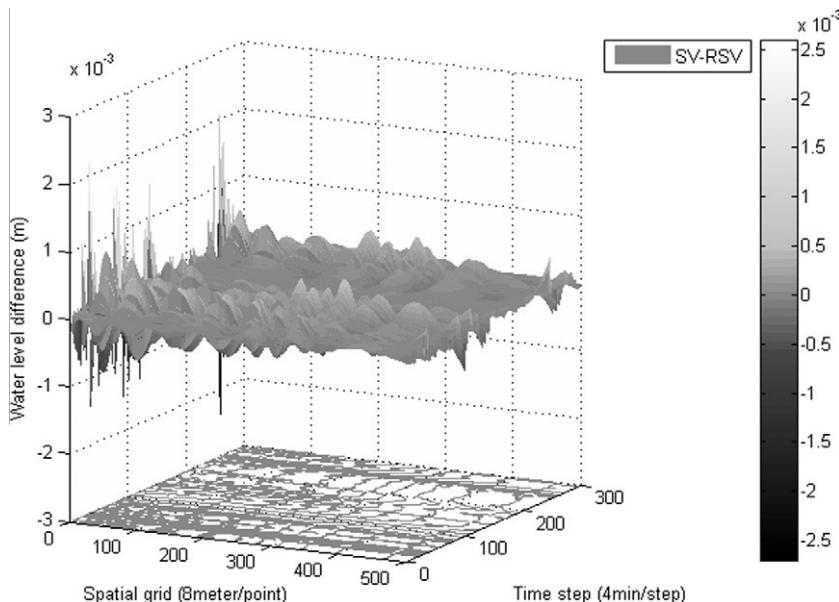


Fig. 7. Water level difference between models.

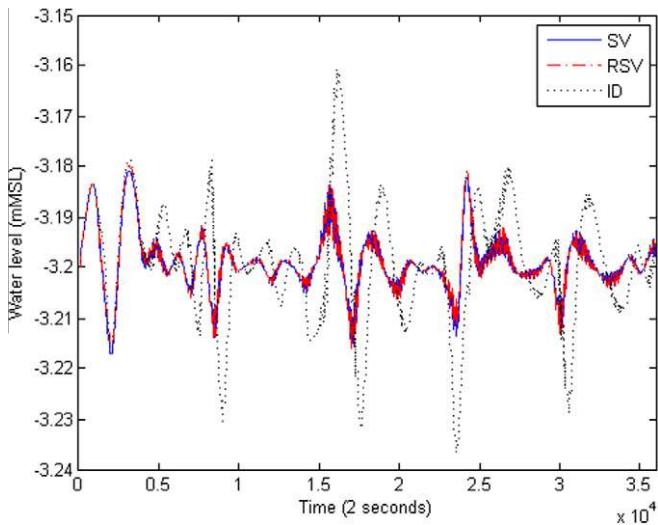


Fig. 9. Water level.

Table 4
Overall performance of MPC.

	Maximum absolute value (MAE) (%)	Integrated absolute discharge change (IAQ) (m^3/s)
SV	-0.60	13.55
RSV	-0.63	13.53
ID	-1.23	16.33

* MAE is negative because the target level is negative.

most accurate, the RSV model follows the SV model track well and they both outperform the ID model significantly.

5.2.2. Computational efficiency

The computational efficiency in MPC is reflected by the computational time. Fig. 10 shows the time spent in the control process with different prediction models. The control process time of using the SV model and RSV model includes the forward estimation and Kalman filter as well. The figure shows that the computational speed of the RSV model is more than 8 times faster than that of the SV model. The control process time of using the ID model is extremely low in Fig. 10. The most important reason is that the ID model is much simpler than the other two and has much less number of states and disturbances. Besides, the controller only calls the ID model once per control step, while the other two models are called n times per control step (n is the number prediction steps) within the controller and the forward estimation, and 120

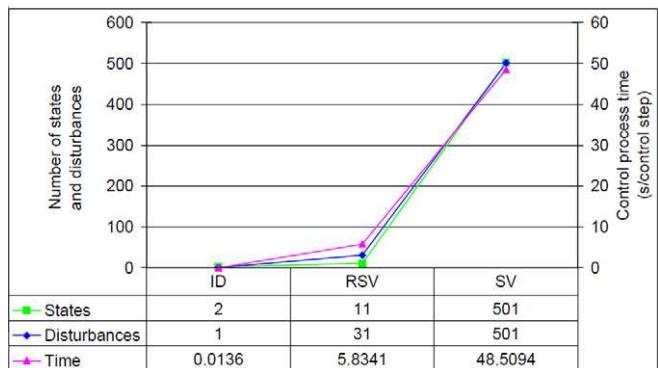


Fig. 10. Computation efficiency.

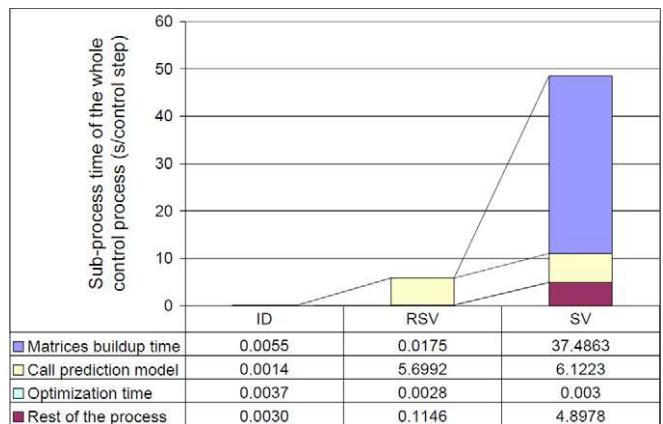


Fig. 11. Influence of matrix size.

times per control step (simulation steps between two control steps) in the Kalman filter.

Most of the control process time of using the SV model is spent on the state matrix, control matrix, and disturbance matrix related matrix multiplication over the prediction horizon. Fig. 11 shows an example of the influence of the matrix size on the control process time. It points out that building up the state matrix, control matrix and disturbance matrix over the prediction horizon in MPC takes 77.3% of the total control process time with the SV model in this case, while it only takes 0.3% with the RSV model due to the reduced matrix size. Most of the time taken (97.7%) with the RSV model is shifted to call the prediction model within the controller, the forward estimation, and the Kalman filter. The rest of the process time is also reduced significantly, since many calculations are related to these large matrix multiplications.

As discussed in Section 2.2, the optimization time is only affected by the search space and the initial search point. Fig. 11 shows that the difference in optimization time consumption with three models is insignificant. In addition, the optimization time is relatively short because the control problem is very simple in this case.

6. Conclusions

This paper explored the application of a model reduction technique on model predictive control. The idea and procedure of proper orthogonal decomposition, implemented with the snapshots method, were illustrated as an effective way of generating a reduced model.

The reduced model RSV is very accurate in describing the flow dynamics. It keeps the model structure of the SV model, conquers the limitations of the ID model, and functions properly over the entire flow range. The generated RSV model is also efficient for large scale problems, in terms of the number of states and disturbances.

Both implementations of MPC with the SV model and RSV model need Kalman filter to estimate the unmeasured states. Thus extra computational time is added to the system, but it is rather limited. Comparing with the MPC of using the SV model, the RSV model significantly reduces the computational time by reducing the matrix size. Although this computational time is higher than the ID model, it is very acceptable. Therefore, it can be concluded that the reduced model is capable of balancing the control effectiveness and computational efficiency in MPC, and the POD model reduction technique is applicable to the MPC prediction model.

In addition, the flow condition used for MPC in Fig. 4 has different ranges and change frequencies from Fig. 5 for generating the RSV model. This shows that the RSV model can deal with extrapol-

lated flow conditions, once the coherent flow structures are determined.

While the approach is very effective, MPC with the RSV model could still be improved by speeding up the SV model calculation. This will dramatically decrease the MPC calculation time, since 97.7% of the control process time is spent on calling the prediction model, although the absolute time consumption is small. This may

$$A_{r,n} = \begin{bmatrix} I \\ A_r(k) \\ A_r(k+1) \cdot A_r(k) \\ \vdots \\ A_r(k+n-1) \cdot A_r(k+n-2) \cdots A_r(k) \end{bmatrix}; \quad C_{r,n} = \begin{bmatrix} C_r & 0 & 0 & 0 \\ 0 & C_r & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & C_r \end{bmatrix}$$

$$Bu_{r,n} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ Bu_r(k) & 0 & \cdots & 0 \\ A_r(k+1) \cdot Bu_r(k) & Bu_r(k+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ A_r(k+n-1) \cdots A_r(k+1) \cdot Bu_r(k) & A_r(k+n-2) \cdots A_r(k+1) \cdot Bu_r(k+1) & \cdots & Bu_r(k+n-1) \end{bmatrix}$$

$$Bd_{r,n} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ Bd_r(k) & 0 & \cdots & 0 \\ A_r(k+1) \cdot Bd_r(k) & Bd_r(k+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ A_r(k+n-1) \cdots A_r(k+1) \cdot Bd_r(k) & A_r(k+n-2) \cdots A_r(k+1) \cdot Bd_r(k+1) & \cdots & Bd_r(k+n-1) \end{bmatrix}$$

be done by optimizing the computer code or changing to a lower level (faster) programming language.

Appendix A. Time variant state-space model over prediction horizon

Note: the subscript ‘r’ represents that the variables are in reduced-order domain. When deleting the subscript ‘r’, they become the original variables in the unreduced model.

The reduced-order time variant state-space model is described as follows:

$$\begin{aligned} \dot{x}_r(k+1) &= A_r(k) \cdot \dot{x}_r(k) + Bu_r(k) \cdot u(k) + Bd_r(k) \cdot d_r(k) \\ \hat{y}_r(k) &= C_r \cdot \hat{x}_r(k) \end{aligned}$$

When the model is written over the prediction horizon, the overall state-space model formulation becomes, following van Overloop [14]:

$$\begin{bmatrix} \dot{x}_r(k) \\ \dot{x}_r(k+1) \\ \vdots \\ \dot{x}_r(k+n) \end{bmatrix} = A_{r,n} \cdot \hat{x}_r(k) + Bu_{r,n} \cdot \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+n-1) \end{bmatrix} + Bd_{r,n} \cdot \begin{bmatrix} d_r(k) \\ d_r(k+1) \\ \vdots \\ d_r(k+n-1) \end{bmatrix}$$

$$\begin{bmatrix} \hat{y}_r(k) \\ \hat{y}_r(k+1) \\ \vdots \\ \hat{y}_r(k+n) \end{bmatrix} = C_{r,n} \cdot \begin{bmatrix} \hat{x}_r(k) \\ \hat{x}_r(k+1) \\ \vdots \\ \hat{x}_r(k+n) \end{bmatrix}$$

The prediction matrices of A_r , Bu_r , Bd_r and C_r are also time-varying and given as:

References

- [1] Clemmens AJ, Schuurmans J. Simple optimal downstream feedback canal controllers: theory. *J Irrig Drain Eng* 2004;130(1):26–34.
- [2] Litrico X, Fromion V. Design of structured multivariable controllers for irrigation canals. In: 44th IEEE conference on decision and control, the European control conference, Seville, Spain; 2005. p. 1881–6.
- [3] Wahlin BT, Clemmens AJ. Automatic downstream water-level feedback control of branching canal networks: theory. *J Irrig Drain Eng* 2006;132:198.
- [4] Schuurmans J, Bosgra OH, Brouwer R. Open-channel flow model approximation for controller design. *Appl Math Model* 1995;19(9):525–30.
- [5] Xu M, Van Overloop PJ. On the application of model reduction in model predictive control of open channel flow. In: 9th International conference on hydroinformatics, HIC 2010, Tianjin, China; 2010.
- [6] Xu M, Van Overloop PJ, Van de Giesen NC, Stelling GS. Real-time control of combined surface water quantity and quality: polder flushing. *Water Sci Technol* 2010;61(4):869.
- [7] Simon D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, Inc; 2006.
- [8] MATLAB User’s Guide. The MathWorks, Inc., Natick, MA; 1998.
- [9] Stelling GS, Duinmeijer SPA. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *Int J Numer Meth Fluids* 2003;43:1329–54.
- [10] Ravindran SS. Proper orthogonal decomposition in optimal control of fluids. *Int J Numer Meth Fluids* 2000;34(425–448):2.
- [11] Ha DM, Lim KM, Khoo BC, Wilcox K. Real-time optimization using proper orthogonal decomposition: free surface shape prediction due to underwater bubble dynamics. *Comput Fluids* 2007;36(3):499–512.
- [12] Ha DM, Tkalich P, Chan ES. Tsunami forecasting using proper orthogonal decomposition method. *J Geophys Res-Oceans* 2008;113(C6):C06019.
- [13] Sirovich L. Turbulence and the dynamics of coherent structures I: Coherent structures. II: Symmetries and transformations. III: Dynamics and scaling. *Q Appl Math* 1987;45:561–90.
- [14] Van Overloop PJ. Model predictive control on open water systems. IOS Press; 2006.
- [15] Clemmens AJ, Kacerek TF, Grawitz B, Schuurmans W. Test cases for canal control algorithms. *J Irrig Drain Eng* 1998;124(1):23–30.