Delft University of Technology

Delft Center for Systems and Control

Technical report 08-018

A distributed model predictive control approach for the control of irrigation canals*

R.R. Negenborn and B. De Schutter

If you want to cite this report, please use the following reference instead:
R.R. Negenborn and B. De Schutter, "A distributed model predictive control approach for the control of irrigation canals," *Proceedings of the International Conference on Infrastructure Systems 2008: Building Networks for a Brighter Future*, Rotterdam, The Netherlands, 6 pp., Nov. 2008. Paper 152.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

A Distributed Model Predictive Control Approach for the Control of Irrigation Canals

Rudy R. Negenborn and Bart De Schutter

***Abstract*— Water networks are large-scale systems, consisting of many interacting components. They are currently typically operated by local decentralized controllers which receive set-points from human operators. We discuss how communication among the local controllers can be included and in particular propose the use of distributed model predictive control for enabling the local controllers to determine set-points autonomously using communication and coordination. We consider the control of a particular class of water networks, viz. irrigation canals. A simulation study on a 7-reach irrigation canal illustrates the potential of the proposed approach.**

I. Introduction

A. Water networks

In the near future the importance of an efficient and reliable flood and water management system will keep on increasing, among others due to the effects of global warming (higher sea levels, more heavy rain during the spring season, but possibly also drier summers). Due to the large scale of water networks, control of such networks in general cannot be done in a centralized way, in which from a single location measurements from the whole system are collected and actions for the whole system are determined. Instead, control is typically decentralized over several local control bodies, each controlling a particular part of the network [1], [2]. Local control actions include activation of pumps or locks, filling or draining of water reservoirs, or controlled flooding of water meadows or of emergency water storage areas.

To each of the actions that can be taken in a water system a certain cost is associated, and the same holds in case of too high water levels (which may result in floods) or too low groundwater or surface water levels (which have a negative impact on agriculture, irrigation, and drinking water supplies). Although the local water management bodies usually only control or manage the water levels in a relatively small region, the evolution of the water levels is influenced by what happens over a much larger region, often extending far beyond the neighborhood of the given region (e.g., due to water arriving via rivers or via subsurface diffusion flows).

To improve the operation of water systems the controllers of different parts of the water network should cooperate and coordinate their local water management actions, and take into account predictions or forecasts of future rain fall, future droughts, future arrival of increased water flow via rivers, etc. (using various weather and hydrological sensors, and prediction models). In this paper we propose the use of distributed model predictive control to obtain more efficient flood and water management with less risks and less costs.

R.R. Negenborn and B. De Schutter are with the Delft Center for Systems and Control of Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, e-mail: `r.r.negenborn@tudelft.nl`, `b.deschutter@dcsc.tudelft.nl`. B. De Schutter is also with the Marine and Transport Technology department of Delft University of Technology.

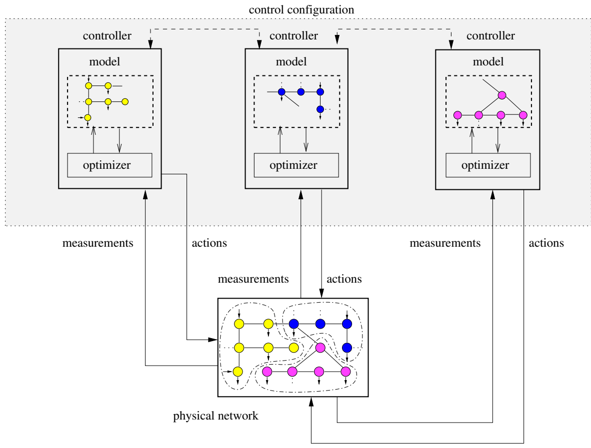


Fig. 1. Illustration of distributed MPC control of a network.

B. Distributed model predictive control

To determine the actions that meet the control objectives of the local control bodies as well as possible, the local controllers have to make a trade-off among the various actions available. An in particular promising form of control for this seems to be model predictive control (MPC) [3]. Over the last decades MPC (also known as receding horizon control or moving horizon control) has become an important strategy for finding control policies for complex, dynamic systems. MPC for centralized control has shown successful application in the process industry [3], and is now gaining increasing attention in fields like power networks [4], road traffic networks [5], and steam networks [6].

In a distributed MPC control configuration, there are multiple controllers, each of them using MPC to control its own subnetwork, i.e., its own part of the overall network, as illustrated in Fig. 1. The challenge in implementing a distributed MPC strategy comes from ensuring that the actions that the individual controllers choose result in a joint performance that ideally is as good as when a hypothetical centralized control configuration in which all information is available at a central location would be used.

Various distributed MPC control schemes have been investigated since the 90s, e.g., in [7], [8], [9]. In this paper we apply a particular distributed MPC scheme, recently proposed in [10], for improving the operation of a particular type of water systems, viz. irrigation canals.

C. Outline

The remainder of this paper is organized as follows. In Section II we discuss a particular distributed MPC scheme. In Section III we discuss the dynamics of a particular water system, viz. an irrigation canal, and set up the distributed MPC control scheme for control of this system. In Section IV we illustrate the potential of the proposed approach through simulation studies. Section V concludes the paper and contains directions for future research.

II. Distributed model predictive control

In distributed MPC control each individual controller is responsible for a particular part of the network. The individual controllers on the one hand obtain measurements from and determine actions for their part of the network, and on the other hand communicate with other controllers in order to obtain coordination and to improve predictions. To actually determine which actions to take each controller uses MPC.

A. General ingredients and structure of distributed MPC

At each control cycle each controller uses the following information:

- a *prediction model* describing the behavior of its subnetwork;
- an *objective function* expressing which subnetwork behavior and actions are desired;
- possibly *constraints* on the local states, the local inputs, and the local outputs;
- possibly known information about future disturbances and exogenous inputs;
- a *measurement* of the state of the subnetwork at the beginning of the current control cycle.

The objective of each controller is to determine those actions that optimize the behavior of the overall network and to minimize costs as specified through the objective function. In order to find the actions that lead to the best performance, each controller uses its prediction model to predict the behavior of its subnetwork under various actions over a certain prediction horizon, starting from the state at the beginning of the control cycle. Once the controller has determined the actions that optimize the performance of its subnetwork over the prediction horizon, it implements these actions until the beginning of the next control cycle, at which point the controller determines new actions over the prediction horizon starting at that point, using updated information. Hence, each controller operates in a receding or rolling horizon fashion to determine its actions.

To make accurate predictions of the evolution of a subnetwork, each controller requires the current state of its subnetwork, a sequence of actions over the prediction horizon, and predictions of the values of variables that interconnect the model of its subnetwork with the model of other subnetworks. The predictions of the values of these so-called *interconnecting variables* are based on the information communicated with the neighboring controllers. One particular class of methods aims at achieving cooperation among controllers in an iterative way in which in each control cycle controllers perform several iterations consisting of local problem solving and communication. In each iteration controllers then obtain information about the plans of neighboring controllers. Ideally, at the end of the iterations the controllers choose overall optimal actions.

In [10] we have proposed a distributed MPC scheme for control of general transportation networks. Water networks are a particular type of transportation networks, and therefore this scheme is also suitable for distributed control of water networks. The actions that the controllers determine using the scheme lead over the iterations to overall optimal performance if certain assumptions on the dynamics, the information available to controllers, and the control objectives are made. Below we briefly outline these assumptions and the steps of the scheme.

B. Dynamics

Let the network be divided into n subnetworks. Assume that the dynamics of subnetwork $i \in \{1, \dots, n\}$ are given by a deterministic linear discrete-time time-invariant model (possibly obtained after symbolic or numerical linearization of a nonlinear model in combination with discretization):

$$\mathbf{x}_i(k+1) = \mathbf{A}_i \mathbf{x}_i(k) + \mathbf{B}_{1,i}\mathbf{u}_i(k) + \mathbf{B}_{2,i}\mathbf{d}_i(k) + \mathbf{B}_{3,i}\mathbf{v}_i(k) \quad (1)$$

$$\mathbf{y}_i(k) = \mathbf{C}_i \mathbf{x}_i(k) + \mathbf{D}_{1,i}\mathbf{u}_i(k) + \mathbf{D}_{2,i}\mathbf{d}_i(k) + \mathbf{D}_{3,i}\mathbf{v}_i(k), \quad (2)$$

where at control cycle k, for subnetwork i, $\mathbf{x}_i(k) \in \mathbb{R}^{n_{\mathbf{x}_i}}$ are the local states, $\mathbf{u}_i(k) \in \mathbb{R}^{n_{\mathbf{u}_i}}$ are the local inputs, $\mathbf{d}_i(k) \in \mathbb{R}^{n_{\mathbf{d}_i}}$ are the local known exogenous inputs, $\mathbf{y}_i(k) \in \mathbb{R}^{n_{\mathbf{y}_i}}$ are the local outputs, $\mathbf{v}_i(k) \in \mathbb{R}^{n_{\mathbf{v}_i}}$ are the remaining variables influencing the local dynamical states and outputs, and $\mathbf{A}_i \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{x}_i}}$, $\mathbf{B}_{1,i} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{u}_i}}$, $\mathbf{B}_{2,i} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{d}_i}}$, $\mathbf{B}_{3,i} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{v}_i}}$, $\mathbf{C}_i \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{x}_i}}$, $\mathbf{D}_{1,i} \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{u}_i}}$, $\mathbf{D}_{2,i} \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{d}_i}}$, $\mathbf{D}_{3,i} \in \mathbb{R}^{n_{\mathbf{y}_i} \times n_{\mathbf{v}_i}}$ determine how the different variables influence the local states and outputs of subnetwork i. The $\mathbf{v}_i(k)$ variables appear due to the fact that a subnetwork is connected to other subnetworks. Hence, the $\mathbf{v}_i(k)$ variables represent the influence of other subnetworks on subnetwork i. If the values of $\mathbf{v}_i(k)$ would be constant, then the dynamics of subnetwork i would be decoupled from the other subnetworks. In practice, however, the variables $\mathbf{v}_i(k)$ are equal to some of the variables of models representing dynamics of neighboring subnetworks. Below we refer to the interconnecting input variables $\mathbf{w}_{\text{in},ji}(k) \in \mathbb{R}^{n_{\mathbf{w}_{\text{in},ji}}}$ as these variables of subnetwork i that are influenced by subnetwork j, i.e., a selection of $\mathbf{v}_i(k)$, and we refer to the interconnecting output variables $\mathbf{w}_{\text{out},ji}(k) \in \mathbb{R}^{n_{\mathbf{w}_{\text{out},ji}}}$ as these variables of subnetwork i that influence a neighboring subnetwork j, i.e., a selection of $\mathbf{x}_i(k)$, $\mathbf{u}_i(k)$, and $\mathbf{y}_i(k)$. Fig. 2 illustrates the relations between the variables of the models of two subnetworks.

C. Available information

Assume that each of the subnetworks $i \in \{1, \dots, n\}$ is controlled by a controller i that:

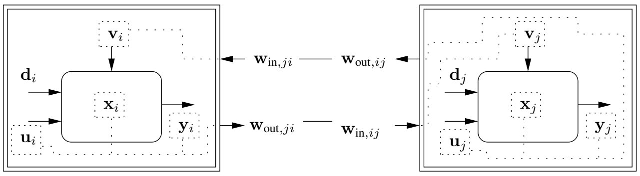


Fig. 2. Illustration of the relation between the models and variables of two subnetworks i and j.

- has a prediction model of the form (1)–(2) of the dynamics of subnetwork i;
- can measure or estimate the state $\mathbf{x}_i(k)$ of its subnetwork;
- can determine settings $\mathbf{u}_i(k)$ for the actuators of its subnetwork;
- can estimate exogenous inputs $\mathbf{d}_i(k+l)$ of its subnetwork over a certain horizon of length N, for $l = \{0, \dots, N-1\}$;
- can communicate with neighboring controllers, i.e., the controllers controlling the subnetworks $j \in \mathcal{N}_i$, where $\mathcal{N}_i = \{j_{i,1}, \dots, j_{i,m_i}\}$ is the set of indexes of the m_i subnetworks connected to subnetwork i, also referred to as the *neighbors* of subnetwork or controller i.

D. Control objectives

We assume that the controllers are cooperative, meaning that the individual controllers strive for the best overall network performance. In addition, we assume that the objectives of the controllers can be represented by convex functions $J_{\text{local},i}$, for $i \in \{1, \dots, n\}$, which are typically linear or quadratic. Such functions are commonly encountered, in particular for systems that can be represented by (1)–(2), as illustrated in Section III-E.

E. Distributed MPC scheme

The distributed MPC scheme that we employ comprises at control cycle k the following steps:

1) For $i = 1, \dots, n$, controller i makes a measurement of the current state of the subnetwork $\mathbf{x}_i(k)$ and estimates the expected exogenous inputs $\mathbf{d}_i(k+l)$, for $l = 0, \dots, N-1$.
2) The controllers cooperatively solve their control problems in the following iterative way[1]:
   a) Set the iteration counter s to 1 and initialize the Lagrange multipliers $\tilde{\boldsymbol{\lambda}}^{(s)}_{\text{in},ji}(k)$, $\tilde{\boldsymbol{\lambda}}^{(s)}_{\text{out},ij}(k)$ arbitrarily[2].
   b) For $i = 1, \dots, n$, one controller i after another determines $\tilde{\mathbf{x}}_i^{(s)}(k+1)$, $\tilde{\mathbf{u}}_i^{(s)}(k)$, $\tilde{\mathbf{w}}^{(s)}_{\text{in},ji}(k)$, $\tilde{\mathbf{w}}^{(s)}_{\text{out},ji}(k)$ as solution of the following optimization problem:
$$\min_{\substack{\tilde{\mathbf{x}}_i(k+1), \tilde{\mathbf{u}}_i(k), \tilde{\mathbf{y}}_i(k), \\ \tilde{\mathbf{w}}_{\text{in},j_{i,1}i}(k), \dots, \tilde{\mathbf{w}}_{\text{in},j_{i,m_i}i}(k), \\ \tilde{\mathbf{w}}_{\text{out},j_{i,1}i}(k), \dots, \tilde{\mathbf{w}}_{\text{out},j_{i,m_i}i}(k)}} J_{\text{local},i}\left(\tilde{\mathbf{x}}_i(k+1), \tilde{\mathbf{u}}_i(k), \tilde{\mathbf{y}}_i(k)\right) + \sum_{j \in \mathcal{N}_i} J^{(s)}_{\text{inter},i}\left(\tilde{\mathbf{w}}_{\text{in},ji}(k), \tilde{\mathbf{w}}_{\text{out},ji}(k)\right), \tag{3}$$
   subject to the local dynamics (1)–(2) of subnetwork i over the horizon, the measurement of the current state $\mathbf{x}_i(k)$, the known exogenous inputs $\tilde{\mathbf{d}}_i(k)$. In this, the additional objective function $J_{\text{inter},i}$ is at iteration s defined as
$$J^{(s)}_{\text{inter},i}\left(\tilde{\mathbf{w}}_{\text{in},ji}(k), \tilde{\mathbf{w}}_{\text{out},ji}(k)\right) = \begin{bmatrix} \tilde{\boldsymbol{\lambda}}^{(s)}_{\text{in},ji}(k) \\ -\tilde{\boldsymbol{\lambda}}^{(s)}_{\text{out},ij}(k) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in},ji}(k) \\ \tilde{\mathbf{w}}_{\text{out},ji}(k) \end{bmatrix} + \frac{\gamma_{\text{c}}}{2} \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{\text{in,prev},ij}(k) - \tilde{\mathbf{w}}_{\text{out},ji}(k) \\ \tilde{\mathbf{w}}_{\text{out,prev},ij}(k) - \tilde{\mathbf{w}}_{\text{in},ji}(k) \end{bmatrix} \right\|_2^2,$$
   where $\tilde{\mathbf{w}}_{\text{in,prev},ij}(k) = \tilde{\mathbf{w}}^{(s)}_{\text{in},ij}(k)$ and $\tilde{\mathbf{w}}_{\text{out,prev},ij}(k) = \tilde{\mathbf{w}}^{(s)}_{\text{out},ij}(k)$ is the information computed at the current iteration s for each controller $j \in \mathcal{N}_i$ that has solved its problem *before* controller i in the *current* iteration s, and where $\tilde{\mathbf{w}}_{\text{in,prev},ij}(k) = \tilde{\mathbf{w}}^{(s-1)}_{\text{in},ij}(k)$ and $\tilde{\mathbf{w}}_{\text{out,prev},ij}(k) = \tilde{\mathbf{w}}^{(s-1)}_{\text{out},ij}(k)$ is the information computed at the *last* iteration $s-1$ for the other controllers. Furthermore, γ_{c} is a positive scalar that penalizes the deviation from the interconnecting variable iterates that were computed by the controllers before controllers i in the current iteration and by the other controllers during the last iteration. The results $\tilde{\mathbf{w}}^{(s)}_{\text{out},ji}(k)$ of the optimization are sent to controller j.
   c) Update the Lagrange multipliers,
$$\tilde{\boldsymbol{\lambda}}^{(s+1)}_{\text{in},ji}(k) = \tilde{\boldsymbol{\lambda}}^{(s)}_{\text{in},ji}(k) + \gamma_{\text{c}}\left(\tilde{\mathbf{w}}^{(s)}_{\text{in},ji}(k) - \tilde{\mathbf{w}}^{(s)}_{\text{out},ij}(k)\right). \tag{4}$$
   d) Move on to the next iteration $s+1$ and repeat steps 2b–2c. The iterations stop when the following stopping condition is satisfied:
$$\left\| \begin{bmatrix} \tilde{\boldsymbol{\lambda}}^{(s+1)}_{\text{in,err},j_{1,1}1}(k) \\ \vdots \\ \tilde{\boldsymbol{\lambda}}^{(s+1)}_{\text{in,err},j_{n,m_n}n}(k) \end{bmatrix} \right\|_\infty \leq \gamma_{\epsilon,\text{term}}, \tag{5}$$
   where $\tilde{\boldsymbol{\lambda}}^{(s+1)}_{\text{in,err},ji}(k) = \tilde{\boldsymbol{\lambda}}^{(s+1)}_{\text{in},ji}(k) - \tilde{\boldsymbol{\lambda}}^{(s)}_{\text{in},ji}(k)$, and $\gamma_{\epsilon,\text{term}}$ is a small positive scalar and $\|\cdot\|_\infty$ denotes the infinity norm. Note that satisfaction of this stopping condition can be determined in a distributed way, since each individual component of the infinity norm depends only on variables of one particular controller.
3) The controllers implement the actions until the beginning of the next control cycle.

[1] The tilde notation is used to represent variables over the prediction horizon. E.g., $\tilde{\mathbf{u}}_i(k) = [\mathbf{u}_i(k)^{\mathrm{T}}, \dots, \mathbf{u}_i(k+N-1)^{\mathrm{T}}]^{\mathrm{T}}$.

[2] The Lagrange multipliers can in principle be initialized arbitrarily. However, initializing them with values close to the optimal Lagrange multipliers will increase the convergence of the decision making process. Therefore, also initializing the Lagrange multipliers with values obtained from the previous control cycle is beneficial, since typically these Lagrange multipliers will be good initial guesses for the new solution. This is referred to as *warm start*.

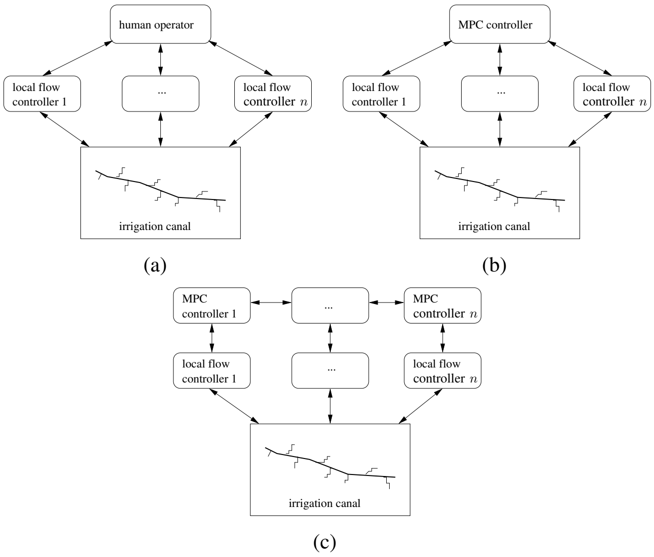


Fig. 3. Three control configurations for the control of irrigation canals.

4) The computations of the next control cycle are started.

Under the assumptions that we have made on the objective functions and prediction models the solution of this scheme converges to the solution that a centralized MPC controller would have obtained, see [10].

III. CONTROL OF AN IRRIGATION CANAL

In this section we describe the dynamics and control of a particular water network, viz. an irrigation canal. Irrigation canals are used mostly to transport water from source nodes, such as lakes, large rivers, etc., to sink nodes, such as small rivers and pipes transporting water to agricultural fields of farmers. Irrigation canals consist of several connected canal reaches, the inflow or outflow of which can be controlled using structures such as so-called overshot or undershot gates, which restrict the flow of water flowing from an upstream canal reach into a downstream canal reach [11]. These structures usually have a local flow controller that regulates the position of the gates in order to obtain a certain flow. We focus on determining the set-points for the local flow controllers at these structures.

A. Control configurations

Fig. 3 illustrates three possible control configurations for irrigation canals. Currently the configuration of Fig. 3(a) is typically used in practice. A human operator manually adjusts the set-points for the local flow controllers at the undershot and overshot gates. This manual process is expensive, since the human operator has to travel from one control structure to the next, possibly several times per day [12]. A more advanced control configuration is depicted in Fig. 3(b). In this case, the determination of the set-points for the local flow controllers has been centralized and automated. Although implementation of such a centralized control configuration may be feasible in practice for relatively small water networks, the increasing computational requirements and required bandwidth prevent application to larger networks. A centralized control configuration is not well scalable and moreover constitutes a single point of failure. In addition, in practice, management of irrigation canals may already be distributed over several control authorities, preventing a centralized control configuration from being implemented. Instead of a centralized control configuration, the control configuration in Fig. 3(c) may be employed, i.e., a distributed control configuration may be installed, in which set-points are autonomously decided upon by the distributed controllers based on local communication and cooperation.

B. Benchmark system

The irrigation canal that we consider is based on the W-M canal, which is a physically existing irrigation canal in the South of Phoenix, Arizona. The canal is used to provide water to farmers. The length of the canal is almost 10 km and the maximum capacity of the head gate is $2.8\,\mathrm{m}^3/\mathrm{s}$ [12]. The irrigation canal that we consider consists of 7 canal reaches. At each of the reaches water can be taken out at offtakes for irrigation purposes. Between each of the reaches control structures are present in the form of undershot gates to control the water flow locally. These control structures are equipped with local flow controllers that adjust the height of the undershot gate in order to meet a set-point for the water flow.

In [12] an MPC scheme is proposed that is used by a single controller to determine in a centralized way the set-points for the local flow controllers, cf. Fig. 3(b). Here we propose to use the distributed MPC scheme of Section II to take over this task. Using a distributed approach there is no need for a central control location in the network. Using a distributed approach only local information available to a local controller and information from neighboring local controllers is used.

C. Dynamics of irrigation canals

The dynamics of irrigation canals can be modeled in detail, e.g., using the Saint Venant equations [11] resulting in systems of highly-nonlinear partial differential-algebraic equations. However, using such highly-detailed models for predictive control results in significant requirements on computational power. Therefore, similarly as in [12], we employ the integral delay model [13] to model the dynamics of a canal reach. This model has shown to adequately capture relevant dynamics [13], and reduces computations required for simulation of the dynamics significantly.

The integrator delay model is a discrete-time model, which models how the water level at particular places in the canal changes over time. Let time be discretized into control cycles $k \in \mathbb{N}^+$ (where $\mathbb{N}^+$ are the positive natural numbers) and let the continuous time between two control cycles k and $k+1$ correspond to $T_\mathrm{c} \in \mathbb{R}^+$ (s) (where $\mathbb{R}^+$ are the positive real numbers). Each canal reach is considered to have an inflow from an upstream canal reach. Let this inflow into reach i be given by $q_{\mathrm{in},i}(k) \in \mathbb{R}^+$ (m^3/s). A canal reach has an outflow to a downstream canal reach. Let $q_{\mathrm{out},i}(k) \in \mathbb{R}^+$ (m^3/s) denote this outflow. In addition to this inflow and outflow due to upstream and downstream canal reaches there can be additional local inflow (e.g., due to rainfall) and outflow (e.g., due to outflow caused by farmers). Let such inflow be represented by $q_{\text{ext,in},i}(k) \in \mathbb{R}^+$ (m^3/s) and such outflow by $q_{\text{ext,out},i}(k) \in \mathbb{R}^+$ (m^3/s). The inflow $q_{\text{ext,in},i}(k)$ and outflow $q_{\text{ext,out},i}(k)$ are assumed to be static and known or predicted accurately in advance.

Depending on how the inflows and outflows change over time, the levels of the water in reaches will change. Instead of considering the levels of the water at each location in the reaches, we only consider the levels of the water at the downstream end of each reach. In addition to the amount of inflow and outflow, also the surface of the reach influences how much the level of the water will change. Let $h_i(k) \in \mathbb{R}^+$ (m) denote the level of the water in canal reach i, and let the surface of reach i be $c_i \in \mathbb{R}^+$ (m^2). It takes some time for a change in the inflow of reach i to result in a change of the water level at the downstream end of the reach. Let this delay be $k_{\text{d},i} \in \mathbb{N}^+$ control cycles for reach i.

Using the variables defined above, the model describing how the level of the water in the canal reach changes from one control cycle k to the next control cycle $k+1$ is given by:

$$\begin{aligned} h_i(k+1) = h_i(k) &+ \frac{T_\text{c}}{c_i} q_{\text{in},i}(k - k_{\text{d},i}) - \frac{T_\text{c}}{c_i} q_{\text{out},i}(k) \\ &+ \frac{T_\text{c}}{c_i} q_{\text{ext,in},i}(k) - \frac{T_\text{c}}{c_i} q_{\text{ext,out},i}(k). \end{aligned} \quad (6)$$

Canal reaches are connected to one another. When two canal reaches are connected to each other, the inflow of one canal reach is equal to the outflow of the other. Hence, for neighboring reaches i and j this interconnection is given by

$$q_{\text{out},i}(k) = q_{\text{in},j}(k). \quad (7)$$

In the state-space form (1)–(2) the dynamics of canal reach i are conveniently written down by defining

$$\mathbf{x}_i(k) = \begin{bmatrix} h_i(k) \\ q_{\text{in},i}(k - k_{\text{d},i}) \\ \vdots \\ q_{\text{in},i}(k-1) \end{bmatrix} \qquad \mathbf{d}_i(k) = \begin{bmatrix} q_{\text{ext,in},i}(k) \\ q_{\text{ext,out},i}(k) \end{bmatrix}$$

$$\mathbf{u}_i(k) = q_{\text{in},i}(k) \qquad \mathbf{v}_i(k) = q_{\text{out},i}(k) \qquad \mathbf{y}_i(k) = \mathbf{x}_i(k)$$

and

$$\mathbf{A}_i = \begin{bmatrix} 1 & \frac{T_\text{c}}{c_i} & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & \dots & \dots & \dots & 0 \end{bmatrix}$$

$$\mathbf{B}_{1,i} = \begin{bmatrix} -\frac{T_\text{c}}{c_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{B}_{2,i} = \begin{bmatrix} \frac{T_\text{c}}{c_i} & -\frac{T_\text{c}}{c_i} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \qquad \mathbf{B}_{3,i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C}_i = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$\mathbf{D}_{1,i} = 0 \qquad \mathbf{D}_{2,i} = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad \mathbf{D}_{3,i} = 0,$$

and

$$\mathbf{w}_{\text{in},j_\text{down} i}(k) = q_{\text{out},i}(k) \qquad \mathbf{w}_{\text{out},j_\text{up} i}(k) = q_{\text{in},i}(k),$$

where j_up and j_down are the index of the upstream and downstream canal reach, respectively.

D. Available information

There are n controllers, and each controller i is responsible for canal reach i. Controller i can measure the water level in its canal reach, can adjust the set-point for the flow controller at its upstream gate, and can communicate with the controllers of the canal reaches immediately upstream and downstream of the canal reach. In addition, controller i can obtain the expected water offtakes and rainfall with respect to its canal reach.

The actions that are optimal for each of the controllers depend on one another, since if one controller decides to increase its inflow, the water level in the upstream reach will decrease and therefore influences the decision making process of the upstream controller.

E. Control objectives

The set-points determined by the controllers and provided to the local flow controllers of the undershot gates should be chosen in such a way that

1) the deviations of water levels h_i from provided set-points $h_{\text{ref},i} \in \mathbb{R}^+$, for $i \in \{1, \dots, n\}$ are minimized;
2) the changes in the water levels h_i from one control cycle to the next are minimized to encourage smooth water level changes;
3) the changes in the set-points u_i provided to the local flow controllers are minimized to reduce equipment wear.

After defining the deviation in the water level $h_{\text{dev},i}(k) \in \mathbb{R}$ as $h_{\text{dev},i}(k) = h_i(k) - h_{\text{ref},i}$, the objective function $J_{\text{local},i}$ can be written as

$$\begin{aligned} J_{\text{local},i}(\cdot) = &\sum_{l=0}^{N-1} p_{h,i} \left(h_{\text{dev},i}(k+1+l) \right)^2 \\ &+ \sum_{l=0}^{N-1} p_{\Delta h,i} \left(h_{\text{dev},i}(k+1+l) - h_{\text{dev},i}(k+l) \right)^2 \\ &+ \sum_{l=0}^{N-1} p_{u,i} \left(u_i(k+l) - u_i(k-1+l) \right)^2 \end{aligned}$$

where for controller i, $p_{h,i} \in \mathbb{R}^+$ is the quadratic cost on the water level deviation, $p_{\Delta h,i} \in \mathbb{R}^+$ is the quadratic cost on a change in the water level deviation, and $p_{u,i} \in \mathbb{R}^+$ is the quadratic cost on a change in the set-point provided to the local flow controller.

IV. Simulation

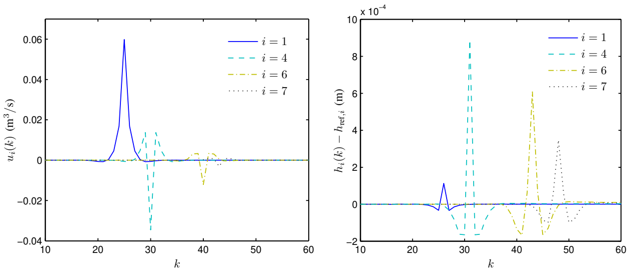
In this section we describe a simulation result to illustrate the performance of the controllers discussed in this paper. We have implemented the model of the benchmark irrigation canal consisting of 7 canal reaches in Matlab v7.3[3]. For

[3]See http://www.mathworks.com/.

TABLE I

VALUES OF THE PARAMETERS OF THE MODEL, TAKEN FROM [12].

i	1	2	3	4	5	6	7
$k_{\mathrm{d},i}$ (steps)	1	3	1	1	9	3	5
c_i (m^2)	397	653	503	1530	1614	2000	1241



(a) (b)

Fig. 4. Evolution for four representative canal reaches of (a) set-points and (b) deviation of the water levels from reference values.

solving the optimization problems at each control sample we use the ILOG CPLEX v10.0 quadratic programming solver through the Tomlab v5.7 interface [14] to Matlab.

We compare the performance of the distributed MPC scheme with a hypothetical centralized scheme, i.e., we compare the performance of a control configuration of Fig. 3(c) with a corresponding control configuration of Fig. 3(b).

A. Scenario

The parameters used for the model of the irrigation canal are shown in Table I. The time T_{c} between two consecutive control cycles is 240 s. The controllers use as parameters $N = 31$, $\gamma_{\mathrm{c}} = 1$, $\gamma_{\epsilon,\mathrm{term}} = 1.10^{-4}$. A prediction horizon length of 31 is chosen to take into account the total delay present in the irrigation canal [12]. The cost coefficients that the controllers use are chosen as $p_{h,i} = 10$, $p_{\Delta h,i} = 1$, $p_{u,i} = 0.01$, for $i \in \{1, \ldots, n\}$.

As scenario we consider a sudden increase in the water offtake of canal reach 3 at $k = 30$ of $0.1\,\mathrm{m}^3/\mathrm{s}$.

B. Results

Fig. 4(a) shows the changes in the set-points decided upon by the controllers. Fig. 4(b) shows the closed-loop evolution of the deviations of the water levels from the reference values. It can be seen that the inflow of canal reach 1 is increased right before the additional offtake increase takes place to prevent having a too low water level after the additional offtake. It can also be observed that the deviations of the water levels after the offtake increase are minimal due to the changes in the set-points. We observe that after about 25 control cycles the set-points settle at a constant value, while the deviations of the water levels from the references are minimal, and that thus the controllers have performed their tasks adequately.

The costs computed over the full simulation using the distributed MPC scheme are 1832.10^{-7}. A centralized MPC controller based on the same objectives obtains costs over the full simulation of 1831.10^{-7}. This difference in performance is negligible, and hence, in this case in which the assumptions made are valid, indeed, the distributed controllers have achieved a performance comparable to the performance obtained by a centralized MPC controller.

V. CONCLUSION AND FUTURE RESEARCH

In this paper we have considered model predictive control (MPC) for distributed control of water networks. In particular, we have discussed the use of a serial, iteration-based, distributed MPC scheme for the control of irrigation canals. We have illustrated the potential of the approach in a simulation study on a 7-reach irrigation canal.

Future work consists of further assessing the performance of the proposed scheme, extending the system model to include constraints on the minimal and maximal flow possible through undershot and overshot gates, assessing the performance of the distributed MPC scheme using linear prediction models on a nonlinear simulation model of the canal, and, based on this assessment, further improving the system model if necessary.

ACKNOWLEDGMENTS

Research supported by the BSIK project "Next Generation Infrastructures (NGI)", the Delft Research Center Next Generation Infrastructures, the European 6th Framework Network of Excellence "HYbrid CONtrol: Taming Heterogeneity and Complexity of Networked Embedded Systems (HYCON)", the European STREP project "Hierarchical and distributed model predictive control (HD-MPC)", and the project "Multi-Agent Control of Large-Scale Hybrid Systems" (DWV.6188) of the Dutch Technology Foundation STW.

REFERENCES

[1] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, Massachusetts: MIT Press, 2000.

[2] D. D. Šiljak, *Decentralized Control of Complex Systems*. Boston, Massachusetts: Academic Press, 1991.

[3] J. M. Maciejowski, *Predictive Control with Constraints*. Harlow, UK: Prentice-Hall, 2002.

[4] T. Geyer, M. Larsson, and M. Morari, "Hybrid emergency voltage control in power systems," in *Proceedings of the European Control Conference 2003*, Cambridge, UK, Sept. 2003.

[5] A. Hegyi, B. De Schutter, and J. Hellendoorn, "Optimal coordination of variable speed limits to suppress shock waves," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 102–112, Mar. 2005.

[6] Y. Majanne, "Model predictive pressure control of steam networks," *Control Engineering Practice*, vol. 13, no. 12, pp. 1499–1505, Dec. 2005.

[7] L. Acar, "Some examples for the decentralized receding horizon control," in *Proceedings of the 31st IEEE Conference on Decision and Control*, Tucson, Arizona, 1992, pp. 1356–1359.

[8] M. R. Katebi and M. A. Johnson, "Predictive control design for large-scale systems," *Automatica*, vol. 33, no. 3, pp. 421–425, 1997.

[9] D. Jia and B. H. Krogh, "Distributed model predictive control," in *Proceedings of the 2001 American Control Conference*, Arlington, Virginia, June 2001, pp. 2767–2772.

[10] R. R. Negenborn, B. De Schutter, and J. Hellendoorn, "Multi-agent model predictive control for transportation networks: Serial versus parallel schemes," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, Apr. 2008.

[11] V. T. Chow, *Open-Channel Hydraulics*. New York, New York: McGraw-Hill Book, 1959.

[12] P.-J. van Overloop, "Model predictive control on open water systems," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands, June 2006.

[13] J. Schuurmans and M. Ellerbeck, "Linear approximation model of the WM canal for controller design," in *Proceedings of the First International Conference on Water Resources Engineering*, San Antonio, Texas, June 1995, pp. 353–357.

[14] K. Holmström, A. O. Göran, and M. M. Edvall, "User's guide for Tomlab /CPLEX," June 2007.