Technical report

A non-iterative cascaded predictive control approach for control of irrigation canals

R.R. Negenborn, A. Sahin, Z. Lukszo, B. De Schutter, M. Morari

Delft University of Technology, Delft, The Netherlands

A non-iterative cascaded predictive control approach for control of irrigation canals

R.R. Negenborn[*], A. Sahin[†], Z. Lukszo[‡], B. De Schutter[*,§], and M. Morari[†]

[*] Delft Center for Systems and Control
[‡] Faculty of Technology and Policy Making
[§] Department of Marine and Transport Technology
Delft University of Technology, Delft, The Netherlands
Email: {r.r.negenborn,z.lukszo}@tudelft.nl, b.deschutter@dcsc.tudelft.nl

[†]Automatic Control Laboratory
ETH Zürich, Zürich, Switzerland
Email: {sahin,morari}@control.ee.ethz.ch

Abstract—Irrigation canals transport water from water sources (such as large rivers and lakes) to water users (such as farmers). Irrigation canals are typically very large in nature, covering vast geographical areas, and involving a significant number of control actuators, such as pumps, gates, and locks. The control of such canals is aimed at guaranteeing the adequate delivery of water with minimal water spillage and with minimal control structure usage. To take into account forecasts on, e.g., water consumption and weather, model predictive control (MPC) can be used to determine which actions to take. For large-scale systems, in which different parts of the canal are owned by different parties, distributed MPC control could then be employed. Although iterative distributed MPC approaches proposed earlier in the literature may yield overall optimal performance, the amount of iterations required before achieving this performance may be large, and thus require a significant amount of time. In this paper, the structure of systems consisting of serially interconnected subsystems is exploited to obtain an efficient non-iterative, cascaded MPC scheme. Simulation studies on a 7-reach irrigation canal illustrate the performance of this non-iterative scheme in comparison with an iterative scheme.

Index Terms—Model predictive control; cascaded optimization; irrigation canals.

I. INTRODUCTION

Irrigation canals, like many other water systems, are large-scale systems that can be considered to consist of several smaller interconnected subsystems, in this case canal reaches. The dynamics of these canal reaches depend on one another, since water flowing out of one reach enters another reach. Hence, to optimize the performance of the system as a whole, the dynamics of each individual reach have to be taken into account. For large-scale systems, implementing a single controller that takes into account the dynamics of all reaches, can be infeasible due to large communication requirements or due to different parts of the canal being owned by different owners. In that case, instead of solving the control problem from a single centralized location, the control problem has to be solved in a distributed way. The actions determined by individual local controllers, each controlling its own canal reach, have to be coordinated. Achieving in such a distributed way performance that is comparable to centralized control is a major challenge.

In most large-scale systems, physical constraints play an important role. In order to take physical constraints into account, so-called model predictive controllers (MPC) [1] can be used. Previous work, such as [2]–[4], has considered MPC control of large-scale water systems using distributed MPC, based on the decomposition of an overall optimization problem and coordination between individual problem solvers to obtain an optimal solution. These schemes can be categorized as *single-level*, *iterative* schemes. Single-level here refers to the characteristic that all controllers are assumed to have the same rights, in the sense that one controller cannot force upon another controller its actions. Iterative here refers to the characteristic that individual controllers perform several iterations of computation and communication before deciding which actions to take. Through these iterations, the controllers obtain the necessary information about how neighboring subsystems will react to their actions. For this scheme, under certain conditions (such as linearity of the subsystem models and convexity of the control objective functions), and given sufficient computation time, the performance that is achieved in a distributed way is even equal to the performance of a centralized controller. The main drawback of such iterative schemes is that the number of iterations required can be significant. This can result in decision making time that is larger than the control sample length.

To reduce the number of computations involved, here, we exploit the structure of systems consisting of serially interconnected subsystems to efficiently solve their distributed control problem using a *cascaded*, *non-iterative* distributed control scheme. In this scheme, the controllers are arranged in a virtual cascade or hierarchy, in which one controller can force upon other controllers its decision. In addition, in this scheme no iterations of computation and communication take place before the controllers decide on their actions. It is hereby assumed that the controllers at the top of the cascade do not force upon lower controllers actions that will block the
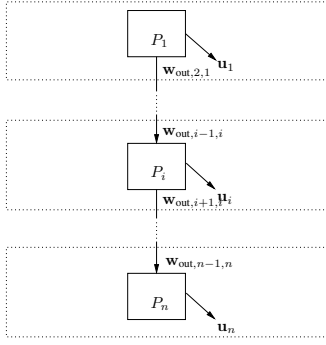
Fig. 1. Illustration of cascaded problems $P_1, \ldots, P_n$, representing the optimization problems of controllers $1, \ldots, n$, which determine actions $\mathbf{u}_1, \ldots, \mathbf{u}_n$.

lower controllers, in the sense that lower controllers should be able to still have feasible actions, after the decisions of the higher layers have been determined. The advantages of this scheme are clear: the computational time requirements are low, since no iterations are performed. However, the accompanying reduction in communication may result in a degradation of the performance, since a controller does not know how the other controllers will react to its actions.

In general, the actions of a particular controller can depend on an arbitrary number of neighboring controllers, both in a higher level and in a lower level. In this paper, however, we consider at most two neighboring subsystems, as illustrated in Figure 1. For the application to irrigation canal control, this is a natural choice, and moreover, this simplifies the explanation of the presented approach. It is noted, though, that extension to more than two neighbors is straightforward.

This paper is organized as follows. In Section II the dynamics of general interconnected linear subsystems are modeled, and we present a non-iterative cascaded distributed MPC solution technique for controlling these subsystems. In Section III the dynamics and control of irrigation canals are formulated in terms of the presented framework. Section IV describes the results of simulation studies on a 7-reach irrigation canal aimed at assessing the performance of the proposed approach. Hereby, a comparison is made with the performance of an iterative scheme. Conclusions and directions for future research are given in Section V.

II. MODELING AND CONTROL

Below we first formulate the model for subsystems as part of a larger network, the interconnections between subsystems, and the objectives for control of these subsystems. We then present an efficient non-iterative cascaded MPC scheme for controlling the subsystems.

A. Subsystem models

Consider a network divided into n subnetworks. Let the dynamics of subnetwork $i \in \{1, \ldots, n\}$ be 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):

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

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

where at control step 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 or measureable 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 that influence 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 various variables influence the local states and outputs of subnetwork i. Variables $\mathbf{v}_i(k)$ of subnetwork i represent the influence of other subnetworks on subnetwork i. These variables should therefore be equal to some of the variables of models representing dynamics of neighboring subnetworks.

Let $i - 1$ and $i + 1$ denote the indices of the neighbors of subsystem i. Define the so-called interconnecting input variables $\mathbf{w}_{\text{in},i-1,i}(k) \in \mathbb{R}^{n_{\mathbf{w}_{\text{in},i-1,i}}}$ as the variables of subnetwork i that are influenced by subnetwork $i - 1$, and let

$$\mathbf{w}_{\text{in},i-1,i}(k) = \mathbf{v}_i(k). \tag{3}$$

Let the so-called interconnecting output variables $\mathbf{w}_{\text{out},i+1,i}(k) \in \mathbb{R}^{n_{\mathbf{w}_{\text{out},i+1,i}}}$ denote the variables of subnetwork i that influence neighboring subnetwork $i + 1$, and let

$$\mathbf{w}_{\text{out},i+1,i}(k) = \mathbf{K}_i \left[\mathbf{x}_i^{\text{T}}(k) \quad \mathbf{u}_i^{\text{T}}(k) \quad \mathbf{y}_i^{\text{T}}(k) \right]^{\text{T}}, \tag{4}$$

where $\mathbf{K}_i$ is an interconnecting output selection matrix that contains zeros everywhere, except for a single 1 per row corresponding to a local variable that corresponds to an interconnecting output variable.

To obtain the dynamics for the whole network, the interconnecting inputs to the model of subnetwork i with respect to the model of subnetwork $i-1$ must be equal to the interconnecting outputs from the model of subnetwork $i - 1$ with respect to subnetwork i, since the variables of both subnetworks model the same quantity. For subnetwork i this thus gives rise to the following *interconnecting constraints*:

$$\mathbf{w}_{\text{in},i-1,i}(k) = \mathbf{w}_{\text{out},i,i-1}(k) \tag{5}$$

$$\mathbf{w}_{\text{out},i+1,i}(k) = \mathbf{w}_{\text{in},i,i+1}(k). \tag{6}$$

B. Subsystem control architecture

Subnetwork $i \in \{1, \ldots, n\}$ is controlled by a controller i that uses an MPC strategy to determine which actions to take. At discrete control cycle k, controller i finds the actions that over a prediction horizon of N steps give the best predicted performance, taking into account the current state of its subnetwork, the dynamics of its subnetwork, operational constraints, and objectives. The controller implements the actions for the coming time step and at the next time step performs a new optimization. It is hereby assumed that controller i:

- 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 estimate exogenous inputs $\mathbf{d}_i(k+l)$ of its subnetwork over a the horizon of length N_p, for $l = 0, \ldots, N_\mathrm{p} - 1$;
- can communicate with its neighboring controllers.

Moreover, the controllers are cooperative in the sense that the individual controllers strive for the best overall network performance. It is hereby assumed that the objectives of the controllers can be represented by convex functions $J_{\mathrm{local},i}$, for $i \in \{1, \ldots, n\}$. Such functions are commonly encountered, e.g., when considering set-point tracking or minimization of actuating effort [1].

C. Cascaded MPC for subnetwork control

1) Motivation: Since the dynamics of subsystems depend on one another, the controllers of the individual subsystems have to take into account one another's actions. If controller i wants to determine actions $\mathbf{u}_i(k)$ that are optimal from an overall point of view, it somehow has to know how the neighboring subsystems will behave over the prediction horizon. Controller i, however, generally does not know this; it does not know the values of the interconnecting variables $\mathbf{w}_{\mathrm{in},i-1,i}(k)$ over the prediction horizon, since these values are determined by controller $i-1$ via $\mathbf{w}_{\mathrm{out},i,i-1}$. Hence, one way or another controller i has to make assumptions on the behavior and capabilities of controller $i-1$.

There are several ways to deal with the unknown values of the interconnecting variables. Two of them are the following:

1) Controllers can determine the values in an iterative manner in the sense that one controller i informs a controller $i+1$ about what it wants, and given what controller i wants controller $i+1$ informs controller i what controller $i+1$ wants. This process is then performed over several iterations until a stopping condition is satisfied. In such a scheme the controllers have to determine values for their interconnecting variables, such that the interconnecting constraints (5)–(6) are satisfied.

2) One controller i can force another controller $i+1$ exactly which interconnecting outputs $\mathbf{w}_{\mathrm{out},i+1,i}(k)$ controller i will provide to controller $i+1$. It is hereby assumed by controller i that controller $i+1$ can deal with the interconnecting output $\mathbf{w}_{\mathrm{out},i+1,i}(k)$ provided by controller i as specified. Generally, this is not the case. However, this is the case, e.g., when there are no hard constraints in the control problem of controller $i+1$, or when hard constraints that are present can be replaced by soft constraints in the objective function of such a controller.

In previous work [2]–[4], iterative scheme have been proposed using the first way to deal with the interconnecting variables. Here, we consider the second way of dealing with the interconnecting variables.

2) Algorithm: Let controllers $\{1, \ldots, n\}$ be ordered in a virtual hierarchy with l_max levels. The controller i at the highest level does not have interconnecting inputs $\mathbf{w}_{\mathrm{in},i-1,i}(k)$.

The controller i at the lowest level does not have interconnecting outputs $\mathbf{w}_{\mathrm{out},i+1,i}(k)$. The ordering is now made in such a way that controller i at a particular level can solve its control problem for a given fixed $\mathbf{w}_{\mathrm{in},i-1,i}(k)$ as determined by the higher level controller. Hence, the virtual hierarchy is a cascade, as illustrated in Figure 1. Such a hierarchy can be naturally constructed for networks with uni-directional flows. In Section III we illustrate this for irrigation canals.

The approach proposed here for solving the MPC problems of the individual controllers at control step k is as follows:

1) Controller i at the highest level l_max of the hierarchy solves its problems. This results in local actions $\mathbf{u}_i(k)$ and interconnecting outputs $\mathbf{w}_{\mathrm{out},i+1,i}(k)$ for controller i at level l_max. Thus controller communicates the values found for its interconnecting outputs to its neighboring controller at level $l_\mathrm{max} - 1$.

2) Then, controller i at level $l_\mathrm{max}-1$ solves its optimization problem subject to the values of the interconnecting outputs $\mathbf{w}_{\mathrm{in},i-1,i}(k)$ forced upon controller i by the higher level controller. This results in local actions $\mathbf{u}_j(k)$ and interconnecting outputs $\mathbf{w}_{\mathrm{out},i+1,i}(k)$. Controller i at level $l_\mathrm{max} - 1$ communicates the values for the interconnecting outputs $\mathbf{w}_{\mathrm{out},i+1,i}(k)$ to its neighboring controller at level $l_\mathrm{max} - 2$.

3) The previous step of computation and communication continues until the controller at level 1 has received the information from its higher-level neighbor. The controller at level 1 solves its problem to obtain the values for its local inputs.

4) Once the controller at level 1 has solved its local optimization problem, the actions that all controllers have decided upon for the coming control cycle are implemented simultaneously.

This scheme is simple and only requires limited computation and communication. However, since the information exchange only flows in one direction, the controller at a higher level does not take into account the consequences of its control actions on the performance of the controller at a lower layer. Therefore, the optimal solution of a higher-level controller could result in infeasibility for a lower-level controller. Whether this will actually occur, depends on the particular series of problems to be solved and constraints to be considered.

In the remainder of this paper we discuss the application of the proposed approach for control of an irrigation canal. In this application, in fact, the optimal solution of a higher-level controller cannot result in infeasibility for a lower-level controller.

III. CONTROL OF AN IRRIGATION CANAL

Several works have been presented on control of irrigation canals [5], including approaches based on centralized MPC, such as [6]–[8]. In [4] we have proposed an iterative distributed MPC scheme for control of such a canal. Here we use the cascaded MPC scheme of Section II-C.
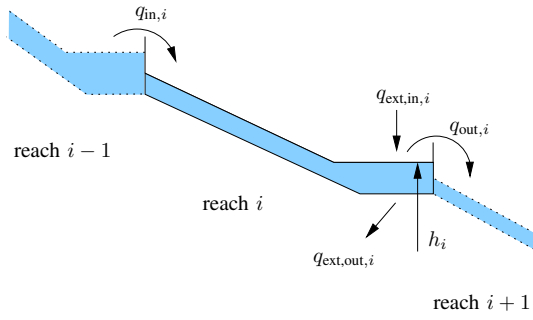
Fig. 2.   Illustration of canal reach i and associated variables.

A. Subnetwork dynamics

To model the dynamics of an irrigation canal with one origin and one destination, without splits and merges, similarly as in [9], we employ the discrete-time linear integral-delay model. Although this is a simplified representation of the real dynamics of irrigation canals, this model has shown to adequately capture relevant dynamics, and it reduces computations required for simulation of the dynamics significantly [10].

Let time be discretized into control cycles k and let the length of one control cycle be T_c (s). Each canal reach is considered to have an inflow from an upstream canal reach, as illustrated in Figure 2. Let this inflow into reach i be given by $q_{\text{in},i}(k) \geq 0$ (m^3/s). Canal reach i has an outflow to a downstream canal reach. Let $q_{\text{out},i}(k) \geq 0$ (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 $q_{\text{ext,in},i}(k) \geq 0$ (m^3/s) and such outflow by $q_{\text{ext,out},i}(k) \geq 0$ (m^3/s). Accurate predictions of the inflow $q_{\text{ext,in},i}(k)$ and the outflow $q_{\text{ext,out},i}(k)$ are assumed to be available in advance.

Depending on the changes in the inflows and outflows over time, the levels of the water in the 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, since that is where the water offtakes take place and where the water levels have to be kept close to the given set-points. 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 $h_i(k) \geq 0$ (m) denote the level of the water in canal reach i, and let the surface of reach i be $c_i \geq 0$ (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 $k_{\text{d},i}$ 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:

$$h_i(k+1) = h_i(k) + \frac{T_c}{c_i} q_{\text{in},i}(k - k_{\text{d},i}) - \frac{T_c}{c_i} q_{\text{out},i}(k)$$
$$+ \frac{T_c}{c_i} q_{\text{ext,in},i}(k) - \frac{T_c}{c_i} q_{\text{ext,out},i}(k). \quad (7)$$

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 $i+1$ this interconnection is expressed by

$$q_{\text{out},i}(k) = q_{\text{in},i+1}(k). \quad (8)$$

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

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

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

$$\mathbf{A}_i = \begin{bmatrix} 1 & \frac{T_c}{c_i} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

$$\mathbf{B}_{1,i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{B}_{2,i} = \begin{bmatrix} \frac{T_c}{c_i} & -\frac{T_c}{c_i} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \quad \mathbf{B}_{3,i} = \begin{bmatrix} -\frac{T_c}{c_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

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

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

B. Controllers

1) Available information: There are n controllers, and 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.

2) 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 $h_{\text{ref},i} \geq 0$, for all $i \in \{1, \ldots, n\}$ are minimized;
2) the changes in the set-points u_i (m) provided to the local flow controllers are minimized to reduce equipment wear.

Hence, the performance criterion $J_{\text{local},i}$ at control step k can be written as

$$J_{\text{local},i}(k) = \sum_{l=0}^{N_p-1} p_{h,i} \left(h_i(k+1+l) - h_{\text{ref},i} \right)^2$$
$$+ \sum_{l=0}^{N_p-1} p_{u,i} \left(u_i(k+l) - u_i(k-1+l) \right)^2,$$

TABLE I
VALUES OF THE PARAMETERS OF THE MODEL, TAKEN FROM [9].

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

where for controller i, $p_{h,i} \geq 0$ is the cost for the water level deviation and $p_{u,i} \geq 0$ is the cost for a change in the set-point provided to the local flow controller.

3) Dependencies: 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. The interconnecting input of controller i is defined as $\mathbf{w}_{\mathrm{in},i+1,i}(k) = q_{\mathrm{out},i}(k)$, and $\mathbf{K}_i$ is defined such that $\mathbf{w}_{\mathrm{out},i-1,i}(k) = q_{\mathrm{in},i}(k)$. Note that the controller of the most downstream canal reach does not have a neighboring downstream controller, and therefore also does not have interconnecting inputs. The controller of the most upstream canal reach does not have a neighboring upstream controller, and therefore does not have interconnecting outputs.

4) Virtual hierarchy of controllers: The controllers are ordered based on the topology of the irrigation canal. In irrigation canals, from the point of view of one canal reach, the control action is taken at the upstream end, and the disturbance appears at the downstream end, as illustrated in Figure 2. Hence, the controllers are ordered bottom up: starting at the downstream end, ending at the upstream end. With this ordering, the scheme presented in Section II-C can be implemented. It is important to note that, since no operational inequality constraints are present in the optimization problems of the individual controllers, these optimization problems can always be solved, and a higher level controller cannot block a lower level controller.

IV. SIMULATIONS

In this section we perform simulation experiments to assess the performance of the proposed scheme. We also compare this non-iterative scheme with a previously proposed iterative scheme. We have implemented the model of a benchmark irrigation canal consisting of 7 serially-connected canal reaches in Matlab v7.3. The parameters used for the model of the irrigation canal are shown in Table I. For solving the optimization problems at each control sample ILOG CPLEX v10.0.

A. Scenario

Consider a typical situation over 70 steps in which the farmers at offtakes 3 and 6 decide unexpectedly to increase their water consumption as in Figure 3. The time T_c between two consecutive control cycles is 240 s. The controllers use $N_p = 31$, to take into account the total delay present in the irrigation canal [9]. The cost coefficients are chosen as $p_{h,i} = 100$ units and $p_{u,i} = 10$ units, for $i \in \{1,\ldots,n\}$.

B. Results

Figure 4 shows the resulting trajectories determined by the cascaded scheme over the full simulation of 70 steps. We
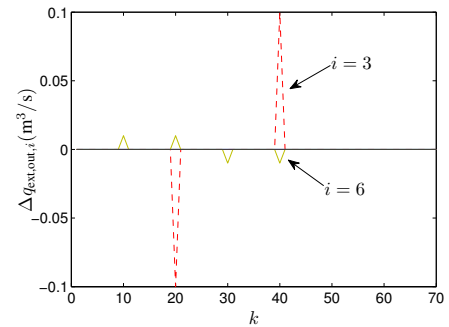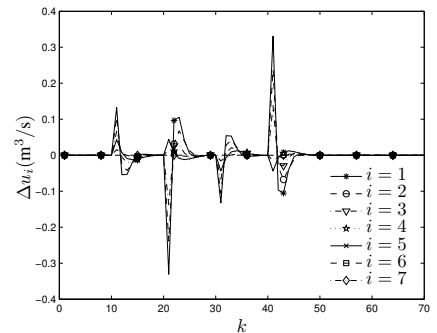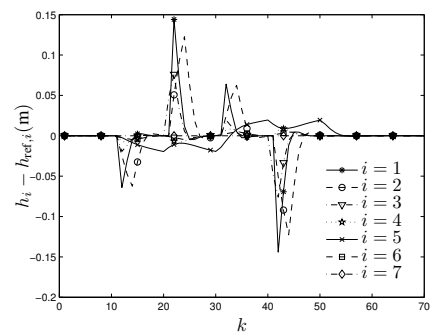

Fig. 3. Profiles of unexpected changes in water offtakes for reaches 3 and 6.
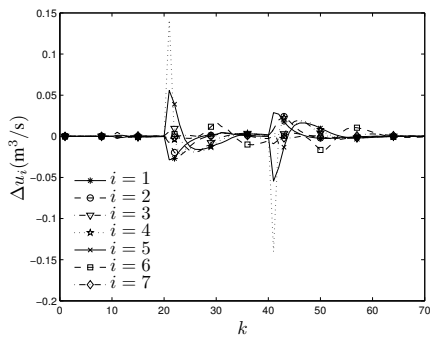

(a) Input changes.
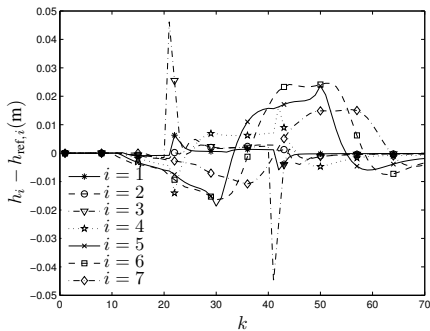

(b) Water level deviations.

Fig. 4. Trajectories over the full simulation with the non-iterative scheme.

observe that when the first disturbance appears (an increase in the amount of water taken out in reach 6), the controllers of reaches 1 until 6 react by increasing the water amount of water taken into the reach. Canal reach 7 takes no action, as is to be expected: The controller of canal reach 7 is at the highest level of the cascade and therefore does not know that a disturbance appeared at canal reach 6. The controller of reach 7 therefore has no incentive to perform any action.

Figure 5 shows the trajectories obtained with the iterative scheme proposed in [4]. The controllers perform iterations of computation and communication until an overall optimal solution has been found, or a maximum number of iterations (here set to 40, to ensure that every 240 s a new decision has been made), has been reached. We observe that here the iterative scheme obtains better performance by taking into

(a) Input changes.

(b) Water level deviations.

Fig. 5.   Trajectories over the full simulation with the iterative scheme.
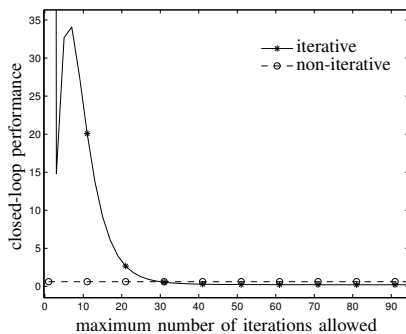

Fig. 6.   Closed-loop performance of the non-iterative and the iterative approach over the full simulation as a function of the number of iterations.

account that sometimes it is beneficial from an overall point of view for a downstream controller to act such that an upstream controller has less problems. The non-iterative scheme does not do this, since a higher-level controller i forces a lower-level controller j to accept what higher-level controller i decides, given only the local subsystem model and information from the higher level of the controller i. In the non-iterative scheme, coordination by the controllers is achieved implicitly (with a delay) via the dynamics of the canal reaches.

If fewer iterations are allowed, then the non-iterative scheme can yield better performance, as seen in Figure 6. This figure illustrates the closed-loop performance over the simulation (i.e., the sum of the objective function values of all controllers,

based on the chosen actions and resulting state trajectories) as a function of the allowed number of iterations for the iterative scheme. If less than 30 iterations are allowed, then the non-iterative scheme performs better than the iterative.

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper we have discussed the use of a non-iterative cascaded model predictive control approach for controlling irrigation canals. Controllers are organized in a cascade, in which higher-level controllers perform their computations before lower-level controllers. The presented scheme is fast, since at each control cycle a controller only performs one optimization and communication step. The drawback is that a higher-level controller does not know what disturbances a lower-level controller faces. Simulations on a 7-reach irrigation canal illustrate this in comparison with an iterative scheme.

Future research addresses including inequality constraints and analyzing its consequences on performance. Also, improving decision making performance by giving local controllers more knowledge about subsystem dynamics and objectives of other parts of the system will be investigated.

REFERENCES

[1] J. M. Maciejowski, *Predictive Control with Constraints*.   Harlow, UK: Prentice-Hall, 2002.
[2] G. Georges, "Decentralized adaptive control for a water distribution system," in *Proceedings of the 3rd IEEE Conference on Control Applications*, Glasgow, UK, 1999, pp. 1411–1416.
[3] H. El Fawal, D. Georges, and G. Bornard, "Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented Lagrangian," in *Proceedings of the 1998 International Conference on Systems, Man, and Cybernetics*, San Diego, California, 1998, pp. 3874–3879.
[4] R. R. Negenborn, P. J. van Overloop, T. Keviczky, and B. De Schutter, "Distributed model predictive control for irrigation canals," *Networks and Heterogeneous Media*, vol. 4, no. 2, pp. 359–380, June 2009.
[5] M. Cantoni, E. Weyer, Y. Li, S. K. Ooi, I. Mareels, and M. Ryan, "Control of large scale irrigation networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 75–91, Jan. 2007.
[6] P. O. Malaterre and J. Rodellar, "Multivariable predictive control of irrigation canals. design and evaluation on a 2-pool model," in *Proceedings of the International Workshop on Regulation of Irrigation Canals: State of the Art of Research and Applications*, Marrakech, Morocco, Apr. 1997, pp. 239–248.
[7] B. T. Wahlin and A. J. Clemmens, "Automatic downstream water-level feedback control of branching canal networks: Simulation results," *Journal of Irrigation and Drainage Engineering*, vol. 132, no. 3, pp. 208–219, May 2006.
[8] O. Begovich, V. M. Ruiz, G. Besançon, C. I. Aldana, and D. Georges, "Predictive control with constraints of a multi-pool irrigation canal prototype," *Latin American Applied Research*, vol. 37, no. 3, pp. 177–185, Aug. 2007.
[9] P. J. van Overloop, "Model predictive control on open water systems," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands, June 2006.
[10] 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.