Distributed Model Predictive Control Methods For Improving Transient Response Of Automated Irrigation Channels

Ali Khodabandehlou, Alireza Farhadi and Ali Parsa

***Abstract*—This paper presents distributed model predictive control methods for improving the transient response of feedback-controlled irrigation channels by managing the water-level reference set points across the irrigation season. The implemented distributed model predictive control methods exploit a two-level and single-level architectures for communication. For an automated irrigation channel, the satisfactory performance of the method with two-level architecture for communication in improving the transient response of automated irrigation channel is illustrated using computer simulation and compared with the performance of the distributed model predictive control method that exploits a single-level architecture for communication. It is illustrated that the distributed model predictive control method that exploits a two-level architecture for communication has a better performance in improving the transient response by better managing communication overhead.**

I. Introduction

An automated irrigation channel is a system of water transfer for agricultural purposes that consists of several pools cascaded by flume gates (see Fig. 1). Each flume gate includes an overshot gate, a modem for wireless communication, sensors, actuators and a processing device for making decision. The adjustment of the flow over upstream gates regulate the downstream water level in each pool under a downstream control. This leads to adequate releasing water from reservoir in the response to changes in the water levels of pools from the reference set points; and hence, near to demand supply that avoids water transmission lost due to oversupply. As the irrigation pools are very often long, delays for responding to the changes ruin the performance of automated irrigation channel and results in upstream transient error propagation and amplification phenomenon. This results in actuators saturation and flooding in long automated irrigation channels.

To mitigate these drawbacks, in [1] a supervisory controller is proposed that exploits the information available from automated irrigation channel and properly manages the reference set points for local controllers of distributed flume gates. To achieve this goal, the supervisory controller solves a constrained linear quadratic optimal control problem. But, in long irrigation channels, the total number of constraints and decision variables are very large. Therefore, to solve this optimal control problem in real time using closed loop control strategy, the receding horizon idea must be used. However, the computation overhead (i.e., the time spent for computing the optimal solution) using centralized model predictive control methods, as used in [1], at each receding horizon will not be practical for long irrigation channels because of large number of constraints and decision variables. For large irrigation channels, Distributed Model Predictive Control (DMPC) methods [2], [4] must be used that have relatively much smaller computation overhead.

Towards overcoming this computational scalability problem, the distributed model predictive control method of [2] can be used. This method solves constrained linear quadratic optimal control problems using a single-level architecture for communication by exploiting the computational power often available at each sub-system in the network. This distributed control method consists of two steps: 1) Initialization and 2) Iterated (parallel) computation and communication for exchanging updates of components of the overall decision variable between distributed computing resources.

To provide scope for managing the communication overheads, motivated by the initial idea presented in [3], in [4] a distributed model predictive control method with hierarchical (two-level) architecture for communication and a three-step algorithm including an extra outer iterate step are presented for solving constrained linear quadratic optimal control problems. Hence, this method seems to be more suitable to overcome the above computational scalability problem. In This method, distributed decision makers are grouped into q disjoint neighborhoods. Exchange of information between decision makers within a neighborhood occurs after each update, whereas the exchange of information between neighborhoods is limited to be less frequent. Within a neighborhood, each decision maker frequently updates its local component of the overall decision variable by solving an optimization problem of reduced size. The updated value is then communicated to all other neighboring decision makers. This intra-neighborhood update and communication is referred as an inner iterate. In addition to inner iterates, updates of decision variables from other neighborhoods are received periodically.

A. Khodabandehlou was M.Sc. student in the Department of Electrical Engineering, Sharif University of Technology.
A. Farhadi is Assistant Professor in the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. Email: afarhadi@sharif.edu.
A. Parsa is currently a Ph.D student in the Department of Electrical Engineering, Sharif University of Technology.
This work was supported by the research office of Sharif University of Technology.

These are referred to as outer iterates. Between outer iterates, distributed decision makers continue to compute and refine the local approximation of the optimal solution, with fixed values for decision variables from outside the neighborhood.

In this paper, the distributed model predictive control methods of [2] and [4] are used to solve the constrained linear quadratic optimal control problem associated with the supervisory controller of [1] for automated irrigation channels. For an automated irrigation channel, the satisfactory performance of the method with two-level architecture for communication [4] in improving the transient response of automated irrigation channel and mitigating the upstream transient error propagation and amplification phenomenon is illustrated using computer simulation and compared with the performance of the distributed model predictive control method that exploits a single-level architecture for communication [2]. It is illustrated that the distributed model predictive control method that exploits a two-level architecture for communication has a better performance by better managing communication overhead.

The paper is organized as follows: Section II briefly describes the distributed model predictive control method of [4] and also the method of [2] (as the method of [2] is a special case of the method of [4]). Section III is devoted to modeling automated irrigation channels, explaining upstream transient error propagation and amplification phenomenon in automated irrigation channels and the formulation of the constrained linear quadratic optimal control problem to be solved for mitigating this phenomenon. Section IV is devoted to simulation study and applications of distributed model predictive control methods of [4] and [2] in automated irrigation channels. In Section V the paper is concluded by summarizing the paper.

II. Distributed MPC with Hierarchical Architecture for Communication

The distributed model predictive controller of [4] is concerned with n interacting sub-systems: S_1, S_2, ... , S_n each equipped with a decision maker with limited computational power for solving the following optimization problem in a distributed fashion at each receding horizon:

$$\min_{(u_1,...,u_n)} \Big\{ J(\mathbf{g}, u_1, ..., u_n),\ u_i \in \mathcal{U}_i, \forall i \Big\}. \tag{1}$$

Here, $\mathbf{g}$ is a collection of known vectors, $J \geq 0$ is a finite-horizon quadratic cost functional of decision variables with horizon length $\bar{N} << L$, for each $i = 1, 2, ..., n$, $u_i \in \mathbb{R}^{\bar{N}m_i}$ is the decision variable associated with sub-system S_i and $\mathcal{U}_i$ is a closed convex subset of the Euclidean space $\mathbb{R}^{\bar{N}m_i}$ that includes zero vector.

For the simplicity of presentation, without loss of generality, the dependency of the cost functional J on $\mathbf{g}$ is dropped. Throughout, it is assumed that decision makers have knowledge of known parameters described by $\mathbf{g}$ and also the expression for the cost functional J in (1) at each receding horizon. To manage the communication overhead, the distributed controller of [4] uses a two-level architecture for exchanging information between distributed decision makers. This communication architecture involves a collection of disjoint neighborhoods of sub-systems. In each neighborhood at least one decision maker is selected as the neighborhood cluster head such that all the sub-systems of the neighborhood and also all the sub-systems of the nearest neighboring neighborhood are within the effective communication range of the neighborhood cluster head so that the communication graph between cluster heads is connected. That is, there is a communication path between a cluster head to any other cluster heads.

Without loss of generality, suppose sub-systems S_1, S_2,...,S_n are distributed into q disjoint neighborhoods, as follows: $\mathcal{N}_1 = \{S_1, .., S_{l_1}\}$, $\mathcal{N}_2 = \{S_{l_1+1}, ..., S_{l_2}\}$, ..., $\mathcal{N}_q = \{S_{l_{q-1}+1}, ..., S_n\}$. Then, the distributed model predictive controller of [4] approximates the solution of the optimization problem (1) for each time instant $k \in \{0, 1, 2, 3, ...\}$ by taking the following three steps:

- *Initialization:* The information exchange between neighborhoods at outer iterate $t \in \{0, 1, 2, ..., \bar{t} - 1\}$ makes it possible for sub-system S_i to initialize its local decision variables as $h_i^0 = u_i^t \in \mathbb{R}^{\bar{N}m_i}, \forall i \in \{1, ..., n\}$. For $k \geq 1$, $u_i^0 = (\, u_i^{\bar{t}'}[1] \quad u_i^{\bar{t}'}[2] \quad ... \quad u_i^{\bar{t}'}[\bar{N}-1] \quad \mathbf{0}'\,)'$ where

  $$u_i^{\bar{t}} = (\, u_i^{\bar{t}'}[0] \quad u_i^{\bar{t}'}[1] \quad ... \quad u_i^{\bar{t}'}[\bar{N}-1]\,)' \in \mathbb{R}^{\bar{N}m_i}$$

  is the approximated solution of the optimization problem (1) for the time instant $k-1$. For the time instant $k = 0$, $u_i^0 \in \mathcal{U}_i$ are chosen arbitrarily. This means that the initialization is based on the warm start for $t = 0$.

- *Inner Iterate:* Between every two successive outer iterates there are $\bar{p}$ inner iterates. Sub-system $S_i \in \mathcal{N}_e$ ($e \in \{1, 2, ..., q\}$) performs $\bar{p}$ inner iterates, as follows:
  For each inner iterate $p \in \{0, 1, ..., \bar{p}-1\}$, sub-system S_i first updates its decision variable via

  $$h_i^{p+1} = \pi_i h_i^* + (1 - \pi_i) h_i^p, \tag{2}$$

  where π_i are chosen subject to $\pi_i > 0$, $\sum_{j=1}^{l_1} \pi_j = 1, \ ... \ , \sum_{j=l_{q-1}+1}^{n} \pi_j = 1$ and $h_i^* \doteq \mathrm{argmin}_{h_i \in \mathcal{U}_i} J(h_1^0, ..., h_{l_{e-1}}^0, h_{l_{e-1}+1}^p, ..., h_i, ..., h_{l_e}^p, h_{l_e+1}^0, ..., h_n^0)$ (note that $l_0 = 0$, $l_q = n$). Then, it trades its updated decision variable h_i^{p+1} with all other sub-systems in its neighborhood $\mathcal{N}_e$.

- *Outer Iterate:* After $\bar{p}$ inner iterates, there is an outer iterate update as follows:

  $$u_i^{t+1} = \lambda_i h_i^{\bar{p}} + (1 - \lambda_i) u_i^t, \tag{3}$$

  where $t \in \{0, 1, 2, ..., \bar{t} - 1\}$ and λ_i, $i = 1, 2, ..., n$, are chosen subject to $\lambda_i > 0$, $\lambda_1 = ... = \lambda_{l_1}$, $\lambda_{l_1+1} = ... = \lambda_{l_2}$, $..., \lambda_{l_{q-1}+1} = ... = \lambda_{l_q}$ $(\lambda_{l_q} = \lambda_n)$, $\lambda_{l_1} + \lambda_{l_2} + ... + \lambda_{l_q} = 1$. Then, there is an outer iterate communication,

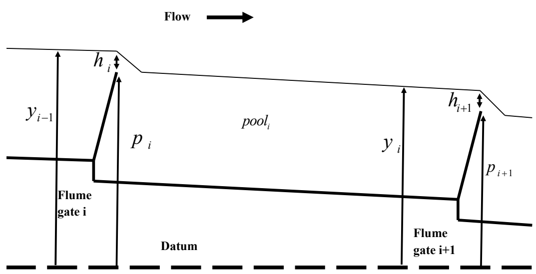

Fig. 1. An automated irrigation channel.

in which the updated decision variables u_i^{t+1} are shared between all neighborhoods; and subsequently, between all sub-systems.

The above three-step algorithm is repeated $\bar{t}$ times. As shown in [4] when $\bar{t} \to \infty$, then $u_i^{\bar{t}}$ converges to the optimal solution of the optimization problem (1). Hence

$$u_i^{\bar{t}} = (\, u_i^{\bar{t}'}[0] \quad u_i^{\bar{t}'}[1] \quad ... \quad u_i^{\bar{t}'}[\bar{N}-1]\,)' \in \mathbb{R}^{\bar{N}m_i}$$

represent the approximated solution of the optimization problem (1) for the time instant k, in which based on the receding horizon idea, its first component, i.e., $u_i^{\bar{t}}[0]$ is applied by the sub-system i to the system as the control input of the time instant k.

III. Automated Irrigation Channel

This section is devoted to modeling automated irrigation channels, explaining upstream transient error propagation and amplification phenomenon in automated irrigation channels and the formulation of the constrained linear quadratic optimal control problem to be solved by the distributed model predictive controllers of [2] and [4] for mitigating this phenomenon.

A. Automated Irrigation Channel Model

Throughout, we use the following model for automated irrigation channel that has been borrowed from [5]. Consider the automated irrigation channel shown in Fig. 1. For the i-th pool of this channel we have the following representation

$$\begin{aligned}\dot{y}_i(t) = C_i^{in} z_i(t-\tau_i) + C_i^{out} z_{i+1}(t) + C_i^{out} d_i(t),\\ i = 1,2,...,n,\\ z_i \doteq h_i^{\frac{3}{2}}, \quad h_i = y_{i-1} - p_i,\\ z_{i+1} \doteq h_{i+1}^{\frac{3}{2}}, \quad h_{i+1} = y_i - p_{i+1},\\ d_i = \frac{\bar{d}_i}{\gamma_{i+1}}, \quad d_n = \frac{\bar{d}_n}{\gamma_n}.\end{aligned} \tag{4}$$

Here, $\alpha_i > 0$ (measured in meter square - m^2) is a constant that depends on the pool surface area, $y_i \geq 0$ (measured in meter Above Height Datum - mAHD) is the downstream water level at the i-th pool, $h_i \geq 0$ (measured in meter) is the head over upstream gate (the i-th gate), h_{i+1} is the head over downstream gate (the i+1-th gate), $p_i \geq 0$ (measured in meter) is the position of the i-th gate, τ_i (measured in minutes) is the fixed transport delay, $\bar{d}_i \geq 0$ (measured in meter cube per minutes - m^3/min) is the known off-take flow rate disturbance taken at the end of pool i by user, and γ_i as well as C_i^{in} (measured in $m^{\frac{-1}{2}}/min$) and C_i^{out} (measured in $m^{-\frac{1}{2}}/min$) are constant.
The equation (4) can be written in terms of the storage (integrator) equation (5) and transport (delay) equation (6), as follows.

$$\dot{y}_i(t) = C_i^{in}\tilde{z}_i(t) + C_i^{out} z_{i+1}(t) + C_i^{out} d_i(t). \tag{5}$$

$$\tilde{z}_i(t) = z_i(t - \tau_i). \tag{6}$$

The storage equation (5) can be directly converted to discrete time model using the zero holder technique, while the transport equation (6) is converted to discrete time model by introducing $\frac{\tau_i}{T}$ states as follows $x_{i,2}(kT) = z_i(kT - \frac{\tau_i}{T})$, ... , $x_{i,\frac{\tau_i}{T}+1}(kT) = z_i(kT-T)$, where the sampling period T is the biggest common factor of pools transport delays (note that $x_{i,1}(kT) = y_i(kT)$, $k \in \{1,2,3,...\}$). Following the above conversions, the equivalent discrete time model describing the dynamics of the i-th sub-system/pool is given by the following:

$$\begin{cases} s_i[k+1] = \check{A}_i s_i[k] + \check{B}_i z_i[k] + \check{D}_i z_{i+1}[k] + \check{F}_i d_i[k], \\ \quad y_i[k] = \check{C}_i s_i[k] \qquad\qquad i = 1,2,...,n \end{cases}$$

In automated irrigation channel, PI controllers $z_i(s) = C_i(s)e_i(s)$, $C_i(s) = \frac{K_iT_is+K_i}{T_iF_is^2+T_is}$, $e_i = u_i - y_i$ are used to stabilize automated irrigation channel around the pre-defined reference signals u_is. Now, by finding the corresponding discrete time transfer function $C_i(z)$ and then the corresponding state space representation, we have the following discrete time representation for the PI controllers

$$\begin{cases} q_i[k+1] = \bar{A}_i q_i[k] + \bar{B}_i e_i[k], \; q_i[0] = 0, \\ \quad z_i[k] = \bar{C}_i q_i[k]. \end{cases}$$

Consequently, by defining the augmented state variable $x_i[k] = \begin{pmatrix} s_i[k] \\ q_i[k] \end{pmatrix}$, the dynamics of the automated irrigation channel is given by (7)

$$S_i : \begin{cases} x_i[k+1] = A_i x_i[k] + B_i u_i[k] + F_i d_i[k] \\ \qquad\qquad + v_i[k], \\ \quad y_i[k] = C_i x_i[k], \\ \quad z_i[k] = D_i x_i[k], \end{cases} \tag{7}$$

for $i = 1,2,...,n$ and $k \in \{0,1,2,...,\}$. In the above dynamic model $v_i[k] = M_i x_{i+1}[k]$ represents the cascade interconnection, $x_i \in \mathbb{R}^{n_i}$ is the state variable of dimension $n_i \in \mathbb{N} \doteq \{1,2,3,...\}$, $u_i \in \mathbb{R}$ is the reference set point, $y_i \in \mathbb{R}$ and $z_i \in \mathbb{R}$ are variables to be controlled, and $d_i \in \mathbb{R}$ is a known off-take disturbance for the i-th sub-system.

Pool	$c_{in}(\frac{m^{-1/2}}{min})$	$c_{out}(\frac{m^{-1/2}}{min})$	$\tau(min)$	$\gamma_i(\frac{m^3}{min})$	K_i	T_i	F_i
1 (pool 2 of the East Goulburn)	0.01072	0.01034	36	2330	0.585	539	47.2
2 (pool 5 of the East Goulburn)	0.01169	0.00833	28	1210	0.679	366	43.4
3 (pool 8 of the East Goulburn)	0.01065	0.01599	15	351	0.892	355	31.2
4 (pool 9 of the East Goulburn)	0.08457	0.0853	1	527	1.31	26.1	3.1

TABLE I. NUMERICAL VALUES FOR PARAMETERS DESCRIBING A FEW POOLS OF THE EAST GOULBURN MAIN IRRIGATION CHANNEL.

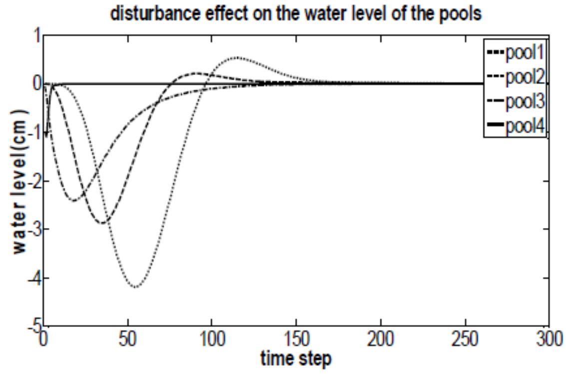

Fig. 2. The upstream transient error propagation and amplification phenomenon.

B. Upstream Transient Error Propagation and Amplification Phenomenon

For the purpose of illustrating the upstream transient error propagation and amplification phenomenon in automated irrigation channels, in this section we consider an automated irrigation channel consisting of pools 2,5,8 and 9 of the East Gourburn main irrigation channel located in Victoria, Australia. Numerical values for parameters describing these automated pools borrowed from [1] are given in Table I.

Fig. 2 illustrates the response of this automated irrigation channel to an off-take disturbance with the value of $\bar{d}_4 = 8\frac{m^3}{min}$ applied to the last pool (pool 9 of the East Goulburn irrigation channel) for the first 15 time steps (note that for simulations, the desired steady state values for water levels are set to be $1m$ above datum in Fig. 1 and for simulations the datum for the water levels are moved to the desired steady state water levels).

Fig. 2 illustrates the upstream transient error propagation and amplification phenomenon due to interplay between off-take flow disturbance and transport delay in this automated irrigation channel. As clear from Fig. 2 for the last pool (pool 4) the transient deviation of the water level from the desired set point is relatively small. However, this transient error propagates towards upstream pools and as we moved towards upstream pools, the deviation of water levels from desired set points increases. In large scale automated irrigation channels with large number of pools, at their worst, these undesirable transient characteristics can result in instability and performance degradation due to actuator limitations.

C. Reference Manager

As shown in [1] one way to mitigate the above effect is to equip automated irrigation channels with a supervisory controller which properly manages reference set points u_is of local PI controllers by solving a quadratic constrained optimal control problem. To formulate this problem note that the distributed dynamic model (7) for automated irrigation channels has the following augmented state space representation

$$\begin{cases} x[k+1] = Ax[k] + Bu[k] + Fd[k], \\ y[k] = Cx[k], \\ z[k] = Dx[k], \end{cases} \tag{8}$$

where $x[k] = (\, x_1'[k] \quad x_2'[k] \quad ... \quad x_n'[k]\,)'$, $u[k] = (\, u_1[k] \quad u_2[k] \quad ... \quad u_n[k]\,)'$, $d[k] = (\, d_1[k] \quad d_2[k] \quad ... \quad d_n[k]\,)'$, $y[k] = (\, y_1[k] \quad y_2[k] \quad ... \quad y_n[k]\,)'$, $z[k] = (\, z_1[k] \quad z_2[k] \quad ... \quad z_n[k]\,)'$. As the supervisory controller can have a larger sampling period than the sampling period of local PI controllers, the sampling period for supervisory controller is set to be ST, $S \in \mathbb{N}$. Hence, the dynamic model for supervisory controller is obtained by taking the model re-sampling approach, which involves holding the inputs to the system constant for the whole new sample period, and aggregating the dynamic (8) across the new sample period, as follows:

$$\begin{cases} x[k+1] = A^S x[k] + (\sum_{j=0}^{S-1} A^{S-j-1}B)u[k] \\ \quad +(\sum_{j=0}^{S-1} A^{S-j-1}Fd[Sk+j]), \\ y[k] = Cx[k], \\ z[k] = Dx[k], \\ k = \{0,1,2,3,...\}. \end{cases} \tag{9}$$

After obtaining the re-sampled model, the number of states in the re-sampled model (9) is reduced while maintaining the input-output behavior using balanced truncation. Consequently, the obtained reduced model for the supervisory controller has the following representation

$$\begin{cases} \hat{x}[k+1] = \hat{A}\hat{x}[k] + \hat{B}u[k] + \hat{d}[k], \\ y[k] = \hat{C}\hat{x}[k], \\ z[k] = \hat{D}\hat{x}[k], \\ k = \{0,1,2,3,...\}, \end{cases} \tag{10}$$

where $\hat{d}[k]$ represents the effect of known off-take disturbances on the reduced model. Now, the supervisory controller manages reference set points u_is of local PI controllers by solving the following quadratic constrained optimal control problem [1].

$$\begin{array}{l} \min\limits_{u=(u_1,\dots,u_n)} J_L(\hat{x}[0], \hat{d}_0^{L-1}, r, u) \\ \text{subject to (10) and} \\ \left\{ \begin{array}{c} y_i[k] \in [L_i, H_i],\ u_i[k] \in [L_i, H_i] \\ z_i[k] \in [E_i, Z_i] \end{array} \right\} \\ \forall i \in [1,n],\ k \in [0, L-1], \end{array} \tag{11}$$

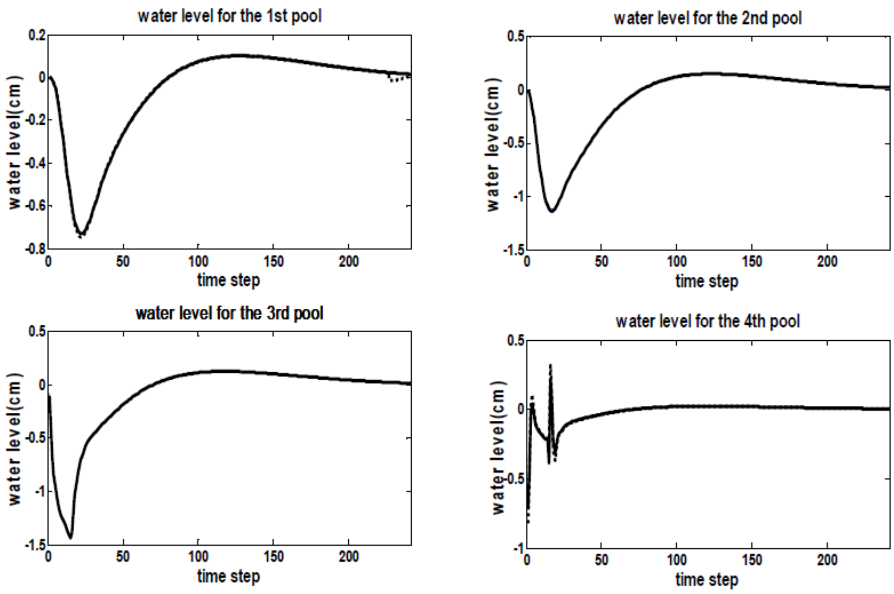

Fig. 3. DMPC with two-level architecture for communication. Solid line: without computational latency, dotted line: with latency.

where L is the irrigation season length, the interval $[L_i, H_i]$ is the admissible region for water-levels y_is and also decision variables u_is, the interval $[E_i, Z_i]$ is the admissible region for variable z_i which is a measure of water flow rate, and

$$J_L(\hat{x}[0], \hat{d}_0^{L-1}, r, u) \doteq \sum_{i=1}^{n} \sum_{k=0}^{L-1} ||y_i[k] - r_i||_Q^2$$
$$+||u_i[k] - u_i[k-1]||_R^2 + ||z_i[k]||_P^2 \quad (u_i[-1] = 0). \quad (12)$$

Here $||.||$ denotes the Euclidean norm (i.e., $||z||_P^2 \doteq z'Pz$), $\hat{x}[0]$ is the vector of known initial states, $\hat{d}_0^{L-1} \doteq \{\hat{d}[k]\}_{k=0,1,\ldots,L-1}$, where $\hat{d}[k] = (\hat{d}_1'[k] \ \ . \ . \ . \ \ \hat{d}_n'[k])'$ is a collection of known vectors that represent the effects of off-take disturbances, $r = (r_1 \ \ \ldots \ \ r_n)'$ is the vector of desired steady state values for y_is, and $Q, P \geq 0$, $R > 0$ are weighting matrices. The first norm in the cost functional (12) penalizes deviation of water levels from the corresponding desired values, and the second norm penalizes large changes in the input vector to the local PI controllers; and therefore, it tries to provide a smooth input trajectory. The last norm tries to minimize the input flow rates as z_is are measures of input flow rates; and therefore, it is desirable to make them as small as possible to keep water in reservoir as much as possible.

Remark 3.1: As the optimal control problem (11) is a constrained problem and the season length L is long, to solve this optimization problem in real time using a closed loop control strategy, the receding horizon idea must be used. In [1] a centralized model predictive control method is used to solve this optimal control problem, in which this method uses a centralized optimization method at each receding horizon. As shown in [6], the computation overhead for solving the optimization problem associated with problem (11) at each receding horizon using centralized optimization methods grows as $\mathcal{O}(n^5)$. However, the overhead using the methods of [2] and [4] grows as $\mathcal{O}(n)$. This results in large computational latency (i.e., time delay between making measurements and applying the corresponding control inputs) when we use centralized model predictive control methods for large scale automated irrigation channels which have large number of sub-systems n. Long computational latency significantly reduces the performance of the supervisory controller. Hence, for large scale automated irrigation channels a practical way to solve the constrained optimal control problem (11) using full computational capacity of existing local decision makers is to implement the distributed model predictive controller with either two-level or single-level architecture for communication. Using these controllers, at each time instant k by solving a constrained optimization problem with horizon length $\bar{N} << L$, the set points u_is for the time instant k is obtained and they are applied to automated irrigation channel by distributed decision makers.

IV. SIMULATION STUDY

In this section, the satisfactory performance of the distributed model predictive controller of [4] in mitigating the upstream transient error propagation and amplification phenomenon is illustrated using computer simulation and compared with the performance of the distributed model predictive controller of [2].

To illustrate the satisfactory performance of the distributed model predictive controller with two-level architecture for communication in improving the transient response of automated irrigation channels, this controller is applied to the optimization problem (11) formulated for the automated irrigation channel with four pools with numerical values as given in Table I. Here, it is assumed that the above automated irrigation channel is subject to an off-take disturbance with the

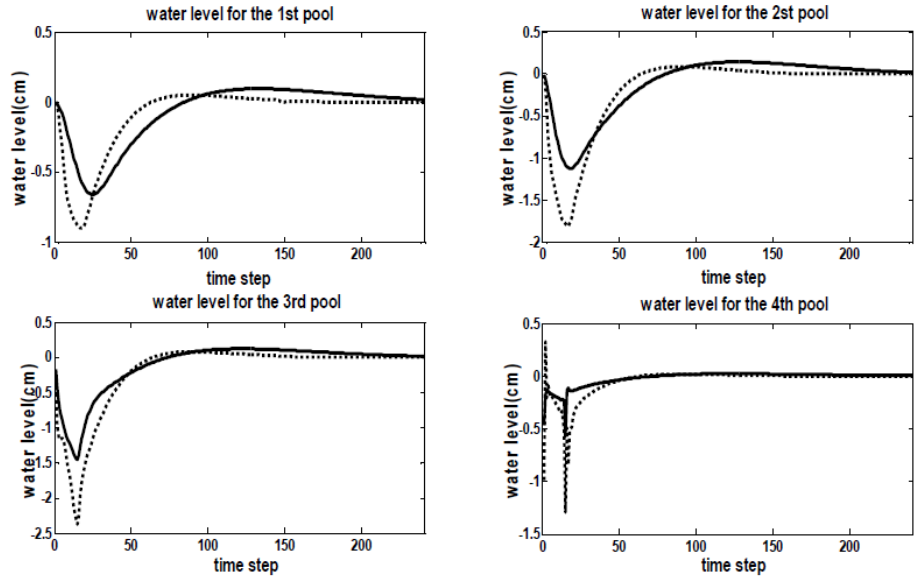

Fig. 4. DMPC with single-level architecture for communication. Solid line: without computational latency, dotted line: with latency.

value of $\bar{d}_4 = 8\frac{m^3}{min}$ for the first 135 min. It is also assumed that $x[0] = \mathbf{0}$, $r = \mathbf{0}$, $S = 9$, $L = 240$, $N = 10$, $[L_i, H_i] = [-0.2m, 0.2m]$ and $[E_i, Z_i] = [0, 0.75^{3/2}]$. Note that by applying balanced truncation, the reduced model has only 22 states instead of 92 states. Also, a similar method as used in [1] is used to guarantee the recursive feasibility of DMPCs. The communication load for each inner iterate communication is assumed to be 5sec., while for each outer iterate communication the load is assumed to be 50 sec. $\bar{p}$ is set to be 10 and $\bar{t} = 1$. Hence, at each receding horizon the communication overhead of the Distributed Model Predictive Control (DMPC) with two-level architecture for communication is $\bar{p} \times 5 + \bar{t} \times 50 = 10 \times 5 + 1 \times 50 = 100$ sec., and the overhead of the DMPC with single-level architecture for communication is $\bar{p} \times 50 = 500$ sec.

Fig. 3 illustrates the response of the DMPC with two-level architecture for communication without and with considering the computational latency. As the computation overhead in average is 13sec., its computational latency in average is $(100 + 13 =)$113sec., which is almost $2/9$th of the sampling period of 9min. As clear from Fig. 3, the response with the computational latency is similar to the response without latency. This result is expected because the computational latency here is almost 4 times smaller than the sampling period. As clear from Fig. 3, the magnitude of transient errors between water levels and the desired values decreases as we move towards upstream pools. This indicates that the DMPC with two-level architecture for communication mitigates the upstream transient error propagation and amplification phenomenon.

Fig. 4 illustrates the response of the DMPC with single-level architecture for communication without and with considering the computational latency. Here, the average computation overhead is 15sec., and hence the computational latency in average is 515sec. Fig. 4 clearly illustrates that the performance of the case with the computational latency in disturbance rejection is worst than the performance of the case without the computational latency. This results is expected because of large computational latency here. Obviously, this reduction in performance is more significant in larger channels that have larger communication overhead. Fig. 3 and Fig. 4 illustrate that the DMPC with two-level architecture has a performance better than the performance of DMPC with single-level architecture by better managing communication overhead.

V. Conclusion

In this paper, it was illustrated that the distributed model predictive control method with two-level architecture for communication is a suitable method for improving the transient response of automated irrigation channels. Hence, this method can be particularly used in large scale automated irrigation channels to mitigate the upstream transient error propagation and amplification phenomenon.

References

[1] M. Kearney and M. Cantoni, MPC-based reference management for automated irrigation channels, *in the Proceedings of 2012 Australian Control Conference*, Sydney, Australia, pp. 349-354, 2012.

[2] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright and G. Pannocchia, Cooperative distributed model predictive control, *Systems and Control Letters*, 59, pp. 460-469, 2010.

[3] B. T. Stewart, A. N. Venkat, J. B. Rawlings and S. J. Wright, Hierarchical cooperative distributed model predictive control, *in the Proceedings of American Control Conference*, Marriott Waterfront, Baltimore, MD, USA, pp. 3963-3968, 2010.

[4] A. Khodabandehlo, *Distributed model predictive control and its application to automated irrigation channels*, MSc. Thesis, Sharif University of Technology, 2014.

[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*, 95(1), pp. 75-91, 2007.

[6] A. Farhadi, P. M. Dower and M. Cantoni, Computation time analysis of a centralized and distributed optimization algorithms applied to automated irrigation networks, *in the Proceedings of 2013 Australian Control Conference*, Perth, Australia, pp. 263-269, November 2013.