Delft University of Technology

Delft Center for Systems and Control

Technical report 12-013

Distributed tree-based model predictive control on an open water system*

J.M. Maestre, L. Raso, P.J. van Overloop, and B. De Schutter

If you want to cite this report, please use the following reference instead:
J.M. Maestre, L. Raso, P.J. van Overloop, and B. De Schutter, "Distributed tree-based model predictive control on an open water system," *Proceedings of the 2012 American Control Conference*, Montréal, Canada, pp. 1985–1990, June 2012.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: http://www.dcsc.tudelft.nl

Distributed Tree-Based Model Predictive Control on an Open Water System

J. M. Maestre, L. Raso, P. J. van Overloop and B. De Schutter

Abstract— **Open water systems are one of the most externally influenced systems due to their size and continuous exposure to uncertain meteorological forces. In this paper we use a stochastic programming approach to control a drainage system in which the weather forecast is modeled as a disturbance tree. A model predictive controller is used to optimize the expected value of the system variables taking into account the disturbance tree. This technique, tree-based model predictive control (TBMPC), is solved in a parallel fashion by means of dual decomposition. In addition, different possibilities are explored to reduce the communicational burden of the parallel algorithm. Finally, the performance of this technique is compared with others such as minmax or multiple model predictive control.**

I. INTRODUCTION

Societies living near open waters strive at managing these waters by trying to control the water levels in them. As being open environmental systems, the open waters are vulnerable to meteorological influences. With the passing of the time, these disturbances can be predicted better and over longer horizons. Over the last decade, as being able to structurally handle anticipation, Model Predictive Control (MPC) has been proposed for control of open water systems in infrastructures with limited capacities and conflicting objectives [18], [21]. MPC is a high-performance control technique that handles in a systematic manner multi-variable interactions, constraints on manipulated inputs and system states, and optimization requirements. To this end, MPC uses a system model to predict the state future evolution along a given prediction horizon. The future predictions of the state, output, and input variables are used to minimize a given performance index, which is a cost function that defines the optimization criterion used to determine the best possible control action sequence. Due to its versatility, MPC has become very popular in the industry and many implementations can be found [2].

In this paper we are especially interested in the way MPC deals with uncertainty. The simplest way is to let the controller work in a nominal and deterministic fashion, which often results in a poor control performance. In general, it is more robust to assume a bounded set of unmeasured disturbances that may affect the system and then use min-max MPC [20], so that the controller is ready to face the worst possible scenario. Nevertheless, this approach has shown to be very conservative and is translated into slow response controllers [3]. An alternative approach to deal with uncertainty is stochastic programming (SP) [8], which models unknown disturbances as random variables and focuses on the control of the expected value of the system variables while guaranteeing robust constraint satisfaction. Under this approach unknown disturbances can be also taken into account by considering different representative realizations of the disturbances. For example, this idea has been previously used for the control of water systems in [19], where multiple MPC (MMPC) is proposed.

In this work we use tree-based MPC (TBMPC) which is an SP-MPC scheme that assumes that the time evolution of the most relevant possible disturbance signals can be synthesized in a rooted tree [13], [9]. Each root-to-leaf path is a possible disturbance scenario, i.e., branches appear in the tree as the different disturbance forecasts diverge along the prediction horizon. As a result of this, the controller provides us with a rooted tree of control actions that are calculated according to the different sequences in the disturbance tree. Consequently, non-anticipativity constraints [14] are introduced in the optimization problem in order to limit the anticipative nature of the control sequences. At this point, it is important to remark that in this paper we assume that all the scenarios contained in a tree have the same value at the first step of the prediction horizon. This assumption is reasonable in the context of open water systems, where the weather forecasts provide a good indication about how the disturbances will be specially at the beginning of the forecasted period.

Probably, the major issue about MPC is its computational burden, which is prohibitive for large-scale systems [10]. In order to overcome this problem, different distributed MPC (DMPC) techniques have been proposed during the last decade; see [15] for an extensive survey on this topic. In general, DMPC focuses on the application of MPC to systems that are composed of several subsystems governed by local controllers or *agents*. Each of these agents implements a local MPC controller and may or may not share information with the other subsystems. In this paper we use dual decomposition [12], [11] in order to distribute the TBMPC optimization problem. This approach has been also previously used in water management, in particular for water resources system planning; see [4].

This research was supported by the BSIK project Next Generation Infrastructures (NGI), the Delft Research Center Next Generation Infrastructures, the European STREP project Hierarchical and distributed model predictive control (HD-MPC, contract number INFSO-ICT-223854) and the EU Network of Excellence Highly-complex and networked control systems (HYCON2, FP7/2007-2013 under grant agreement no. 257462).

J. M. Maestre and B. De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands, j.m.maestretorreblanca@tudelft.nl, b.deschutter@tudelft.nl

L. Raso and P. J. van Overloop are with the Department of Water Management, Delft University of Technology, The Netherlands, l.raso@tudelft.nl, P.J.A.T.M.vanOverloop@tudelft.nl

Besides the application of distributed control techniques in order to consider different scenarios in the control of water systems, this paper studies the reduction of the communicational burden imposed by the use of dual decomposition. Specifically, we show that the limitation of the non-anticipativity constraints in the prediction horizon provides similar closed-loop performance with significantly less communicational burden. Likewise, we study different thresholds as stopping condition in the parallel algorithm. Finally, the performance of the controller is compared with other techniques that has been previously used in the literature such as MMPC [19].

II. DMPC BASED ON SCENARIO DECOMPOSITION

In this section we provide with the grounds for understanding how the scenario trees are generated in the first place. Next, we deal with the basic formulation of the MPC problem and explain how its computation can be distributed by means of dual decomposition.

A. Ensemble forecasting

In open water systems, the uncertainty is generally introduced by the unpredictable nature of the weather. Specifically, runoff derived from rainfall are the major source of uncertainty in this context. An ensemble forecast (EF) is a weather prediction composed of possible trajectories of its evolution. It is generated by a model that is run using different initial conditions or different numerical representations of the atmosphere, accounting for the major sources of forecast uncertainty [5]. These trajectories have generally small differences at the initial stage of the forecast; then they tend to diverge, because of the chaotic nature of the underlying model. Finally, notice that each of the trajectories in the EF has a certain associated probability.

For simulation purposes, a large number of scenarios improves the accuracy of the stochastic approach. However, this may lead to an excessive growth of the computational burden. In order to avoid this problem, a *representative* subset of scenarios can be chosen using a scenario reduction algorithm [6]. This reduced ensemble is bundled into a tree that will allow us to set up a multistage stochastic programming problem. In general, the tree is generated from the ensemble by aggregating trajectories over time until the difference between them becomes such that they can be no longer assumed to be similar. At such a point, a bifurcation is produced and the tree branches. The operation of tree generation has been the subject of numerous studies, for example [6], [17].

B. Model Predictive Control

At first, we begin analyzing a standard MPC controller. Let the model of the system be described by the following discrete time equation:

$$x(k+1) = Ax(k) + Bu(k) + Ew(k) \tag{1}$$

where $x(k) \in \mathbb{R}^q$ is the state of the system at time step k, $u \in \mathbb{R}^r$ is the vector of manipulated variables, and $w \in \mathbb{R}^s$ is a vector of measurable disturbances. A, B, and E are matrices of proper dimensions. We consider linear constraints in the states and the inputs, i.e., $x \in \mathcal{X}$ and $u \in \mathcal{U}$, where $\mathcal{X}$ and $\mathcal{U}$ are closed polyhedra defined by a system of linear inequalities. Without loss of generality, we can assume that the control objective consists in regulating the state vector to the origin. To this end, it is possible to define the following performance index

$$\begin{aligned} J(U, x_0) = &\sum_{k=0}^{N_h-1} \left(x^T(k)Qx(k) + Q_l x(k) + u^T(k)Ru(k) \right) \\ &+ x^T(N_h)Qx(N_h) + Q_l x(N_h) \end{aligned} \tag{2}$$

where Q, Q_l, and R are constant weighting matrices of the proper size and N_h is the prediction horizon. Taking into account equation (1), it can be seen that function J depends on the control input sequence $U = (u(0), \ldots, u(N_h - 1))$ and the value of the state at time step $k = 0$, x_0. The MPC controller calculates the optimal control action by minimizing this cost:

$$\begin{aligned} &U^* = \arg\min_U J(U, x_0) \\ &\text{s.t.} \\ &x(k+1) = Ax(k) + Bu(k) + Ew(k) \\ &x(k) \in \mathcal{X} \ \forall k \in \{1, \ldots, N_h\} \\ &u(k) \in \mathcal{U} \ \forall k \in \{0, \ldots, N_h - 1\} \\ &x(0) = x_0 \\ &w(k) = w_k \end{aligned} \tag{3}$$

where $w_k = (w_0, w_1, w_2, \ldots, w_{N_h})$ is a deterministic sequence composed by the present and future values of the disturbances. The result of this optimization problem is the optimal control sequence $U^* = (u^*(0), \ldots, u^*(N_h - 1))$. Only the first component of U^* is applied. The rest of the sequence provides information about the expected evolution of the manipulated variables in the future but is not implemented during the next sample times. At the next sample instant, the optimization problem (3) is solved using the current state at that time and the most recent disturbance forecast. The first component of the resulting control sequence is implemented again. This procedure is repeated every sample time in what is usually denominated as receding horizon strategy. Distributed TBMPC uses the tree provided by the ensemble forecasting and solves the problem (3) taking into account the sequences contained in the tree. In particular, problem (3) is solved for each possible disturbance sequence in the tree. Notice that the resulting set of problems can be solved by a set of agents in parallel. Thus, if there are N_s different scenarios in the tree, there are N_s different agents working in parallel. Nevertheless, the problem faced by the agents has more constraints than (3). Additional restrictions have to be imposed in order to account for the non-anticipativity constraints in the optimization procedure. To this end, we define the boolean auxiliary function $\delta_{i,j}(k)$ so that $\delta_{i,j}(k) = 1$ iff $u_i(k) = u_j(k)$, i.e., the control input of agent i must have the same value of the control input of agent j at time k.

In order to have an input tree with the same structure as the disturbance tree, $\delta_{i,j}(k)$ must be 1 for $k = 0, \ldots, k_{ij}$, where k_{ij} is first the time step at which the disturbance sequences corresponding to agents i and j are different. From a centralized point of view, the problem solved by a $TBMPC$ controller is the following:

$$
\begin{aligned}
&\min_{U_1,\ldots,U_{N_s}} \sum_{i=1}^{N_s} \alpha_i J(U_i, x_0)\\
&\text{s.t.}\\
&\left.\begin{array}{l}
x_i(k+1) = Ax_i(k) + Bu_i(k) + Ew_i(k)\\
x_i(k) \in \mathcal{X}\ \forall k \in \{1,\ldots,N_h\}\\
u_i(k) \in \mathcal{U}\ \forall k \in \{0,\ldots,N_h-1\}\\
x_i(0) = x_0\\
w_i(k) = w_{i,k}\\
\delta_{i,j}(k)u_i(k) = \delta_{i,j}(k)u_j(k)\\
\ \forall j \in \{1,..,N_s\},\ k \in \{0,\ldots,N_h-1\}
\end{array}\right\} \forall i \in [1, N_s]
\end{aligned}
\tag{4}
$$

where $w_{i,k} = (w_{i,0}, w_{i,1}, \ldots, w_{i,N_h})$ is a deterministic sequence composed by the present and future values of the disturbances faced by agent i and α_i is the probability assigned to the disturbance sequence $w_{i,k}$.

C. Dual Decomposition

In order to solve problem (4) in a distributed fashion, it is necessary to remove the coupling constraints of the type $u_i(k) = u_j(k)$. Dual decomposition can be used to this end [1]. In particular, the introduction of Lagrange multipliers $\lambda_{i,j}(k)$ allows us to separate the problem:

$$
\begin{aligned}
&\max_{\Lambda} \min_{U_1,\ldots,U_{N_s}} \sum_{i=1}^{N_s} \Bigg(\alpha_i J(U_i, x_{i,0})\\
&+ \sum_{j=1}^{N_s} \sum_{k=0}^{N_a} \delta_{i,j}(k)\lambda_{i,j}(k)(u_i(k) - u_j(k)) \Bigg)\\
&\text{s.t.}\\
&\left.\begin{array}{l}
x_i(k+1) = Ax_i(k) + Bu_i(k) + Ew_i(k)\\
x_i(k) \in \mathcal{X}\ \forall k \in \{1,\ldots,N_h\}\\
u_i(k) \in \mathcal{U}_i\ \forall k \in \{0,\ldots,N_h-1\}\\
x_i(0) = x_0\\
w_i(k) = w_{i,k}
\end{array}\right\} \forall i \in [1, N_s]
\end{aligned}
\tag{5}
$$

where $\Lambda = \{\lambda_{i,j}(k),\ \forall i,j \in \{1,..,N_s\},\ k \in \{0, N_a\} \mid \delta_{i,j}(k) = 1\}$ is the set of all the Lagrange multipliers, sometimes referred to as set of prices. Notice that we have introduced a new parameter, N_a, which is the agreement horizon. Ideally, $N_a = N_h - 1$, i.e, the agents have to respect the non-anticipativity constraints defined over the entire horizon. Nevertheless it may be desirable to limit the effect of these constraints in time so that the number of coupling constraints is reduced.

As it can be seen, problem (5) can be solved in a distributed fashion. The following algorithm shows the distributed optimization procedure that takes place at time step k:

- Step 0: Let l be the index used to count the number of iterations of the procedure. Initially $l = 0$ and an initial set of prices Λ^0 is given.
- Step 1: At each iteration l, each agent i calculates its own optimal input trajectory solving the following problem for a particular set of values of the Lagrange multipliers Λ^l:

$$
\begin{aligned}
&U_i^* = \arg\min_{U_i} \Bigg(\alpha_i J(U_i, x_0)\\
&+ \sum_{k=0}^{N_a} u_i(k) \sum_{j \neq i} \delta_{i,j}(k)\lambda_{i,j}^l(k) \Bigg)\\
&\text{s.t.}\\
&x_i(k+1) = Ax_i(k) + Bu_i(k) + Ew_i(k)\\
&x_i(k) \in \mathcal{X}\ \forall k \in \{1,\ldots,N_h\}\\
&u_i(k) \in \mathcal{U}\ \forall k \in \{0,\ldots,N_h-1\}\\
&x_i(0) = x_0\\
&w_i(k) = w_{i,k}
\end{aligned}
\tag{6}
$$

  Notice that problem (6) corresponds to the part of problem (5) that corresponds to agent i.
- Step 2: Once the input trajectories have been calculated, the prices of agent i are updated by a gradient step: $\lambda_{i,j}^{l+1}(k) = \lambda_{i,j}^l(k) + \gamma^l(u_i(k) - u_j(k))$. Notice that a price is only changed whenever there is a disagreement about the value of the variable between the agents. Convergence of these gradient algorithms has been proven under different type of assumptions on the step size sequence γ^l [16]. Note that in order to update the prices, the agents must communicate. See [7] for alternatives in the way the prices can be updated.
- Step 3: Let $\Delta u(k) = \max_{i,j} |u_i(k) - u_j(k)|$. The algorithm stops if $\Delta u(k) \leq \Delta u_{max}$, where Δu_{max} is a parameter that represents the maximum allowable difference between the proposals of any two agents. In case that $u(k)$ is a vector, this criterion is applied componentwise. Alternatively, the algorithm also stops if the number of iterations l exceeds a given threshold $l_{\max}$. Otherwise, the process is repeated from step 1 for $l = l + 1$.

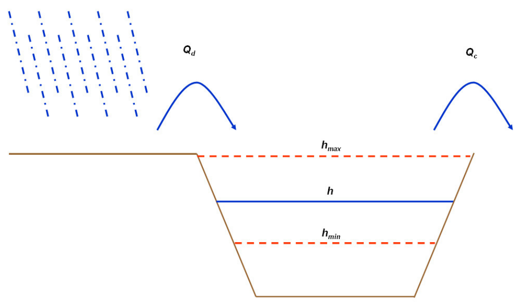
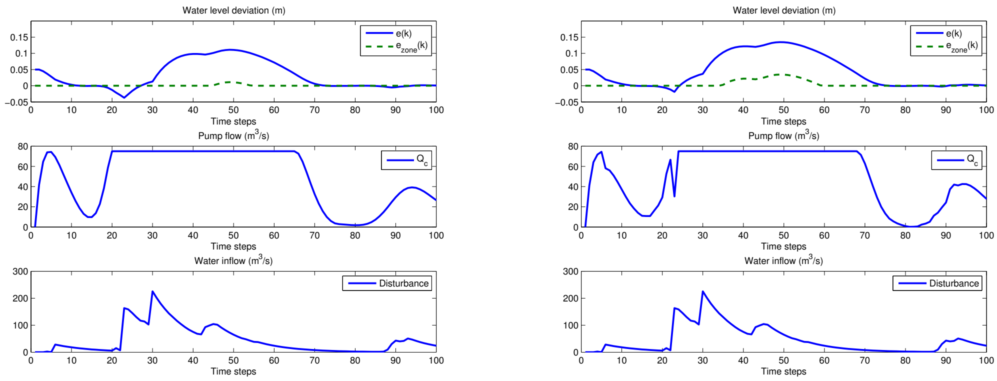


Fig. 1. Schematization of drainage water system.

III. Results and Discussion

We have applied the distributed TBMPC controller to a model of a real drainage system described in [19]. In Figure 1 a sketch of the water system can be seen. There are three important variables in this figure. In the first place we have



(a) (b)

Fig. 2. Simulation cases: (a) MPC with perfect forecast, (b) TBMPC.

the controlled variable, h (m), which is the average water level with respect to average sea level. The second variable is Q_c (m^3/s), which represents the effect of pumping water out of the system and is the manipulated variable. Finally, there is a disturbance term given by Q_d (m^3/s), which stands for the inflow of water due to rainfall. The discrete-time model used to represent the dynamics of the drainage canal system is:

$$h(k+1) = h(k) - \frac{T_{\mathrm{c}}}{A_{\mathrm{s}}} Q_{\mathrm{c}}(k - k_{\mathrm{d}}) + \frac{T_{\mathrm{c}}}{A_s} Q_{\mathrm{d}}(k) \tag{7}$$

where A_s is the average storage area (m^2), T_c is the control time step (s), k_d is the number of delay steps between control action and change in average water level, and k is the time step index. A state space model based on equation (7) will be used for the controller. In particular, the model will focus on the error between the current water level and the water level setpoint. Let h_{ref} be the constant water level setpoint. Thus, the error $e(k)$ in the water level can be calculated as $e(k) = h(k) - h_{\mathrm{ref}}$. In addition, the state vector will be expanded with the state variable $e_{\mathrm{zone}}(k)$ in order to represent explicitly how much the error $e(k)$ is above or below of a given safety margin defined around the water level setpoint. Taking all of this into consideration, we can define the following space state model:

$$\begin{aligned}
\begin{bmatrix} e(k+1) \\ e_{zone}(k) \\ Q_c(k) \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -\frac{T_c}{A_s} \\ 1 & 0 & -\frac{T_c}{A_s} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e(k) \\ e_{zone}(k-1) \\ Q_c(k-1) \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta Q_c(k) \\ u_{zone}(k) \end{bmatrix} + \begin{bmatrix} \frac{T_c}{A_s} \\ \frac{T_c}{A_s} \\ 0 \end{bmatrix} \cdot [Q_d(k)]
\end{aligned} \tag{8}$$

Note that a new variable has been introduced in this model, $u_{zone}(k)$, which is related to the constraints of the optimization problem the MPC controller solves. In particular, hard constraints are defined on the input of the system and a soft constraints are introduced on the state:

$$\begin{aligned}
&Q_c(k) \in [0, Q_{c,max}] \\
&u_{zone}(k) \in [h_{min} - h_{ref}, h_{max} - h_{ref}]
\end{aligned}$$

These constraints help us explain better the meaning of $u_{zone}(k)$. According to (8), $e_{zone}(k)$ may seem equal to a duplicated version of $e(k+1)$. The difference between them is that $e_{zone}(k)$ is affected by the auxiliary manipulated variable $u_{zone}(k)$, so that $e_{zone}(k) = 0$ as long as $e(k+1)$ stays between h_{min} and h_{max}. Note that $u_{zone}(k)$ is introduced only in order to avoid infeasibility problems.

The behavior of the disturbance $Q_d(k)$ in (8) is modeled as a tree that contains its most representative possible trajectories along the prediction horizon. The data contained in the tree that is given to the controller each time sample k is obtained from artificial ensemble forecasts that are generated using meteorological models.

In Figures 2(a) and 2(b), we show the results of a closed-loop simulation of the proposed scheme during 25 hours, which corresponds to a 100 time steps. The simulation takes place during a stormy event that tests the controller capability to keep the water level within the desired margin. The numerical values of the parameters that characterize the system and the cost function of the controller can be seen in Table I. In addition, we have carried out simulations with values different from the ones shown in Table I in order to see their influence in the controller performance. In particular, these additional simulations were carried out for different values of the agreement horizon, N_a, and the disagreement threshold, Δu_{max}.

At first, in Figure 2(a), the results of a centralized MPC controller with a perfect forecast are depicted. Notice that this case is given only as a reference since a perfect forecast, i.e. an exact knowledge about the future evolution of weather, cannot be obtained. However, this case gives us an upper bound for the performance of the rest of the controllers. In Figure 2(b) the centralized TBMPC configured with the parameters of Table I can be seen.

In general, it is not possible to calculate the exact number of optimization variables that the centralized TBMPC prob-

TABLE I

MODEL AND CONTROLLER PARAMETERS

Parameter	Symbol	Value
Storage area	A_{s}	7.3e6 (m^2)
Control time step	T_{c}	900 (s)
Number of scenarios in the tree	N_{s}	6
Prediction horizon	N_{h}	16
Control horizon	N_{c}	16
Maximum pump capacity	$Q_{\max}$	75 (m^3/s)
Delay in the actuation	k_{d}	1
Maximum number of iterations	$l_{\max}$	2000
Agreement horizon	N_{a}	10
Disagreement threshold	$\Delta u_{\max}$	1 (m^3/s)
Quadratic penalty on e	Q_{e}	180
Quadratic penalty on e_{zone}	$Q_{\mathrm{e_{zone}}}$	180e5
Linear penalty on Q_{c}	$Q_{1,\mathrm{Q_c}}$	130e-5
Quadratic penalty on ΔQ_{c}	$R_{\Delta Q_{\mathrm{c}}}$	1e-6
Quadratic penalty on u_{zone}	$R_{\mathrm{u_{zone}}}$	1e-16

lem has; it depends on the structure of the tree. An upper bound is $r \times N_s \times (N_c - 1) + r$, where r is the dimension of the manipulated variables vector, N_s is the number of scenarios and N_c is the control horizon. This corresponds to the case in which the tree branches completely after the first time step in the horizon. A lower bound is given simply by $r \times N_c$, assuming that there is only one scenario in the tree. The distributed algorithm that we use in this paper substitutes a problem with up to $r \times N_s \times (N_c - 1) + 1$ optimization variables by N_s separable problems with $r \times N_c$ variables that have to be solved iteratively until an agreement has been reached. The size of the problem that we use in this paper as an example is not big enough to demand the use of parallel computation, but it helps us illustrate how the problem is distributed and the iterative procedure that takes place for the convergence. In our example, the number of optimization variables is bounded between 32 and 182, and its distributed version is composed of 6 problems with 32 optimization variables for each one.

The separation of the original optimization problem into several separable optimization problem comes at a price: these problems have to be solved iteratively until an agreement has been obtained. In Table II, the number of iterations needed for convergence are shown for different values of the agreement horizon, N_a. As expected, a greater value of N_a implies a higher number of iterations, which is logical since it implies a higher number of variables on which the agents must reach an agreement. A question that arises naturally is how a change of N_a affects the performance of the controller. Table II also helps to answer this question. The pump flow difference between the centralized TBMPC and its distributed versions have been analyzed in order to calculate its mean and standard deviation along the 100 time steps of the simulation. As it can be seen, the results are quite similar. These results suggest that the completeness of

TABLE II

IMPACT ANALYSIS OF AN AGREEMENT HORIZON VARIATION ($\Delta u_{max} = 1$).

N_a	Iterations l for convergence: Mean	Iterations l for convergence: Std. dev.	Pump flow diff. (m^3/s): Mean	Pump flow diff. (m^3/s): Std. dev.	Cum. cost
Cen.	-	-	0	0	2.31e5
1	16.56	27.44	0.00	0.58	2.44e5
3	38.17	64.49	-0.01	0.51	2.35e5
5	70.84	121.57	-0.02	0.35	2.37e5

TABLE III

IMPACT ANALYSIS OF A DISAGREEMENT THRESHOLD Δu_{max} VARIATION ($N_a = 16$).

Δu_{max}	Iterations l for convergence: Mean	Iterations l for convergence: Std. dev.	Pump flow diff. (m^3/s): Mean	Pump flow diff. (m^3/s): Std. dev.	Cum. cost
Cen.	-	-	0	0	2.31e5
0.5	297.01	461.25	-0.00	0.12	2.32e5
1	169.82	279.41	-0.03	0.39	2.32e5
2	79.15	141.83	-0.00	0.63	2.42e5

the set of non-anticipativity constraints is not so relevant in the calculation of the first component of the control vector. In addition, Table II shows in its last column the value of the cumulated cost of the closed-loop system at the end of the simulation. This value is calculated according to the performance index defined in (2) and the parameters given in Table I. It is interesting to observe how the performance of these controllers was very similar to the performance shown by the centralized TBMPC.

Table II shows that the maximum deviation from the control signal provided by the centralized TBMPC is always below 3%, but this value can be reduced with a proper adjustment of the disagreement threshold Δu_{max}, which is another degree of freedom with a strong influence on the number of iterations. For this reason, Table III shows the impact of a variation of Δu_{max}. As Δu_{max} grows, the number of iterations and the quality of the solution decrease.

Finally, we have performed closed-loop simulations with several controllers for the sake of comparison. In particular, we have tested the following controllers:

- PFMPC: MPC with perfect forecast.
- TBMPC: Centralized TBMPC controller.
- DTBMPCx: Distributed implementation of the TBMPC controller where x is the value of the agreement horizon. The following values were tested: $N_a = 1, 3, 5$.
- ProbMPC: MPC with a single forecast consisting on the weighted average of all the scenarios.
- MMPC: Multiple MPC, which is presented in [19].

TABLE IV

TOTAL CUMULATED COST COMPARISON

Controller	Total cumulated cost	
	Mean	Std. Dev
PFMPC	1.42e7	4.77e7
TBMPC	1.56e7	4.87e7
DTBMPC1	1.57e7	4.86e7
DTBMPC3	1.56e7	4.87e7
DTBMPC5	1.58e7	4.86e7
ProbMPC	1.77e7	5.21e7
MMPC	1.77e7	5.21e7
MinmaxMPC	1.86e7	5.46e7

- MinmaxMPC: Min-Max MPC implemented as in [3].

A total number of a 100 different simulations were carried out for the test. The disturbances used in the simulations were based on real rainfall data from the Netherlands and their severity ranges over the whole spectrum of possible scenarios. Specifically, the maximum inflow into the drainage system of each simulation was chosen between 0 and 500 m^3/s. For each one of these simulations, the drainage system was controlled in closed-loop during 25 hours by each controller. The total cumulated cost of the 25 hours was computed according to the performance index defined in (2) and the parameters given in Table I. The results can be seen in Table IV and are consistent with the expectations about the performance of the controllers. Naturally, the PFMPC outperforms the rest of the controllers with a great difference. Then it can be seen how the TBMPC and its distributed versions offer the best performance of the rest of the controllers. The probMPC and the MMPC controllers are a step behind these controllers. Finally, the minmax offers the worst performance of all the controllers tested.

IV. CONCLUSIONS

In this paper we have presented an implementation of TBMPC in a distributed fashion by means of dual decomposition. The distributed formulation allows us to apply TBMPC to problems with a higher number of optimization variables than traditional MPC controllers and sets the basis for distributed TBMPC. In addition, we have carried out experiments using a drainage water system as a benchmark in order to test how the number of iterations needed for convergence vary as different parameters change. It has been shown that a proper choice of the agreement horizon N_a and the disagreement threshold Δu_{max} can reduce the number of iterations in a significant manner, while keeping almost constant control performance. Our results suggest that the relevance of the non-anticipativity constraints is low from a control point of view. Nevertheless, the absence of these constraints makes the control signals calculated by the controller unrealistic for those time steps different from the first one. Finally, the TBMPC and its distributed versions have been compared to other controllers in the same benchmark. The results of our simulations showed that the TBMPC outperformed the rest of the alternatives considered, such as minmax MPC or MMPC, and it can be concluded that the TBMPC and its distributed versions are suitable controllers to deal with the uncertain inflows that are typical for this kind of open water systems.

REFERENCES

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] E. F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry. Second Edition.* Springer-Verlag, London, England, 2004.

[3] D.M. de la Peña, T. Alamo, D.R. Ramirez, and E.F. Camacho. Min-max model predictive control as a quadratic program. *IET Control Theory Applications*, 1(1):328 –333, January 2007.

[4] L. F. Escudero. Warsyp: a robust modeling approach for water resources system planning under uncertainty. *Annals of Operation Research*, 95:313–339, 2000.

[5] T. Gneiting and A. E Raftery. Weather forecasting with ensemble methods. *Science*, 310(5746):248, 2005.

[6] N. Growe-Kuska, H. Heitsch, and W. Romisch. Scenario reduction and scenario tree construction for power management problems. In *2003 IEEE Bologna Power Tech Conference Proceedings*, June 2003.

[7] J.M. Maestre, P. Giselsson, and A. Rantzer. Distributed receding horizon kalman filter. In *2010 49th IEEE Conference on Decision and Control (CDC)*, pages 5068 –5073, Atlanta, Dec. 2010.

[8] D. Muñoz de la Peña, A. Bemporad, and T. Alamo. Stochastic programming applied to model predictive control. In *CDC-ECC '05. 44th IEEE Conference on Decision and Control 2005 and 2005 European Control Conference*, pages 1361 – 1366, Seville, Dec. 2005.

[9] J. M. Mulvey and A. Ruszczynski. A new scenario decomposition method for large-scale stochastic optimization. *Operations Research*, 43(3):477–490, 1995.

[10] R. R. Negenborn, B. De Schutter, and H. Hellendoorn. Multi-agent model predictive control of transportation networks. In *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006)*, pages pp. 296–301, Ft. Lauderdale, Florida, April 2006.

[11] R. R. Negenborn, B. De Schutter, and H. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, April 2008.

[12] A. Rantzer. Dynamic dual decomposition for distributed control. In *American Control Conference, 2009. ACC '09*, pages 884 –888, St. Louis, MO, June 2009.

[13] L. Raso, P. J. van Overloop, and D. Schwanenberg. Decisions under uncertainty: Use of flexible model predictive control on a drainage canal system. In *Proceedings of the 9th Conference on Hydroinformatics*, Tianjin, China, 2009.

[14] R.T. Rockafellar and R.J.-B. Wets. Scenario and policy aggregation in optimization under uncertainty. *Mathematics of Operation Research*, 16(1):119–147, 1991.

[15] R. Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *Journal of Process Control*, 19:723–731, 2009.

[16] N. Z. Shor. *Minimization Methods for Nondifferentiable functions*. Springer, 1985.

[17] K. Sutiene, D. Makackas, and H. Pranevicius. Multistage k-means clustering for scenario tree construction. *Informatica*, 21(1):123–138, 2010.

[18] P. J. van Overloop. *Model Predictive Control on Open Water Systems*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2006.

[19] P. J. van overloop, S. Weijs, and S. Dijkstra. Multiple model predictive control on a drainage canal system. *Control Engineering Practice*, 16(5):531–540, 2008.

[20] W. S. Witsenhausen. A minimax control problem for sampled linear systems. *IEEE Transactions on Automatic Control*, 13(1):5–21, 1968.

[21] A. Zafra-Cabeza, J. M. Maestre, M. A. Ridao, E. F. Camacho, and L. Sánchez. A hierarchical distributed model predictive control approach in irrigation canals: A risk mitigation perspective. *Journal of Process Control*, 21(5):787–799, 2011.