Delft University of Technology

Delft Center for Systems and Control

Technical report 14-032

Gradient-based hybrid model predictive control using time instant optimization for Dutch regional water systems*

B. Dekens, A.D. Sadowska, P.J. van Overloop, D. Schwanenberg, and B. De Schutter

If you want to cite this report, please use the following reference instead:
B. Dekens, A.D. Sadowska, P.J. van Overloop, D. Schwanenberg, and B. De Schutter, "Gradient-based hybrid model predictive control using time instant optimization for Dutch regional water systems," *Proceedings of the 2014 European Control Conference*, Strasbourg, France, pp. 1343–1348, June 2014.

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

Gradient-Based Hybrid Model Predictive Control using Time Instant Optimization for Dutch regional water systems

B. Dekens[1], A.D. Sadowska[2], P.J. van Overloop[3], D. Schwanenberg[4] and B. De Schutter[2]

Abstract— **We present a novel hybrid nonlinear Model Predictive Control (MPC) algorithm for real-time control of hydraulic structures in water systems. These systems can be regarded as hybrid systems because they involve both continuous and discrete elements. The algorithm uses Time Instant Optimization for the control of discrete variables. The presented method introduces a procedure to obtain continuous time instants, allowing the derivation of a gradient of the objective function. Reverse-mode algorithmic differentiation is applied to obtain the analytical gradient of the objective function. The gradient allows the use of efficient gradient-based optimizers, making the approach suitable for real-time control applications. Furthermore, hybrid schemes including the optimization of time instants and standard MPC can be easily integrated. We illustrate an application of the algorithm to the control of five hydraulic structures in the Fivelingo water system, a water system consisting of discrete and continuous variables that is located in the North-East of Netherlands.**

I. INTRODUCTION

Water is an essential resource for life. People use water for consumption, sanitation, navigation, agriculture, hydropower and leisure. Throughout history, people have always tended to live close to water resources. However, water can also be experienced as a burden. Especially in low-lying polder systems, floods can occur. In order to manage the water levels and flows in these systems, infrastructures such as gates and pumps are installed. These hydraulic structures can be adjusted according to the objectives imposed by society, e.g. to deliver water for agriculture or to protect the land against flooding. Some of these structures are operated manually, but more and more the control has been automated. Supervisory control of these hydraulic structures is in many cases done locally by rule-based (if-then) operators [12], [4] that base their control actions on a comparison of the current state (e.g. water levels) with the desired state. The field of operational water management focuses on optimizing control actions of these structures. It does not only deal with the mitigation of extreme events, but also with complex, interdependent and often conflicting water quality and quantity objectives. Another important goal in e.g. polder systems, where surplus water has to be evacuated by pumps, is reducing the energy consumption.

Model Predictive Control (MPC) [11], [16], an anticipatory control methodology that originated in the process industry, is a control method that applies the concepts of feedback and feedforward control, while taking constraints into account. This methodology uses a model of the controlled process and forecasts of future process states and external disturbances to determine the optimal sequence of control actions by using an optimization algorithm, taking into account physical and operational constraints.

MPC has gained increasing attention in the field of water management over the last years. Some controllers have successfully been implemented in water systems [21], [13], [2], [8]. However there are some drawbacks to the methodology when discrete actuators such as pumps are present, especially concerning the computational effort. Continuous and discrete elements are not easily combined in MPC, since the discrete elements give rise to a combinatorial optimization problem.

To reduce the computational effort, Time Instant Optimization MPC (TIO-MPC) [5], [20] has been proposed to control systems with discrete-state elements. TIO-MPC involves the optimization of an (a priori determined) number of time instants, which are the moments that a discrete variable changes its state, e.g. when a pump should be switched on or off. The advantage of this method is that the computational effort is reduced because instead of deciding at each time step whether to switch the pump on or off, the decision making process involves optimizing the discrete moments at which control actions should take place. Therefore, the number of optimization variables can be significantly reduced.

However, the method described in [20] requires a multi-start pattern search optimization algorithm to converge to a solution since the discrete-time rounded equivalents of the continuous time instants have to agree with the model sampling time. The use of discrete time instants as control input rules out the use of a gradient search, since the objective function is stepwise constant. Pattern search methods are known to exhibit slower convergence compared to gradient-based methods [10].

Note that gradient-based solvers using Sequential Quadratic Programming (SQP) or Interior Point (IP) algorithms require a gradient vector of the objective function for efficient performance. These gradients can be calculated by means of finite differences, but this requires many function evaluations. As a result, gradient-based methods become computationally inefficient for problems with hundreds of dimensions, and hence disqualify them for

[1]B. Dekens is with the Department of Water Management, Witteveen+Bos Consulting Engineers, Rotterdam, The Netherlands b.dekens@witteveenbos.nl

[2]A.D. Sadowska and B. De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands {a.d.sadowska, b.deschutter}@tudelft.nl

[2]P.J. van Overloop is with the Department of Water Management, Delft University of Technology, Delft, The Netherlands p.j.a.t.m.vanoverloop@tudelft.nl

[4]D. Schwanenberg is with the Department of Operational Water Management, Deltares, Delft, The Netherlands dirk.schwanenberg@deltares.nl

being used within a real-time setting.

In this paper, we present an extension of the current TIO-MPC in the way that the time instants become continuous. This way, the gradient of the objective function can be calculated and efficient gradient-based optimizers can be used. This will decrease the computational effort of the TIO-MPC controller. We use algorithmic differentiation [9] to determine the gradients of the objective function that MPC uses. Hence, by combining these two techniques, i.e., introducing continuous time instants and supplying the solver with analytical gradients, the computational load of the TIO-MPC controller is decreased, making it suitable for a real-time application. Furthermore, hybrid schemes including the optimization of time instants and standard MPC can be easily integrated.

We test the proposed TIO-MPC algorithm with continuous time instants and supplied analytical gradient using a simulation-based case study of the Fivelingo water system. This is an existing polder system in the North-East of the Netherlands, which is drained by pumps. This model consists of both continuous and discrete variables and can thus be regarded as a hybrid system.

This paper is organized as follows. Section II discusses TIO-MPC and the dynamic model that was used. Section III proposes gradient-based TIO-MPC. Section IV demonstrates the potential of the novel approach in a case study. Section V concludes the paper.

II. Preliminaries

In this section we first discuss the dynamic model of the water system. Then we introduce the principles of the existing TIO-MPC method.

A. Process models

The flow-governing equations describing one-dimensional gradually varying non-steady flow in prismatic channels are the dynamic wave equations, also referred to as the De Saint-Venant (SV) equations or shallow-water equations [3]. The SV equations are coupled nonlinear hyperbolic partial differential equations that are derived from equations of conservation of mass (continuity) and momentum, respectively. The equations of De Saint-Venant are [3]:

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q_{\text{lat}}, \tag{1a}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(\frac{Q}{A})^2 + gA\frac{\partial h}{\partial x} + \frac{gQ|Q|}{C^2RA} = 0, \tag{1b}$$

where Q is the flow $[\text{m}^3\,\text{s}^{-1}]$, A is the average cross-sectional area of flow $[\text{m}^2]$, q_{lat} is the lateral inflow per unit length $[\text{m}^2\,\text{s}^{-1}]$, h is the water depth [m], g is the acceleration due to gravity $[\text{m}\,\text{s}^{-2}]$, C is the Chézy roughness coefficient $[\text{m}^{1/2}\,\text{s}^{-1}]$, R [m] is the hydraulic radius [m], t is the time [s] and x is the longitudinal distance [m]. Unfortunately there is no known analytical solution of the De Saint-Venant equations in real geometry [12] so the system of equations has to be solved numerically. Depending on the characteristics of the flow and the required accuracy, different one-dimensional distributed flow routing equations can be derived by using the full continuity equation (1a) while neglecting some terms of the momentum equation (1b) (see [3] and references therein). The diffusive wave model neglects the local and convective acceleration terms of the momentum equation. Some rewriting of (1b) yields:

$$Q = -\text{sign}\left(\frac{\partial h}{\partial x}\right) CA\sqrt{\left|\frac{\partial h}{\partial x}\right| R}. \tag{2}$$

The diffusive wave model can be spatially discretized on a staggered grid [19], [18], where the dependent variables (water level and discharge, respectively) are carried at alternating grid points. The model can be schematized as a system of nodes and branches. The discharge is schematized in branches between upstream and downstream storage nodes h_{up} and h_{down}. The water levels in the storage nodes are calculated from the continuity equation, while the flow in the branches that connect the storage nodes is described by the diffusive wave model (2). If we define the distance between the storage nodes as Δx, we can rewrite (2) into a function of the upstream and downstream water levels h_{up} and h_{down} by applying central differences:

$$Q^{k+1} = f(h^k_{\text{up}}, h^k_{\text{down}}) = -\text{sign}(\frac{h^k_{\text{up}} - h^k_{\text{down}}}{\Delta x}) C(\overline{h}^{\,k}) A(\overline{h}^{\,k}) \cdot \sqrt{\left|\frac{h^k_{\text{up}} - h^k_{\text{down}}}{\Delta x}\right| R(\overline{h}^{\,k})}, \tag{3}$$

where k denotes the time step and the variables C, A and R are functions of the mean water level $\overline{h}^{\,k} = \frac{h^k_{\text{up}} + h^k_{\text{down}}}{2}$ [m] in a representative cross-section between storage nodes.

Hydraulic structures between nodes can be represented by a general flow equation for a hydraulic structure, where the flow is a function of upstream and downstream water levels and a gate or weir setting d_{g}, according to:

$$Q^{k+1} = f_{\text{Q}}(h^k_{\text{up}}, h^k_{\text{down}}, d^{k+1}_{\text{g}}). \tag{4}$$

The gate setting d_{g} can usually not be written as an explicit function of the discharge and has to be solved iteratively.

By using (3), applying the (implicit) Euler Backward scheme to (3), multiplying by Δx and substituting $s(h) = A(h)\Delta x$, the continuity equation can be written as a water balance in the domain of a node [1]:

$$s^{k+1} = s^k + \Delta t(Q^{k+1}_{\text{up}} - Q^{k+1}_{\text{down}} + Q^{k+1}_{\text{lat}}), \tag{5}$$

where s^k is the storage $[\text{m}^3]$ at the node and Q^{k+1}_{lat} is the aggregated lateral inflow $[\text{m}^3\,\text{s}^{-1}]$ flowing into the domain of the node, and Q^{k+1}_{up} and Q^{k+1}_{down} are the upstream and downstream discharge at the intermediate reaches connected to the node.

B. Time Instant Optimization MPC

Water systems often comprise both continuous and discrete elements. For instance, pumps are either on or off. For these variables, an MPC algorithm would have to decide whether or not to change the state of the binary variable for every time step of the prediction horizon. For one binary input variable and a prediction horizon of N_p steps, this leads to a combinatorial optimization problem with 2^{N_p} possible solutions.[1] When there are more binary input variables and the prediction horizon is long, the complexity of the optimization problem increases rapidly and this may disqualify the method from being used in an operational setting.

Another MPC approach to deal with discrete elements in the system is Time Instant Optimization Model Predictive Control (TIO-MPC), a method that has been first been used for traffic control [5]. TIO-MPC has been applied for the control of hydraulic structures in [20], with a focus on storm surge barriers in the Dutch Rhine-Meuse delta. The TIO-MPC algorithm optimizes n time instants $t_1, \ldots, t_n$ over the prediction horizon N_p, where the time instants are the discrete time steps at which the discrete variables change their state. The rationale behind this approach is that from a practical point of view, it is often undesired to have too many on/off switching of actuators and therefore it makes sense to define a priori how many switches are allowed within the prediction horizon. The result of this approach is that the amount of control variables is reduced.

A TIO-MPC prediction model can be described as [20]:

$$\tilde{x}^k = f_\mathrm{x}(\tilde{t}^k, \tilde{u}^k, x^k), \tag{6}$$

with:

$$\tilde{x}^k = [(x^{k+1|k})^\top \quad (x^{k+2|k})^\top \quad \ldots \quad (x^{k+N_\mathrm{p}|k})^\top]^\top, \tag{7}$$

$$\tilde{u}^k = [(u^{k|k})^\top \quad (u^{k+1|k})^\top \quad \ldots \quad (u^{k+N_\mathrm{p}-1|k})^\top]^\top, \tag{8}$$

$$\tilde{t}^k = [t_1^k \quad t_2^k \quad t_3^k \quad t_4^k]^\top, \tag{9}$$

where x^k is the state at time step k, $\tilde{u}^k$ and $\tilde{x}^k$ are the input variables and state variables, respectively, and $\tilde{t}^k$ is a vector that contains the time instants that need to be optimized within the prediction horizon N_p. The notation $k+i|k$ for $i = 1, \ldots, N_\mathrm{p}$ describes the sequence of optimized variables over the prediction horizon N_p, evaluated at control step k. As can be seen in equation 9, four time instants are defined. To be able to have time instants as decision variables to be determined by solving the optimization problem, we define:

$$u^{k+i|k} = \begin{cases} u_\mathrm{max} & \text{if } i \leq k_1 \text{ or } k_2 \leq i \leq k_3 \text{ or } i \geq k_4 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

for $i = 0, \ldots, N_\mathrm{p} - 1$ and where k_1, k_2, k_3 and k_4 are discrete-time rounded equivalents of the continuous time instants.

The objective function that needs to be minimized in TIO-MPC at every time step k is, in a generalized formulation:

$$J = f_\mathrm{J}(\tilde{t}^k, \tilde{u}^k, \tilde{x}^k), \tag{11}$$

where $\tilde{x}$ and $\tilde{u}$ are functions of the decision vector $\tilde{t}$. With TIO-MPC [20], constraints prescribe the required sequence of discrete state switching (i.e. $t_1 < t_2 < t_3 < t_4$) and the minimum and maximum time between time instants. A change is implemented if t_ℓ^k, $\ell = 1, \ldots, n$, is scheduled to occur in the current sample step. Otherwise, no changes are implemented and the optimization is repeated in the next sample step in a receding horizon manner.

[1] For the sake of simplicity, the control horizon is taken to be equal to the prediction horizon in this paper.

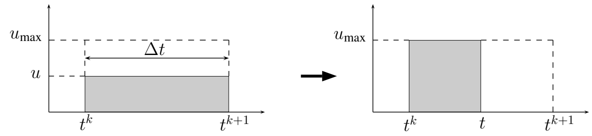

Fig. 1: Towards continuous Time Instant Optimization

Given the existence of discrete-time equivalents of the actual continuous-time instants, the MPC formulation in [20] ceased to be smooth. Thus, the objective function is minimized using a derivative-free optimizer such as a pattern search algorithm. A gradient-based optimizer cannot be used efficiently because the time instants are in fact discrete values, causing a stepwise objective function. In this work, we report on a way to tackle this problem of continuous time instants, for a discrete-time system, to improve computational efficiency of the controller.

III. Gradient-based TIO-MPC

In this section, we introduce a gradient-based implementation of TIO-MPC for the application in a water system. We first discuss the concept of TIO-MPC with continuous time instants. Afterwards, we elaborate on the numerical implementation of the proposed scheme.

A. Theoretical analysis

The contribution of the work presented here is the extension of the standard TIO-MPC in such a way that the time instants become continuous. In this way, the gradient of the objective function can be derived. The gradient allows the use of efficient gradient-based solvers. Furthermore, hybrid schemes including the optimization of time instants and standard MPC can be easily integrated.

The derivation of continuous time instants is based on a mass balance between two discrete time steps. In Fig. 1, the translation from discrete to continuous time instants is shown. On the left, the control input (the discharge of a pump) can be freely chosen between 0 and u_max on the discrete interval $[t^k, t^{k+1}]$. On the right, the pump is either off or operating with discharge u_max on the interval $[t^k, t]$ where t is (in this case) the continuous time instant that the pump is switched off. The water balance on the interval $[t^k, t^{k+1}]$ is equal for both cases, so we can write:

$$u \cdot (t^{k+1} - t^k) = u_\mathrm{max} \cdot (t - t^k), \tag{12}$$

or:

$$u = \frac{u_\mathrm{max}}{t^{k+1} - t^k} \cdot (t - t^k). \tag{13}$$

By applying the chain rule we obtain:

$$\frac{\mathrm{d}J}{\mathrm{d}t} = \frac{\mathrm{d}J}{\mathrm{d}u}\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\mathrm{d}J}{\mathrm{d}u}\frac{u_\mathrm{max}}{t^{k+1} - t^k} \tag{14}$$

and, since $\Delta t = t^{k+1} - t^k$, this results in:

$$\frac{\mathrm{d}J}{\mathrm{d}t} = \frac{\mathrm{d}J}{\mathrm{d}u}\frac{u_{\text{max}}}{\Delta t}. \tag{15}$$

With MPC, the model calculations are only carried out at the discrete time steps k. To account for this in TIO-MPC, we need to perform a transformation of the control vector u similar to the procedure in Fig. 1 in order to be able to optimize continuous time instants as control input. This is done by defining the following: let t_1 and t_2 be the continuous time instants at which the pumps are switched on and off respectively, in different intervals $[a, b]$ and $[c, d]$ where the intervals $[a, b]$ and $[c, d]$ satisfy $[a, b] \cap [c, d] = \emptyset$ and they denote particular sampling intervals when t_1 and t_2 occur, respectively. To compensate for the pump being switched on or off in the middle of a sampling time, which is not reflected in the discrete-time model of the system, the pump is assumed to provide a discharge of a part of its maximum capacity. This way, the overall discharge remains the same for the discrete-time model and the actual system with continuous time instants. Consequently, the value of the flow of a pump within the corresponding sampling interval is:

$$u_{\text{ab}} = \frac{b - t_1}{b - a}\, u_{\text{max}} \tag{16}$$

for t_1, and

$$u_{\text{cd}} = \frac{t_2 - c}{d - c}\, u_{\text{max}} \tag{17}$$

for t_2 and $u_{\text{bc}} = u_{\text{max}}$. When $t_1, t_2 \in [a, b]$, i.e. they occur in the same interval, the control corresponds to:

$$u_{\text{ab}} = \frac{t_2 - t_1}{b - a}\, u_{\text{max}}. \tag{18}$$

The transformed discharge profile corresponds to the situation where $u(t) = u_{\text{max}}$ for $t_1 \leq t \leq t_2$, and 0 otherwise. By using this control vector, the solver is able to optimize continuous time instants. The Jacobian or gradient vector can now be derived, which allows the use of efficient gradient-based solvers. Note that the real system still can only pump at maximum discharge or not pump at all. Therefore, once the optimal continuous time instants have been calculated, the actuator switches its state at discrete time instants obtained through rounding the continuous optimized time instants.

B. *Numerical implementation*

The process model and the objective function have been configured in RTC-Tools [1], an open source software package developed by the research institute Deltares for modeling flow routing processes in real-time control applications. RTC-Tools applies reverse-mode algorithmic differentiation [9], [14] to evaluate an adjoint model, resulting in the analytical gradient of the objective function with respect to the continuous input variables. The model is linked to an external solver to obtain the gradient with respect to the time instants, hence allowing the optimization of time instants as control inputs. We used Matlab's constrained nonlinear solver fmincon, using the interior-point algorithm. As an initial guess, the time instants were divided equidistantly over the prediction horizon N_{p}. Using zeros as an initial guess, the solver is likely to get stuck in (suboptimal) local minima [6].

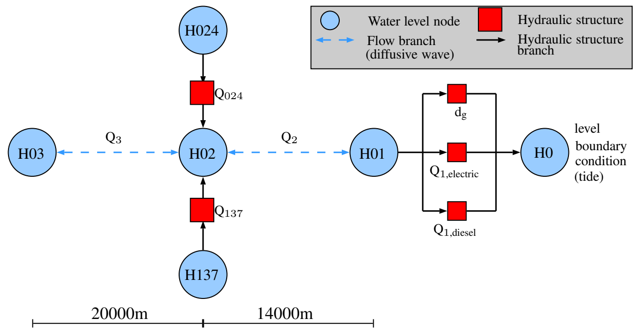

Fig. 2: Schematic overview of the Fivelingo water system. The two dashed arrows together represent the Damsterdiep canal

IV. Case study

The Fivelingo water system is located in the North-East of the Netherlands and is managed by the regional water board Noorderzijlvest. It is a low-lying polder system, where pumps discharge water from secondary canals into a so-called belt canal. This man-made canal, also known as the Damsterdiep, is about 25 kilometers long and connects the city of Groningen with the sea at the harbor of Delfzijl. There is a pumping station at the outlet, consisting of an electric and a diesel pump. During low tide, water can be released through an undershot gate.

A. *Process model*

The coarse nonlinear MPC model we present here is an extension of the model used in [17]. The model consists of storage nodes and branches with a spatial discretization on a staggered grid, as described in Section II. The Fivelingo water system consists of three storage nodes, in order to take into account water level gradients between the central node (representing the average water level in the Damsterdiep canal) and the storage node upstream of the outlet to the sea at the harbor of Delfzijl, representing the tide. The level-storage relations of the storage nodes are derived from a detailed SOBEK hydrodynamic model [7] by aggregating the available storage from the primary canals. See [6] and references therein for a full description of the model used.

In the work presented here, the model is extended with two polder canals that discharge water into the Damsterdiep canal using pumps that are operated either on or off, making it a hybrid system. In the MPC model, this was modeled by adding two storage nodes and two pumps. Time Instant Optimization MPC is applied for these two pumps. A schematization of the resulting model is given in Fig. 2.

Since the water system is a typical polder-belt canal system, the main goal of the controller is to minimize the water level deviations from setpoint in the belt and polder canals, while avoiding potential flooding events by anticipating on expected disturbances. A second goal, which applies mainly to day-to-day operations, is to achieve energy and cost savings on pumping by:

1) Making efficient use of the available storage in the system.
2) Creating storage in the system by pre-releasing water prior to an expected disturbance that coincides with an 'expensive' pumping period.
3) Releasing as much of the water surplus as possible through the gate at low tide.

A third goal is to minimize wear and tear of hydraulic structures by limiting the changes in pump flow and gate height. All goals can be described mathematically in the following objective function that needs to be minimized:

$$\begin{aligned}\min_{\tilde{t}_{\mathrm{n}},\mathrm{Q},\Delta\mathrm{d}_g} \quad J^k = & \sum_{i=1}^{N_\mathrm{p}}\sum_{j=1}^{m} W_{\mathrm{e},j}\ (e_j^{k+i|k})^2 \\ & + \sum_{j=1}^{m} W_{\mathrm{e}_{N_\mathrm{p}},j}\ (e_j^{k+N_\mathrm{p}|k})^2 \\ & + \sum_{i=1}^{N_\mathrm{p}}\sum_{j=1}^{l} W_{\Delta\mathrm{Q},j}\ (\Delta Q_j^{k+i|k})^2 \\ & + \sum_{i=1}^{N_\mathrm{p}}\sum_{j=1}^{l} W_{\mathrm{Q},j}\ Q_j^{k+i|k} \\ & + \sum_{i=1}^{N_\mathrm{p}}\sum_{j=1}^{p} W_{\Delta\mathrm{dg},j}\ (\Delta d_{\mathrm{g},j}^{k+i|k})^2,\end{aligned} \tag{19}$$

where $Q_j^{k+i|k}$ is the pump flow $[\mathrm{m}^3\,\mathrm{s}^{-1}]$ that follows from the time instants, $\tilde{t}_\mathrm{n} = (t_1, \ldots, t_\mathrm{n})$ is the vector of time instants, where n denotes the number of time instants used, $e_j^{k+i|k}$ is the water level deviation from the setpoint [m], $\Delta Q_j^{k+i|k}$ is the change in pump flow $[\mathrm{m}^3\,\mathrm{s}^{-1}]$, $\Delta d_{\mathrm{g},j}^{k+i|k}$ is the change in gate setting [m], m the number of flow branches between storage nodes $[-]$, l is the number of pumps $[-]$, p is the number of gates $[-]$, $W_{\mathrm{e},j}$ is the penalty on the water level deviation from setpoint for a storage node, $e_j^{k+N_\mathrm{p}|k}$ is the water level error [m] at the end of the prediction horizon, $W_{\mathrm{e}_{N_\mathrm{p}},j}$ is the penalty on this error $[-]$, $W_{\mathrm{Q},j}$ is the penalty on the change in pump flow $[-]$, $W_{\mathrm{Q},j}$ is a penalty on pump flow and $W_{\Delta\mathrm{dg},j}$ is a penalty on the change of the gate height setting $[-]$. The objective function is subject to physical and operational constraints, which limit the (change in) pump flow and gate settings.

The use of a nonlinear process model in the TIO-MPC problem leads to a loss of convexity [15]. For this reason it may be harder to find a (sufficiently good) solution, and if a solution is found it is not guaranteed to be the global optimum. Therefore, the gradient-based solver uses a multi-start optimization algorithm.

B. Control setup

The prediction model uses historical data of the tide and lateral inflow over the period February 17-27, 2012. With 48-hour time series and a control time step of 5 minutes, this results in prediction horizon $N_\mathrm{p} = 576$. With only continuous optimization variables, this would lead to an optimization problem with 2880 variables. However, since we use TIO-MPC the amount of optimization variables is reduced.

The objective function value, root mean square error (RMSE) of the predicted water levels and CPU time serve as performance indicators. The computation time depends on the specific computer that is being used for the optimization, see Table I. In this article, only the results of one open-loop experiment is shown. For additional results, the reader is referred to [6].

TABLE I: Specifications of computer and software used for simulations

Processor	Intel Core i3 350M @ 2.27 GHz
Memory	4.00 GB DDR3
Operating system	Windows 7 Home Premium SP1 64-bit
Matlab version	R2013a 32-bit
Solver	`fmincon`, using Interior-Point algorithm

TABLE II: Simulation results

n_total $[-]$	n_TIO $[-]$	J $[-]$	RMSE_{137} [m]	CPU time [s]
1732	2	1994.5	0.1297	76.7
1736	4	742.2	0.0773	106.1
1744	8	367.2	0.0471	35.1
1760	16	190.0	0.0241	57.6

C. Results

The novel continuous TIO-MPC algorithm is able to adequately optimize both continuous and discrete control input. The results of experiments using different numbers of time instants are summarized in Table II, where n_total is the total amount of (continuous and discrete) variables, n_TIO is the number of time instants used per discrete variable, J is the objective function value and RMSE_{137} is the root mean square error of the water level in node H137.

It was observed that the CPU time is not necessarily higher with increasing numbers of time instants. This can be caused by non-convexity of the optimization problem, causing the solver to end up in a local minimum based on the supplied initial value. Fig. 3 shows the results of an experiment with 8 time instants for the discrete control input, where the time instants have been divided equidistantly over the prediction horizon as an initial guess.

It can be observed from Fig. 3 that the controller exploits the gravity flow through the gates during low tide. For each discrete variable, eight optimized time instants translate into four pumping intervals. Using these intervals, the deviation from setpoint is minimized over the prediction horizon N_p.

The experiments show that using more time instants generally leads to better performance in terms of the objective function value. This makes sense because there are more degrees of freedom. Using a multi-start optimization and supplying the solver with an educated guess of an initial point for the optimization may prevent the solver from getting "stuck" in the first (suboptimal) local minimum it encounters.

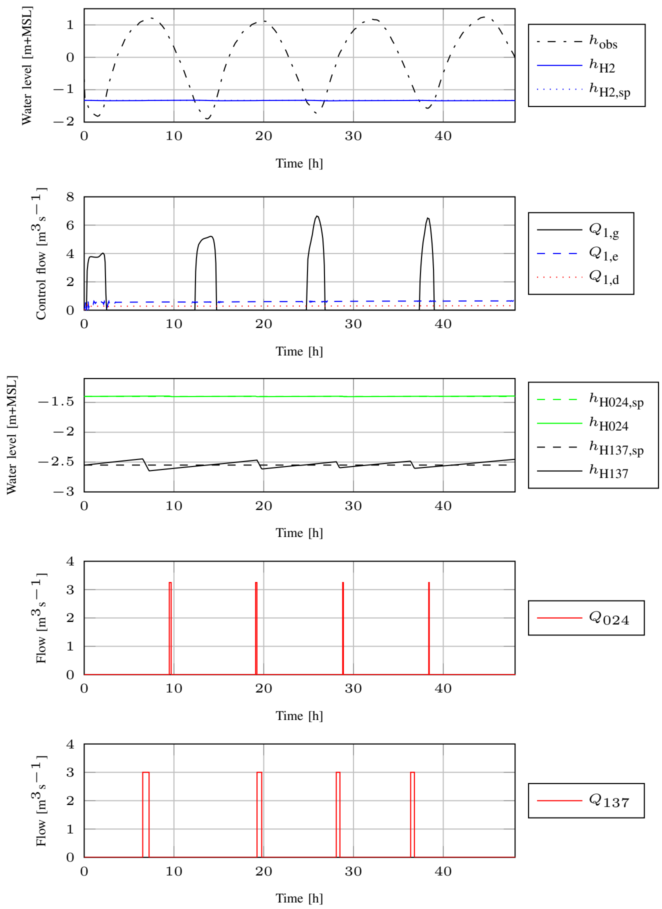

Fig. 3: Open loop simulation results of the gradient-based TIO-MPC experiment using 8 time instants, where h_{obs} is the observed water level at sea, h and Q are the water levels and flows in various nodes, $Q_{1,\mathrm{g}}$, $Q_{1,\mathrm{e}}$ and $Q_{1,\mathrm{d}}$ represent the flows at the outlet from the gate, electric pump and diesel pump, respectively. The subscript sp denotes a setpoint.

It was observed that a better solution in terms of the objective function J, RMSE and CPU time is found in the experiments where an initial guess has been supplied.

V. CONCLUSIONS

In this paper, we have derived a method to employ continuous time instants for a discrete-time model. The continuous time instants allow the use of efficient gradient-based solvers. The potential of the proposed TIO-MPC algorithm with continuous time instants and supplied analytical gradient has been illustrated using a simulation-based case study of the Fivelingo water system. The approach is very promising regarding computational efforts, making it suitable for real-time control applications in hybrid systems.

Future work will focus on closed-loop simulations to assess the effects that the tuning parameters such as N_{p}, N_{c}, sampling time and weighting gains have on the performance. Also, performance comparison with alternative approaches will be done in the future.

REFERENCES

[1] B. Becker and D. Schwanenberg, *RTC-Tools: Software Tools for Modeling Real-Time Control. Reference Manual*, Deltares, Delft, the Netherlands, December 2012.

[2] T. B. Blanco, P. Willems, P.-K. Chiang, N. Haverbeke, J. Berlamont, and B. D. Moor, “Flood regulation using nonlinear model predictive control,” *Control Engineering Practice*, vol. 18, no. 10, pp. 1147 – 1157, 2010.

[3] V. T. Chow, D. R. Maidment, and L. W. Mays, *Applied Hydrology*, ser. McGraw-Hill series in Water Resources and Environmental Engineering. McGraw-Hill Book Company, 1988.

[4] A. Clemmens and J. Schuurmans, “Simple optimal downstream feedback canal controllers: Theory,” *Journal of Irrigation and Drainage Engineering*, vol. 130, no. 1, pp. 26–34, 2004.

[5] B. De Schutter and B. De Moor, “Optimal traffic light control for a single intersection,” *European Journal of Control*, vol. 4, no. 3, pp. 260–276, 1998.

[6] B. Dekens, “Gradient-based hybrid model predictive control using time instant optimization for Dutch regional water systems,” Master’s thesis, Delft University of Technology, Delft, the Netherlands, 2013.

[7] Deltares, *SOBEK Technical Reference Manual*, Delft, the Netherlands, 2013.

[8] G. Glanzmann, M. von Siebenthal, T. Geyer, G. Papafotiou, and M. Morari, “Supervisory water level control for cascaded river power plants,” in *Proceedings of the Hydropower Conference, Stavanger, Norway*, 2005.

[9] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed. Society for Industrial and Applied Mathematics (SIAM), 2008.

[10] T. G. Kolda, R. M. Lewis, and V. Torczon, “Optimization by direct search: New perspectives on some classical and modern methods,” *SIAM review*, vol. 45, no. 3, pp. 385–482, 2003.

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

[12] P.-O. Malaterre and J.-P. Baume, “Modeling and regulation of irrigation canals: existing applications and ongoing researches,” in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, 1998, pp. 3850–3855.

[13] 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, 2009.

[14] B. A. Pearlmutter and J. M. Siskind, “Reverse-mode AD in a functional framework: Lambda the ultimate backpropagator,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 30, no. 2, p. 7, 2008.

[15] S. J. Qin and T. A. Badgwell, “An overview of nonlinear model predictive control applications,” in *Nonlinear model predictive control*. Springer, 2000, pp. 369–392.

[16] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Madison, Wisconsin: Nob Hill Publishing, 2009.

[17] D. Schwanenberg and K.-J. van Heeringen, “2011.06.08 Pilot RTC Noorderzijlvest,” Flood Control 2015, Tech. Rep., February 2012.

[18] D. Schwanenberg, S. Galelli, and I. Sheret, “Nonlinear model predictive control for heterogeneous process models in water resources,” in *Proceedings of the 18th IFAC World Congress*, vol. 18, no. 1, 2011, pp. 10 565–10 570.

[19] G. S. Stelling and S. P. A. Duinmeijer, “A staggered conservative scheme for every Froude number in rapidly varied shallow water flows,” *International Journal for Numerical Methods in Fluids*, vol. 43, no. 12, pp. 1329–1354, 2003.

[20] H. van Ekeren, R. Negenborn, P. van Overloop, and B. De Schutter, “Time-instant optimization for hybrid model predictive control of the Rhine-Meuse Delta,” *Journal of Hydroinformatics*, vol. 15, no. 2, pp. 271–292, 2013.

[21] P. J. van Overloop, “Model predictive control on open water systems,” Ph.D. dissertation, Delft University of Technology, Delft, the Netherlands, 2006.