

Continuous nonlinear model predictive control of a hybrid water system

Eelco Nederkoorn, Jan Schuurmans, Joep Grispen
and Wytze Schuurmans

ABSTRACT

Incorporating weather forecasts in the control of land surface water levels requires predictions of the net inflow to the water system. This net inflow is the combined flow of an incoming load (rain, evaporation, etc.) and outgoing pump rates. Because the pump costs are considerable, optimal pump schedules have minimal energy consumption. Model predictive control (MPC) is able to compute, revise and apply such optimized schedules by incorporating a model of the water system. The pumps typically cause discontinuities in the model, which leads to mathematical complications. Avoiding advanced solving techniques for these hybrid systems, this paper introduces an alternative that enables pure continuous MPC by smoothing the jumps. Although the resulting underlying model is continuous, it is also highly nonlinear. This requires use of the specialized class of nonlinear model predictive control (NMPC), which is able to cope with the arising nonlinearities. Control inputs computed by these methods can be translated to the original hybrid system by a final post-processing step. This paper presents the outlined scheme, and verifies it by applying an optimized NMPC implementation (the DotX nonlinear predictive controller, DNPC), equipped with the approximated continuous nonlinear model, to a real-life hybrid water system.

Key words | hybrid systems, nonlinear model predictive control, online pump scheduling, water management, weather predictions

Eelco Nederkoorn (corresponding author)
Jan Schuurmans

DotX Control Solutions,
James Wattstraat 23,
1817 DC Alkmaar,
The Netherlands
E-mail: e.nederkoorn@dotxcontrol.com

Joep Grispen

Wytze Schuurmans
Nelen & Schuurmans,
Zakkendragershof 34,
3500 BE Utrecht,
The Netherlands

INTRODUCTION

Controlling land surface water levels is essential for avoiding floods and droughts. The Netherlands, with half of its land lower than 1 m above sea level, strives for a balanced surface water level by using an extensive system of pumps and maneuverable dams. This work focuses on controlling the water level in a specific region (a Dutch polder) in the north-west of the country, called the *Waterlandse Boezem*. This area is equipped with a set of pumps, all with their own characteristic capacity, which are used to deal with excesses of rain water.

Recently, new initiatives have emerged, like Delft-FEWS and the controlNEXT project (Roos *et al.* 2010), which facilitate data exchange between different hardware systems and data pools such as weather predictions.

Control systems responsible for activating the pumps now actually have access to such data. Due to advances in meteorological modeling, increasing interest is now given to the incorporation of weather forecasts into pump control decisions. For example, when a storm is imminent, it is beneficial to start draining in advance to avoid floods. Similarly, in the case of an upcoming drought, it is desirable to halt pumping, even during momentary rainfall. To achieve such planning objectives, the controller needs to predict and incorporate the net inflow to the system given a specific weather prediction. Classical feedback controllers do not have this capacity and are inadequate to achieve these objectives. The more advanced class of model predictive control (MPC) methods typically predict

water levels based on a continuous linear model of the system (Jowitt & Germanopoulos 1994; Ormsbee & Lansley 1994; Yu *et al.* 1994; McCormick & Powell 2003). Based on these online model predictions, MPC methods are able to compute, revise and apply optimized pump schedules.

Increasing the complexity of the mentioned planning issue, the pump's practical limitations impose constraints on the schedules, which the controller must obey.

Actuator constraints

The pumps at the Waterlandse Boezem have a few characteristics that are very common in pump scheduling problems. They can, in generalized form, be summarized by the following:

- A1. There are a discrete number of pumps (say N), each having a different capacity (in m^3/s).
- A2. Pumps need to be activated in a predetermined order, i.e. when pumps are ordered with index $k = 1, \dots, N$, then pump k can only be turned on when pump $k - 1$ is active.
- A3. Each pump needs to be turned on or off for a time interval lasting at least D (for instance, $D = 4$ hr).

The total pump flow over time, defined by the sum of the rates of all individual activated pumps, becomes inherently discontinuous by these three restrictions. A pump either pumps at zero or full capacity, and, when turned on, makes this transition instantly. Hence, a model capturing these dynamics exhibits both continuous and discontinuous behavior and is often referred to as a hybrid system.

Besides these constraints, the controller needs to have clearly defined control objectives, so it knows which behavior to strive for.

Control objectives

At the Waterlandse Boezem these objectives, which are typical for hybrid water systems in general, are the following.

- B1. The water level needs to stay between predetermined boundaries.

- B2. The energy consumption, caused by active pumps, needs to be minimal.
- B3. Pumping is preferred during low-cost energy hours (typically at night). However, this should not cause the water level to flow out of its bounds.

Note that these sub-objectives are possibly conflicting: pumping is needed after a heavy downpour to avoid a flood; however, the energy consumption will increase when pumps are turned on. By careful selection of key parameters, MPC can be tuned such that it weighs the importance of each goal and strives for balanced pump schedules.

The resulting optimization problem in MPC, defined by minimizing a quantification of the aforementioned objectives in terms of the hybrid system, is typically posed as a mixed integer program (Van Overloop *et al.* 2010; Van Ekeren *et al.* 2011). Solving these problems requires advanced techniques (De Schutter & van den Boom 2003; Axehill & Hansson 2007; Axehill 2008). This paper introduces an alternative for such methods by smoothing the jumps and applying standard continuous MPC to the resulting continuous system. Such approximations yield nonlinear terms in the model equations, which implies a specialized subclass of continuous nonlinear model predictive control (NMPC) (Kouvaritakis *et al.* 1999; Tenny *et al.* 2004; Bacic *et al.* 2005) is needed. Recent developments have given NMPC guaranteed robustness and stability (Findeisen & Allgöwer 2002; Findeisen *et al.* 2003a, 2003b), which are essential characteristics for feedback controllers.

This paper presents how the constraints (A1–A3) and objectives (B1–B3) can be translated to an MPC framework (based on the approximated smooth model). In order to illustrate the effectiveness of the presented scheme, we have utilized a commercially developed NMPC implementation, called the DotX nonlinear predictive controller (DNPC) (Schuermans 2009a, 2009b; Nederkoorn *et al.* 2011a, 2011b), to generate pump schedules for the Waterlandse Boezem in a closed-loop simulation environment. The results of the simulations indicate that the outlined scheme can yield solutions to the pump scheduling problem satisfying the constraints and objectives. Moreover, in this simulation environment, there are potential advantages

over other (classical) control methods in terms of energy use and water-level control.

Outline

This paper first introduces a continuous, nonlinear reservoir model of the hybrid system. This system can be used as the underlying model for NMPC, as explained in the section on applying nonlinear MPC. The outlined strategy is evaluated by numerical simulations of the Waterlandse Boezem in the section on numerical results.

NONLINEAR RESERVOIR MODEL

The rate at which water is added or drained from a water system on a time interval $[0, T]$ (by, for instance, rainfall or evaporation) is called the load prediction $Q_l(t)$ (in m^3/s) and is a key forecast parameter in water management. Let the surface water level itself (in meters) be defined by $w(t)$. This quantity is usually measured with respect to some standardized level (the Nederlands Algemeen Peil (NAP) is the standard level at the Waterlandse Boezem). The rate at which the pumps drain water from the system at time $t \in [0, T]$ is defined by $Q_p(t)$ (in m^3/s). Given the total area of the water surface A (in m^2), consider the following reservoir model of the water system:

$$\dot{w}(t) = \frac{1}{A} (Q_l(t) - Q_p(t)), \quad w(0) = w_0 \tag{1}$$

where w_0 is a given measurement of the water level at $t = 0$. Let a pump have a fixed capacity c (in m^3/s). Recall that a pump either operates at zero or full capacity, which makes $Q_p(t)$ a discontinuous function with jumps when a pump is (de)activated. Therefore, even though the underlying model for $w(t)$ in Equation (1) is continuous, the dynamics of the right-hand side function are inherently discontinuous. As mentioned, continuous MPC methods fail to cope with such hybrid models.

In order to circumvent these obstacles, this work splits $Q_p(t)$ into smooth building blocks. A properly chosen combination of these blocks, or ‘pump actions’, yields a continuous approximation of the pump demand. Because

the discontinuities no longer exist in the approximated model, it can be used for standard continuous MPC.

Pump action

The pump demand caused by turning a pump on and off on $[0, T]$ is called a pump action. Note that pumps can have multiple pump actions on the specified interval. Let $s \in [0, T]$ be the moment the pump action is initiated, and let $e \in [0, T]$ be the moment the pumping stops. Recall that a pump either pumps at zero or full capacity c , which yields a discontinuity in the model that needs to be addressed. Consider function $F_c: \mathbb{R} \rightarrow [0, c]$, defined by:

$$F_c(p) \equiv \begin{cases} 0 & p \leq 0 \\ \frac{2c}{L^2}p^2 & p \leq \frac{1}{2}L \\ -\frac{2c}{L^2}p^2 + \frac{4c}{L}p - c & p \leq L \\ c & p > L \end{cases} \tag{2}$$

for some fixed $L \in \mathbb{R}^+$. Figure 1 shows this function for fixed L and c . Note how this function enables a smooth transition from 0 (when $p \leq 0$) to c (when $p \geq L$). Given a value of $p \in \mathbb{R}$, let the rate of the pump action be defined by $F_c(p)$. This rate smoothly transitions from zero to full capacity when p is increased from a negative value to a value bigger than L . Hence, the variable p enables a continuous switch for the pump flow.

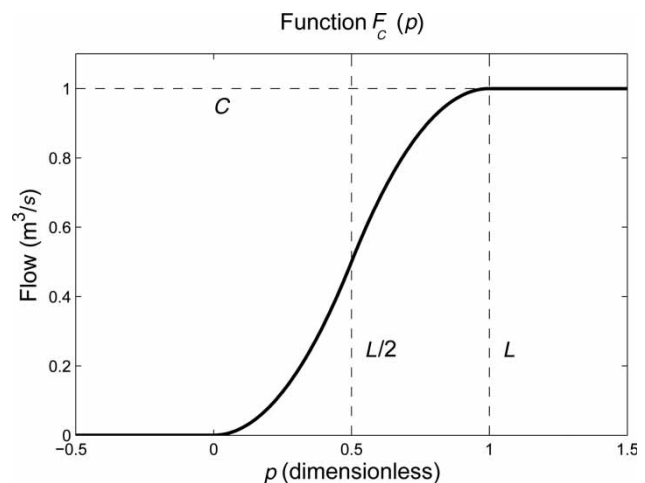


Figure 1 | Function $F_c(p)$ depends on the dimensionless parameter $p \in \mathbb{R}$ and realizes a smooth transition between 0 and c capacity. In this example $L = 1, c = 1$.

Combining these variables and functions: $F_c(p)$ determines the rate of the pump, which is turned on at time s (hr) and turned off at time e (hr). Note that if $p > 0$, and therefore $F_c(p) > 0$, the pump flow still contains discontinuities in time at s and e . Though these jumps do not cause a jump in the modeled water level (because of the continuity of the differential Equation (1)), they do limit the applicability of MPC, which often needs continuous Jacobians of all the functions in the model equations. Hence, in a second smoothing step, these discontinuities at s and e need to be removed.

Consider the following smooth approximation of the Heaviside step function:

$$\Phi_\alpha(t) \equiv \frac{1}{1 + \exp(-2\alpha t)} \tag{3}$$

with parameter $\alpha > 0$. Applying this function to smooth the jumps at s and e and combining this with the pump rate $F_c(p)$ yields the following pump rate for a pump action:

$$q_\alpha(s, e, p, c, t) = F_c(p)\Phi_\alpha(t - s)\Phi_\alpha(-t + e) \tag{4}$$

Expressing the pump flow in such a way allows both the jumps between pumping at zero or full capacity (modeled by $F_c(p)$), and the jumps at times s and e (modeled by $\Phi_\alpha(t - s)$ and $\Phi_\alpha(-t + e)$), to be approximated continuously. Figure 2 illustrates such a smooth pump action.

The next step in forming a continuous approximation of $Q_p(t)$ is combining several of these smooth building blocks.

Combining pump actions

Let a pump schedule contain M pump actions, distributed over N individual pumps. A pump action k is defined by the set of variables (s_k, e_k, p_k) and given fixed capacity c_k . The entire schedule can be written by the triplet $\mathbf{s} = (s_1, \dots, s_M)$, $\mathbf{e} = (e_1, \dots, e_M)$, $\mathbf{p} = (p_1, \dots, p_M)$ and given fixed $\mathbf{c} = (c_1, \dots, c_M)$. The combined pump rate at time t is defined by:

$$Q_p^\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}, \mathbf{c}, t) = \sum_{k=1}^M q_\alpha(s_k, e_k, p_k, c_k, t) \tag{5}$$

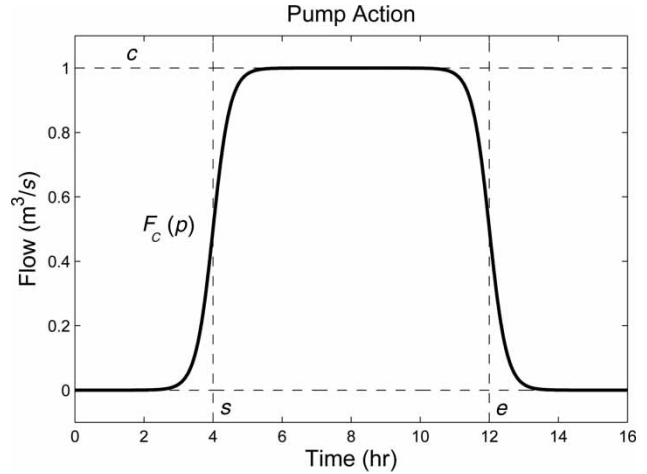


Figure 2 | Example of a pump action, with variables $s = 4$, $e = 12$, $p = L$ and parameters $L = 1$, $c = 1$. Note how the pump flow is a continuous function in time. Since $p = L$, we have $F_c(p) = c$ by construction of the pump flow function in Equation (2).

For simplicity, consider the case $M = N$, where each pump has its own unique pump action on interval $[0, T]$. In later sections, the outlined procedure will be extended to the situation $M > N$.

In order for a pump schedule $(\mathbf{s}, \mathbf{e}, \mathbf{p})$ to match all requirements (A1–A3), some constraints need to be imposed on the schedule. Recall that a pump needs to be turned on or off for at least D hours (A3). This implies that the following must hold:

$$e_k \geq s_k + D \tag{6a}$$

for $k = 1, \dots, M$. Moreover, pumps need to be turned on in a fixed order (A2): pump two can only be active if pump one is turned on, pump three only when pump two is active, etc. Translating this requirement to restrictions on \mathbf{s} , \mathbf{e} and \mathbf{p} is threefold. Firstly, a pump $k + 1$ can only be activated if pump k is active:

$$s_{k+1} \geq s_k \tag{6b}$$

for $1 \leq k < M$. In a similar fashion, pump $k + 1$ can only be deactivated if pump k is still pumping:

$$e_k \geq e_{k+1} \tag{6c}$$

for $1 \leq k < M$. Though this ensures that all pump actions occur simultaneously, it does not guarantee that all pumps are

activated in order. In addition to the previous constraints, set:

$$p_k \geq p_{k+1} + L \quad (6d)$$

for $1 \leq k < M$. Note that $F_{c_{k+1}}(p_{k+1}) > 0$ can occur if and only if $F_{c_k}(p_k) = c_k$ (Equation (2)). This implies that no pump can be turned on if its predecessor is not, which is exactly the required behavior.

Let $(\mathbf{s}, \mathbf{e}, \mathbf{p})$ be vectors obeying the constraints in Equations (6). By the limit of the Heaviside approximation of Equation (3), the corresponding discrete combined pump flow satisfies:

$$Q_p(t) = \lim_{\alpha \rightarrow \infty} Q_p^\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}, \mathbf{c}, t)$$

This suggests that the continuous approximation of the pump flow is suitable for MPC when α is chosen large enough.

Resulting model

Substituting $Q_p^\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}, \mathbf{c}, t)$ in the original model in Equation (1) leads to the following system on $[0, T]$:

$$\dot{w}(t) = \frac{1}{A} (Q_l(t) - Q_p^\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}, \mathbf{c}, t)), \quad w(0) = w_0 \quad (7)$$

Given variables $(\mathbf{s}, \mathbf{e}, \mathbf{p})$, let $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ be the solution to (7). Though the differential equation is linear (there are no cross terms of \dot{w} and w), the solution $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ is not linearly dependent on the input parametrization $(\mathbf{s}, \mathbf{e}, \mathbf{p})$. Hence, by smoothing the jumps in $Q_p(t)$, we have sacrificed the linearity of the system. Controlling the model prediction $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ of a pump schedule $(\mathbf{s}, \mathbf{e}, \mathbf{p})$ requires the use of a specialized class of NMPC, which is, as the name suggests, able to deal with these nonlinearities.

APPLYING NONLINEAR MPC

In the MPC framework, we can restate the pump scheduling problem as: given a load prediction $Q_l(t)$ on $[0, T]$ and a measured water level w_0 at $t = 0$, find $(\mathbf{s}, \mathbf{e}, \mathbf{p})$, obeying constraints (6), such that the corresponding solution $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ of Equation (7) satisfies the control objectives (B1–B3) on

$[0, T]$. In NMPC, these goals are quantified in an objective function, often referred to as the cost function.

Cost function

This function $J_\alpha: (\mathbf{s}, \mathbf{e}, \mathbf{p}) \rightarrow \mathbb{R}^+$ assigns high values to unwanted pump demands and low values to schedules satisfying the control objectives. Therefore, the cost function is the objective for the overall optimal control problem:

$$\min_{\mathbf{s}, \mathbf{e}, \mathbf{p} \in \mathbb{R}^M} J_\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}) \quad (8)$$

subject to constraints (6). The cost function has one non-negative output value; however, there are multiple optimality requirements (B1–B3). This implies $J_\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p})$ must make a trade-off between the different, possibly conflicting, optimization goals.

Recall the first objective: keeping the water level between predetermined boundaries (B1). In terms of the model:

$$w^- \leq w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t) \leq w^+ \quad (9)$$

for $t \in [0, T]$, where w^- is the lower and w^+ is the upper boundary. A well-designed cost function has a large value when $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ does not satisfy (9), and a low value (preferably zero) when it does. Let $R: \mathbb{R} \rightarrow \mathbb{R}^+$ be a smooth function upholding this property (an example is illustrated in Figure 3), then $R(w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t))$ for $t \in [0, T]$ is a suitable quantification of this control objective.

The cost function also needs to cover the energy consumption (B2). Assuming there is a direct relationship between the pumping rate and energy usage, then function $Q_p^\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}, \mathbf{c}, t)$, defined in Equation (5), is an ideal cost measure for energy consumption. Additionally, in an optimized schedule, most of the pump actions are allocated at low-cost energy hours (B3). Let there be K energy peak intervals on $[0, T]$ (typically daytime hours), defined by $[t_j^s, t_j^e]$ (the intervals start and end time in hours) for $j = 1, \dots, K$. Consider the weighting function:

$$W^\alpha(t) \equiv \sum_{j=1}^K \Phi_\alpha(t - t_j^s) \Phi_\alpha(-t + t_j^e)$$

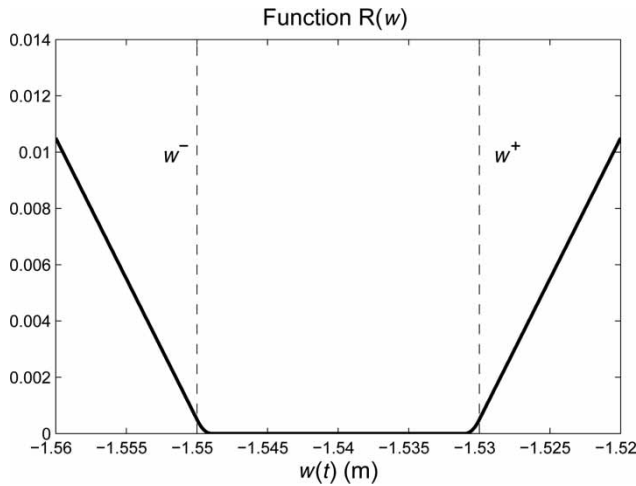


Figure 3 | An example of a smooth function $R(w(t))$ with $w^- = -1.55$ m and $w^+ = -1.53$ m. Note that R is zero almost everywhere when $w^- \leq w(t) \leq w^+$, and becomes large rapidly when $w(t)$ approaches or passes through the boundaries.

which is 1 during peak hours and 0 almost everywhere else. Note that α corresponds to the same smoothing parameter chosen earlier in this work. Figure 4 shows this function on a typical prediction window. A suitable quantification of the energy usage requirements B2 and B3 is the product between W^α and Q_p^α .

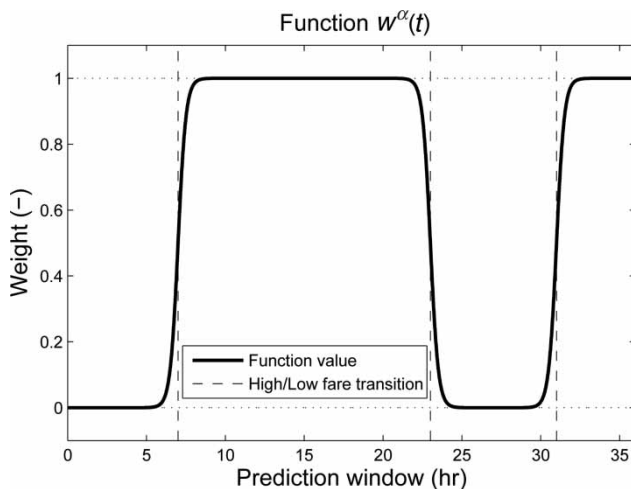


Figure 4 | Function $W^\alpha(t)$ on a prediction window $T = 36$ hr (with $\alpha = 2$). High-cost hours are during the day between 7 am and 11 pm, resulting in intervals [7, 23] and [31, 36].

Combining the three objectives B1 and B2–B3 yields the following cost function:

$$J_\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}) = \frac{\omega_1}{2} \int_{t=0}^T R(w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t))^2 dt + \frac{\omega_2}{2} \int_{t=0}^T \left(W^\alpha(t) Q_p^\alpha(\mathbf{s}, \mathbf{e}, \mathbf{p}, t) \right)^2 dt \quad (10)$$

where fixed penalty parameters $\omega_1, \omega_2 \in \mathbb{R}^+$ can be used to shift the focus between the sub-objectives, thereby ensuring a balanced pump schedule.

Applying NMPC for solving (8) with cost function (10), constraints (6) and underlying model (7) yields optimal pump schedules with respect to the approximated model. However, the computed water level $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ might not match the solution $w(t)$ to the original hybrid system (1). This difference can be annihilated by adjusting the schedule slightly.

Post-processing

The points at which the original hybrid system differs from the approximated system are the discontinuities in the pump demand. The pump schedule computed by NMPC allows pump actions to pump at half capacity (i.e. $0 < p_k < L$), whereas the hybrid system has pumps either turned on or off. Due to the design of the last of the constraints in (6), this can only happen to the pump action turned on last. Such behavior violates the initial constraints A1–A3 and needs to be addressed.

Several strategies can be deployed to solve this issue. Firstly, after obtaining the total pumped volume of this pump k from $\hat{w}[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$, we can easily extend the pump time of pump $k - 1$ accordingly, causing the total pumped volume to remain the same. A slightly more elegant technique adjusts the number of pumps to $N = k - 1$ and requests the NMPC solver to come up with a new optimal solution. This avoids pump action k to be activated at all, and will drain the water system with fewer pumps' actions, all at full capacity. In practice, these two methods yield very similar pump schedules.

The smoothed discontinuities at times s_k and e_k cause almost no difference between the two models when α is chosen large enough. The symmetry of a pump action, as

illustrated in Figure 2, ensures that the total pumped volume over time is the same in both the discontinuous and the smooth pump demand. After the pump schedule $(\mathbf{s}, \mathbf{e}, \mathbf{p})$ has been adjusted for p , it can be used for a forward simulation of the approximated model with $\alpha \gg 1$. The resulting $\hat{w}[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$ corresponds to the original hybrid system (within a negligible error margin).

The scheme presented so far is able to generate optimized pump schedules at a time $t = 0$. In real-time control, such predictions need to be made at a sample frequency. This requires a few extensions to the scheme.

Real-time control

Let the computed scheme (based on load prediction $\bar{Q}_1(t)$ and measured water level \bar{w}_0) at a moment \bar{t} be defined by $(\bar{\mathbf{s}}, \bar{\mathbf{e}}, \bar{\mathbf{p}})$. In real-time control, this schedule needs to be revised at a subsequent time instance $\tau = \bar{t} + \Delta t$ using an updated forecast $Q_1(t)$ and water level w_0 . The sample rate is typically chosen such that $\Delta t \ll D$ to ensure complete pump actions do not fall between two consecutive schedule updates. The results section contains more on computational aspects of choosing Δt .

Note that, when at time \bar{t} , pumps have already been turned on, they do not need to be active for D hours on the new prediction window. Hence, we need to incorporate activation data from the schedule at \bar{t} in order to compute a new schedule at τ .

Let $m \leq M$ pump actions be active at τ , i.e.

$$\bar{s}_k < \tau, \quad \bar{e}_k > \tau \quad \text{and} \quad \bar{p}_k \geq L$$

for all $k = 1, \dots, m$. The new schedule cannot alter the start times of these pumps anymore since they have been fixed. For feasibility, we need to add the constraint:

$$s_k = \bar{s}_k \tag{11a}$$

for all $k = 1, \dots, m$. This might seem odd because s_k appears to no longer exist in the prediction interval $[\tau, \tau + T]$. However, this restriction has never been required in this scheme and we can safely impose constraint (11a). The original constraints (6) will ensure that e_k is chosen such that no pump is turned on shorter than D hours in real time.

Though this forces pump actions to have the proper activation time and length, it does not guarantee that the new schedule will remain pumping them at full capacity. Consider the second additional constraint:

$$p_k \geq L \tag{11b}$$

which, in combination with the last constraint in (6), sets $p_k \geq L$ for $k = 1, \dots, m$. By construction of the flow function (2), we have $F_{c_k}(p_k) = c_k$, which implies all desired pump actions are at full capacity.

This concludes the theoretical outline of the receding horizon scheme for real-time control of hybrid water systems. The last step in this work is to verify the method by applying it to several load cases.

NUMERICAL RESULTS

The Waterlandse Boezem, an area of low-lying land in the north-west of The Netherlands, would easily flood if its surface water level were not controlled. Such regions are often called *polder* (a word that originates from Dutch). There are four pumps linked to the water system with an ordered pump activation capacity $\{5.835, 4.25, 5.835, 4.25\} \text{ m}^3/\text{s}$. The area of the water surface is roughly $A = 1,059 \times 10^4 \text{ m}^2$. The water level has bounds $w^- = -1.55 \text{ m}$ and $w^+ = -1.53 \text{ m}$ (with respect to NAP). In The Netherlands, peak energy hours are between 7 am and 11 pm. Pumps should be (de)activated for at least $D = 4 \text{ hr}$.

Setting up the scheme

For the following computations, let $L = 1$. The Heaviside parameter α needs to be chosen sufficiently large for the approximation to align with the original hybrid system. In practice, we let $\{\alpha_j\}$ be a monotonically increasing sequence, and consecutively solve the MPC problem for each α_j . With each iteration, let the starting point be the solution computed in the previous step. Such strategies avoid local minima of the optimization problem caused by starting out with large α . After the final iteration is done, we can apply the post-processing step of the post-processing subsection.

The prediction horizon in these examples is $T = 36$ hr and the sample rate is 0.25 hr. The larger the prediction window, the larger the solution space of the optimization problem, which leads to increased performance. In practice, however, the window is determined by the availability of reliable load predictions (which are typically around 24–48 hr). The sample rate might seem large for online control, but keep in mind that the process dynamics of these water systems are inherently slow.

It seems reasonable that, in an optimal solution, a pump k will have more than one pump action on the prediction window. In the computations described below, we have assumed a pump has no more than two active intervals. Hence, there are eight pump actions to be planned ($M = 2N = 8$). Let action $k \in [1, 4]$ be the first set of actions and $k \in [5, 8]$ be the second. The first pump is activated at s_1 , deactivated at e_1 and turned on and off for a second time at s_5 and e_5 , respectively. The second pump is activated on $[s_2, e_2]$ and $[s_6, e_6]$, etc. In order to maintain a minimum of D hours between a pump being deactivated, add:

$$s_5 \geq e_1 + D$$

to the set of constraints. Keeping constraints (6) and (11) for $k \in \{1, 2, 3, 5, 6, 7\}$ will ensure that all pump actions are planned under restrictions A1–A3.

Now the scheme has been set up, it needs to prove itself numerically. This work deploys a commercial NMPC implementation for computing the control schedules.

DNPC

DNPC is an optimized implementation of a stable and robust NMPC scheme with applications ranging from the steel industry to wind turbine design. It can compute control actions based on any nonlinear white model in state-space form:

$$\dot{x} = f(x(t), u(t), t) \quad (12a)$$

$$y = h(x(t), u(t), t) \quad (12b)$$

with state $x(t) \in \mathbb{R}^{N_x}$, control input $u(t) \in \mathbb{R}^{N_u}$ and output $y(t) \in \mathbb{R}^{N_y}$. Additionally, DNPC has the option to parametrize the input with variables $\mathbf{d} \in \mathbb{R}^M$, which determine the control,

now denoted by $u(\mathbf{d}, t)$. The benefit of using such a parametrization is twofold. Firstly, it allows the user to incorporate more complex control definitions and secondly, DNPC is able to use the input parametrization as optimization variables of the underlying optimal control problem:

$$\min_{\mathbf{d} \in \mathbb{R}^M} J(y(t), u(\mathbf{d}, t), t) \quad (13a)$$

subject to:

$$\mathbf{a}_i^T \mathbf{d} \leq \mathbf{b}_i, \quad i = 1, \dots, K \quad (13b)$$

where $\mathbf{a}_i \in \mathbb{R}^M$ and $\mathbf{b}_i \in \mathbb{R}$ define constraints on the input parametrization \mathbf{d} .

The approximated model and cost function presented in this work exactly match this description. The input parametrization corresponds to the triplet $(\mathbf{s}, \mathbf{e}, \mathbf{p})$, the control equals the pump demand $Q_p(\mathbf{s}, \mathbf{e}, \mathbf{p}, \mathbf{c}, t)$ and the state and output matches the water level $w[\mathbf{s}, \mathbf{e}, \mathbf{p}](t)$. This makes DNPC a suitable tool for applying our numerical scheme. Moreover, even though the scheme is computationally intensive, the computation of a schedule using DNPC typically takes 10 s on a moderate central processing unit, which is well within the defined sample rate Δt (of 15 min).

Sample loads

Consider a sample load case with constant rainfall over the prediction window such that $Q_1(t) = 5.835 \text{ m}^3/\text{s}$ for $t \in [0, 36]$. Note that this rate equals the capacity of the first pump exactly. A classical feedback controller, which keeps the water level at set point, would activate the first pump over the entire prediction window. Hence, this controller would pump a total quantity of $761,467.5 \text{ m}^3$, of which $435,874.5 \text{ m}^3$ are during high-cost energy hours.

Figure 5 shows the results computed by DNPC, equipped with the scheme presented in this work, for the same load case. First of all, note that the modeled water level never flows out of the specified limits. Though the feedback controller ensured a constant water level, our method uses the entire bandwidth. Note that the schedule computed by DNPC allocates most of the pump actions during low-cost hours. However, it does start the second pump action of the first pump (s_5, e_5, p_5) in the middle of the day in

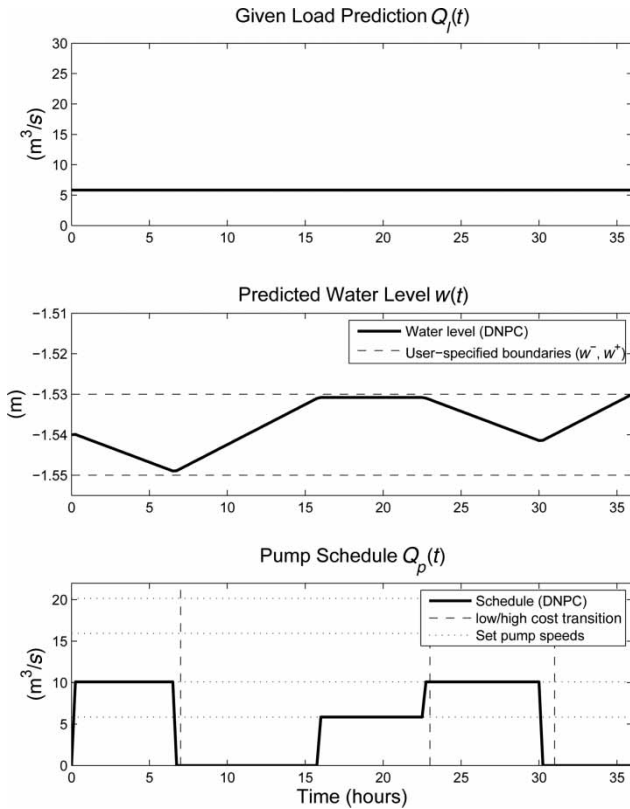


Figure 5 | Results of a pump schedule computation with a given load prediction $Q_1(t)$ on an interval $t \in [0, 36]$ (upper graph), with constant rainfall over the entire prediction window. High-cost energy hours are between 7 am and 11 pm (intervals [7, 23] and [31, 36]). The lower graph contains the computed pump schedule where the sequence of pumps has fixed cumulative capacity (dotted lines). The middle graph presents the corresponding water-level prediction with its boundaries [-1.55, -1.53].

order to prevent the water flowing over its bound. This schedule pumps a total of $650,074.5 \text{ m}^3$, of which $150,867.0 \text{ m}^3$ are allocated during high-cost energy hours. Compared with the feedback controller, this implies a reduction of 14.6% of the total pumped quantity and 65.4% of the pumped volume at peak energy costs.

A next load case involves a heavy downpour in the middle of the day:

$$Q_1(t) = \begin{cases} 15.92 & \text{if } 16 \leq t \leq 20 \\ 0.00 & \text{else} \end{cases}$$

Note that during this heavy rainfall, the incoming rate is equal to the combined rate of the first three pumps. The feedback controller from the previous example would maintain the water level on target by activating three pumps on

this exact interval. The corresponding pump schedule pumps a total of $214,920.0 \text{ m}^3$, all during peak energy hours.

Applying DNPC to this load case yields the results shown in Figure 6. Note how the schedule only activates a single pump twice, but still keeps the water level in its bandwidth. The night before the downpour, the pump already decreases the water level to the lower boundary. The pump is activated a second time as soon as the water level has reached the upper boundary, which prevents a flood. Though the pump is activated during high-cost hours, it only needs to pump a minor quantity during these intervals. DNPC pumps a total of $220,563.0 \text{ m}^3$, of which $78,772.5 \text{ m}^3$ are at high-cost hours. Comparing this performance with the feedback controller, DNPC has realized a reduction of 63.3% of the volume that needs to be pumped at peak energy costs, while the total volume remains similar.

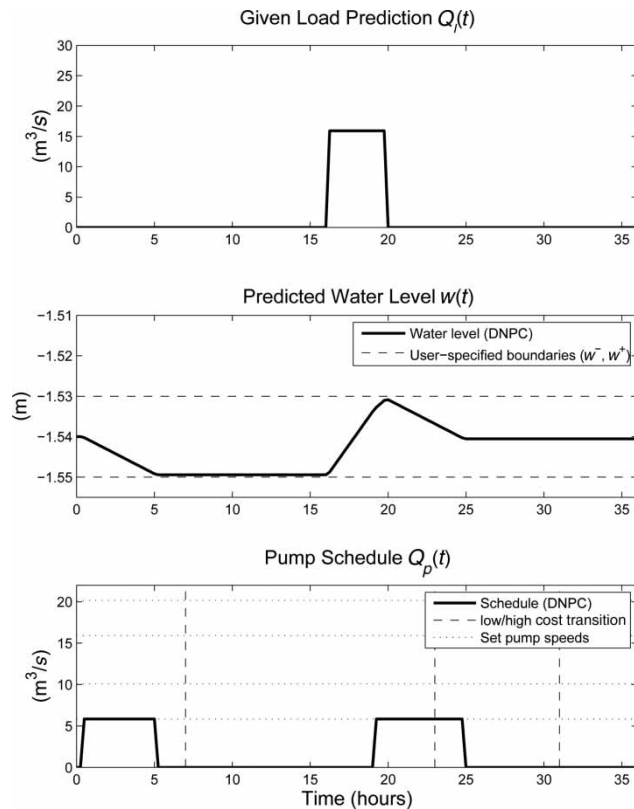


Figure 6 | Results of a pump schedule computation with a given load prediction $Q_1(t)$ on an interval $t \in [0, 36]$ (upper graph), with heavy rainfall during a short interval during the day. High-cost energy hours are between 7 am and 11 pm (intervals [7, 23] and [31, 36]). The lower graph contains the computed pump schedule where the sequence of pumps has fixed cumulative capacity (dotted lines). The middle graph presents the corresponding water-level prediction with its boundaries [-1.55, -1.53].

These two examples illustrate how the method outlined in this paper uses the user-specified bandwidth of the water level to compute an optimal pump schedule satisfying A1–A3 and B1–B3. Moreover, it shows that a significant reduction of energy costs over a feedback controller can be achieved. This latter controller is not able to deal with the complex requirements listed in this paper and is not an adequate comparison with more complex load predictions. In order to further test the scheme, it needs to prove itself with data from more realistic situations.

Logic-based controller

At the time of writing, the outlined scheme has not been implemented in the control system of the Waterlandse Boezem. Instead, a logic-based controller operates the pumps. Although constraints and objectives for this controller are (similar to) A1–A3 and B1–B3, they have been interpreted slightly differently.

First of all, the controller is designed to maintain the water at a predetermined set point $\bar{w} \in [w^-, w^+]$. This set point differs throughout the day: it is closer to w^+ during high-cost energy hours and closer to w^- otherwise. Such strategy forces the pumps to pump less at peak energy costs. The controller operates at a sample rate of 3 hr, which avoids frequent pump (de)activations. At these instances, a measurement of the current water level is compared with the set point. Pump(s) are turned on if the level exceeds the set point and turned off if the opposite is true. Moreover, at every update instance, a load prediction of 24 hr is obtained. Additional resources are activated if the total volume of the forecast exceeds a threshold.

Between two updates the schedule is reevaluated at a fine sample rate of 0.25 hr by comparing a newly obtained water-level measurement to its set point. If the water level is not behaving as scheduled in the plan (for example, when the water level is not going down though pumps have just been activated), the next/last pump in the activation sequence is (de)activated. The operator can intervene in this entire process at all times.

Note that, though this method avoids frequent pump activations and pumping at peak energy costs, it does not guarantee such behavior. Moreover, the amount of

parameters that need to be manually chosen is large, which could potentially lead to suboptimal performance.

Historical data

In order to compare the method presented in this work to the logic-based controller of the Waterlandse Boezem, we have acquired historical rainfall and the corresponding pump activation (computed by the logic-based controller). Figure 7 shows these measurement data of the Waterlandse Boezem over a window of 36 hr, starting from midnight 24 August 2011. Note that there has been considerable rainfall over this window, which makes the data suitable for a simulation study. In these computations the total simulation time is equal to the prediction window.

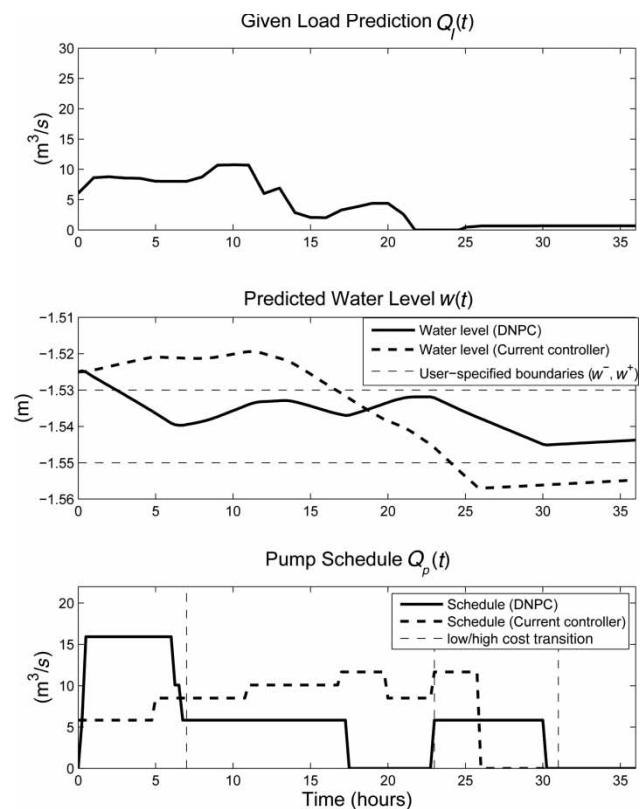


Figure 7 | Results of a pump schedule computation with a given load prediction $Q(t)$ on an interval $t \in [0, 36]$ (upper graph), measured on 24 August 2011. High-cost energy hours are between 7 am and 11 pm (intervals [7, 23] and [31, 36]). The lower graph shows the pump schedule computed by DNPC and the current controller. The middle graph presents the corresponding water-level predictions with its boundaries $[-1.55, -1.53]$.

The water levels over the prediction window, simulated by the continuous model (1), have been computed for the pump flow computed by the logic-based controller and the NMPC scheme. At time zero, the measured water level is higher than the upper bound w^+ , which forces both methods to pump a considerable amount more than the total volume of rainfall. The logic-based controller pumps a total of 849,924.0 m³, of which 731,056.5 m³ are during high-cost energy hours. The proposed NMPC method pumps a total of 550,206.0 m³, where 215,311.5 m³ are allocated during peak hours. The latter approach pumps a total of 14.0% less, and pumps 60.9% less at the highest energy cost. Such a performance indicates that our controller can be successful for real-time control of the Waterlandse Boezem. However, these simulations have been carried out with the internal model of the NMPC scheme, while the pump flow of the logic-based controller was obtained from measurements. It can be expected that the gap in performance between these two controllers is reduced when applying them both in practice.

CONCLUSION AND DISCUSSION

Applying MPC to hybrid water systems involves dealing with discontinuities in the modeled pump demand caused by requirements A1–A3 and B1–B3. An alternative to advanced optimization techniques approximates the system by smoothing these jumps. Such an approach yields a continuous, though highly nonlinear model. Optimal pump schedules of the approximated system can be computed by NMPC. In a final post-processing step, these schedules can be altered so they correspond to the original hybrid system. The NMPC method proposed in this work computes, revises and applies these optimal pump schedules at a fixed sample rate.

As shown, an optimized NMPC implementation called DNPC can be used to execute the presented scheme. The NMPC scheme showed significant improvements over a classical feedback and a logic-based controller (in terms of energy use) in numerical simulations of the Waterlandse Boezem. Though these results suggest that the outlined controller would be successful in a real-life implementation, there are a few factors that could potentially diminish its performance (compared with other methods). The simulations

are carried out with the internal model of the NMPC method, i.e. the process is assumed to behave exactly as the reservoir model. Moreover, the load prediction is assumed to be exact, which implies weather predictions are always correct. In practice, neither of these assumptions is valid: there are many physical aspects not taken into account by the simple reservoir model (1), and the load forecasts have a stochastic component. Though the NMPC scheme has guaranteed robustness and stability, these simplifications might have amplified its performance.

A situation that has not been addressed in this paper is the handling of extreme events. MPC methods, like the one introduced here, typically excel in such cases, since they can spread resources to avoid pumping at maximum capacity. For instance, in the case of an upcoming storm with net inflow larger than the combined flow of all pumps, the controller will initiate pumping before the storm arrives, and thereby avoids a flood caused by pumping after the fact at full capacity.

At the time of writing, the presented scheme has not been fully operational at the Waterlandse Boezem. The true performance of the NMPC scheme will be evident after it has actively controlled the system and measurement data are available. As mentioned, similar MPC methods have been developed (Van Overloop *et al.* 2010; Van Ekeren *et al.* 2011). A comparison with these methods, in terms of performance indicators such as energy consumption, water level and computation time, will lead to a more decisive evaluation of the NMPC method outlined in this work.

ACKNOWLEDGEMENTS

This research has been carried out as part of the controlNEXT project (Roos *et al.* 2010). DNPC was developed by DotX Control Solutions (Schuurmans 2009a, 2009b; Nederkoorn *et al.* 2011a, 2011b). The authors wish to thank the AgentschapNL of the Dutch Ministry of Economic Affairs for financial support.

REFERENCES

- Axehill, D. 2008 Integer Quadratic Programming for Control and Communication. Linköping Studies in Science and Technology, Dissertation no. 1158.

- Axehill, D. & Hansson, A. 2007 A mixed integer dual quadratic programming algorithm tailored for MPC. Technical Report LiTH-ISY-R-2761, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.
- Bacic, M., Cannon, M. & Kouvaritakis, B. 2005 [Extension of efficient predictive control to the nonlinear case](#). *International Journal of Robust Nonlinear Control* **15**, 219–231.
- De Schutter, B. & van den Boom, T. 2003 [MPC for discrete-event systems with soft and hard synchronisation constraints](#). *International Journal of Control* **76** (1), 82–94.
- Findeisen, R. & Allgöwer, F. 2002 An introduction to nonlinear model predictive control. In: *21st Benelux Meeting on Systems and Control*, Veldhoven, pp. 1–23.
- Findeisen, R., Imstand, L., Allgöwer, F. & Foss, B. 2003a [Output feedback stabilization for constrained systems with nonlinear model predictive control](#). *International Journal of Robust Nonlinear Control* **13**, 211.
- Findeisen, R., Imstand, L., Allgöwer, F. & Foss, B. 2003b [State and output feedback nonlinear model predictive control: an overview](#). *European Journal of Control* **9**, 179–195.
- Jowitt, P. W. & Germanopoulos, G. 1994 Optimal pump scheduling in water-supply networks. *Journal of Water Resources Planning and Management* **120** (2).
- Kouvaritakis, B., Cannon, M. & Rossiter, J. A. 1999 [Non-linear model based predictive control](#). *International Journal of Control* **72** (10), 919–928.
- McCormick, G. & Powell, R. S. 2003 [Optimal pump scheduling in water supply systems with maximum demand charges](#). *Journal of Water Resources Planning and Management* **129**, 372–379.
- Nederkoorn, E., Schuurmans, J. & Schuurmans, W. 2011a Continuous nonlinear model predictive control of a hybrid water system – application of DNPC to a Dutch Polder. In: *2011 IEEE International Conference on Networks, Sensors and Control (ICNSC 2011)*, Delft, The Netherlands, pp. 210–215.
- Nederkoorn, E., Van Wilgen, P. & Schuurmans, J. 2011b Nonlinear model predictive control of walking beam furnaces. In: *METEC InsteelCon*, Düsseldorf, Germany.
- Ormsbee, L. E. & Lansey, K. E. 1994 [Optimal control of water supply pumping systems](#). *Journal of Water Resources Planning and Management* **120** (2), 237–252.
- Roos, B., Nederkoorn, E. & de Hart, K. 2010 Real-time control in het (afval)waterbeheer (Dutch). www.control-next.nl.
- Schuurmans, J. 2009a DotX nonlinear predictive controller (DNPC). www.dotxcontrol.com.
- Schuurmans, J. 2009b Report of benchmark simulations with DNPC. www.dotxcontrol.com.
- Tenny, M., Rawlings, J. & Wright, S. 2004 [Closed-loop behavior of nonlinear model predictive control](#). *AIChE Journal* **50**, 2142–2154.
- Van Ekeren, H., Negenborn, R., Van Overloop, P. & De Schutter, B. 2011 Hybrid model predictive control using time-instant optimization for the Rhine-Meuse delta. In: *2011 IEEE Int. Conf. on Netw., Sens. and Control (ICNSC 2011)*, Delft, The Netherlands, pp. 216–221.
- Van Overloop, P., Negenborn, R., De Schutter, B. & Van de Giesen, N. 2010 Predictive control for national water flow optimization in The Netherlands. In: *Intelligent Infrastructures* (R. R. Negenborn, Z. Lukszo & H. Hellendoorn, eds), Springer, Dordrecht, The Netherlands, pp. 439–461.
- Yu, G., Powell, S. & Sterling, M. 1994 [Optimized pump scheduling in water distribution systems](#). *Journal of Optimization Theory and Applications* **83** (3), 463–488.

First received 23 October 2011; accepted in revised form 2 July 2012. Available online 13 December 2012