Formal Controller Synthesis for Wastewater Systems with Signal Temporal Logic Constraints: The Barcelona Case Study

Samira S. Farahani [*], **Sadegh Esmaeil Zadeh Soudjani** [*], **Rupak Majumdar** [*], **Carlos Ocampo-Martinez** [**]

[*] *Max Planck Institute for Software Systems, Kaiserslautern, Germany (e-mail: farahani,sadegh,rupak@mpi-sws.org).*
[**] *Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), 08028 Barcelona, Spain (e-mail: cocampo@iri.upc.edu)*

Abstract: We present an approach for formal controller synthesis of the Barcelona wastewater system. The goal of the controller is to minimize overflow in the system and to reduce environmental contamination (pollution). Due to the influence of sudden and unpredictable weather changes within the Mediterranean climate, we propose robust model predictive control strategy. This approach synthesizes control inputs (i.e., flows through network actuators) that make the system robust to uncertainties in the weather forecast; control inputs are updated in an online fashion to incorporate the newly available measurements from the system and the disturbances. We employ signal temporal logic as a formal mechanism to express the desired behavior of the system. The quantitative semantics of the logic is then used to encode the desired behavior in both the set of constraints and the objective function of the optimization problem. We propose a solution approach for the obtained worst-case optimization, which is based on transforming the nonlinear dynamics of the system into a mixed logical dynamical model. Then, we employ Monte Carlo sampling and dual reformulation to get a mixed integer linear or quadratic programming problem. The proposed approach is applied to a catchment of the Barcelona wastewater system to illustrate its effectiveness.

Keywords: Formal Synthesis, Signal Temporal Logic, Model Predictive Control, Robust Control, Mixed Integer Linear/Quadratic Programing, Wastewater Management

1. INTRODUCTION

The infrastructure for water and wastewater management is being continuously upgraded due to the constant increase in demand for water and wastewater services as a result of population growth. In order to support this upgrade, the water industry has been investigating the potential benefits of using more advanced automatic control strategies. The design and automatic control of sewer networks pose new challenges to the control community. The newly designed methodologies should be able to handle the effect of uncertainties in the amount of precipitation, the physical and operational constraints of the network, and the effects of delays and nonlinearities in the dynamics of the system. These challenges require improving performance of the traditional control strategies such as on-off and PID controllers, which are not capable of handling such issues. Model predictive control (MPC) seems to be a suitable methodology to control sewer networks as it can deal with these particular challenges associated with such systems. MPC is an online control technique that uses a mathematical model of the considered system to compute the control inputs by minimizing a cost function (Bemporad et al., 2002b; Lazar et al., 2006; Maciejowski, 2002; Rawlings and Mayne, 2009). Moreover, it is capable of incorporating either linear or nonlinear dynamics of the system as well as handling constraints on inputs, states and outputs. Hence, the MPC methodology is quite suitable for

the global control of urban sewage systems within a hierarchical control structure (Schütze et al., 2004; Marinaki and Papageorgiou, 2005).

The system under investigation in this paper is part of the Barcelona wastewater system, which is subject to sudden weather-change events within the Mediterranean climate. We consider the Barcelona test catchment (BTC) that covers a surface area of 22.6 km^2 and represents all the typical elements of the whole network. The application of deterministic MPC to Barcelona wastewater system has been investigated in Ocampo-Martinez (2010) for a portion of this system and its benefits have been examined toward the potential percentage reductions in both flooding and pollution in Barcelona sewage network. In this paper, we build on the work of Ocampo-Martinez (2010) by including uncertainty in the amount of precipitation as a bounded disturbance and by formulating a robust MPC optimization problem (Allwright, 1994; Badgwell, 1997; Campo and Morari, 1987; Kothare et al., 1996; Mayne et al., 2005) to synthesize control inputs.

In order to specify the desired behavior of a system with continuous dynamics, signal temporal logic (STL) is one of the most useful languages. In comparison with other temporal logic formalisms, STL has the advantage of naturally admitting a quantitative semantics. As such, in addition to the binary answer to the satisfaction question of the specifications, it provides a real

number that indicates the extent to which the specification is either satisfied or violated. This quantitative semantics associated to the STL specification is referred to as the *robustness function*. Incorporating such temporal specification in the optimization problem formulation enforces the closed-loop system to satisfy the desired temporal behavior, as it is confirmed by the simulation results of this paper.

Considering the nonlinear (or hybrid) nature of the network model, we show that the proposed robust MPC optimization problem can be formulated as mixed integer linear or quadratic programming (MILP or MIQP) problems as follows. First, the nonlinear dynamics of the wastewater network are transformed into a mixed logical dynamical (MLD) model. Then, the nonlinear expressions in the objective function and the STL constraints are transformed into mixed integer linear terms and constraints, respectively. Finally, we employ either dual reformulation or Monte Carlo method in the inner optimization problem, i.e., the maximization problem, to get either an MIQP problem or an MILP problem. In the case of MIQP, we obtain a non-convex optimization problem, which we solve iteratively by linear approximation of the quadratic objective function. In the simulation results, we compare the performance of the dual reformulation with the Monte Carlo approach and we show the effect of STL specifications on system behavior.

Related Work. STL has been used for controller synthesis in a variety of domains for uncertain systems using receding horizon control techniques (Farahani et al., 2017a, 2015; Raman et al., 2015). Transforming STL constraints into mixed integer linear constraints has been used in Raman et al. (2014). Several works related to this wastewater system consider different models and cope with the design of alternative MPC approaches, e.g., Joseph-Duran et al. (2014, 2015) and references therein. Recent works have proposed different approaches for handling uncertainties in process control. The work reported in Ricardez-Sandoval et al. (2010) proposes a two-level method to first estimate the worst-case disturbance profile using an uncertain finite impulse response (FIR) model. This profile is then employed to simulate the closed-loop nonlinear dynamic process model for obtaining the worst-case output variability and checking the feasibility of constraints. Likewise, the work reported in Gutierrez et al. (2014) proposes an MPC strategy that relies on nonlinear optimizations. This approach incorporates integer variables towards performing a modeling selection within the control structure. In our previous work (Farahani et al., 2017b), we studied a small part of BTC with only 3 tanks. In the current paper, we consider the full model of BTC, as presented in Ocampo-Martinez (2010), to show that our method is both scalable and efficient for formally synthesizing control inputs for the system.

Our work is distinct from the previous works on wastewater systems in a) considering uncertainty in the amount of precipitation both in the model and in the controller design; b) employing STL to encode desired properties of the closed-loop trajectories; c) proposing an approximate solution for the formulated optimization problem that is scalable and can be applied to the large dimensional model of the BTC.

The remainder of this paper is organized as follows. In Section 2, the considered model of the BTC is described. In Section 3, the robust MPC formulation is presented together with the constraints induced by both the model of the system and the STL specifications. In Section 4, we discuss the MLD model of
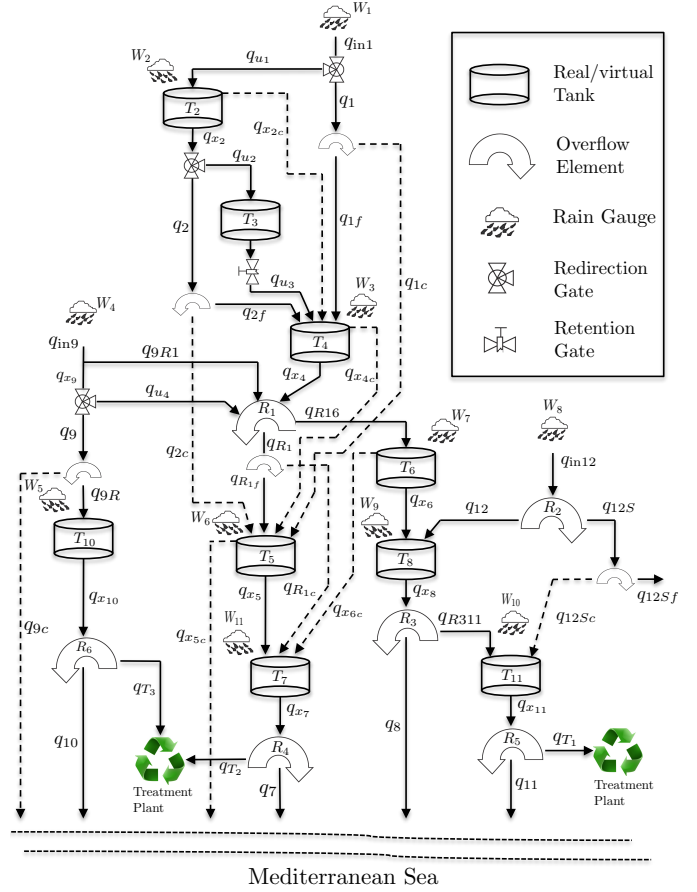

Fig. 1. Considered 9-tank catchment scheme

the system and propose solution approaches to solve the mixed integer robust MPC optimization problem. In Section 5, the proposed control approach is applied to the BTC and the main results are proposed and discussed. Finally, Section 6 draws the main conclusions of the paper and the possible lines of future research. In order to keep the discussion of the paper focused, we summarize STL semantics and the notion of robustness in the appendices.

2. BARCELONA TEST CATCHMENT MODEL

We consider a portion of the sewer network of Barcelona that is representative, as it exhibits the main phenomena and the most common characteristics found in the entire network. The network consists of nine tanks, four control inputs corresponding to the manipulated flows, and eleven measured disturbances corresponding to the measurements of rain precipitation. Two wastewater treatment plants (WWTP) are used to treat the sewage before it is released to the receiving environment. Figure 1 shows the part of Barcelona test catchment (BTC) area considered in this paper. There are two types of tanks in the model: one *real* tank (T_3), and eight *virtual* tanks (all tanks except T_3). The BTC has six weir overflow devices R_i, $i \in \{1, 2, \ldots, 6\}$, three redirection gates, one retention gate, and five T-pipes.

Description of the components. A virtual tank is a storage element that represents the total volume of sewage inside the sewer mains associated with a determined sub-catchment (Gelormino and Ricker, 1994). A real tank is a buffer that stores the wastewater and allows to redirect it towards different pipes in the
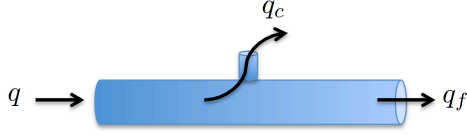
Fig. 2. Flow direction in a T-pipe

network. Redirection gates are used to change the direction of the sewage while retention gates are used to retain the sewage flow at a certain point in the network. Weir overflow devices are used to specify the desired direction of the flow while taking into account the capacity of the pipes.

The role of T-pipes is to merge or split the sewage flows (Figure 2). The equations of flow inside T-pipes at time step $k \in \mathbb{Z}_+$ can be written as $q_i(k) = q_{c,i}(k) + q_{f,i}(k)$ with

$$q_{c,i}(k) = \begin{cases} q_i(k) - \bar{q}_i, & q_i(k) \geq \bar{q}_i \\ 0, & q_i(k) < \bar{q}_i \end{cases} \text{ for } i = 1, \ldots, 5, \quad (1)$$

where $\bar{q}_i$ denotes the maximum flow through pipe i. The outflows from redirection gates satisfy the mass conservation equation $q_i = q_{C_j,\text{in}} - q_{ui}$, where $q_{C_j,\text{in}}$ is the inflow to the redirection gate C_j, $j = 1, 2, 3$. Moreover, the outflow of the virtual tank i is proportional to the tank volume, i.e., $q_{x_i}(k) = \beta_i x_i(k)$ with β_i denoting the volume/flow conversion coefficient and $x_i(k)$ denoting the volume of tank T_i at time step k.

The flow equations in the weir overflow devices R_i, $i \in \{1, \ldots, 6\}$, can be defined as

$$q_{R_i,\text{out}}(k) = \begin{cases} q_{R_i,\text{in}}(k), & q_{R_i,\text{in}}(k) \leq \bar{q}_{R_i} \\ \bar{q}_{R_i}, & q_{R_i,\text{in}}(k) > \bar{q}_{R_i} \end{cases} \text{ for } i = 1, \ldots, 6, \quad (2)$$

where $q_{R_i,\text{in}}(k)$ denotes the sum of inflows entering weir overflow device R_i at time step k, $q_{R_i,\text{out}}(k)$ denotes the outflow from R_i in the *desired* direction, and $\bar{q}_{R_i}$ denotes the maximum capacity of the pipe in the desired direction. Accordingly, the flow equation of the weir overflow device R_i in the *undesired* direction can be obtained by $q_{R_i,\text{in}}(k) - q_{R_i,\text{out}}(k)$. For instance in the case of R_1 in Figure 1, $q_{R_1,\text{out}} = q_{R16}$ is the outflow and $\bar{q}_{R_1} = \bar{q}_{R16}$. The undesired flow direction of R_1 is q_{R1}, which can be obtained as $q_{R1}(k) = q_{R_1,\text{in}}(k) - q_{R16}$, with $q_{R_1,\text{in}}(k) = q_{u_4}(k) + q_{9R1}(k) + q_{x_4}(k)$. Unlike the real tank T_3, a virtual tank T_i may have overflow, which is denoted by $q_{x_{ic}}$, and can be computed as

$$q_{x_{ic}}(k) = \begin{cases} q_{x_{ic},\text{tot}}(k) - \frac{1}{\Delta t}\bar{x}_i, & q_{x_{ic},\text{tot}}(k) \geq \bar{x}_i/\Delta t, \\ 0, & q_{x_{ic},\text{tot}}(k) < \bar{x}_i/\Delta t, \end{cases} \quad (3)$$

where $q_{x_{ic},\text{tot}}(k)$ denotes the sum of inflow and the current volume rate minus the outflow of tank T_i at each time step k, $\bar{x}_i$ denotes the maximum capacity of tank T_i, and Δt is the sampling time. We consider such overflows only in virtual tanks T_2, T_4, T_5, and T_6, as other tanks do not have overflow in practice due to their large storing capacities.

Construction of the dynamical model. Changes in the volume of a tank are proportional to the difference between its inflows and outflows. The dynamical evolution of the volume of tank T_i in discrete time is

$$x_i(k+1) = x_i(k) + \Delta t (q_{\text{in}}(k) - q_{\text{out}}(k)),$$

where $x_i(k)$ is the volume of tank T_i at time step k. For the system shown in Figure 1, the state equations can be then written as

$$\begin{aligned}
x_2(k+1) &= x_2(k) + \Delta t (W_2(k) + q_{u1}(k) - q_{x_1}(k) - q_{x_{2c}}(k)), \\
x_3(k+1) &= x_3(k) + \Delta t (q_{u2}(k) - q_{u3}(k)), \\
x_4(k+1) &= x_4(k) + \Delta t (W_3(k) + q_{1f}(k) + q_{2f}(k) \\
&\quad + q_{u3}(k) + q_{x_{2c}} - q_{x_4}(k) - q_{x_{4c}}(k)), \\
x_5(k+1) &= x_5(k) + \Delta t (W_6(k) + q_{R1f}(k) + q_{2c}(k) \\
&\quad + q_{x_{4c}}(k) + q_{1c} - q_{x_5}(k) - q_{x_{5c}}(k)), \\
x_6(k+1) &= x_6(k) + \Delta t (W_7(k) + q_{R16}(k) - q_{x_6}(k) - q_{x_{6c}}(k)), \\
x_7(k+1) &= x_7(k) + \Delta t (W_{11}(k) + q_{x_5}(k) + q_{R1c}(k) + q_{x_{6c}}(k) \\
&\quad - q_{x_7}(k)), \\
x_8(k+1) &= x_8(k) + \Delta t (W_9(k) + q_{x_6}(k) + q_{12}(k) - q_{x_{8c}}(k)), \\
x_{10}(k+1) &= x_{10}(k) + \Delta t (W_5(k) + q_{9f}(k) - q_{x_{10}}(k)), \\
x_{11}(k+1) &= x_{11}(k) + \Delta t (W_{10}(k) + q_{12Sc}(k) + q_{R311} - q_{x_{11}}(k)),
\end{aligned} \quad (4)$$

where $W_i(k) = \varphi_i S_i P_i(k)$, $i \in \mathcal{W} \triangleq \{2, 3, 5, 6, 7, 9, 10, 11\}$, specifies the amount of rainfall entered in tank T_i, with φ_i denoting the ground absorption coefficient of the i-th catchment, S_i denoting the corresponding surface area, and P_i denoting the rain intensity. Since the amount of rainfall is uncertain, we treat each $W_i(k)$ as a disturbance defined as $W_i(k) = W_{\text{ref},i}(k) + e_i(k)$, where $W_{\text{ref},i}(k)$ is a known nominal value obtained through the weather forecast and $e_i(k)$ is an unknown bounded quantity.

Remark 1. The BTC of Ocampo-Martinez (2010) has twelve tanks (one real and 11 virtual tanks). In our model, we have not considered tanks T_1, T_9, T_{12} corresponding to W_1, W_4, W_8 (cf. Figure 1) since their dynamics are not influenced by any gate. As a result of eliminating these three tanks, the corresponding disturbances are defined as $W_j(k) = P_j(k)$ since there is no surface to absorb the rain anymore. The inflows from eliminated tanks entering the redirection gates are also considered as $q_{j,\text{in}}(k) = W_j(k)$.

Having the flow equations for each element of the network via (1)-(3), we can replace them in the state equations (4), which results in a nonlinear model for the wastewater system of the form

$$x(k+1) = f_d(x(k), u(k), W(k)), \quad (5)$$

where $x \in \mathbb{R}^9$ denote the state vector, $u = [q_{u_1}, q_{u_2}, q_{u_3}, q_{u_4}]^T \in \mathbb{R}^4$ is the control input vector, and $W \in \mathbb{R}^8$ is the disturbance vector. According to the equations (1)-(4), $f_d(\cdot, \cdot, \cdot)$ will be a piecewise affine function of the state, input, and disturbance at each time step k.

We explain the controller synthesis problem for this model in the next section.

3. ROBUST MODEL PREDICTIVE CONTROL

In the BTC, the goal is to control the inflow and outflow in (both virtual and real) tanks in order to avoid flooding and contaminating Mediterranean sea. The uncertainty in the wastewater system is in the amount of precipitation that we consider to be a bounded quantity. The control objectives are to minimize both flooding on streets (overflows $q_{1c}, q_{2c}, q_{x_{2c}}, q_{x_{4c}}, q_{x_{6c}}, q_{R1c}$ and q_{12Sc}) and the pollution entering the sea (overflows q_{9c} and $q_{x_{5c}}$ and flows q_7, q_8, q_{10}, and q_{11} in Figure 1) as well as to minimize the control inputs to save energy consumption in opening and closing gates. It is also desirable that the overflows in the controlled network are reduced to zero as soon as possible whenever they occur.

For this purposes, we employ the control approach of robust (worst-case) MPC in a shrinking horizon fashion. The choice for applying shrinking horizon fashion has been made due to the

fact that we are interested in the behavior of the system only in a given finite time-interval. This approach can be summarized as follows: at time step one, we obtain a sequence of control inputs with length N (prediction horizon) to optimize the objective function; we only apply the first component of the obtained control sequence to the system and update its state. At the next time step, the first component of the control sequence is fixed by the one of the previously calculated optimal control sequence, and we only optimize for a control sequence of length $N-1$. Hence, the size of control sequence decreases by 1 at each time step.

In the sequel, we first formulate the objectives as a cost function. Then, we write the desired temporal property in STL language and formulate the closed-loop worst-case optimization problem. In order to keep the discussion focused, we directly give the STL specification and refer the reader to Appendix I for a formal description of the syntax and semantics of STL.

Cost function. The cost function for the sewage network at each time step k is defined as

$$
\begin{aligned}
J(k) = &\|\bar{u}(k)\|_1 + q_{1c}(k) + q_{2c}(k) + q_{x_{2c}}(k) + q_{x_{4c}}(k) \\
&+ q_{x_{6c}}(k) + q_{R1c}(k) + q_{12Sc}(k) + q_{9c}(k) + q_{x_{5c}}(k) \quad (6) \\
&+ q_7(k) + q_8(k) + q_{10}(k) + q_{11}(k),
\end{aligned}
$$

which includes the control inputs, the overflows to the street, and the pollution entering the sea. We have chosen the one-norm of the control input, however, it is also possible to choose quadratic or infinity norms. Note that minimizing $q_7(k), q_8(k), q_{10}(k), q_{11}(k)$ results in maximizing the amount of sewage treatments $q_{T1}(k), q_{T2}(k)$ and $q_{T3}(k)$ at each time step k.

STL constraint. We require the closed-loop system to have the following property: *always during the time interval* $[0,N]$*, if there is an overflow in any of the pipes* 1 *or* 2*, then the overflow in that pipe should eventually be zero within the next* k' *time steps.* This desired temporal behavior can be expressed by the STL specification

$$
\begin{aligned}
\varphi := \Box_{[0,N]} \Big[&\Big(q_{1c} > 0 \rightarrow \Diamond_{[1,k']} q_{1c} \leq 0 \Big) \wedge \\
&\Big(q_{2c} > 0 \rightarrow \Diamond_{[1,k']} q_{2c} \leq 0 \Big) \Big], \quad (7)
\end{aligned}
$$

where *always* is denoted by $\Box$, $q_{1c} > 0$ and $q_{2c} > 0$ are the amount of overflow in pipes 1 and 2, *eventually* is denoted by $\Diamond$, and time intervals are written as subscript (cf. Appendix I for a formal treatment of STL). The aim of our control problem is to synthesize an input sequence such that the closed-loop trajectory satisfies φ. We have selected this temporal specification to avoid the street flooding as much as possible. Note that a considerable amount of flow enters the streets through these two pipes. However, it is possible to define the temporal specification for other flows and/or overflows in the network depending on the priorities of the network, the predicted rain profile, or other criteria.

Disturbance set. We assume that there is an uncertainty in the amount of rain, i.e., $W_i(k) = W_{\text{ref},i}(k) + e(k)$, where $W_{\text{ref},i}(k)$ is the amount of rainfall entered tank T_i or a gate at time k for $i = 1, \ldots, 11$ (cf. Section 2). We gather the uncertainty for time steps $k, k+1, \ldots, N$ in a vector $\bar{e}(k) = [e^T(k), \ldots, e^T(N)]^T$ such that, for all k, each component of $\bar{e}(k)$ belongs to $\mathscr{E} = \{e : Se \leq q\}$, which is a bounded polyhedral set.

Suppose that we have solved the robust MPC optimization problem up to time step $k-1$ obtaining the optimal control

inputs $u_0^*, \ldots, u_{k-1}^*$ and the observed states $x_0^*, \ldots, x_k^*$. In the following, we formulate the optimization problem that needs to be solved at time step k. Define $\tilde{u}(0 : k : N) := [u_0^*, \ldots, u_{k-1}^*, \bar{u}(k)]^T$ to be the vector of input variables with $\bar{u}(k) := [u(k), \ldots, u(N-1)]^T$ being the vector of input variables to be optimized over. Let $\tilde{x}(0 : k : N) = [x_0^*, \ldots, x_k^*, \bar{x}(k+1)]^T$ be the vector of states such that $\bar{x}(k+1) = [x(k+1), \ldots, x(N)]$ is the vector of future states of the system satisfying (5). Substituting state equations (5) in the trajectory $\tilde{x}(0 : k : N)$ makes it a function of unknown vectors $\bar{u}(k)$ and $\bar{e}(k)$. To emphasize this, we denote the trajectory at time step k by $\xi_N(\bar{u}(k), \bar{e}(k))$. Accordingly, for $0 \leq k \leq N$, we can define the robust MPC optimization problem at time step k as

$$
\min_{\bar{u}(k)} \max_{\bar{e}(k) \in \mathscr{E}} \sum_{j=k}^{N} J(j) \quad (8a)
$$

subject to

$$
\begin{aligned}
x(\tau+1) &= f_{\text{d}}(x(\tau), u(\tau), e(\tau)), \quad &(8b) \\
P\bar{x}(k) + Q\bar{u}(k) + R\bar{e}(k) + h &\leq 0, \quad &(8c) \\
\xi_N(\bar{u}(k), \bar{e}(k)) &\models \varphi, \ \forall \, \bar{e}(k) \in \mathscr{E}, \quad &(8d)
\end{aligned}
$$

for all $k \leq \tau \leq N$, where (8c) includes the constraints on state, input, and disturbance, in which P, Q, R are appropriately defined constant matrices of suitable dimensions and h is a vector of known quantities. Also in (8d), the STL specification (7) is denoted by φ. The symbol $\models$ indicates that the trajectory ξ_N should satisfy φ. Note that (8c) is related to the physical constraints of the system, which appear in the inner optimization problem, while the STL constraint (8d) belongs to the outer optimization problem.

Remark 2. The optimization problem (8) includes the specification (7) as a hard constraint. Alternatively, one may use the quantitative semantics of the specification known as the robustness function (cf. Appendix I) and include it into the objective function (6). Adding the robustness function to the objective function (6) allows us to maximize the robust satisfaction of the specification. In this way, we have the STL specification not only as hard constraint, which needs to be satisfied for all values of e, but also we maximize the robustness of satisfaction of the STL specification.

The optimization problem (8) is nonlinear due to the hybrid nature of the wastewater system. We explain in the next section how to deal with this issue and propose different methods to solve the formulated robust MPC optimization problem.

4. SOLVING THE ROBUST MPC PROBLEM

In order to solve the optimization problem (8), we transform the hybrid system model into its equivalent MLD form. An MLD model is a linear system model with both continuous and binary variables while having affine constraints on these variables. The MLD formalism allows the transformation of logical statements involving continuous variables into mixed integer linear inequalities. We employ the following equivalences (Bemporad et al., 2002a) to transform the nonlinear dynamics of the system and nonlinear terms in the objective function into linear functions and linear constraints:

$$[f(x(k)) \leq 0] \leftrightarrow [\delta(k) = 1] \text{ iff } \begin{cases} f(x(k)) \leq M(1 - \delta(k)), \\ f(x(k)) \geq \varepsilon + (m - \varepsilon)\delta(k), \end{cases}$$

$$z(k) = \delta(k)f(x(k)) \text{ iff } \begin{cases} z(k) \leq M\delta(k), \\ z(k) \geq m\delta(k), \\ z(k) \leq f(x(k)) - m(1 - \delta(k)), \\ z(k) \geq f(x(k)) - M(1 - \delta(k)), \end{cases}$$

$$(9)$$

where $M, m \in \mathbb{R}$ are the upper and lower bounds on the linear function $f(x(k))$ and ε is the machine precision.

Based on the equivalence relations (9), the MLD model of (1)-(3) can be obtained by defining the following auxiliary variables:

- $[\delta_i(k) = 1] \leftrightarrow [q_i(k) \geq \bar{q}_i]$, $z_i(k) = \delta_i(k)q_i(k)$
  for $i = 1, \ldots, 5$,
- $[\delta_j(k) = 1] \leftrightarrow [q_{R_i,\text{in}}(k) \leq \bar{q}_{(\cdot)}]$, $z_j(k) = \delta_j(k)q_{R_i,\text{in}}(k)$
  for $i = 1, \ldots, 6$ and $j = 6, \ldots, 11$,
- $[\delta_j(k) = 1] \leftrightarrow [q_{x_{ic},\text{tot}}(k) \geq \bar{x}_i/\Delta t]$, $z_j(k) = \delta_j(k)q_{x_{ic},\text{tot}}(k)$
  for $i = 2, 4, 5, 6$ and $j = 12, \ldots, 15$.

The inequality constraints corresponding to the above auxiliary variables and logical statements can be obtained according to (9).

Example 1. Consider the above definition of δ_1 and z_1 with the functions $f_1(x(k)) := \bar{q}_1 - q_1(k)$ and $f_2(x(k)) := q_1(k)$, where $q_1(k) = W_{\text{ref},1}(k) + e_1(k) - u_1(k)$. The associated inequality constraints can be written as

$$\bar{q}_1 - W_{\text{ref},1}(k) - e_1(k) + u_1(k) \leq M(1 - \delta_1(k)),$$
$$\bar{q}_1 - W_{\text{ref},1}(k) - e_1(k) + u_1(k) \geq \varepsilon + (m - \varepsilon)\delta_1(k),$$
$$z_1(k) \leq M'\delta_1, \quad z_1(k) \geq m'\delta_1,$$
$$z_1(k) \leq W_{\text{ref},1}(k) + e_1(k) - u_1(k) - m'(1 - \delta_1(k)),$$
$$z_1(k) \geq W_{\text{ref},1}(k) + e_1(k) - u_1(k) - M'(1 - \delta_1(k)),$$

with $m = \bar{q}_1 - W_{\text{ref},1}(k) - \hat{e}_1$ and $M = \bar{q}_1 + \hat{u}_1$ as the lower and upper bounds on f_1, where $\hat{u}_1$ is the upper bound of u_1 and $\hat{e}_1$ is the upper bound of e_1. The lower and upper bounds on f_2 can also be defined as $m' = 0$ and $M' = W_{\text{ref},1}(k) + \hat{e}_1$.

We also transform the STL constraints into mixed-integer linear constraints by introducing continuous and binary auxiliary variables (cf. (Raman et al., 2014)). Denote by $z(k) = [z_1(k), \ldots, z_r(k)]^T$ and $\delta(k) = [\delta_1(k), \ldots, \delta_s(k)]^T$ the vectors that contain all continuous and binary auxiliary variables, respectively, for the MLD model, the objective function, and the STL constraints. Using these two vectors, the state equations and constraints of the MLD model can be written as

$$x(k+1) = [A \quad B_1 \quad B_2 \quad B_3 \quad B_4]\chi(k) + B_5, \quad (10a)$$
$$[E_1 \quad E_2 \quad E_3 \quad E_4 \quad E_5]\chi(k) \leq g, \quad (10b)$$

with $\chi(k) \triangleq [x(k) \quad u(k) \quad \delta(k) \quad z(k) \quad W(k)]^T$, where A and B_i, $i = 1, 2, 3, 4$, are system matrices of suitable dimensions (see Appendix II) and B_5, g are vectors of constant entries. Matrices E_i, $i = 1, \ldots, 5$, are related to the MLD constraints, the physical constraints of the system (in this case flow constraints on the input variables), and the constraints obtained from the STL transformation.

Let $\tilde{z}(0 : k : N) = [z_0^*, \ldots, z_{k-1}^*, \bar{z}(k)]$ such that $z_0^*, \ldots, z_{k-1}^*$ are the auxiliary variables up to time $k-1$ uniquely specified based on $x_\tau^*, u_\tau^*, \tau < k$, and $\bar{z}(k) = [z(k), \ldots, z(N-1)]$ are the auxiliary variables to be optimized over at time step k ($\tilde{\delta}(0 : k : N)$ and $\bar{\delta}(k)$ are defined similarly). Using these auxiliary variables, the

cost function (6) can be rewritten as
$$J(\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \bar{e}(k)) = C_1^T\bar{u}(k) + C_2^T\bar{z}(k) + C_3^T\bar{\delta}(k) + C_4^T\bar{e}(k),$$
where C_i, $i = 1, \ldots, 4$, are properly defined weighting matrices.

Putting all these transformation together, the worst-case MPC optimization problem in (8) considering the MLD model of the system can be written as

$$\min_{\bar{u}(k), \bar{z}(k), \bar{\delta}(k)} \max_{\bar{e}(k) \in \mathscr{E}} J(\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \bar{e}(k)) \quad (11a)$$

subject to

$$\tilde{E}_1 x(0) + \tilde{E}_2\bar{u}(0 : k : N) + \tilde{E}_3\tilde{\delta}(0 : k : N) + \tilde{E}_4\tilde{z}(0 : k : N)$$
$$+ \tilde{E}_5\bar{e}(0 : k : N) \leq \tilde{g}, \quad (11b)$$

where $\tilde{E}_i$, $i = 1, \ldots, 5$ and $\tilde{g}$ are appropriately defined constraint matrices and vector, respectively. Note that the state constraints (8b) are now incorporated in the inequality constraints (11b). Note also that the STL constraints, which are now part of the constraints in (11b), should hold for all values of $\bar{e}(k) \in \mathscr{E}$ (cf. (8d)). In the following theorem, we prove that the closed-loop system satisfies the STL specification by using the shrinking-horizon technique and by keeping track of the control input and observed states.

Theorem 1. For the STL formula φ and ε, if the optimization problem (11) is feasible at each time step k, the optimal control sequence $\tau^*(0 : N) = [u^*(0), \ldots, u^*(N-1), z^*(0), \ldots, z^*(N-1), \delta^*(0), \ldots, \delta^*(N-1)]$ computed on a machine with precision ε ensures that the closed-loop system satisfies φ.

Proof. We have chosen the prediction horizon N such that $N \geq \text{len}(\varphi)$, where $\text{len}(\varphi)$ is defined as the maximum over the sums of all nested upper bounds on the temporal operators (cf. Appendix I for details). Since we apply a shrinking-horizon approach, at each time step k, we fix the previously obtained optimal input variables. As such, at time step $N - 1$, which is the last time step in the closed-loop optimization procedure, the vector of decision variables has the following form: $\tau(0 : N) = [u^*(0), \ldots, u^*(N-2), u(N-1), z^*(0), \ldots, z^*(N-2), z(N-1), \delta^*(0), \ldots, \delta^*(N-2), \delta(N-1)]^T$, in which the only unknown variables are $u(N-1), z(N-1)$ and $\delta(N-1)$. Hence, if at this step an optimal input sequence $[u^*(N-1), z^*(N-1), \delta^*(N-1)]$ is obtained, we are assured that the STL specification is satisfied. ∎

There are two approaches in the literature for solving robust MPC optimization problems in the presence of STL specifications. The first approach, from Raman et al. (2015), is based on a counterexampleguided inductive synthesis. In this approach, at each time step, the algorithm starts by guessing the value of the disturbance signal over the current horizon. It then tries to synthesize a control input that satisfies the specification in the face of this disturbance. Once such control input is found, a new disturbance is sought that thwarts this control input, by minimizing the robustness of satisfaction to a level at which the specification is not satisfied. The process repeats until a control input is found such that there is no disturbance that can prevent the specification from being satisfied. The major disadvantage of this approach is that the algorithm may never terminate if the set of disturbances $\mathscr{E}$ is not finite.

The second approach for solving a robust MPC optimization problem is based on multi-parametric mixed integer linear programming (mp-MILP) (Dua and Pistikopoulos, 2000). The approach tries to find the explicit solution of the inner optimization problem as a function of $\bar{u}(k), \bar{z}(k), \bar{\delta}(k)$ and then solves

the outer optimization as an MILP problem. As expected, this approach is not quite efficient when either the size of the optimization variables vector or the prediction horizon N is large. Moreover, the number of integer variables in the MILP problem adversely affects the computational time.

For non-trivial STL formulas, the number of integer variables is large in general, as binary variables are introduced to encode each min and max operation in the robustness function of the formula (cf. Appendix I). The number of states and integer variables related to the MLD model of BTC is also large, which makes the mp-MILP approach unsuitable. In the following, we propose two different approaches to solve the worst-case MPC optimization problem (11): Monte Carlo-based optimization and dual reformulation of the inner optimization.

4.1 Monte Carlo-based optimization

We use a sampling approach to get an approximate solution for the inner optimization and solve the outer optimization based on that (Pardalos and Resende, 2002). Let $\bar{e}^{(1)}(k), \ldots, \bar{e}^{(L)}(k)$ denote L different realizations of the disturbance belonging to the set $\mathscr{E} = \{e | Se \leq q\}$ and let

$$t(k) = \max_{\bar{e}^{(1)}(k), \ldots, \bar{e}^{(L)}(k)} \Big(J(\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \bar{e}^{(1)}(k)), \cdots,$$
$$J(\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \bar{e}^{(L)}(k)) \Big). \quad (12)$$

The number of realizations L can be selected based on the desired level of accuracy and computational efficiency (Agili et al., 2012). The optimization problem (11) can be then rewritten as

$$\min_{\bar{u}(k), \bar{z}(k), \bar{\delta}(k), t(k)} t(k) \quad (13)$$

subject to

$$t(k) \geq J(\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \bar{e}^{(1)}(k))$$
$$\vdots$$
$$t(k) \geq J(\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \bar{e}^{(L)}(k))$$
$$\tilde{E}_1 x(0) + \tilde{E}_2 \tilde{u}(0:k:N) + \tilde{E}_3 \tilde{\delta}(0:k:N) + \tilde{E}_4 \tilde{z}(0:k:N)$$
$$+ \tilde{E}_5 \tilde{e}^{(1)}(0:k:N) \leq \tilde{g},$$
$$\vdots$$
$$\tilde{E}_1 x(0) + \tilde{E}_2 \tilde{u}(0:k:N) + \tilde{E}_3 \tilde{\delta}(0:k:N) + \tilde{E}_4 \tilde{z}(0:k:N)$$
$$+ \tilde{E}_5 \tilde{e}^{(L)}(0:k:N) \leq \tilde{g},$$

which can be solved as a single MILP problem.

Note that the above Monte Carlo approach is an approximation method that guarantees satisfaction of the specification with a high probability and not with probability equals one. This is due to the fact that it relies on a finite number of disturbance realizations. Moreover, computational efficiency of the Monte Carlo approach degrades with the number of sampled realizations, increasing the number of realizations results in larger number of constraints and thus a higher computational effort. In order to address these issues, we propose an alternative approach in the following to solve optimization problem (11).

4.2 Dual reformulation of the inner optimization problem

We transform the min-max optimization problem into a minimization one utilizing the (weak) dual reformulation of the inner optimization problem (M.Geoffrion, 1971; Wright, 1997). Such transformation results in an optimization problem that gives an upper bound on the original problem. As such, the overall optimization problem can be recast as an MIQP problem.

Note that encoding the STL specification (7) as a hard constraint induces linear constraints in the optimization problem (11) that should hold for all $\bar{e}(k) \in \mathscr{E}$ (cf. (8d)), and implicitly for all $\bar{\delta}_i(k)$, and $\bar{z}_i(k)$ for $i = 1, \ldots, 15$, since these variables are uniquely defined as a function of $\bar{e}(k)$ and $\bar{u}(k)$. A conventional way of dealing with constraints having universal quantifiers is the use of Farkas' lemma (Boyd and Vandenberghe, 2004) to replace them by equivalent constraints having existential quantifiers. However, expressing STL specifications as hard constraints prevents us to have such a transformation since Farkas' lemma does not apply to binary variables. One way to deal with this issue is to relax the binary variables and assume that they belong to the interval of $[0,1]$. Alternatively, by referring to Remark 2, we can only use robustness of the STL specification in the objective function and we do not add the specifications as hard constraints. Therefore, the optimization problem maximizes the robust satisfaction of the specification, which results in satisfaction of the STL specification if the optimal value of robustness is positive. Note that this way has an advantage over hard-constraint encoding of the specification. The optimization procedure does not terminate if the specification is not satisfiable for the closed-loop system. Instead, it tries to find control inputs that violate the specification the least.

Hence, we assume in the following that the robustness function of the STL specification φ is included in the objective function without φ being encoded as hard constraint in (11). As such, we first write the dual reformulation of the inner maximization problem in the following form:

$$\max_{\bar{e}(k)} \ C_1^T \bar{u}(k) + C_2^T \bar{z}(k) + C_3^T \bar{\delta}(k) + C_4^T \bar{e}(k) + \mu^T (q - S\bar{e}(k)) +$$
$$+ \lambda^T \big(\tilde{g} - \tilde{E}_1 x(0) - \tilde{E}_2 \tilde{u}(0:k:N) - \tilde{E}_3 \tilde{\delta}(0:k:N)$$
$$- \tilde{E}_4 \tilde{z}(0:k:N) - \tilde{E}_5 \tilde{e}(0:k:N) \big), \quad (14)$$

where μ and λ are the Lagrange multipliers. Note that for any choice of $\mu, \lambda \geq 0$, the solution of (14) is always greater than or equal to the solution of the inner maximization problem. Therefore, we over-approximate the inner optimization problem (maximization) by its (weak) dual formulation as

$$\min_{\mu, \lambda} \ \big(C_1^T - \lambda^T \tilde{E}_2 \big) \bar{u}(k) + \big(C_2^T - \lambda^T \tilde{E}_4 \big) \bar{z}(k)$$
$$+ \big(C_3^T - \lambda^T \tilde{E}_3 \big) \bar{\delta}(k) + \mu^T q + \lambda^T G \quad (15a)$$

subject to

$$S^T \mu + \tilde{E}_5^T \lambda = C_4, \quad \mu, \lambda \geq 0, \quad (15b)$$

where G contains all the constant terms that appear in the multiplier of λ in (14). Using the dual formulation of the inner optimization problem, (11) can be then replaced by

$$\min_{\bar{u}(k), \bar{z}(k), \bar{\delta}(k), \mu, \lambda} \ \big(C_1^T - \lambda^T \tilde{E}_2 \big) \bar{u}(k) + \big(C_2^T - \lambda^T \tilde{E}_4 \big) \bar{z}(k)$$
$$+ \big(C_3^T - \lambda^T \tilde{E}_3 \big) \bar{\delta}(k) + \mu^T q + \lambda^T G \quad (16a)$$

subject to

$$S^T \mu + \tilde{E}_5^T \lambda = C_4, \quad \mu, \lambda \geq 0, \quad (16b)$$

which is a (non-convex) MIQP problem due to terms $\lambda^T\big(\tilde{E}_2\bar{u}(k)+\tilde{E}_4\bar{z}(k)+\tilde{E}_3\bar{\delta}(k)\big)$ in (16a). To the best of our knowledge, the available solvers are not capable of handling non-convex MIQP problems. Therefore, we apply an approximation technique that is based on iteratively approximating the non-convex MIQP problem by an MILP problem using the cutting plane algorithm reported in Kelley (1960). Note that, since we are solving (16) using an approximation algorithm, the obtained results are only suboptimal and, in the case of having an infeasible solution, we cannot conclude anything about the feasibility of the optimization problem (16).

In the next section, we compare the dual reformulation with the Monte Carlo-based approach. We apply both approaches to the model of BTC with and without the STL specification to demonstrate the effectiveness of including the desired temporal behavior in the optimization using logical specifications.

5. SIMULATION RESULTS

We apply our proposed synthesis techniques to the model of the BTC presented in Figure 1. The system's matrices for the MLD representation of the network (10) are given in Appendix II. We have chosen the sampling time $\Delta t = 300$ s, according to the evolution of the real system. The chosen sampling time is related to the *time of concentration* [1] determined for the BTC by its management company.
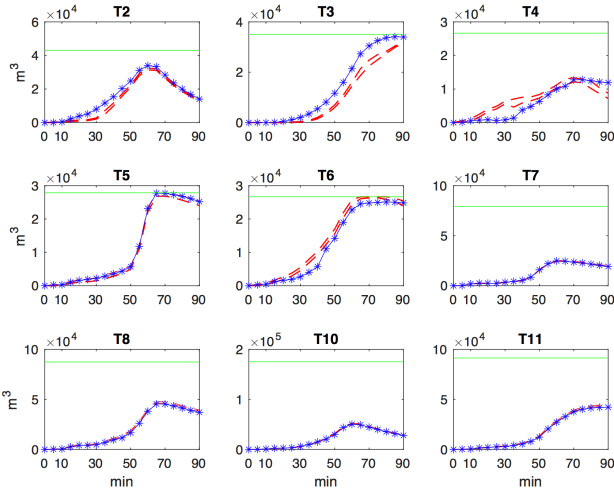
Fig. 3. Volumes of tanks $2,\ldots,11$. The dashed red lines correspond to the minimum and maximum trajectories of the Monte Carlo-based approach over 100 simulations; the starred blue line corresponds to the dual approach; the solid green line corresponds to the upper bound of the tank volumes.

Recall that the amount of rain entering the systems is defined as $W(k) = W_{\mathrm{ref}}(k)+e(k)$. The nominal values of the rain are $W_{\mathrm{ref}}(k) = [P_1(k),\alpha_2 P_2(k),\alpha_4 P_3(k),P_2(k),\alpha_{10}P_2(k),\alpha_5 P_3(k),\alpha_6 P_3(k),P_3(k),\alpha_8 P_3(k),\alpha_{11}P_4(k),\alpha_7 P_5(k)]^T$ with the rain intensities $P_i(k), i = 1,\ldots,5$ obtained based on the available data from the rain gauges in the real system. The parameter values $\alpha_2 = 0.5715, \alpha_4 = 0.1207, \alpha_5 = 0.3152, \alpha_6 = 0.1573, \alpha_7 = 0.6806, \alpha_8 = 0.1570, \alpha_{10} = 0.6935, \alpha_{11} = 0.6377$ are taken from (Ocampo-Martinez, 2010, Table 3.1). We assume

[1] Defined as the time required for a water drop to travel from the most remote catchment to its outlet to the receiving environment (Mays, 2004).
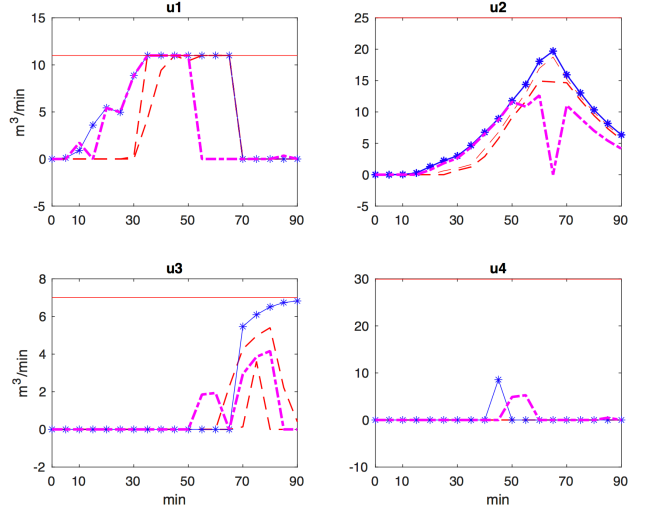
Fig. 4. Control (input) flows in the network. The dashed red lines correspond to the minimum and maximum trajectories of the Monte Carlo-based approach over 100 simulations; the starred blue line corresponds to the dual approach; the dash-dotted magenta line corresponds to the optimization problem without having STL specification; the solid red line corresponds to input upper bounds.

the unknown-but-bounded disturbances $e_i(k)$ are uniformly distributed over the interval $[0,1]$. Moreover, we have the following bounds on states and inputs:

$x_2(k) \in [0,43000], x_3(k) \in [0,35000], x_4(k) \in [0,26659],$
$x_5(k) \in [0,27854], x_6(k) \in [0,26659], x_7(k) \in [0,79229],$
$x_8(k) \in [0,87407], x_{10}(k) \in [0,175220], x_{11}(k) \in [0,91442],$
$u_1(k) \in [0,11], u_2(k) \in [0,25], u_3(k) \in [0,7], u_4(k) \in [0,29.93],$

where the states are in $\left[\mathrm{m}^3\right]$ and the flows are in $\left[\mathrm{m}^3/\mathrm{min}\right]$. Considering the MLD model, we select the objective function of the optimization problem (11) at each time step k as

$$J(k) = 0.1C_1'^T u(k) + 0.9\big(C_2'^T z(k) + C_3'^T \delta(k)\big) - 5\rho(x,k),$$

where C_1', C_2', C_3' denote the weighting matrices and $\rho(x,k)$ is the robustness function associated with the STL specification (7). We select $k' = 5$ for the STL specification, which means once the overflow occurs, the system should eliminate it within the next 5 time steps (20 minutes). We also select the prediction horizon $N = 18$ (90 minutes), which results in 92 input variables, 382 continuous auxiliary variables, and 597 binary auxiliary variables at the beginning of the optimization procedure.

The Monte Carlo approach of Section 4.1 and the dual reformulation of Section 4.2 are applied to the worst-case MPC optimization problem (11). The optimization problems are solved using the MILP solver from Gurobi in Matlab R2014b on a 2.6 GHz Intel Core i5 processor. The simulation results are presented in Figures 3-7.

Figures 3 and 4 illustrate the state and input (control) flow for each tank using both proposed approaches. The closed-loop simulation using the dual-reformulation approach takes 236.4 s for the first iteration. Note that the first time step is the longest one in the shrinking-horizon fashion; as the horizon decreases at each time step, the computational time reduces as well. In order to have a fair comparison, we fix the same simulation time for the Monte Carlo-based approach, which requires 150

samples from the uncertainty vector e. Hence, we repeated the Monte Carlo simulation 100 times, each time with 150 samples from the uncertainty vector e. The minimum and maximum of the resulting 100 trajectories are presented in Figures 3 and 4, and they indicate that the difference between these two trajectories is quite significant. Making this difference smaller needs increasing the number of samples, which results in a considerable increase in the computational time.

As shown in Figure 3, both states and inputs satisfy the constraints. Also, flows are guided such that volumes of most of the tanks, except tank T_3, eventually decrease while the one of tank T_3 increases. This is expected as those former tanks are virtual and it is preferred to keep them as empty as possible. However, tank T_3 is real and should work as a buffer in the wastewater system to handle the flow routing downstream while avoiding both overflow and pollution downstream. This fact, in turn, implies the maximization of the WWTP inflow. Note that the control flow trajectories obtained from the dual-reformulation approach are not as smooth as the ones obtained from the Monte Carlo-based approach. This effect could be the result of approximating the quadratic objective function and can be overcome either by adding an extra term in the cost function where the slew rate is penalized or by adding hard constraints to the optimization.

Remark 3. The longest computation time to obtain the optimal control belongs to the first iteration of the whole optimization problem along the considered simulation scenario (with the largest number of variables over N), which is 236.4 s. Since this computational time is less than $\Delta t = 300$ s, our method is applicable in practice. Note that the computational time can be further decreased for instance by having a more efficient solver for a non-convex MIQP problem instead of the iterative approximation with an MILP problem.
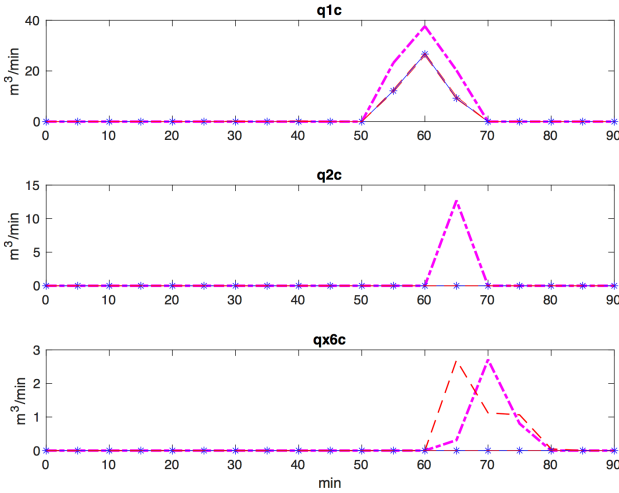

Fig. 5. Overflows in the network that enter the streets. The dashed red lines correspond to the minimum and maximum trajectories of the Monte Carlo-based approach over the 100 simulations; the starred blue line corresponds to the dual approach; the dash-dotted magenta line corresponds to thesolution of the optimization problem without having STL specification.

Figure 5 presents the overflows q_{1c}, q_{2c}, and $q_{x_{6c}}$ through streets. The other overflows, i.e., $q_{x_{2c}}, q_{x_{4c}}, q_{R1c}$, and q_{12Sc} are zero and are not plotted. In this figure, we present the tra-

jectories obtained using the dual-reformulation approach, the minimum and maximum trajectories obtained using the Monte Carlo-based approach, and the trajectories obtained without including the STL specification. The advantage of having an STL specification can be seen in this figure. Although, for the given rain profile, none of these overflows lasts longer than 25 minutes (equivalent to 5 time steps). In the case of having STL specification, the overflow in q_{1c} is much less than the case without STL specification and the overflow in q_{2c} is completely zero once we have STL specification. Additionally, all the other overflows become much less when we have STL specification. The overflow $q_{x_{6c}}$ is also zero in the dual-reformulation approach and varies between zero and the maximum trajectory in the Monte Carlo approach. If we compare the control flow of the cases with and without STL specification in Figure 4, we see that the decrease in the amount of overflows in the entire network happens by the increase in the amount of control flow once we have the STL specification. This increase in the control input is acceptable since our goal is to have as less flooding as possible in the network.

Figure 6 illustrates the flows q_7, q_{10}, q_{11} and the overflow $q_{x_{5c}}$ that pollute the Mediterranean sea. Again, we have not plotted q_{9c} and q_8 since they are null. The presented trajectories are similar to the ones in Figure 5 and confirm that the overflows are much less in the case of considering the STL specification in comparison with the case without the STL specification.
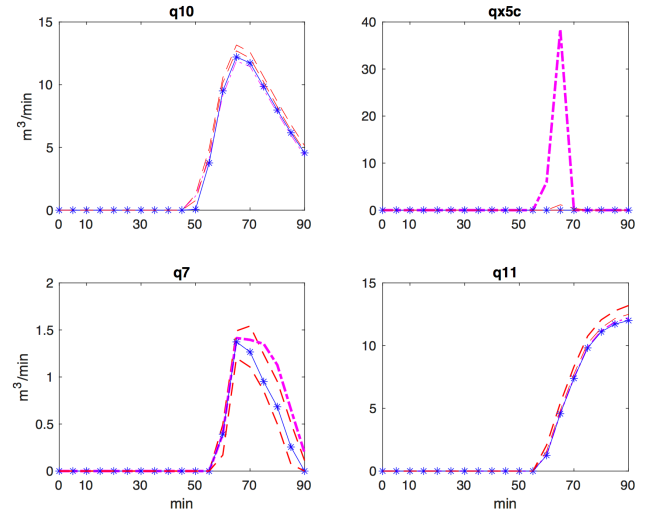

Fig. 6. Overflows and flows that enter Mediterranean sea. The dashed red lines correspond to the minimum and maximum trajectories of the Monte Carlo-based approach over the 100 simulations; the starred blue line corresponds to the dual approach; the dash-dotted magenta line corresponds to the optimization without having STL specification.

Finally, Figure 7 shows the WWTP inflows. The obtained trajectories have a similar trend of increasing and reaching their maximum in the last 30 to 40 minutes before the end of the simulation.

6. CONCLUSION

We proposed an effective control approach to manage the flow in a representative fragment of Barcelona wastewater network. We formulated the nonlinear dynamics of the flow
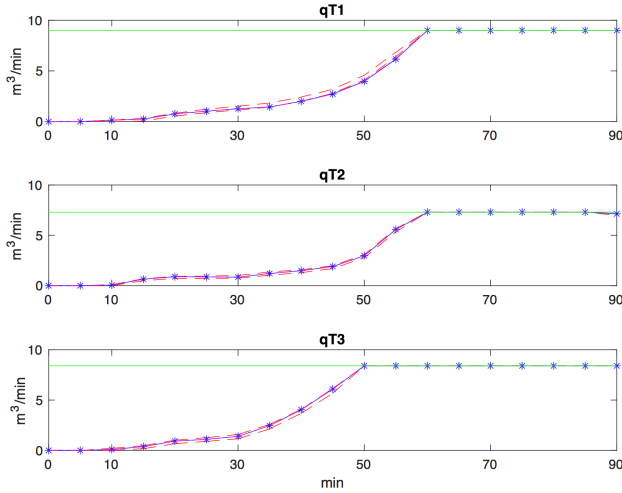
Fig. 7. Flows sent to the WWTPs. The dashed red lines corresponds to the minimum and maximum trajectories of the Monte Carlo-based approach over the 100 simulations, the starred blue line corresponds to the dual-reformulation approach, and the dash-dotted magenta line corresponds to the solution of the solution of the optimization problem without considering the STL specification.

network including the possible overflows in the considered catchment. The goal of the control scheme is to minimize the overflow in the main pipes and to maximize the amount of water treatment. We have employed model predictive control (MPC) to optimally manage the flow into the network. We have used signal temporal logic (STL) to specify the desired temporal behavior of the system with respect to the overflows. Additionally, we have included the uncertainty in the amount of precipitation as a bounded disturbance in the nonlinear model.

In order to solve the formulated worst-case MPC optimization problem with nonlinear state equations and objective function, we represented the dynamics of the network as a mixed logical dynamical (MLD) model. Moreover, we proposed two different techniques to approximately solve the optimization problem with the MLD model: (i) a Monte Carlo method, which approximates the solution of the inner optimization problem by taking samples from the disturbance set, and (ii) the dual reformulation, which transforms the min-max optimization problem into a minimization one. We showed that, by using these approaches, the resulting optimization problem could be recast as a mixed integer linear programming problem or as a non-convex mixed integer quadratic programming problem. We have further compared both proposed approaches in the simulations and have shown that the synthesized closed-loop system exhibits the desired temporal behavior.

In our future work, we consider performing a fair comparison between available optimization solvers, interfaced via Matlab or GAMS, in order to improve the computational efficiency of solving the formulated optimization problems.

ACKNOWLEDGEMENTS

REFERENCES

Agili, S.S., Morales, A.W., Li, J., and Resso, M. (2012). Finding the probability distribution functions of s-parameters and their Monte Carlo simulation. *IEEE Transactions on Instrumentation and Measurement*, 61(11), 2993–3002.

Allwright, J.C. (1994). On min-max model-based predictive control. In *Advances in Model-Based Predictive Control*, Oxford Press Inc., 415–426. New York, New York.

Badgwell, T.A. (1997). Robust model predictive control of stable linear systems. *International Journal of Control*, 68(4), 797–818.

Bemporad, A., Borrelli, F., and Morari, M. (2002a). Model predictive control based on linear programming — The explicit solution. *IEEE Transactions on Automatic Control*, 47(12), 1974–1985.

Bemporad, A., Heemels, W.P.M.H., and De Schutter, B. (2002b). On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control*, 47(5), 863–869.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.

Campo, P.J. and Morari, M. (1987). Robust model predictive control. In *Proceedings of American Control Conference (ACC)*, volume 2, 1021–1026.

Dua, V. and Pistikopoulos, E.N. (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99(1-4), 123–139.

Farahani, S.S., Majumdar, R., Prabhu, V., and Soudjani, S.E.Z. (2017a). Shrinking horizon model predictive control with chance-constrained signal temporal logic specifications. In *Proceedings of American Control Conference (ACC)*, 1740–1746. Seattle, WA.

Farahani, S.S., Raman, V., and Murray, R.M. (2015). Robust model predictive control for signal temporal logic synthesis. In *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*. Atlanta, Georgia.

Farahani, S.S., Soudjani, S.E.Z., Majumdar, R., and Ocampo-Martinez, C. (2017b). Robust model predictive control with signal temporal logic constraints for Barcelona wastewater system. In *Proceedings of IFAC World Congress*, 6778–6784. Toulouse, France.

Gelormino, M. and Ricker, N. (1994). Model predictive control of a combined sewer system. *International Journal of Control*, 59, 793–816.

Gutierrez, G., Ricardez-Sandoval, L.A., Budman, H., and Prada, C. (2014). An MPC-based control structure selection approach for simultaneous process and control design. *Computers & Chemical Engineering*, 70, 11–21.

Jin, X., Donzé, A., Deshmukh, J.V., and Seshia, S.A. (2013). Mining requirements from closed-loop control models. In *Hybrid Systems: Computation and Control, HSCC 2013, April 8-11, 2013, Philadelphia, PA, USA*, 43–52.

Joseph-Duran, B., Ocampo-Martinez, C., and Cembrano, G. (2014). Hybrid modeling and receding horizon control of sewer networks. *Water Resources Research*, 50(11), 8497–8514. doi:10.1002/2013WR015119.

Joseph-Duran, B., Ocampo-Martinez, C., and Cembrano, G. (2015). Output-feedback control of combined sewer networks through receding horizon control with moving horizon estimation. *Water Resources Research*, 51(10), 8129–8145.

Kelley, J.E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4), 703–712.

Kothare, M.V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10), 1361–1379.

Lazar, M., Heemels, M., Weiland, S., and Bemporad, A. (2006). Stability of hybrid model predictive control. *IEEE Transactions on Automatic Control*, 15(11), 1813–1818.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, England.

Maler, O. and Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. Lecture Notes in Computer Science*, volume 3253, 152–166. Springer, Berlin-Heidelberg, Germany.

Marinaki, M. and Papageorgiou, M. (2005). *Optimal Real-Time Control of Sewer Networks*. Springer-Verlag, London, UK.

Mayne, D.Q., Seron, M.M., and Raković, S.V. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2), 219–224.

Mays, L. (2004). *Urban stormwater management tools*. McGraw-Hill, New York, NY.

M.Geoffrion, A. (1971). Duality in nonlinear programming: A simplified applications-oriented development. *SIAM Review*, 13(1), 1–37.

Ocampo-Martinez, C. (2010). *Model Predictive Control of Wastewater Systems*. Springer-Verlag, London, UK.

Pardalos, P. and Resende, M. (2002). *Handbook of Applied Optimization*. Oxford University Press, Oxford, UK.

Raman, V., Donzé, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A.L., and Seshia, S.A. (2014). Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, December 15-17, 2014*, 81–87.

Raman, V., Donzé, A., Sadigh, D., Murray, R.M., and Seshia, S.A. (2015). Reactive synthesis from signal temporal logic specifications. In *Hybrid Systems: Computation and Control, HSCC 2015, Seattle, WA, USA, April 14-16, 2015*, 239–248.

Rawlings, J. and Mayne, D. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Pub.

Ricardez-Sandoval, L.A., Budman, H.M., and Douglas, P.L. (2010). Simultaneous design and control: A new approach and comparisons with existing methodologies. *Industrial & Engineering Chemistry Research*, 49, 2822–2833.

Schütze, M., Campisano, A., Colas, H., Schilling, W., and Vanrolleghem, P. (2004). Real time control of urban wastewater systems: where do we stand today. *Journal of Hydrology*, 299, 335–348.

Wright, S.J. (1997). *Primal-Dual Interior Point Methods*. SIAM, Philadelphia, Pennsylvania.

APPENDIX I: SIGNAL TEMPORAL LOGIC

A *run* of system (5) is defined as a signal $\xi = x(0)x(1)x(2)\dots$, which is an infinite sequence of states satisfying (5). Hence, a finite run of system (5) for the time interval $[0:N]$ can be defined as $\xi_N = x(0)x(1)\dots x(N)$. We consider STL formulas with bounded-time temporal operators defined recursively according to the grammar (Maler and Nickovic, 2004)

$$\varphi ::= \top \mid \pi \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi\mathscr{U}_{[a,b]}\psi,$$

where $\top$ is the *true* predicate, π is a predicate whose truth value is determined by the sign of a function, i.e., $\pi = \{\alpha(x) \geq 0\}$ with $\alpha : \mathbb{R}^n \to \mathbb{R}$ being an affine function of state variables; ψ is an STL formula; $\neg$ and $\wedge$ show negation and conjunction of formulas; and $\mathscr{U}_{[a,b]}$ is the *until* operator with $a,b \in \mathbb{R}_{\geq 0}$.

A run ξ satisfies φ at time k, denoted by $(\xi,k) \models \varphi$, if the sequence $x(k)x(k+1)\dots$ satisfies φ. Accordingly, ξ satisfies φ, if $(\xi,0) \models \varphi$.

Semantics of STL formulas are defined as follows. Every run satisfies $\top$. The run ξ satisfies $\neg\varphi$ if it does not satisfy φ; it satisfies $\varphi \wedge \psi$ if both φ and ψ hold. For a run $\xi = x(0)x(1)x(2)\dots$ and a predicate $\pi = \{\alpha(x) \geq 0\}$, we have $(\xi,k) \models \pi$ if $\alpha(x(k)) \geq 0$. Finally, $(\xi,k) \models \varphi\mathscr{U}_{[a,b]}\psi$ if φ holds at every time step starting from time k before ψ holds, and additionally ψ holds at some time instant between $a+k$ and $b+k$. Additionally, we derive the other standard operators as follows. *Disjunction* $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, the *eventually* operator as $\diamondsuit_{[a,b]} \varphi := \top\mathscr{U}_{[a,b]}\varphi$, and the *always* operator as $\square_{[a,b]} \varphi := \neg\diamondsuit_{[a,b]} \neg\varphi$. Thus $(\xi,t) \models \diamondsuit_{[a,b]} \varphi$ if φ holds at some time instant between $a+k$ and $b+k$ and $(\xi,k) \models \square_{[a,b]} \varphi$ if φ holds at every time instant between $a+k$ and $b+k$.

Formula Horizon. The *horizon* of an STL formula φ is the smallest $n \in \mathbb{N}$ such that the following holds for all signals $\xi = x(0)x(1)x(2)\dots$ and $\xi' = x'(0)x'(1)x'(2)\dots$:

$$\text{If } x(k+i) = x'(k+i) \text{ for all } i \in \{0,\dots,n\}$$
$$\text{Then } (\xi,k) \models \varphi \text{ iff } (\xi',k) \models \varphi.$$

Thus, in order to determine whether signal ξ satisfies an STL formula φ, we can restrict our attention to the signal prefix $x(0),\dots,x(\Delta)$, where Δ is the horizon of φ. This horizon can be upper-approximated by a bound, denoted by $\text{len}(\varphi)$, defined to be the maximum over the sums of all nested upper bounds on the temporal operators. The *bound* of φ, denoted by $\text{len}(\varphi)$, is defined as the maximum over the sums of all nested upper bounds on the temporal operators. Formally, $\text{len}(\varphi)$ is defined recursively as

$$\varphi := \top \Rightarrow \text{len}(\varphi) = 0, \qquad \varphi := \pi \Rightarrow \text{len}(\varphi) = 0,$$
$$\varphi := \neg\varphi_1 \Rightarrow \text{len}(\varphi) = \text{len}(\varphi_1),$$
$$\varphi := \varphi_1 \wedge \varphi_2 \Rightarrow \text{len}(\varphi) = \max(\text{len}(\varphi_1),\text{len}(\varphi_2)),$$
$$\varphi := \varphi_1 \mathscr{U}_{[a,b]} \varphi_2 \Rightarrow \text{len}(\varphi) = b + \max(\text{len}(\varphi_1),\text{len}(\varphi_2)),$$

where φ_1, φ_2 and ψ are STL formulas. For example, for $\varphi = \square_{[0,4]}\diamondsuit_{[3,6]} \pi$, we have $\text{len}(\varphi) = 4 + 6 = 10$. For a given STL formula φ, it is possible to verify that $\xi \models \varphi$ using only the finite run $x(0)x(1)\dots x(N)$, where N is equal to $\text{len}(\varphi)$.

STL Robustness. In contrast to the above Boolean semantics, the quantitative semantics of STL (Jin et al., 2013) assigns to each formula φ a real-valued function ρ^φ of signal ξ and k such that $\rho^\varphi(\xi,k) > 0$ implies $(\xi,k) \models \varphi$. Robustness of a formula φ with respect to a run ξ at time k is defined recursively as

$$\rho^\top(\xi,k) = +\infty,$$
$$\rho^\pi(\xi,k) = \alpha(x(k)) \text{ with } \pi = \{\alpha(x) \geq 0\},$$
$$\rho^{\neg\varphi}(\xi,k) = -\rho^\varphi(\xi,k)$$
$$\rho^{\varphi \wedge \psi}(\xi,k) = \min(\rho^\varphi(\xi,k),\rho^\psi(\xi,k)),$$
$$\rho^{\varphi \mathscr{U}_{[a,b]}\psi}(\xi,k) = \max_{i\in[a,b]} \left(\min(\rho^\psi(\xi,k+i), \min_{j\in[0,i)} \rho^\varphi(\xi,k+j))\right),$$

where $x(k)$ refers to signal ξ at time k. The robustness of the derived formula $\diamondsuit_{[a,b]} \varphi$ can be worked out to be $\rho^{\diamondsuit_{[a,b]} \varphi}(\xi,k) = \max_{i\in[a,b]} \rho^\varphi(\xi,k+i)$; and similarly for $\square_{[a,b]} \varphi$ as $\rho^{\square_{[a,b]} \varphi}(\xi,k) = \min_{i\in[a,b]} \rho^\varphi(\xi,k+i)$. The robustness of an arbitrary STL formula is computed recursively on the structure of the formula according to the above definition.

Mixed Integer Linear Encoding. To synthesize a run that satisfies an STL formula φ, we employ the *robustness-based* encoding of STL constraints to a mixed integer linear formulation, as in (Raman et al., 2014). We first represent the system trajectory as a finite sequence of states satisfying the model dynamics in (5). Then, we encode the formula φ with a set of mixed integer linear constraints. This encoding in possible due to the assumption that $\alpha(x)$ are affine functions of x.

Recall that the robustness function of an STL specification φ can be computed recursively on the structure of the formula. The max and min operations can be expressed in a mixed integer linear formulation using additional binary variables and a large constant M (commonly called big-M). The interested reader is referred to Raman et al. (2014) for details of this encoding, the gist of which follows. For brevity, denote $\rho^{\varphi}(\mathbf{x}, k)$ by ρ_k^{φ}; for a given formula φ, the mixed integer linear representation is extended with a variable ρ_k^{φ} and an associated set of constraints such that having $\rho_k^{\varphi} > 0$ under the added constraints is equivalent to the satisfaction of φ at time step k. This is accomplished by recursively generating mixed integer linear constraints for every subformula of φ according to its structure. In contrast to these STL constraints, the *system constraints* encode valid finite trajectories for a system of the form (5), and are designed to be satisfied if and only if the trajectory $\xi(x(0), \bar{u}(0), \bar{d}(0))$ obeys the dynamics in (5).

$$B_4 = \Delta t \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0 & 0 & \bar{x}_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta t\bar{q}_1 & \Delta t\bar{q}_2 & -\bar{x}_2 & \bar{x}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\Delta t\bar{q}_1 & -\Delta t\bar{q}_2 & 0 & \bar{x}_4 & 0 & \Delta t\bar{q}_{R1} & 0 & 0 & \bar{x}_5 & \Delta t\bar{q}_{R16} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{x}_6 & 0 & -\Delta t\bar{q}_{R16} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\Delta t\bar{q}_{R1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\Delta t\bar{q}_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta t\bar{q}_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\Delta t\bar{q}_{12S} & 0 & 0 & 0 & 0 & -\Delta t q_{R311} & 0 & 0 & 0 \end{bmatrix},$$

where I_9 denotes the 9×9 identity matrix. All the parameters are taken from Ocampo-Martinez (2010), Table 3.1.

APPENDIX II: MLD MODEL MATRICES

The system matrices of (10a) and related to the simulation results in Section 5 are as follows:

$$A = I_9 - \Delta t \begin{bmatrix} \beta_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\beta_2 & 0 & \beta_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\beta_4 & \beta_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\beta_5 & 0 & \beta_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\beta_6 & 0 & \beta_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{10} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{11} \end{bmatrix},$$

$$B_1 = \begin{bmatrix} \Delta t & 0 & 0 & 0 \\ 0 & \Delta t & -\Delta t & 0 \\ -\Delta t & -\Delta t & \Delta t & 0 \\ 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 - \Delta t \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad B_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\Delta t\bar{q}_{R16} \\ \Delta t\bar{q}_{R16} \\ 0 \\ \Delta t\bar{q}_{12} \\ 0 \\ \Delta t\bar{q}_{R311} \end{bmatrix},$$

$$B_3 = \Delta t \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \beta_8 & 0 & 0 & 0 \end{bmatrix},$$