Decentralized Control for Urban Drainage Systems using Replicator Dynamics

GERMÁN OBANDO[1], NICANOR QUIJANO[2], (IEEE Senior Member), and CARLOS OCAMPO–MARTINEZ[3], (IEEE Senior Member)

[1]School of Engineering, Science and Technology, Universidad del Rosario, Bogotá, Colombia (e-mail: german.obando@urosario.edu.co)
[2]Department of Electric and Electronics Engineering, Universidad de los Andes, Bogotá, Colombia (e-mail: nquijano@uniandes.edu.co)
[3]Automatic Control Department, Universitat Politècnica de Catalunya, Barcelona, Spain (e-mail: carlos.ocampo@upc.edu)

Corresponding author: G. Obando (e-mail: german.obando@urosario.edu.co).

ABSTRACT This paper proposes a decentralized control scheme that mitigates floods in urban drainage systems (UDSs). First, we develop a partitioning algorithm of the UDS relying on a graph model of the system. Once this is done, we design a local controller for each partition based on the replicator dynamics model (a set of differential equations that describes the evolution of a population of players involved in a strategic game). The decentralized nature of the proposed strategy makes it suitable for applying it in large-scale systems. Stability of the closed-loop system is proved by using Lyapunov theory. Furthermore, we simulate the performance of the decentralized control scheme in two case studies. One of them models part of the Bogotá (Colombia) stormwater UDS. Finally, we compare the proposed technique with two widely used methods for-real time control of UDSs, i.e., constrained linear quadratic regulator (LQR) and model predictive control (MPC).

INDEX TERMS Urban drainage systems, population dynamics, decentralized control, replicator dynamics, resource allocation.

I. INTRODUCTION

Urban drainage systems (UDSs) collect, store, and transport residual water to wastewater treatment plants and water bodies. UDSs are composed of an arrangement of channels which are connected by inspection and collection chambers. The wastewater flow along UDS is regulated by using measuring equipment (e.g., level and flow sensors) and actuators (e.g., valves and gates).

Normally, UDSs are designed for critical rain and demand scenarios up to certain magnitude [1]. However, in some situations, the wastewater flow that is conveyed by the UDS surpasses its capacity. For this reason, the system becomes saturated and flooding happens. In fact, some recent studies have shown that flood risk is increasing, specially in large cities [2], due to several factors, e.g., an increment of magnitude of extreme precipitation as a result of global warming, a rapid growth of population that stresses drainage infrastructures, and construction of civil infrastructure that has reduced the soil capacity to absorb rainwater [3]. Floods have severe impacts including loss of human life, destruction of property, and health problems.

For the reasons described above, flooding mitigation has gained broad research attention. There are two main approaches: the first one consists in upgrading UDSs infrastructure [4] (e.g., by the construction of new sewer collectors or re-sizing of the existing ones). The second approach is to use real-time control techniques that take advantage of the system's actuators to modify flow conditions throughout the UDS. Implementation of real-time controllers is appealing because it is more cost effective than enlarging existing drainage infrastructures. Therefore, several control techniques have been proposed [5]–[9]. One common approach is to develop simulation tools of UDSs and employing them as black-box models in heuristic-based optimization algorithms [10]–[13]. However, these algorithms lack a rigorous analysis that guarantees, for example, their stability. Another strategy relies on optimal control. For instance, [14] employs a multivariable linear quadratic regulator (LQR) that uses all the available storage capacity of the system and maximizes its outflow to treatment plants. Other authors have also used model predictive control (MPC) for UDSs management [15]–[19].

A widespread problem of most of the aforementioned techniques is that they require the design of a central authority in charge of the whole UDS. This centralized controller is responsible for handling all the actuators and processing information from all system's sensors. Since the size of UDSs is generally of large-scale nature, implementing a centralized controller is problematic. Instead, it is desirable to design decentralized control schemes, i.e., schemes with multiple controllers, where each one of them has partial information of the UDS. Decentralized control of UDS has also been tackled in the literature. For instance, using rule based controllers, as in [20], where fuzzy logic is employed to manage the wastewater level of storage units. Nevertheless, this approach does not consider any coordination of storage units, which largely reduces the benefits of the strategy [21]. On the other hand, some authors have used distributed versions of model predictive control to overcome the requirement of a centralized authority (e.g., see [22]). However, in these schemes the computation of control signals requires high-performance processors because of the complexity of existing algorithms.

Recently, game theory and population dynamics based controllers have gained relevance due to their low computational burden and simplicity for designing decoupled schemes using them (e.g., see [23]–[29]). In fact, there are multiple engineering problems that have been addressed by means of game theory and population dynamics. For instance, coordination of robot networks [30], wind farm optimization [31], demand response in electrical grids [32], traffic assignment [33], charging of large populations of electric vehicles [34], control of epidemics [35], and so forth. It is important to point out that population dynamics based controllers have been applied to water allocation problems. For instance, [36] poses a game between the water resource manufacturers and regulators to find an efficient water allocation in a river basin. In particular, population dynamics have also been used to control UDSs. As described in [37], this kind of control strategy has shown suitable performance for mitigating floods.

Population dynamics play the key role in the control strategy that we propose in this article. Specifically, we develop a decentralized architecture that divides the UDS in subsystems handled by local controllers. Each one of these controllers implements a population dynamics algorithm, which is designed to make an efficient wastewater allocation among the storing units of the sub-system. Efficiency of the proposed strategy is proved using Lyapunov analysis. Furthermore, we test the performance of the controlled UDS by means of simulations under two scenarios. One of them is a case study that uses the model of part of the Bogotá (Colombia) stormwater UDS.

It is worth mentioning that, compared to [23] and [37], where a similar control strategy is used, our paper has three main advantages: First, the formalization of a partitioning algorithm that covers a wider range of UDSs topologies. Second, the wastewater allocation process performed by each local controller takes into account the available capacity of all storing units of the handled sub-system (this is not the case of the strategy proposed in [23] and [37] where only upstream storing units are involved in the resource allocation process). Finally, we have removed avoidable constraints on the control signals that are often imposed when implementing population dynamics based algorithms (specifically, in population dynamics based controllers, the sum of control actions are generally required to remain constant along the time). These constraints unnecessarily reduce the degrees of freedom of control actions.

Regarding partitioning methods that divide UDSs into subsystems, some procedures have been reported in the literature. For instance, [38] proposes a cascade configuration of MPC controllers. First, the UDS is divided into subsystems from downstream to upstream reservoirs. Then, the control strategy starts by computing the outputs of controllers in charge of downstream sub-systems. Once this is done, information is sent to upstream sub-systems so that their controllers can compute the control signals. A drawback of this strategy is that parallelization is not possible since upstream controllers require the information of downstream controllers. Another partitioning method is the one described in [39], where the authors develop an algorithm based on the Strahler number. The idea is to decompose the UDS in hierarchically interconnected sub-systems to increase the system's resilience. This strategy is intended to be implemented at the planning stage of the UDS rather than in its operation stage. Hence, the resulting partitions do not fit well with real-time controllers. On the other hand, [40] develops a gossip-based algorithm that coordinates local controllers in a UDS divided into sub-systems. However, the partitioning method employed by the authors can only be applied to UDSs with tree topologies. Furthermore, [41] devises graph partitioning algorithms to group sections of a UDS into semi-distributed sub-systems. Although these algorithms perform well to accelerate accurate simulations of UDSs, they are not designed for control purposes. Other authors use a geographical proximity criterion for UDSs partitioning [42], which minimizes the cost of the communication infrastructure required by each local controller. Nevertheless, this criterion does not take into account the information requirements of the controllers. Hence, the performance of the closed-loop system may worsen. Another partitioning paradigm that emerges in optimal control strategies for UDSs consists in dividing the optimization problem into sub-problems (which are related to sub-systems of the UDS) [22]. Each sub-problem is solved by a local controller while all the solutions are coordinated by means of a distributed algorithm (e.g., the alternating direction method of multipliers (ADMM) [43]). The main drawback of this paradigm is that coordination algorithms require a communication network that connects all subsystems (generally, such networks are modeled as connected graphs). In addition, convergence rates of distributed methods are quite lower compared to centralized approaches. Finally, some papers also propose to separate the collection of wastewater and surface run-off at the planning stage (see e.g.,

[44]). This division alleviates the risk of overflows. However, separate drainage systems require higher infrastructure and maintenance costs [45]. Beyond the drawbacks that we have pointed out for each of the cited partitioning methods, the key problem is that none of these methods produces subsystems with topologies that fit the controller proposed in this article. Thus, our partitioning algorithm is a central piece in the design of the management strategy presented in this paper. We want to highlight that our partitioning algorithm is oriented to decompose the UDS into sub-systems that use a local control policy, where each sub-system is capable to compute the control signals independently of the state of the other sub-systems. This fact offers inherent resilience to failures (which is typical of decentralized control schemes) because the malfunction of a subsystem does not compromise the performance of the entire UDS.

In addition, the fact that the control strategy developed in this paper uses population dynamics leads to advantages compared to other decentralized methods. On the one hand, population dynamics only requires the evaluation of a set of algebraic expressions to update the control actions. Therefore, compared to optimization-based techniques, our approach avoids numerical issues raised by the employment of optimization solvers. Indeed, in Section VII, we show by means of simulations that the strategy proposed in this article outperforms some optimization-based approaches and decreases the computational burden because of the different control paradigm employed. On the other hand, the analytical nature of population dynamics allows us to perform mathematical analysis for guaranteeing key features of the closed-loop controlled system such as stability. This feature is missing in some UDS control methods, especially in those based on heuristics.

The remainder of this paper is organized as follows: Section II covers notation and some general concepts about population dynamics. In Section III, we describe the problem statement and the model of the UDS employed for developing the control strategy. Sections IV and V show the partitioning algorithm and the design of local controllers using population dynamics, respectively. In addition, a stability analysis of the controlled system is presented in Section VI. Furthermore, Section VII discusses the simulation results obtained when the proposed strategy is implemented in two case studies. This section also presents comparisons with commonly used methods for real-time control of UDSs. Specifically, we compare our method with two optimization-based controllers, i.e., constrained LQR and MPC. Finally, some conclusions are drawn in Section VIII.

II. PRELIMINARIES

A. NOTATION

Throughout this article, bold and calligraphic letters denote vectors and sets, respectively. Variables that depend on time are explicitly written as functions of t, e.g., $x(t)$. As usual, the cartesian product of two sets $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} \times \mathcal{B}$, and the number of elements of the set $\mathcal{A}$ is denoted by $|\mathcal{A}|$. Nonnegative and positive real numbers are denoted by $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$, respectively. Finally, $\mathbb{R}^n_{\geq 0}$ is the set of vectors of size n whose entries belong to $\mathbb{R}_{\geq 0}$.

B. REPLICATOR DYNAMICS MODEL

The problem of flooding can be mitigated if wastewater is efficiently distributed in the storing units (chambers) of the UDSs. Consequently, an appropriate wastewater allocation mechanism is required. If we consider the sewage in the UDS as a resource, traditional resource allocation methods (such as those based on static optimization) are not suitable to address the problem under consideration because the resource is not static. Indeed, UDS is continuously receiving (e.g., from rain) and discharging sewage. Thus, a dynamic resource allocation mechanism, as the case of the replicator dynamics, is required. Broadly speaking, replicator dynamics describe how a mass of individuals (e.g., animals, agents) tries to efficiently distribute itself among different strategies (e.g., habitats, actions). Efficiency of the distribution is measured by means of payoff functions known as fitness functions.

For a better understanding of the role that replicator dynamics play in wastewater allocation, let us consider the following analogy. Assume that the mass of individuals is the wastewater of the UDS and the strategies are the chambers of the system. Under these assumptions, the key idea is to design proper fitness functions in such a way that the replicator dynamics allocate the sewage (mass of individuals) in emptier chambers (more efficient strategies).

Since the control strategy proposed in this article is based on the replicator dynamics, it is necessary to describe its mathematical model (introduced by Taylor and Jonker in 1978 [46]). The replicator equation is a classic model that captures the natural selection process in a population of individuals that can choose between different habitats or strategies. The key elements of the model are: A set of available strategies, which is denoted by $\mathcal{S} = \{1, \dots, s\}$ (in this case we have s strategies); a population mass P; a set of variables $p_1(t), \dots, p_s(t)$, where $p_i(t)$ describes the portion of individuals choosing the i-th strategy at time t; and a population state $\boldsymbol{p}(t) = [p_1(t), \dots, p_s(t)]^\top$. Formally, the replicator dynamics model is given by the following differential equation:

$$\dot{p}_i(t) = \beta p_i(t)\Big(f_i(t) - \bar{f}(t)\Big), \quad i \in \mathcal{S}, \tag{1}$$

where $f_i(t)$ is a real-valued fitness function, which captures the payoff perceived by the individuals playing the i-th strategy, $\bar{f}(t) = \frac{1}{P}\sum_{j\in\mathcal{S}} p_j(t) f_j(t)$ is a weighted average fitness, and $\beta \in \mathbb{R}_{>0}$ is the population growth rate. Notice that according to (1), populations of most successful individuals (i.e., individuals that play strategies with higher-than-the-average fitness functions) tend to grow, while least successful populations decrease. This behavior is closely related with the biological mechanism of natural selection, i.e., survival-of-the-fittest. One important property of this model is next described in Lemma 1.

Lemma 1 ([47]). *The simplex*

$$\Delta = \left\{ \boldsymbol{p}(t) \in \mathbb{R}^n_{\geq 0} : \sum_{i\in\mathcal{S}} p_i(t) = P \right\}$$

is positively invariant under replicator dynamics (1).

The importance of Lemma 1 lies in the fact that it guarantees the evolution of the population state $\boldsymbol{p}(t)$ in the nonnegative orthant provided that $\boldsymbol{p}(0) \in \mathbb{R}^n_{\geq 0}$. Furthermore, the population mass P is preserved along the time. These two properties of $\boldsymbol{p}(t)$ will be exploited in the controller design.

Another key feature of the replicator equation is that the fitness functions of non-extinct strategies[1] reach the same value at equilibrium. Formally, if $p_i^*, p_j^* \in \mathbb{R}_{>0}$, then $f_i^* = f_j^*$, for all $i, j \in \mathcal{S}$, where p_i^* and f_i^* denote the values of $p_i(t)$ and $f_i(t)$ in steady-state, respectively. This fact motivates the use of replicator dynamics in problems where the objective is to attain consensus with respect to a certain variable. For instance, the temperature of the rooms in a building [48], the frequency of electric generators [49], the illuminance level in offices with several lamps [23], or the remaining storage capacity in multi-tank systems, which is the case of this article.

Including Populations Upper Bounds

The population of individuals playing each one of the available strategies is lower-bounded by zero under the replicator dynamics in (1) (cf., Lemma 1). Nevertheless, this portion is not bounded above by any quantity lower than the population mass P. Although upper bounds are not considered in (1), they can be included by adding in the formulation of the replicator equation a term related to the maximum size of each population, i.e., a term that prevents $p_i(t)$ to exceed an upper bound $\bar{p}_i$. The described modification, which has been explored in [50], is specifically done in the fitness functions as follows:

$$f_i(t) = \hat{f}_i(t) + \varepsilon b_i(p_i(t), \bar{p}_i), \quad i \in \mathcal{S}, \tag{2}$$

where $\hat{f}_i(t)$ is the payoff associated with the i-th strategy without considering upper bound, $\varepsilon \in \mathbb{R}_{\geq 0}$ is a small nonnegative scale factor, and $b_i(p_i(t), \bar{p}_i)$ is a function that prevents $p_i(t)$ to exceed the upper bound $\bar{p}_i$. Modified fitness function has the following characteristics:

P1. $b_i : [0, \bar{p}_i) \times (0, m) \mapsto \mathbb{R}$ is a Lipschitz continuous and monotonically decreasing function with respect to its first argument.

P2. $b_i(p_i(t), \bar{p}_i) \to -\infty$ as $p_i(t) \to \bar{p}_i$.

The behavior of the population playing the i-th strategy (i.e., $p_i(t)$) under the fitness functions in (2) can be analyzed considering two cases. The first case is when the variable $p_i(t)$ is far from $\bar{p}_i$. Notice that it is possible to choose a sufficient small value for ε such that the effect of $b_i(p_i(t), \bar{p}_i)$ on the fitness $f_i(t)$ is negligible if $p_i(t)$ is far from $\bar{p}_i$. In this case, the fitness function is quite close to the payoff associated with the i-th strategy without considering an upper bound, i.e., $f_i(t) \approx \hat{f}_i(t)$ according to (2). The second case is when $p_i(t)$ tends to the value $\bar{p}_i$. Under this scenario, although the value of ε is small, the function $\varepsilon b_i(p_i(t), \bar{p}_i)$ takes an arbitrarily large negative value due to P2. Therefore, the fitness $f_i(t) \to -\infty$. Notice that since $f_i(t) \to -\infty$, the payoff perceived by the i-th population is lower than the average, i.e., $f_i(t) < \bar{f}(t)$. Hence, $\dot{p}_i(t) < 0$ $\big($cf., (1)$\big)$, and thus $p_i(t)$ does not exceed its corresponding upper bound $\bar{p}_i$. Summarizing, $p_i(t) \in [0, \bar{p}_i)$. The previous discussion is illustrated in Figure 1.

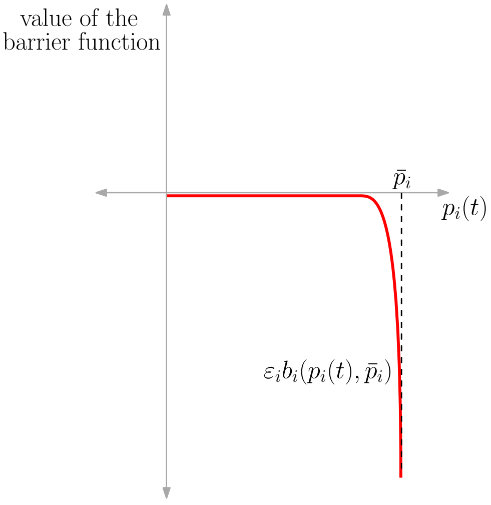


Figure 1. Behavior of the function $b_i(p_i(t), \bar{p}_i)$, which prevents the population playing the i-th strategy moves outside its upper bound $\bar{p}_i$.

III. URBAN DRAINAGE SYSTEMS

A. PROBLEM STATEMENT AND OVERVIEW OF THE PROPOSED SOLUTION

We address the problem of overflooding in UDSs. This problem occurs when the wastewater exceeds the storage capacity of any of the UDS's collectors. Therefore, overflooding can be avoided if wastewater is efficiently allocated among all the UDS's collectors by handling the system's actuators (valves in our case). Reaching an efficient wastewater allocation is the key idea of the proposed control strategy. Specifically, we seek that all collectors have the same remaining capacity for storing wastewater. In other words, our objective is trying that all UDS's collectors have the same empty space during mostly of the time. If this happens, no collector will be overloaded compared to the other ones. Therefore, no flooding occurs provided that there is available storage capacity in the UDS. On the other hand, UDSs are of large scale nature. Hence, decentralized control strategies suit best for these kinds of systems, i.e., we are looking to design a set of controllers, where each controller is in charge of part of the UDS. A constraint that must be satisfied by local controllers is that they are only able to use information from the subsystem they are handling. For attaining the control objective

[1] Let $\delta \in \mathbb{R}_{>0}$. The strategy $i \in \mathcal{S}$ is called non-extinct (resp. extinct) if $p_i(t) \to \delta$ (resp. $p_i(t) \to 0$) as $t \to +\infty$.

while meeting the local information constraint, we divide our strategy in two main stages: first, the UDS is partitioned in sub-systems that have some desired characteristics, and then a local controller based on population dynamics is designed for each sub-system obtained in the partitioning stage.

The partitioning stage uses a graph model of the UDS while the controllers design is based on a dynamic model derived from the mass conservation principle. Let us describe these two models.

B. GRAPH MODEL OF A UDS

Using the ideas in [23], we represent each collector of the UDS as a reservoir. A UDS with m reservoirs is modeled as a connected and directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, m\}$ is the set of nodes of the graph, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. In our case, the nodes are the reservoirs of UDS, and the edges represent connections between these reservoirs, e.g., the edge $(i, j) \in \mathcal{E}$ models that there is a connection allowing wastewater flow from the i-th reservoir to the j-th reservoir. We also assume that each edge has a valve that controls wastewater flow from one reservoir to another.

Assumption 1. *The UDS does not have bidirectional flows, i.e., if $(i, j) \in \mathcal{E}$, then $(j, i) \notin \mathcal{E}$.*

The UDS can be divided into sub-systems. Each sub-system belongs to one of the following two topologies.

Definition 1. ***(Divergent topology)*** *The divergent topology involves $n + 1$ reservoirs, and represents a situation in which there are $n > 1$ receptor reservoirs and only one source reservoir. Consequently, in a divergent topology there are n manipulated flows diverging from a unique source reservoir. Figure 2(a) shows a scheme representing such a topology. Formally, a sub-system with divergent topology can be represented as a graph where the nodes are the source reservoir and the receptor reservoirs, and the edges are the links between the source and receptor reservoirs.* ◇

Definition 2. ***(Convergent topology)*** *The convergent topology involves $n + 1$ reservoirs, and represents a situation in which there are $n \geq 1$ source reservoirs and only one receptor reservoir. Consequently, in a convergent topology there are n manipulated flows converging to a unique receptor reservoir. Figure 2(b) shows a convergent topology. Formally, a sub-system with divergent topology can be represented as a graph where the nodes are the receptor reservoir and the source reservoirs, and the edges are the links between the source and receptor reservoirs.* ◇

Remark 1. *If one sub-system has only one source reservoir and one receptor reservoir (see Figure 2(c)), then it can be represented as a convergent topology. Notice that divergent topologies require at least two receptor reservoirs.*

C. DYNAMIC MODEL OF RESERVOIRS

We use the Muskingum model [51], which is based on the principle of mass conservation (this principle is widely used in the literature for control-oriented models [52]). Specifically, we represent each channel of the UDS as a reservoir containing a wastewater volume $v_i(t)$. Besides, we assume that each channel has a set of output gates whose opening percentage can be controlled. Dynamics of the wastewater volume of the i-th reservoir can be described by the following differential equation:

$$\dot{v}_i(t) = q_{in,i}(t) - q_{out,i}(t), \tag{3}$$

where $q_{in,i}(t)$ and $q_{out,i}(t)$ are the reservoir inflow and outflow, respectively. Modeling of inflow and outflow depends on the topology of the sub-system to which the reservoir belongs. In Section V, we delve into the dynamic model of reservoirs depending on sub-systems topologies.

It is worth noting that convergent topologies have been addressed before in [23] and [37]. However, these papers do not consider the possibility that a UDS has sub-systems with divergent topology. Therefore, our approach covers a wider range of UDSs architectures.

IV. SYSTEM PARTITIONING

Partitioning consists in dividing the whole UDS in sub-systems belonging to the two topologies described before. Besides, partitioning must satisfy the following constraints:

C1. All resulting sub-systems must belong to one and only one topology (convergent or divergent). This is required because we only design two types of controllers, one per each topology.

C2. All reservoirs of the UDS must belong to at least one sub-system. If one reservoir does not belong to any sub-system, its wastewater volume would not be controlled.

C3. Each edge of the graph that models the UDS must belong to one and only one sub-system. This constraint prevents ambiguity in control actions as explained in Section V.

Any partitioning algorithm satisfying these three constraints is suitable for the splitting of UDS. In particular, we propose an algorithm consisting in three main steps: *1)* identification of divergent topologies; *2)* removal of the edges that belong to divergent topologies; and *3)* identification of convergent topologies.

Before stating the partitioning algorithm, let us define two concepts related to directed graphs.

Definition 3. ***In-neighborhood of a node:*** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph. The in–neighborhood of the node $i \in \mathcal{V}$, which is denoted by $\mathcal{N}_i^{in}(\mathcal{G})$, is the set of nodes satisfying the following property:*

$$\mathcal{N}_i^{in}(\mathcal{G}) = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}.$$

Notice that if $\mathcal{G}$ is a graph that models a UDS, then $\mathcal{N}_i^{in}(\mathcal{G})$ is the collection of all reservoirs having an outflow that goes to the i-th reservoir.

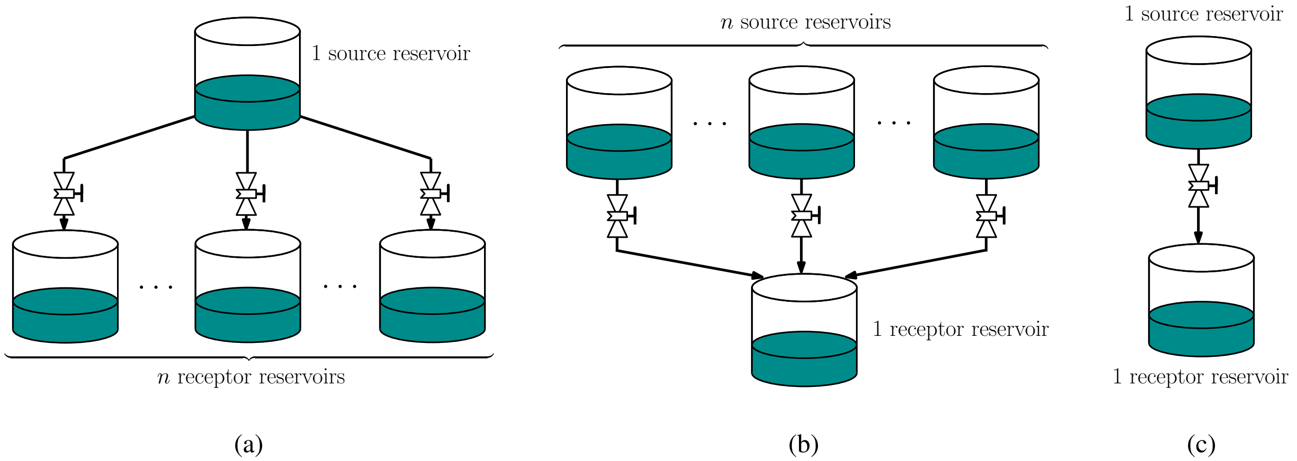


Figure 2. Types of possible sub-systems: (a) divergent topology, and (b) convergent topology. (c) Special case associated with the convergent topology.

Definition 4. ***Out-neighborhood of a node:*** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph. The out–neighborhood of the node $i \in \mathcal{V}$, which is denoted by $\mathcal{N}_i^{out}(\mathcal{G})$, is the set of nodes satisfying the following property:*

$$\mathcal{N}_i^{out}(\mathcal{G}) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}.$$

Notice that $\mathcal{N}_i^{out}(\mathcal{G})$ is the collection of all reservoirs having an inflow that comes from the i-th reservoir.

Using these definitions, we describe the three stages of our partitioning algorithm.

A. STEP 1 (IDENTIFICATION OF DIVERGENT TOPOLOGIES)

According to Definition 1, a sub-system having divergent topology has one source node and several (at least two) receptor nodes. Hence, for identifying all the UDS subsystems with divergent topology, it is enough to find all the source nodes having more than one out-neighbor.

Formally, the set of source nodes of divergent topologies, which we denote by $\mathcal{V}_s$, is

$$\mathcal{V}_s = \left\{i \in \mathcal{V} : \left|\mathcal{N}_i^{out}(\mathcal{G})\right| > 1\right\}.$$

The receptor nodes associated with node $i \in \mathcal{V}_s$ are $\mathcal{N}_i^{out}(\mathcal{G})$.

Thus, a sub-system with divergent topology associated to the source node $i \in \mathcal{V}_s$ is the graph $\mathcal{G}_i^{div} = \left(\mathcal{V}_i^{div}, \mathcal{E}_i^{div}\right)$. The nodes of $\mathcal{G}_i^{div}$ are the source node and its receptors, i.e., $\mathcal{V}_i^{div} = \{i\} \cup \mathcal{N}_i^{out}(\mathcal{G})$. Besides, the edges of $\mathcal{G}_i^{div}$ are the connections between the source node and its receptors, i.e., $\mathcal{E}_i^{div} = \{i\} \times \mathcal{N}_i^{out}(\mathcal{G})$.

B. STEP 2 (REMOVAL OF THE EDGES BELONGING TO DIVERGENT TOPOLOGIES)

From the graph that models the UDS, we create a new graph subtracting all edges belonging to divergent topologies. Formally, the new graph is $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$, where the set of nodes is the same as the one of $\mathcal{G}$, and the set of edges is $\mathcal{E}' = \mathcal{E} / \left(\bigcup_{i \in \mathcal{V}_s} \mathcal{E}_i^{div}\right)$.

C. STEP 3 (IDENTIFICATION OF CONVERGENT TOPOLOGIES)

Similarly as in Step 1, we identify the sub-systems having convergent topology. However, we now use the new graph $\mathcal{G}'$. We recall that a sub-system having divergent topology has one receptor node and at least one source node (see Definition 2). Hence, for identifying all the UDS subsystems with convergent topology, it is enough to find all the receptor nodes having at least one in-neighbor.

Formally, the set of receptor nodes of convergent topologies, which we denote by $\mathcal{V}_r$, is

$$\mathcal{V}_r = \left\{i \in \mathcal{V} : \left|\mathcal{N}_i^{in}(\mathcal{G}')\right| \geq 1\right\}.$$

The source nodes associated with node $i \in \mathcal{V}_s$ are $\mathcal{N}_i^{in}(\mathcal{G}')$.

Hence, a sub-system with convergent topology associated to the receptor node $i \in \mathcal{V}_r$ is the graph $\mathcal{G}_i^{cnv} = (\mathcal{V}_i^{cnv}, \mathcal{E}_i^{cnv})$. The nodes of $\mathcal{G}_i^{cnv}$ are the receptor node and its sources, i.e., $\mathcal{V}_i^{cnv} = \{i\} \cup \mathcal{N}_i^{in}(\mathcal{G}')$. Besides, the edges of $\mathcal{G}_i^{cnv}$ are the connections between the receptor node and its sources, i.e., $\mathcal{E}_i^{cnv} = \{i\} \times \mathcal{N}_i^{in}(\mathcal{G}')$.

All the partitioning algorithm steps described above are summarized in Figure 3.

D. PARTITIONING ALGORITHM EXAMPLE

To see how the partitioning algorithm works, we apply it in the UDS example shown in Figure 4(a), whose associated graph is depicted in Figure 4(b). The three steps of the partitioning algorithm are illustrated in Figure 5. Next, we describe each step.

Figure 5(a) shows the result of applying the first step of the algorithm. All nodes having more than one out-neighbor (in this case, nodes 1 and 4) are source nodes of a divergent topology, i.e., $\mathcal{V}_s = \{1, 4\}$. Receptors of node 1 are nodes 3 and 4, and receptors of node 4 are nodes 5 and 6. Therefore, there are two sub-systems with divergent topology: $\mathcal{G}_1^{div}$ shown in blue, and $\mathcal{G}_4^{div}$ shown in red.

Figure 5(b) shows the result of applying the second step of the algorithm. All edges of such sub-systems with diver-

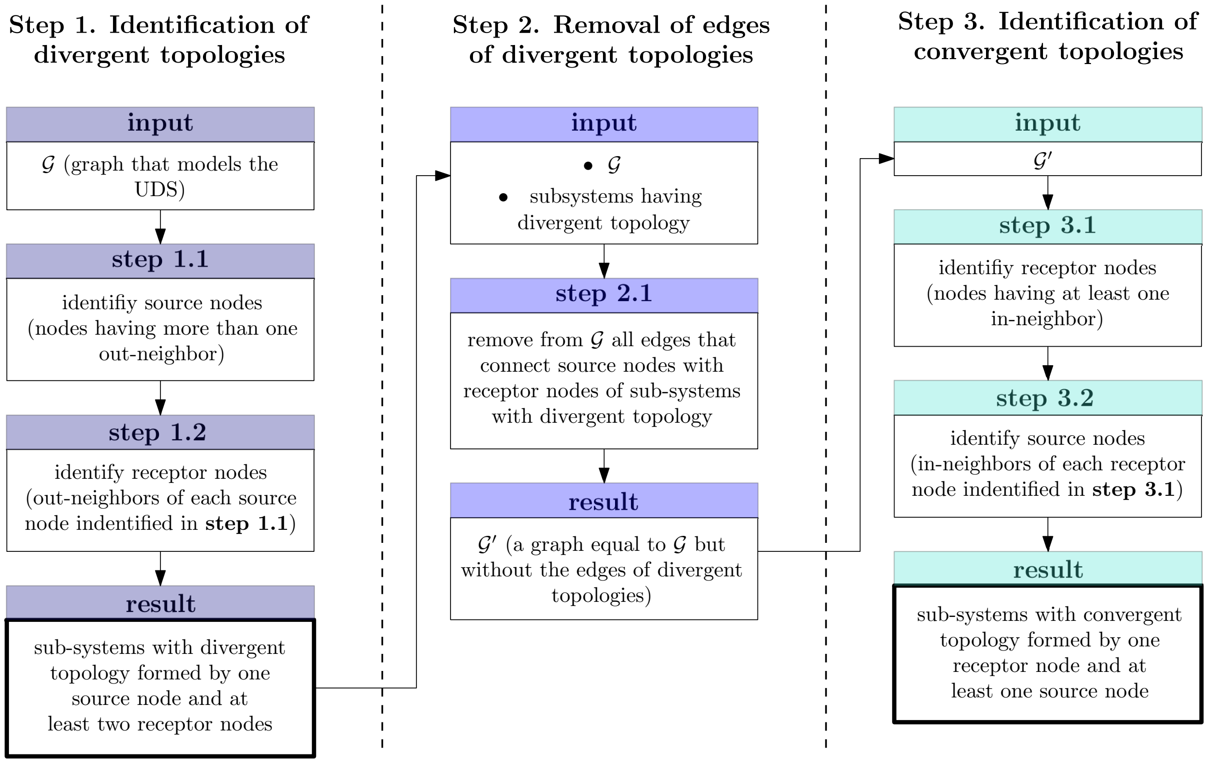

Step 1. Identification of divergent topologies
input
$\mathcal{G}$ (graph that models the UDS)
step 1.1
identifiy source nodes (nodes having more than one out-neighbor)
step 1.2
identify receptor nodes (out-neighbors of each source node indentified in **step 1.1**)
result
sub-systems with divergent topology formed by one source node and at least two receptor nodes
Step 2. Removal of edges of divergent topologies
input
• $\mathcal{G}$
• subsystems having divergent topology
step 2.1
remove from $\mathcal{G}$ all edges that connect source nodes with receptor nodes of sub-systems with divergent topology
result
$\mathcal{G}'$ (a graph equal to $\mathcal{G}$ but without the edges of divergent topologies)
Step 3. Identification of convergent topologies
input
$\mathcal{G}'$
step 3.1
identifiy receptor nodes (nodes having at least one in-neighbor)
step 3.2
identify source nodes (in-neighbors of each receptor node indentified in **step 3.1**)
result
sub-systems with convergent topology formed by one receptor node and at least one source node

Figure 3. Diagram of the proposed partition algorithm. Algorithm's outputs are enclosed in thick line boxes.

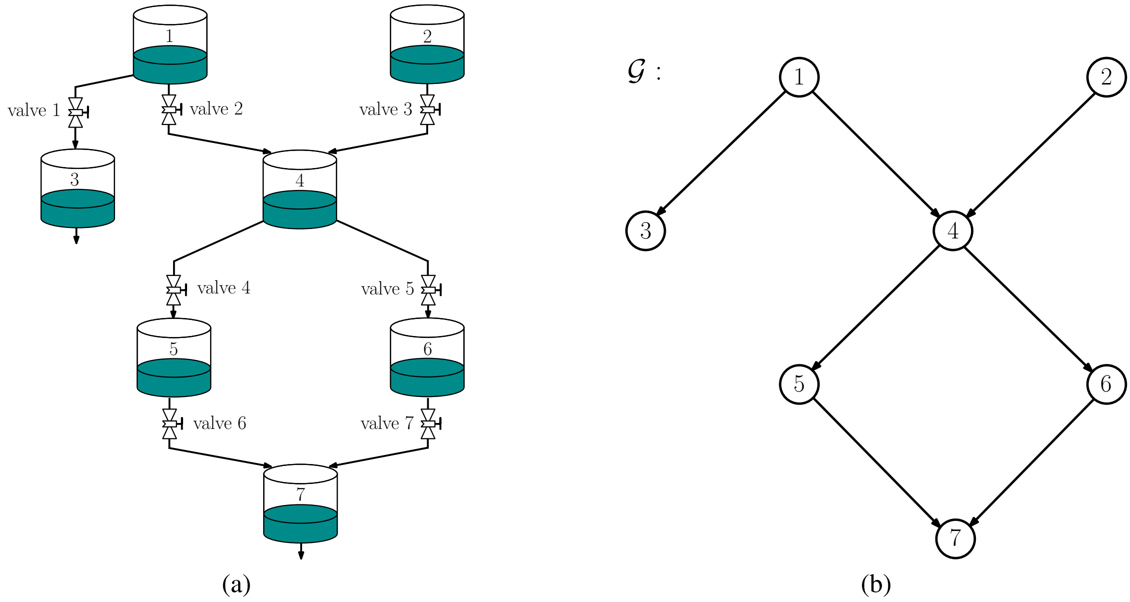


Figure 4. (a) UDS topology of the illustrative example, and (b) its graph representation.

gent topology (which are depicted using dashed lines) are subtracted from the original graph.

Finally, Figure 5(c) shows the result of applying the third step of the algorithm. Nodes having at least one in-neighbor (in this case, nodes 4 and 7) are receptor nodes of convergent topologies, i.e., $\mathcal{V}_r = \{4, 7\}$. Node 2 is the source of node 4, and nodes 5 and 6 are sources of node 7. Thus, there are two sub-systems with convergent topology: $\mathcal{G}_4^{cnv}$ shown in green, and $\mathcal{G}_7^{cnv}$ shown in purple. Notice that, for this step, we use the graph $\mathcal{G}'$, which is obtained after subtracting the edges belonging to divergent topologies from the graph that models the UDS.

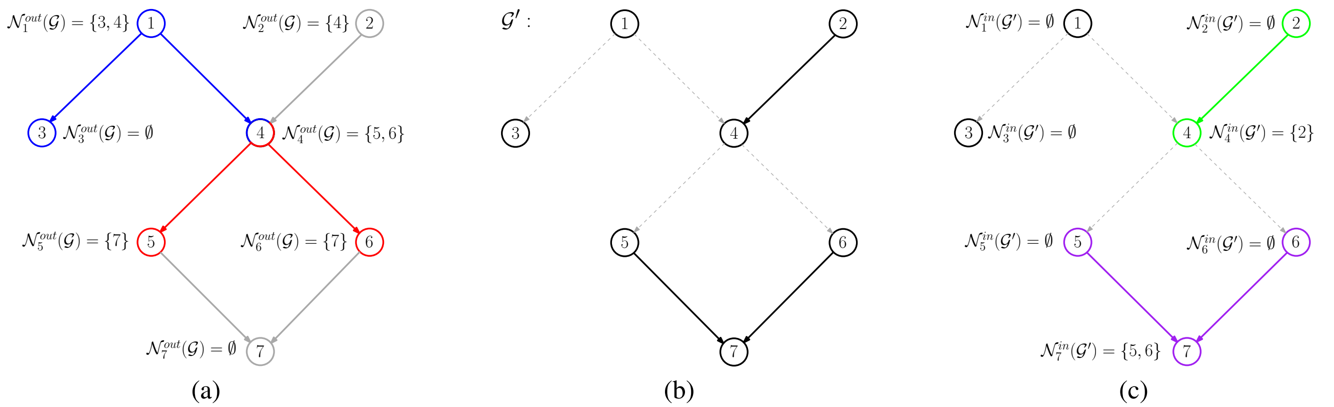

$\mathcal{N}_1^{out}(\mathcal{G}) = \{3,4\}$
$\mathcal{N}_2^{out}(\mathcal{G}) = \{4\}$
$\mathcal{N}_3^{out}(\mathcal{G}) = \emptyset$
$\mathcal{N}_4^{out}(\mathcal{G}) = \{5,6\}$
$\mathcal{N}_5^{out}(\mathcal{G}) = \{7\}$
$\mathcal{N}_6^{out}(\mathcal{G}) = \{7\}$
$\mathcal{N}_7^{out}(\mathcal{G}) = \emptyset$
(a)
$\mathcal{G}'$:
(b)
$\mathcal{N}_1^{in}(\mathcal{G}') = \emptyset$
$\mathcal{N}_2^{in}(\mathcal{G}') = \emptyset$
$\mathcal{N}_3^{in}(\mathcal{G}') = \emptyset$
$\mathcal{N}_4^{in}(\mathcal{G}') = \{2\}$
$\mathcal{N}_5^{in}(\mathcal{G}') = \emptyset$
$\mathcal{N}_6^{in}(\mathcal{G}') = \emptyset$
$\mathcal{N}_7^{in}(\mathcal{G}') = \{5,6\}$
(c)
1
2
3
4
5
6
7

Figure 5. Steps of the partition algorithm: (a) identification of divergent topologies; (b) removal of edges of divergent topologies; and (c) identification of convergent topologies.

V. DECENTRALIZED CONTROL OF UDS

Once the partition of the UDS is done, a controller is synthesized for each sub-system. As discussed in Section III-A, the control goal is to prevent overflows. Hence, wastewater must be efficiently allocated among the available reservoirs by controlling the system's valves. In our case, efficient wastewater allocation is achieved when the reservoirs of UDS have the same remaining capacity.

A. LOCAL CONTROLLERS FOR SUB-SYSTEMS WITH FLOW DIVERGENT TOPOLOGY

1) Dynamic model of sub-systems with flow divergent topology

As stated in Definition 1, sub-systems with flow divergent topology have one source reservoir and n receptor reservoirs (see Figure 2(a)). For convention, we use the indices $1, \dots, n$, for identifying receptor reservoirs, while the source reservoir is marked with the index $n+1$.

As explained in Section III-C, we represent each collector of the UDS as a reservoir of volume $\bar{v}_i$ containing a wastewater volume $v_i(t)$. The model assumes that the outflows of a reservoir are proportional to its wastewater volume (for more details see [23]).

The differential equation describing the change along the time of the source reservoir's wastewater volume is

$$\begin{aligned}\dot{v}_{n+1}(t) \quad = \quad & \underbrace{q_{n+1}^{in}(t)}_{\text{external inflow}} - \underbrace{\gamma_{n+1}^{out}(t) v_{n+1}(t)}_{\text{external outflow}} \\ & - \sum_{i=1}^{n} \underbrace{k_{n+1} x_i(t) v_{n+1}(t)}_{\substack{\text{outflow to } i\text{-th} \\ \text{receptor reservoir}}},\end{aligned} \tag{4}$$

where $k_{n+1} > 0$ is a parameter that depends on the geometry of the source reservoir, $x_i(t) \in [0,1]$ is the opening percentage of the valve that connects the source reservoir with the i-th receptor reservoir $\big(x_i(t) = 0$ means the valve is completely closed, and $x_i(t) = 1$ means a completely opened valve$\big)$, $q_{n+1}^{in}(t) \geq 0$ is the external inflow, and $\gamma_{n+1}^{out}(t) \geq 0$ is a coefficient related to the external outflow of the reservoir[2].

The dynamic model of the receptor reservoirs' wastewater volume is given by

$$\dot{v}_i(t) = \underbrace{k_{n+1} x_i(t) v_{n+1}(t)}_{\text{inflow from source reservoir}} + \underbrace{q_i^{in}(t)}_{\text{external inflow}} - \underbrace{\gamma_i^{out}(t) v_i(t)}_{\text{external outflow}}, \tag{5}$$

for all $i = 1, \dots, n$. Again, $q_i^{in}(t) \geq 0$ is the external inflow, and $\gamma_i^{out}(t) \geq 0$ is a coefficient related to the external outflow of the i-th receptor reservoir.

Figure 6 shows a schematic representation of a sub-system with flow divergent topology. Notice that all the involved variables and parameters of the dynamic model described before are depicted in this figure.

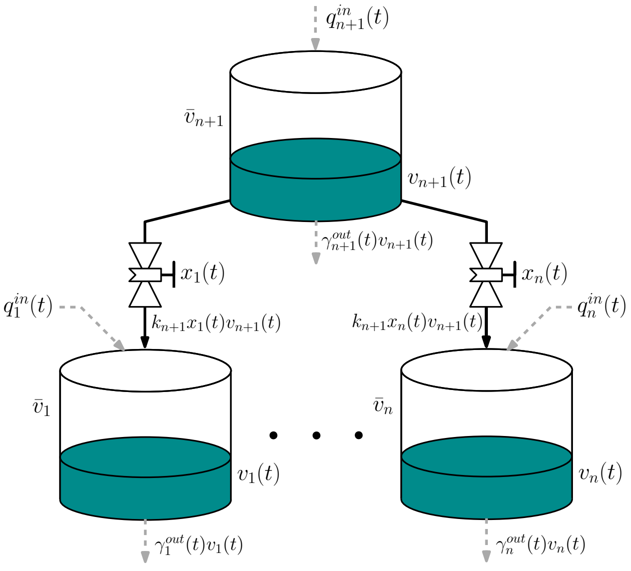

$q_{n+1}^{in}(t)$
$\bar{v}_{n+1}$
$v_{n+1}(t)$
$\gamma_{n+1}^{out}(t)v_{n+1}(t)$
$x_1(t)$
$x_n(t)$
$q_1^{in}(t)$
$q_n^{in}(t)$
$k_{n+1}x_1(t)v_{n+1}(t)$
$k_{n+1}x_n(t)v_{n+1}(t)$
$\bar{v}_1$
$\bar{v}_n$
$v_1(t)$
$v_n(t)$
$\gamma_1^{out}(t)v_1(t)$
$\gamma_n^{out}(t)v_n(t)$

Figure 6. A sub-system with divergent topology.

[2] External inflows and outflows are wastewater flows coming/going from/to either other sub-systems or the environment, e.g., due to raining events or leakages.

2) Replicator dynamics-based controller for flow divergent topology

Based on the replicator dynamics model introduced in Section II-B, we design local controllers for each UDS sub-system having a *flow divergent* topology. Replicator dynamics-based controllers are fully characterized by a set of strategies, a population mass and a set of fitness functions. Let us define each one of these elements.

In our game, each player has two options: increase the opening percentage of one of the tanks' valves, or decrease such a percentage for all the valves. If the player chooses the first option, then it must also select one of the n valves that connect the source tank with its n receptor tanks. Thus, we have $n+1$ available strategies, i.e., the strategy set is $S = \{1, \dots, n+1\}$.

Strategies $1, \dots, n$, increase the opening percentage of the corresponding receptor tank, i.e., if a player selects the i-th strategy, where $i = 1, \dots, n$, then the opening percentage of the i-th receptor tank's valve increases. In this regard, the population playing this i-th strategy is exactly the opening percentage of the i-th receptor tank's valve, i.e., $x_i(t)$.

On the other hand, if a player adopts the strategy $n+1$, it decreases the opening percentage of all valves. In this case, the population playing the strategy $n+1$, i.e., $x_{n+1}(t)$, does not have a physical meaning. However, it is equal to the sum of the remaining opening percentages of all valves.

According to previous definitions, the mass of the population involved in the replicator dynamics model is the sum of the opening capacity of the n valves that connect the source tank with its receptor tanks. Since we assume that $x_i(t) \in [0,1]$, for all $i = 1, \dots, n$, then the population mass is equal to n.

Regarding fitness functions, the higher the remaining volume of a receptor tank, the more appealing to increase its wastewater inflow by increasing its valve opening percentage. This fact is captured in the following fitness functions:

$$f_i(t) = \begin{cases} \bar{v}_i - v_i(t) + \varepsilon b_i\left(x_i(t), 1\right) & \text{if} \quad i = 1, \dots, n \\ \bar{v}_i - v_i(t) & \text{if} \quad i = n+1. \end{cases} \tag{6}$$

Notice that we have added a barrier term $\varepsilon b_i\left(x_i(t), 1\right)$ (as explained in Section II-B) to the first n fitness functions. In this barrier term, the upper bound of $x_i(t)$ is 1, because it is the maximum opening percentage. Notice also that the fitness function of the strategy $n+1$ models the fact that if the remaining capacity of the source tank is high, then it is appealing to increase the wastewater stored in the source tank by decreasing the opening percentage of all valves.

Under previous assumptions, the replicator dynamics-based controller is given by the following differential equations:

$$\dot{x}_i(t) = \beta x_i(t)\left(f_i(t) - \bar{f}(t)\right), \quad i = 1, \dots, n+1, \tag{7}$$

where β is a tuning parameter of the controller, and $\bar{f}(t) = \frac{1}{n}\sum_{j=1}^{n+1} x_j(t) f_j(t)$.

B. LOCAL CONTROLLERS FOR SUB-SYSTEMS WITH FLOW CONVERGENT TOPOLOGY

1) Dynamic model of sub-systems with flow convergent topology

As stated in Definition 2, sub-systems with flow convergent topology have n source reservoirs and one receptor reservoir (see Figure 2(b)). For convention, we use the indices $1, \dots, n$, for identifying source reservoirs, while the receptor reservoir is marked with the index $n+1$.

The differential equations describing the change along the time of the source reservoirs' wastewater volume are

$$\dot{v}_i(t) = \underbrace{q_i^{in}(t)}_{\substack{\text{external}\\\text{inflow}}} - \underbrace{\gamma_i^{out}(t) v_i(t)}_{\text{external outflow}} - \underbrace{k_i x_i(t) v_i(t)}_{\substack{\text{outflow to}\\\text{receptor reservoir}}}, \quad i = 1, \dots, n, \tag{8}$$

where k_i is a parameter that depends on the geometry of the i-th source reservoir, and $x_i(t) \in [0,1]$ is the opening percentage of the valve that connects the i-th source reservoir with the receptor reservoir.

On the other hand, the dynamic model of the receptor reservoir's wastewater volume is given by

$$\dot{v}_{n+1}(t) = \sum_{i=1}^{n} \underbrace{k_i x_i(t) v_i(t)}_{\substack{\text{inflow form } i\text{-th}\\\text{source reservoir}}} + \underbrace{q_{n+1}^{in}(t)}_{\text{external inflow}} - \underbrace{\gamma_{n+1}^{out}(t) v_{n+1}(t).}_{\text{external outflow}} \tag{9}$$

Figure 7 shows a schematic representation of a sub-system with flow convergent topology. Notice that all the involved variables and parameters of the dynamic model described before are depicted in this figure.

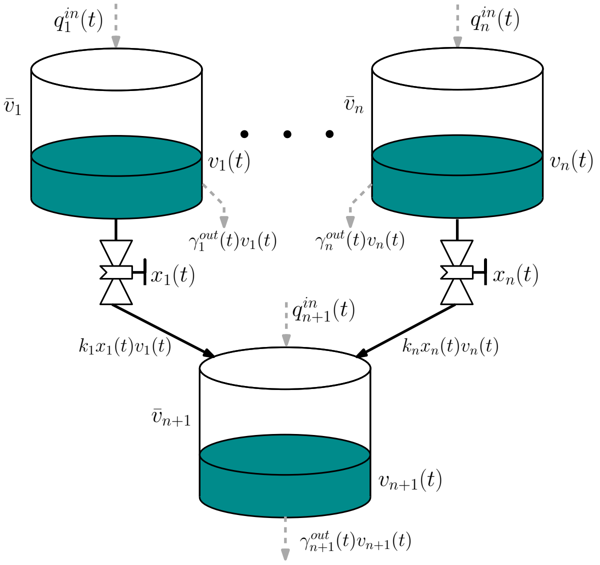


Figure 7. A sub-system with convergent topology.

Remark 2. *The variables of the divergent topology model (see Figure 6) are the same as the ones used in the divergent case (see Figure 7). Nonetheless, the equations that describe both models ((4) and (5) for divergent topologies, and (8) and*

(9) for convergent ones) are different. This fact is because while in the convergent model we have n receptor reservoirs and one source, in the divergent model we have n sources and only one receptor reservoir.

2) Replicator dynamics-based controller for flow convergent topology

Let us define the parameters of the replicator dynamics-based controller for each sub-system having a *flow convergent* topology.

Similarly as in the flow divergent case, we have a strategy set given by $S = \{1, \dots, n+1\}$. The population playing the i-th strategy, where $i = 1, \dots, n$, is the opening percentage of the i-th source tank's valve, i.e., $x_i(t)$. Besides, the population playing the strategy $n+1$, i.e., $x_{n+1}(t)$, is equal to the sum of the remaining opening percentages of all valves. Therefore, the population mass is again equal to n.

Fitness functions must capture the fact that if the remaining volume of a source tank is low, then it is more appealing to increase the opening percentage of its valve. Having this idea in mind, we define the fitness functions as minus the remaining volume of each tank, i.e.,

$$f_i(t) = \begin{cases} -\left(\bar{v}_i - v_i(t)\right) + \varepsilon b_i\left(x_i(t), 1\right) & \text{if} \quad i = 1, \dots, n \\ -\left(\bar{v}_i - v_i(t)\right) & \text{if} \quad i = n+1. \end{cases} \tag{10}$$

As in the flow divergent case, we have added the same barrier term $\varepsilon b_i\left(x_i(t), 1\right)$ to the first n fitness functions.

Finally, the replicator dynamics-based controller for sub-systems with flow convergent topology is given by the equations in (7), with the same average fitness.

Remark 3. *The local controller of each sub-system handles the valves that belong to the subsystem. Therefore, if a valve (or what is the same, an edge of the graph that models the UDS) belongs to more than one sub-system, there would be ambiguity because more than one controller would make decisions about the valve opening percentage. This is the reason for the partitioning constraint C3 (see Section IV).*

VI. STABILITY ANALYSIS

This section analyzes the stability of each sub-system controlled via replicator dynamics. We focus our analysis in a special case where the opening percentages of the sub-systems' valves satisfy the following assumptions:

Assumption 2. *Let $x_i^*, \dots, x_n^*$, be the opening percentages of a sub-system's valves at equilibrium. The value $x_i^* > 0$, for all $i = 1, \dots, n$. Thus, none of the sub-system's valves are fully closed at equilibrium.*

Assumption 3. *The value of x_i^* is such that the barrier terms in (6) and (10) can be neglected in the computation of the fitness functions. Hence, x_i^* is far from being fully opened.*

Furthermore, we make the following assumption on the external inflows and outflows of sub-systems.

Assumption 4. *Consider a sub-system having either convergent or divergent topology. External inflows and coefficients related to external outflows are constants, i.e., $q_i^{in}(t) = \hat{q}_i^{in}$ and $\gamma_i^{out}(t) = \hat{\gamma}_i^{out}$, for all $i = 1, \dots, n+1$.*

A. FLOW DIVERGENT TOPOLOGY CASE

1) Equilibrium point

Under Assumptions 2-4, the equilibrium point of a sub-system with flow divergent topology controlled via population dynamics has an important property, which is given in the following proposition.

Proposition 1. *Let Assumptions 2–4 hold. If $\sum_{i=1}^{n+1} x_i(0) = n$, then the equilibrium point of the closed-loop system given by (4), (5) and (7) is*

$$\begin{aligned} v_i^* &= \bar{v}_i - \frac{\sum_{j=1}^{n+1}\left(\hat{\gamma}_j^{out}\bar{v}_j - \hat{q}_j^{in}\right)}{\sum_{j=1}^{n+1}\hat{\gamma}_j^{out}}, && i = 1, \dots, n+1, \\ x_i^* &= \frac{\hat{\gamma}_i^{out} v_i^* - \hat{q}_i^{in}}{k_{n+1} v_{n+1}^*}, && i = 1, \dots, n, \\ x_{n+1}^* &= n - \sum_{j=1}^{n} x_j^*, \end{aligned} \tag{11}$$

where v_i^* is the wastewater volume of the i-th reservoir at equilibrium. Notice that the remaining capacities of all reservoirs are the same at equilibrium, i.e., $\bar{v}_i - v_i^* = \frac{\sum_{j=1}^{n+1}\left(\hat{\gamma}_j^{out}\bar{v}_j - \hat{q}_j^{in}\right)}{\sum_{j=1}^{n+1}\hat{\gamma}_j^{out}}$, for all $i = 1, \dots, n+1$.

Proof. Equilibrium condition for (4) and (5) is given by

$$\begin{aligned} 0 &= \hat{q}_{n+1}^{in} - \hat{\gamma}_{n+1}^{out} v_{n+1}^* - k_{n+1} v_{n+1}^* \sum_{j=1}^{n} x_j^* \\ 0 &= k_{n+1} x_i^* v_{n+1}^* + \hat{q}_i^{in} - \hat{\gamma}_i^{out} v_i^*, && i = 1, \dots, n, \end{aligned} \tag{12}$$

where we have used the values of inflows and outflows given in Assumption 4.

On the other hand, due to Assumption 2, all fitness functions reach the same value at equilibrium (see (7)). Moreover, fitness functions are equal to the remaining capacity of each reservoir for Assumption 3 (see (6)). Therefore, all sub-system's reservoirs have the same remaining capacity at equilibrium, i.e., $\bar{v}_i - v_i^* = \bar{v}_j - v_j^*$, for all $i, j = 1, \dots, n+1$. Using this fact and after some algebraic manipulations on (12), we obtain the first equations of the equilibrium point given in (11), i.e., v_i^*, for all $i = 1, \dots, n+1$, and x_i^*, for all $i = 1, \dots, n$. The last equation of (11), i.e., the equilibrium state x_{n+1}^*, is computed using the fact that $\sum_{j=1}^{n+1} x_j(0)$ by assumption and $\sum_{j=1}^{n+1} x_j^* = \sum_{j=1}^{n+1} x_j(0)$ (according to Lemma 1). □

The fact that all reservoirs have the same remaining capacity at equilibrium implies a fairly distribution of wastewater throughout the sub-system.

2) Stability

The next result guarantees the stability of the equilibrium point given in Proposition 1 for a sub-system with divergent topology controlled via replicator dynamics.

Theorem 1. *Let Assumptions 2–4 hold. Besides, assume that $k_{n+1} < 4\hat{\gamma}_i^{out}$, for all $i = 1, \ldots, n$. If the initial conditions of the controller in (7) satisfy $\sum_{i=1}^{n+1} x_i(0) = n$ and $x_i(0) > 0$, for all $i = 1, \ldots, n+1$, then the equilibrium point stated in (11) is asymptotically stable under the dynamics of the closed-loop system given in (4), (5) and (7).*

Proof. The sub-system controlled via replicator dynamics, which is given in (4), (5) and (7), can be written in error coordinates with respect to the equilibrium point in (11) as follows:

$$\Sigma_1 : \begin{cases} \dot{e}_{v_i}(t) = -\,\hat{\gamma}_i^{out} e_{v_i}(t) + k_{n+1} e_{v_{n+1}}(t)\Big(e_{x_i}(t) + x_i^*\Big) \\ \qquad\qquad + k_{n+1} v_{n+1}^* e_{x_i}(t), \qquad\qquad i = 1, \ldots, n, \\ \\ \dot{e}_{v_{n+1}}(t) = -\,\hat{\gamma}_{n+1}^{out} e_{v_{n+1}}(t) \\ \qquad\qquad - k_{n+1} e_{v_{n+1}}(t) \displaystyle\sum_{j=1}^{n} \Big(e_{x_j}(t) + x_j^*\Big) \\ \qquad\qquad - k_{n+1} v_{n+1}^* \displaystyle\sum_{j=1}^{n} e_{x_j}(t), \end{cases}$$

$$\Sigma_2 : \begin{cases} \dot{e}_{x_i}(t) = \beta \left(e_{x_i}(t) + x_i^*\right)\Big(f_i(t) \\ \qquad\qquad - \dfrac{1}{n} \displaystyle\sum_{j=1}^{n+1} \left(e_{x_j}(t) + x_j^*\right) f_j(t)\Big), \\ \qquad\qquad\qquad\qquad\qquad i = 1, \ldots, n+1, \end{cases} \tag{13}$$

where $e_{v_i}(t) = v_i(t) - v_i^*$ and $e_{x_i}(t) = x_i(t) - x_i^*$, for all $i = 1, \ldots, n+1$. Notice that Σ_1 is the UDS sub-system and Σ_2 is the controller.

We use Lyapunov theory to prove stability of the equilibrium point. For Σ_1, consider the positive definite function

$$V_1\Big(\boldsymbol{e}_v(t)\Big) = \frac{1}{2k_{n+1}v_{n+1}^*} \sum_{i=1}^{n+1} e_{v_i}^2(t),$$

where $\boldsymbol{e}_v(t) = \Big[e_{v_1}(t), \ldots, e_{v_{n+1}}(t)\Big]^\top$. The derivative of $V_1\Big(\boldsymbol{e}_v(t)\Big)$ along the trajectories of Σ_1 is $\dot{V}_1\Big(\boldsymbol{e}_v(t)\Big) = \Psi_1\Big(\boldsymbol{e}_v(t)\Big) + \sum_{i=1}^{n+1} e_{v_i}(t) e_{x_i}(t)$, where

$$\begin{aligned} \Psi_1\Big(\boldsymbol{e}_v(t)\Big) = &-\frac{\hat{\gamma}_{n+1}^{out}}{v_{n+1}^* k_{n+1}} e_{v_{n+1}}^2(t) \\ &- \frac{1}{v_{n+1}^*} \sum_{i=1}^{n} \left(\frac{4\hat{\gamma}_i^{out} - x_i(t)k_{n+1}}{4k_{n+1}}\right) e_{v_i}^2(t) \\ &- \frac{1}{v_{n+1}^*} \sum_{i=1}^{n} x_i(t) \Big(e_{v_{n+1}}(t) - e_{v_i}(t)\Big)^2. \end{aligned}$$

For computing $\dot{V}_1\Big(\boldsymbol{e}_v(t)\Big)$, we have used the fact that $\sum_{i=1}^{n+1} x_i(t) = n$, for all $t \geq 0$ (Lemma 1 guarantees this property since $\sum_{i=1}^{n+1} x_i(0) = n$ by assumption). The function $\Psi_1\Big(\boldsymbol{e}_v(t)\Big)$ is negative definite since $x_i(t) \in [0,1]$ (this constraint also follows from Lemma 1 and by the effect of the barrier functions given in (6)) and $k_{n+1} < 4\hat{\gamma}_i^{out}$ by assumption.

On the other hand, consider the following positive definite function (which is based on the relative entropy function [53]) for Σ_2:

$$V_2\Big(\boldsymbol{e}_x(t)\Big) = -\frac{1}{\beta} \sum_{i=1}^{n+1} x_i^* \ln\left(\frac{e_{x_i}(t) + x_i^*}{x_i^*}\right), \tag{14}$$

where $\boldsymbol{e}_x(t) = \Big[e_{x_1}(t), \ldots, e_{x_{n+1}}(t)\Big]^\top$. Positive definiteness of $V_2\Big(\boldsymbol{e}_x(t)\Big)$ can be proven using Jensen's inequality as follows: since the logarithm function is strictly concave and given the fact that $\sum_{i=1}^{n+1} x_i(t) = \sum_{i=1}^{n+1} x_i^* = n$, we have

$$\begin{aligned} V_2\Big(\boldsymbol{e}_x(t)\Big) \geq \quad & -\sum_{i=1}^{n+1} x_i^* \ln\left(\frac{\displaystyle\sum_{i=1}^{n+1} e_{x_i}(t) + \sum_{i=1}^{n+1} x_i^*}{\displaystyle\sum_{i=1}^{n+1} x_i^*}\right) \\ = \quad & -n \ln\left(\frac{\displaystyle\sum_{i=1}^{n+1} \left(x_i(t) - x_i^*\right) + \sum_{i=1}^{n+1} x_i^*}{\displaystyle\sum_{i=1}^{n+1} x_i^*}\right) \\ = \quad & -n \ln(1) = 0. \end{aligned}$$

In addition, according to Jensen's inequality, $V_2\Big(\boldsymbol{e}_x(t)\Big) = 0$ only at $x_i(t) = x_i^*$, for all $i = 1, \ldots, n+1$. Therefore, V_2 is positive definite.

The derivative of $V_2\Big(\boldsymbol{e}_x(t)\Big)$ along the trajectories of Σ_2 is $\dot{V}_2\Big(\boldsymbol{e}_x(t)\Big) = \Psi_2\Big(\boldsymbol{e}_x(t)\Big) - \sum_{i=1}^{n+1} e_{v_i}(t) e_{x_i}(t)$, where

$$\Psi_2\Big(\boldsymbol{e}_x(t)\Big) = \sum_{i=1}^{n} \varepsilon_i \Big(b_i\left(x_i(t), 1\right) - b_i\left(x_i^*, 1\right)\Big) e_{x_i}(t).$$

The function $\Psi_2\Big(\boldsymbol{e}_x(t)\Big) \leq 0$ since $b_i\left(x_i(t), 1\right)$ is monotonically decreasing with respect to $x_i(t)$. Furthermore, although the value of $\Psi_2\Big(\boldsymbol{e}_x(t)\Big)$ does not depend on $e_{x_{n+1}}(t)$, $\Psi_2\Big(\boldsymbol{e}_x(t)\Big)$ is negative definite. Notice that $\Psi_2\Big(\boldsymbol{e}_x(t)\Big) = 0$ only if $e_{x_i}(t) = 0$, for all $i = 1, \ldots, n$. This condition implies that $e_{x_{n+1}}(t) = 0$ (which follows from the fact that $e_{x_{n+1}}(t) = -\sum_{i=1}^{n} e_{x_i}(t)$ according to Lemma 1).

Based on the results described above, we propose the Lyapunov function candidate $V\Big(\boldsymbol{e}_v(t), \boldsymbol{e}_x(t)\Big) = V_1\Big(\boldsymbol{e}_v(t)\Big) + V_2\Big(\boldsymbol{e}_x(t)\Big)$. The derivative of this function along the trajectories of the system in (13) is $\dot{V}\Big(\boldsymbol{e}_v(t), \boldsymbol{e}_x(t)\Big) = \Psi_1\Big(\boldsymbol{e}_v(t)\Big) +$

$\Psi_2\Big(\boldsymbol{e}_x(t)\Big)$, which is negative definite. Thus, we can conclude that the equilibrium point in (11) is asymptotically stable. □

Theorem 1 is relevant since it guarantees an appropriate performance of the proposed controller. Notice that the result stated in this theorem implies that if a reservoir has a lower remaining capacity than the other reservoirs (e.g., if a reservoir is going to overflow), then the controller tries to mitigate this undesirable situation by reallocating the wastewater in all reservoirs of the sub-system. This wastewater reallocation is performed in such a way that all reservoirs have the same remaining capacity.

B. FLOW CONVERGENT TOPOLOGY CASE

For sub-systems with flow convergent topology, we have similar results than those shown for the flow divergent case.

1) Equilibrium point

As stated in the next proposition, a sub-system with flow convergent topology controlled via replicator dynamics has an equilibrium point in which the remaining capacities of all reservoirs are the same.

Proposition 2. *Let Assumptions 2–4 hold. If $\sum_{i=1}^{n+1} x_i(0) = n$, then the equilibrium point of the closed-loop system given by (7)–(9) is*

$$\begin{aligned} v_i^* &= \bar{v}_i - \frac{\sum_{j=1}^{n+1}\left(\hat{\gamma}_j^{out}\bar{v}_j - \hat{q}_j^{in}\right)}{\sum_{j=1}^{n+1}\hat{\gamma}_j^{out}}, && i = 1,\ldots,n+1,\\ x_i^* &= \frac{\hat{q}_i^{in} - \hat{\gamma}_i^{out}v_i^*}{k_i v_i^*}, && i = 1,\ldots,n,\\ x_{n+1}^* &= n - \sum_{j=1}^{n} x_j^*. \end{aligned} \tag{15}$$

Notice that, in this equilibrium point, the remaining capacities of all reservoirs (i.e., $\bar{v}_i - v_i^*$, for all $i = 1,\ldots,n+1$) are the same.

Proof. The proof follows similar ideas that the ones used in the proof of Proposition 1. □

As in the flow divergent case, the equilibrium point in (15) implies a well-balanced wastewater distribution among the reservoirs of the sub-system. Indeed, notice that if all reservoirs have the same remaining capacity, there are no overflows in any reservoir.

2) Stability

The next theorem shows that the replicator dynamics based controller drives the sub-system to the equilibrium point given in (15).

Theorem 2. *Let Assumptions 2–4 hold. In addition, let $\bar{v}$ and k be positive real values, and assume that $\bar{v}_i = \bar{v}$, for all $i = 1,\ldots,n+1$, and $k_i = k$, for all $i = 1,\ldots,n$. Furthermore, assume that $k < \frac{4\hat{\gamma}_{n+1}^{out}}{n}$. If the initial conditions of the controller in (7) satisfy $\sum_{i=1}^{n+1} x_i(0) = n$ and $x_i(0) > 0$, for all $i = 1,\ldots,n+1$, then the equilibrium point stated in (15) is asymptotically stable under the dynamics of the closed-loop system given in (7)–(5).*

Proof. Since all reservoir volumes are the same by assumption, then there exists a positive real value v^* such that $v_i^* = v^*$, for all $i = 1,\ldots,n+1$ (see (15)). Taking

$$V\Big(\boldsymbol{e}_v(t), \boldsymbol{e}_x(t)\Big) = \frac{1}{2kv^*}\sum_{i=1}^{n+1} e_{v_i}^2(t) + V_2\Big(\boldsymbol{e}_x(t)\Big),$$

where $V_2\Big(\boldsymbol{e}_x(t)\Big)$ is defined in (14), as Lyapunov function candidate and following the same procedure of the proof of Theorem 1, we can conclude that the derivative of $V\Big(\boldsymbol{e}_v(t)\Big)$ along the trajectories of the closed-loop system is negative definite. Therefore the equilibrium point is asymptotically stable. □

For the case of flow convergent topology, notice that we impose additional constraints (compared to the flow divergent case) to guarantee stability of the desired equilibrium point. Specifically, we require that all reservoirs have the same volumes $\bar{v}_i$ and the same coefficients k_i. These constraints are met for sub-systems where all reservoirs have the same size and geometry.

Once the stability has been proven for both divergent and convergent topology sub-systems, we test the proposed control strategy in a case study.

VII. CASE STUDIES AND DISCUSSION

We test the performance of the proposed decentralized controller in two simulation scenarios. Furthermore, we compare our method with two techniques that have been employed in real-time control for UDSs: decentralized LQR with constraints (e.g., see [14], [54]) and centralized MPC (e.g., see [55]).

For synthesizing the decentralized LQR scheme, we use the partitioning algorithm described in Section IV and design local LQRs per each sub-system. On the other hand, for designing the MPC strategy, we use the information of the whole UDS to compute the control signals. Hence, we notice that the technique developed in this article and the decentralized LQR scheme have disadvantages compared to MPC in terms of the information availability (i.e., local vs global). However, there are two reasons for using centralized MPC in our comparison. First, we want to compare our method with another that might guarantee an optimal performance as is the case of MPC. Second, the global information requirement of centralized MPC can be relaxed by decoupling the optimization problem that MPC solves at each step and then using a distributed optimization algorithm (e.g., ADMM [43]). This relaxation allows MPC to use local information at the cost of increasing the computation time since distributed optimization algorithms have slower convergence rates than their centralized counterparts.

A. CASE STUDY 1

The first scenario corresponds to the UDS shown in Figure 4(a). This UDS has seven reservoirs and four sub-systems (as explained in Section IV-D). Half of the sub-systems has flow divergent topology and the other half has flow convergent topology (see Figure 5). For this case study, all reservoirs have the same parameters, i.e., $k_i = 0.005\ \mathrm{s}^{-1}$ and $\bar{v}_i = 274.80\ \mathrm{m}^3$, for all $i = 1, \ldots, 7$ (these parameters have been taken from [23]). We simulate a rainfall scenario where flooding typically occurs. The direct runoff hydrograph for each reservoir, which models the external inflows, is shown in Figure 8(c). These runoff hydrographs are Gaussian functions whose parameters are randomly chosen as follows: center is in the interval 0.5–0.75 hours, width is in the interval 0–0.5 hours, and amplitude is in the interval 0–0.6 m^3/s. We also assume that there are no leakages at reservoirs. Therefore, external outflows of reservoirs 1, 2, 4, 5, and 6 are only due to wastewater flowing to other sub-systems of the UDS, while external outflows of reservoirs 3 and 7 correspond to the wastewater flowing outside the UDS (e.g., to wastewater treatment plants or to water bodies). In this case, we assume that $\gamma_3^{out}(t) = \gamma_7^{out}(t) = 0.005\ \mathrm{s}^{-1}$.

Evolution of the remaining capacities of each UDS reservoir is shown in Figure 8(a). In this case, all valves of the system are completely opened without any controller acting on them. If the remaining capacity of some reservoir reaches the dotted line, it indicates that the reservoir is full of wastewater. On the other hand, if the remaining capacity of some reservoir is below the dotted line, it implies that wastewater exceeds the capacity of the reservoir, and thus flooding occurs. Notice that, for the uncontrolled case shown in Figure 8(a), there is an overflow in reservoir 7. In fact, the total overflow is around 71.3 m^3. This undesirable situation is avoided by using the decentralized controllers based on replicator dynamics with parameters $\beta = 0.0006$, $\varepsilon = 0.0001$, and barrier functions $b_i = \frac{1}{x^i(t)-1}$. Evolution of the remaining capacities of reservoirs under the proposed control strategy is shown in Figure 8(b), while the control signals (which correspond to the opening percentages of system's valves) are depicted in Figure 8(d). Notice that, under the replicator dynamics-based controller, there are no overflows in the UDS. The reason for this improved performance is that the wastewater is properly allocated among all the UDS reservoirs. The key point of the achieved wastewater allocation is that the proposed controller takes advantage of the capacities of upstream reservoirs. In the uncontrolled scenario, upstream reservoirs are promptly emptied causing downstream reservoirs receive a large amount of wastewater. Hence, reservoir 7 gets overloaded. On the contrary, when the controller is applied, valves of upstream reservoirs are closed (e.g., see valve 1 and valve 3 in Figure 8(d)). Therefore, these reservoirs are more filled than in the uncontrolled scenario causing that the flow downstream is relieved and flooding is avoided.

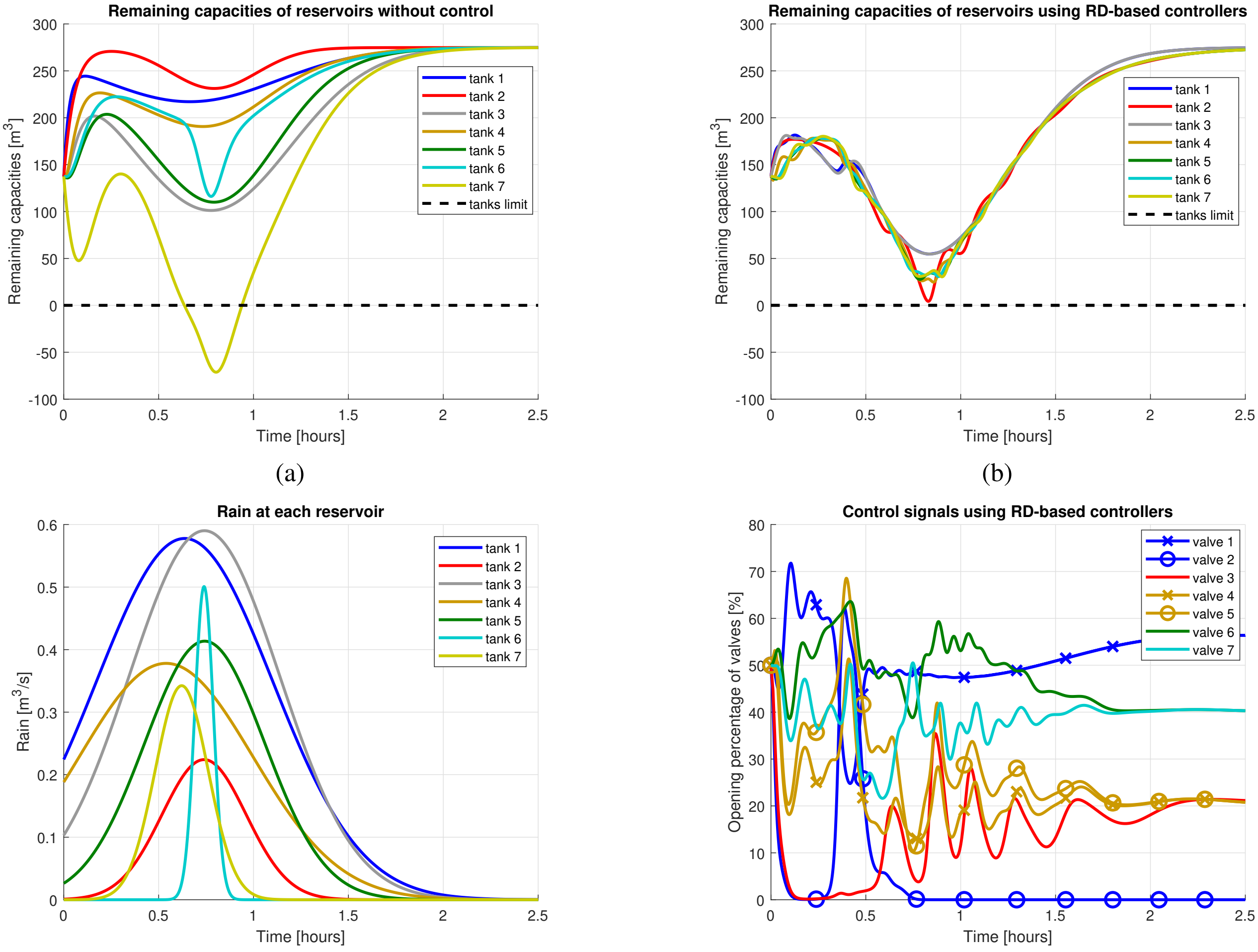
For this case study, we also test a decentralized LQR scheme and a centralized MPC synthesis. For both controllers the cost function is defined by $\sum_i v_i^\top v_i$, where the sum is over the reservoirs that belong to the corresponding sub-system in the case of LQR, and over all the reservoirs of the UDS in the case of MPC. Notice that this cost function penalizes fuller reservoirs. For both controllers, we constraint the opening percentage of valves to belong to the interval $[0, 1]$, i.e., $x_i \in [0, 1]$. Moreover, the sample time is 5 seconds for LQR, while for MPC, the sample time is 1 minute, the prediction horizon is 60 steps, and the control horizon is 10 steps. The performance of the controlled UDS is shown in Figure 9. Capacities of the UDS's reservoirs are depicted in Figure 9(a) (for LQR) and in Figure 9(b) (for MPC). In addition, control signals are shown in Figure 9(c) (for LQR) and in Figure 9(d) (for MPC).

Using the decentralized LQR, flooding occurs. However, it is significantly lower than the one obtained without using any controller. Specifically, the flooding is 18.2 m^3, which corresponds to a decrease of flooding of 75% compared to the baseline scenario (no control). Although the system behaves better under LQR than without control, the performance of LQR is worse than the decentralized controller based on replicator dynamics, which completely eliminates flooding under the same simulation conditions.

MPC also avoids flooding. Indeed, it is worth noting that the control signals generated when the centralized MPC is employed (Figure 9(d)) are quite similar to those produced under the decentralized controller based on replicator dynamics. This fact occurs even when the centralized MPC uses global information (i.e., information of the whole UDS) to compute the control inputs while the replicator dynamics only employ local information (i.e., information of each subsystem).

Furthermore, we want to highlight that both LQR and MPC strategies try to equalize the remaining capacities of reservoirs (see Figures 9(a) and 9(b)) in the same way as the replicator dynamics do. This happens even though neither LQR nor MPC controllers are designed using a cost function that explicitly penalizes the differences between the remaining capacities of reservoirs (the employed cost function only penalizes fuller reservoirs as was explained when the design parameters of LQR and MPC were presented).

Finally, to provide a quantitative comparison of the reduction of flooding using the different control techniques, i.e., local controllers based on replicator dynamics, decentralized LQR and centralized MPC, we simulate the UDS of case study 1 under the same conditions described before, but increasing the rain intensity. To do so, we amplify the peak of the Gaussian functions that model the rain by the factors given in Table 1. The obtained flooding is also shown in that table. Notice that comparing both decentralized strategies, the controller based on replicator dynamics outperforms LQR. For instance, in the heaviest rain scenario, the percentage of flooding reduction under replicator dynamics is 51% compared to LQR. On the other hand, although centralized MPC shows the best performance, the decentralized control strategy based on replicator dynamics keeps flooding at levels



(a) (b) (c) (d)

Figure 8. (a) Remaining capacities of the UDS reservoirs when no controller is used; (b) remaining capacities of reservoirs when the decentralized controller based on replicator dynamics is applied; (c) rainfall at each reservoir; and (d) opening percentages of valves for the replicator dynamics based controller.

close to the optimal. At this point, it is worth mentioning that the replicator dynamics based controller benefits from lower computation requirements compared to MPC. In fact, the replicator dynamics use a simple model given by a set of first-order ordinary differential equations, while MPC requires solving an optimization problem at each time step. Moreover, features such as modularity provided by a decentralized scheme also represent a key advantage from the controller based on replicator dynamics with respect to the centralized MPC approach. Finally, the need of merging information of the entire system in a centralized controller could represent a drawback with respect to a non-centralized control approach, which does not need such a strong condition over the information availability.

B. CASE STUDY 2

We also analyze the performance of the proposed control technique in a case study based on a real scenario. To do that, we simulate a portion of the Bogotá (Colombia) stormwater UDS whose description is presented in [37]. Figure 10 shows the structure of this UDS, which has 16 reservoirs. Parameters of each reservoir are given in Table 2.

Table 1. Comparison of controllers' performance for case study 1

Rain amplifying factor	Flooding [m^3]			
	Without control	Replicator dynamics	LQR	MPC
0.95	55.02	0	0	0
1.00	71.28	0	18.19	0
1.05	87.54	13.61	51.93	0
1.10	103.80	39.67	80.17	15.23

Simulation conditions reproduce a heavy rains scenario, where the external inflow of each reservoir is due to direct runoff. Figure 11 shows the hydrographs that models these runoffs. Similarly as in the previous case study, hydrographs are Gaussian functions whose parameters lie in the following intervals: center is between 0.5–0.75 hours; width is between

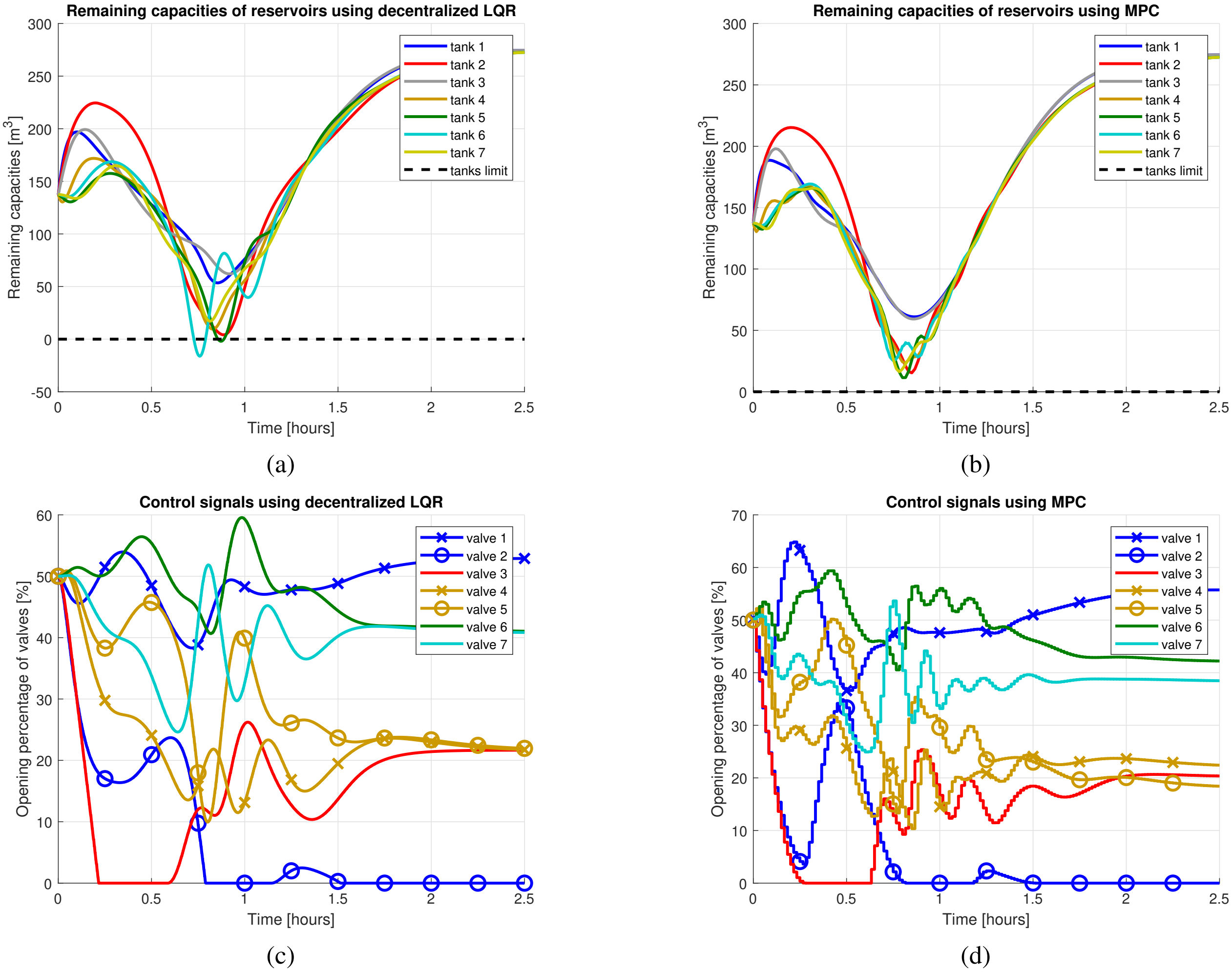


Figure 9. (a) Remaining capacities of the UDS reservoirs under decentralized LQR; (b) remaining capacities of reservoirs under centralized MPC; (c) opening percentages of valves for decentralized LQR; and (d) opening percentages of valves for centralized MPC.

0–0.5 hours; and amplitude is between 0–0.22 m^3/s. Figure 12 shows the behavior of the UDS without control (i.e., reservoirs' valves are completely open during all the time) for a time window of 3.5 hours. Throughout the simulation, wastewater only fills half of most reservoirs. In contrast, reservoirs 12 and 14 are overloaded during the first half of the simulation. Indeed, the remaining capacity of reservoir 14 is negative during this first half implying that flooding occurs. The total flooding is around 797.1 m^3. This situation is mainly due to two facts: first, reservoir 14 is a downstream reservoir, which receive wastewater from a big portion of the UDS; second, reservoir 14 is the one with the lowest coefficient k_i, i.e., it has the lowest capacity to evacuate wastewater (according to (8), the larger the coefficient k_i, the larger the outflow of the i-th reservoir trough its output valve).

Once the baseline (corresponding to the behavior of the system without control) has been established, we implement the control strategy proposed in Section V. First, we apply the partitioning algorithm of Section IV and identify 6 sub-systems with flow convergent topology (each sub-system is associated with a color in Figure 10). Table 3 summarizes the structure of the identified sub-systems. For each one of them, we implement a replicator dynamics based controller with parameters $\beta = 0.0003$, $\varepsilon = 5$, and barrier functions $\frac{1}{x_i(t)-1}$. Simulation results of the controlled UDS are shown in Figure 13. Additionally, the control inputs (opening percentage of the reservoirs' valves) are shown in Figure 14. The first thing that we notice in Figure 13 is that, unlike the uncontrolled case, there are no big differences in the remaining capacities of reservoirs. This behavior is due to the fact that the replicator dynamics based controllers seek to equalize the remaining capacities as proven in Section VI. For instance, see the behavior of reservoirs 14 (brown line) and 15 (orange line), and their corresponding valves. From hour 1–2, reservoir 14 has lower remaining capacity than reservoir 15. Thus, during this time, the valve that allows reservoir 14 to evacuate wastewater is almost fully opened while valve of reservoir 15 is closed. This fact allows reservoirs 14 and 15 to reach almost the same remaining

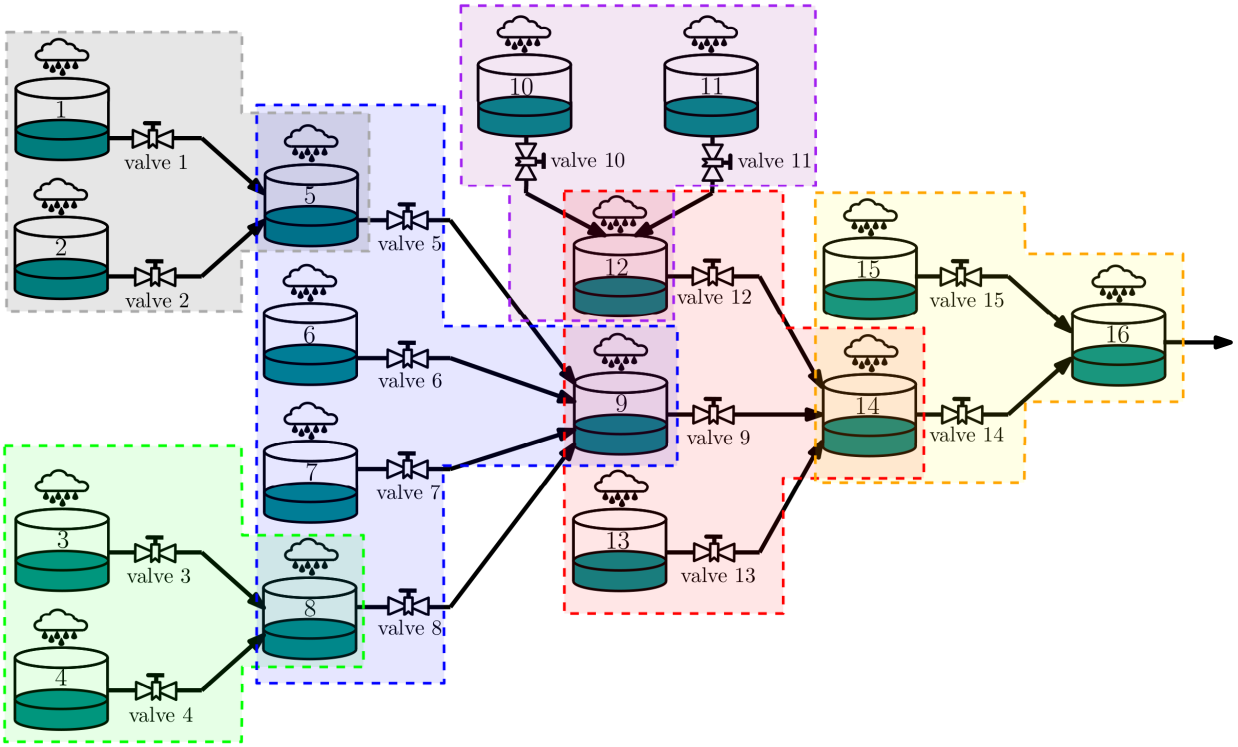


Figure 10. Part of the Bogotá (Colombia) stormwater UDS having 6 sub-systems: s1 (grey), s2 (green), s3 (blue), s4 (purple), s5 (red), and s6 (yellow).

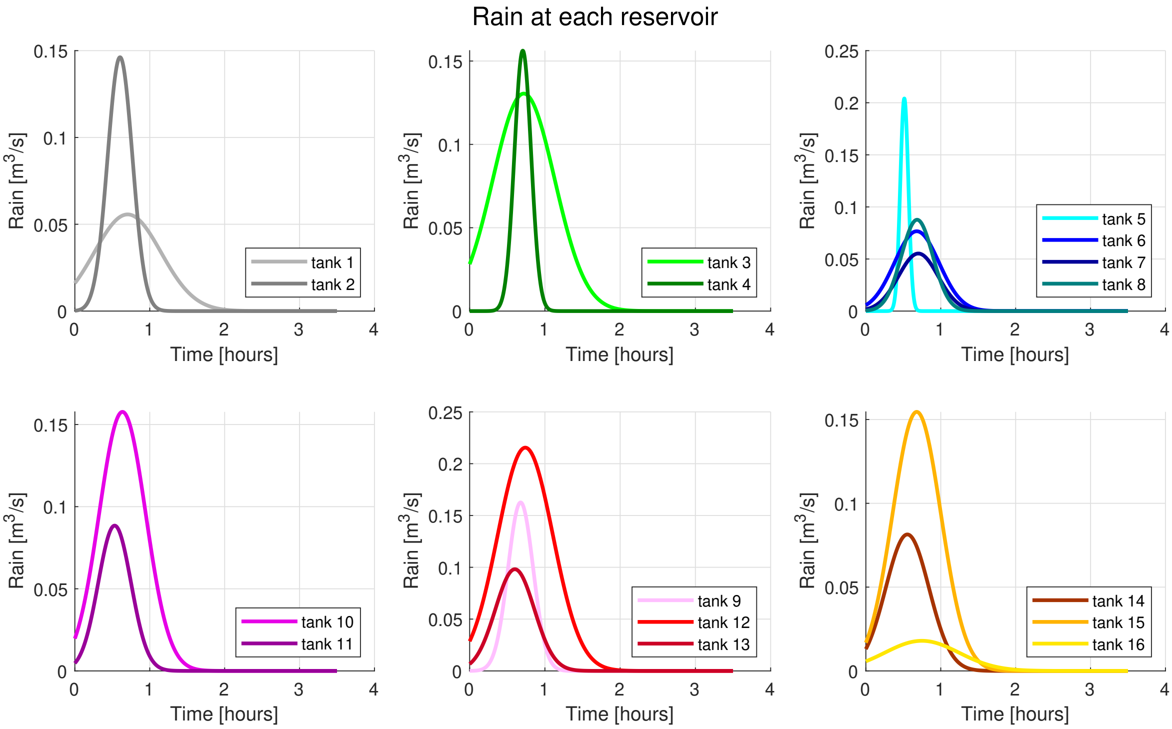


Figure 11. Hydrographs modeling the direct runoff for each reservoir of the Bogotá (Colombia) case study.

capacity from hour 2 onwards. Summarizing, when using the local controllers based on replicator dynamics, the total capacity of the UDS is better used because wastewater is stored in underloaded reservoirs. On the other hand, the total

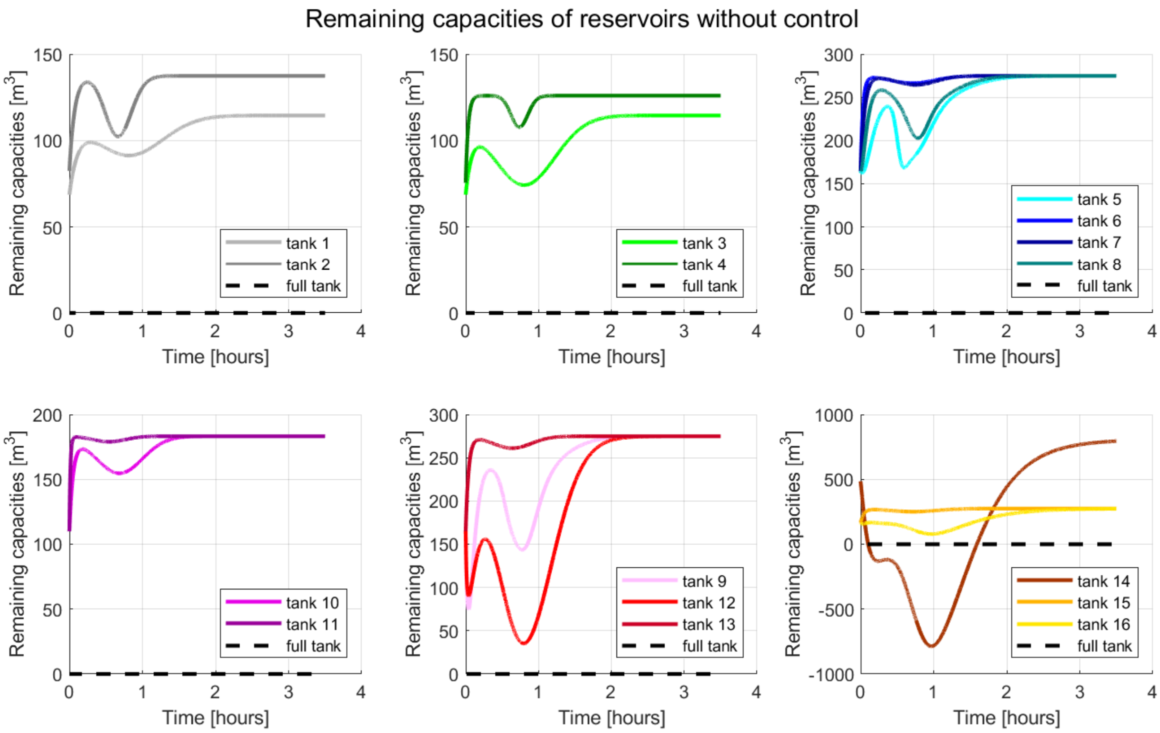


Figure 12. Evolution of the remaining capacities of reservoirs of the Bogotá (Colombia) case study when no control is applied.

Table 2. Parameters of reservoirs for the Bogotá (Colombia) case study

Reservoir label	Volume $\bar{v}_i$ [m^3]	Coefficient k_i [s^{-1}]
1	114.50	0.002332
2	137.40	0.003870
3	114.50	0.003170
4	125.95	0.008239
5	274.80	0.002217
6	274.80	0.008975
7	274.80	0.005185
8	274.80	0.004764
9	274.80	0.006147
10	183.20	0.005446
11	183.20	0.020703
12	274.80	0.001693
13	274.80	0.007026
14	807.22	0.000632
15	274.80	0.006319
16	274.80	0.005782

flooding for the controlled case is 29.4 m^3, which is only 4% of the flooding that occurs for the uncontrolled case. Finally, contrary to what happens in the uncontrolled scenario where high flooding is concentrated in reservoir 14, the controlled case shows that small floods occur in reservoirs 3, 4, 8 and 11. The fact that flooding is spread over the UDS's reservoirs mitigates the impact in the zone where reservoir 14 is located.

Table 3. Sub-systems of the Bogotá (Colombia) case study

Sub-system's label	Source reservoirs	Receptor reservoir
s1	1, 2	5
s2	3, 4	8
s3	5, 6, 7, 8	9
s4	10, 11	12
s5	9, 12, 13	14
s6	14, 15	16

As in the first case study, we implement a decentralized LQR and a centralized MPC to compare the performance of the controller based on replicator dynamics. The design process of LQR and MPC is the same described in case study 1. Besides, the sample time of LQR is 2 seconds, while MPC parameters are chosen as follows: sample time is 10 seconds, prediction horizon is 360, and control horizon is 60. Simulation results under LQR are shown in Figure 15 (reservoirs' capacities) and Figure 16 (control signals). In addition, simulation results under MPC are shown in Figure 17 (reservoirs' capacities) and Figure 18 (control signals). Finally, we implement the same quantitative comparison

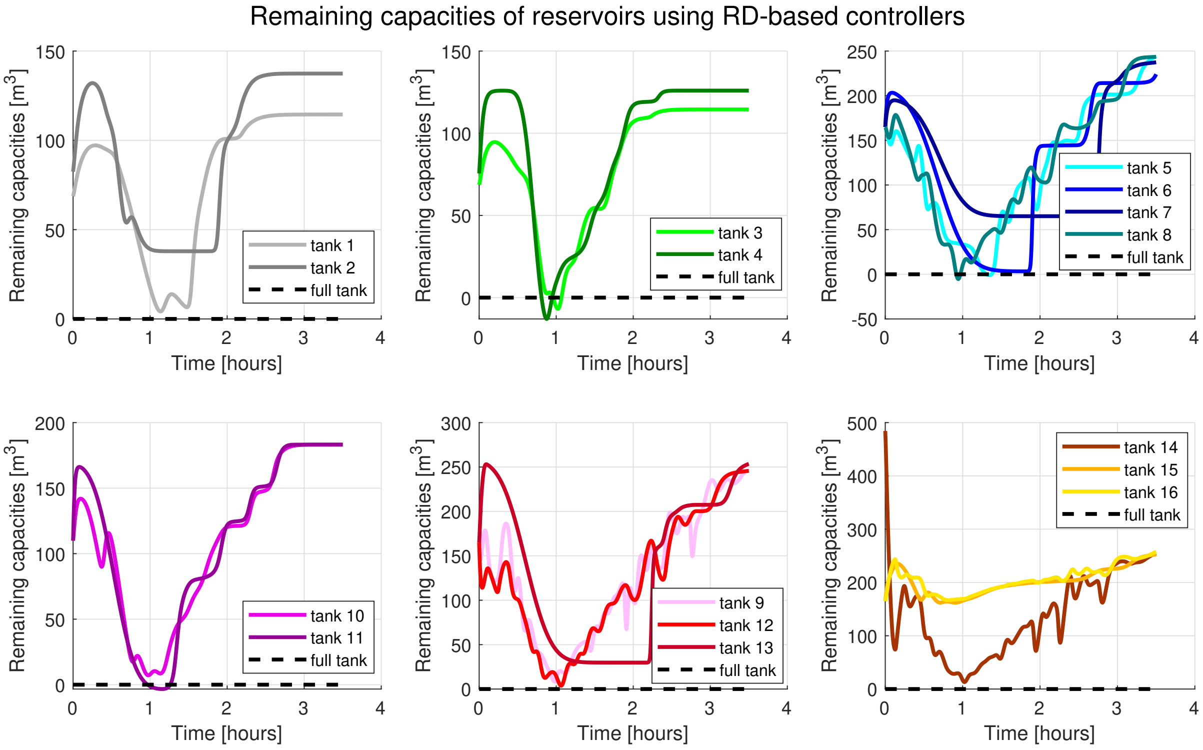


Figure 13. Evolution of the remaining capacities of reservoirs of the Bogotá (Colombia) case study under replicator dynamics based controllers.

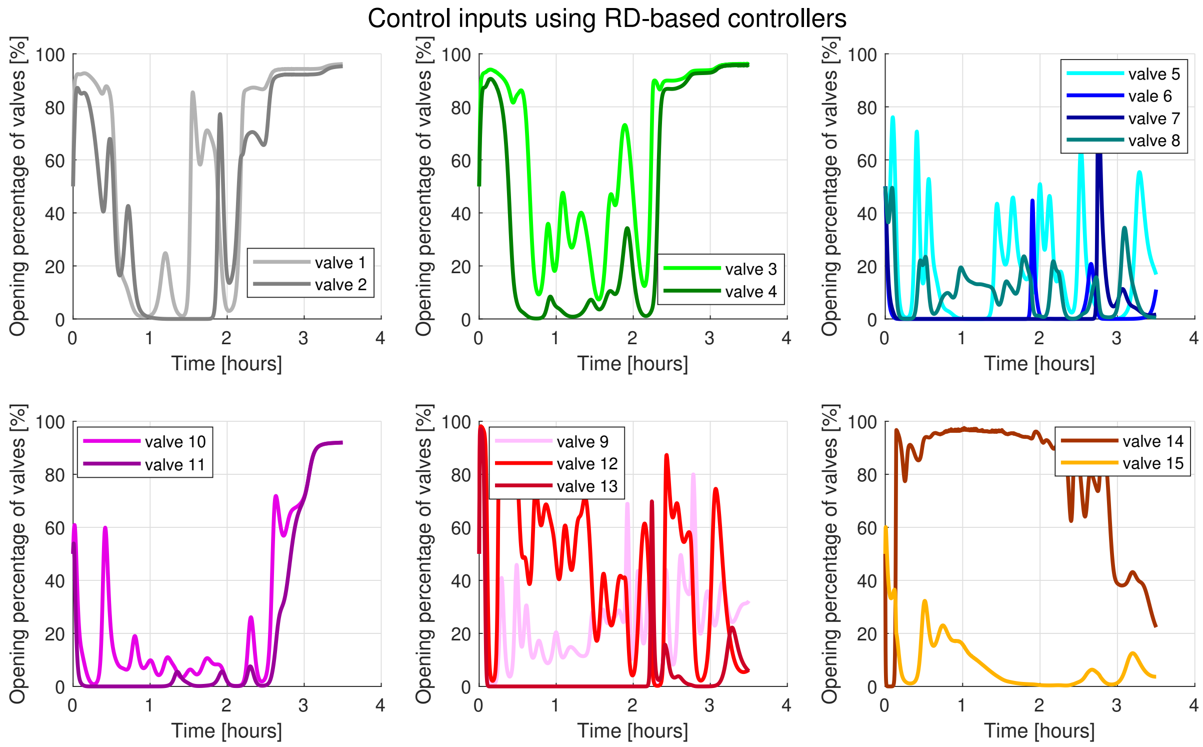


Figure 14. Opening percentage of reservoirs' valves for the Bogotá (Colombia) case study under replicator dynamics based controllers.

performed in the first case study, the results are summarized in Table 4. Notice that we obtain similar results as the ones of case study 1, i.e., replicator dynamics controller outperforms LQR and its behavior is quite close to centralized MPC, but with computational advantages as discussed next.

Table 4. Comparison of controllers' performance for case study 2

Rain amplifying factor	Flooding [m^3]			
	Without control	Replicator dynamics	LQR	MPC
0.90	672.33	0	0	0
0.95	734.08	0	77.99	0
1.00	797.06	29.38	149.40	1.03
1.05	861.29	112.78	261.45	78.89
1.10	926.86	243.28	401.72	223.20

C. COMPUTATIONAL BURDEN COMPARISON

An important advantage of the proposed control technique is its low computational burden, especially when we compare replicator dynamics with optimization-based controllers such as LQR and MPC. This advantage rests on the philosophy behind the controllers. While techniques based on optimization should minimize a cost function subject to constraints at each control step, replicator dynamics only require the solution of a set of ordinary differential equations. The key point is that iterative algorithms are required for solving constrained optimization problems. This fact implies that, for each control step, optimization-based controllers should perform several iterations, which increases the computational burden. In contrast, each time step of replicator dynamics only requires the evaluation of the right-hand side of the replicator equation in (7) to update the control input (using, for instance, the Euler approximation). Clearly, the computational complexity of replicator dynamics is significantly lower. To demonstrate this fact, we make a comparison of the computational time taken for simulating each case study under the three different control strategies implemented in Sections VII-A and VII-B, i.e., decentralized replicator dynamics, decentralized LQR and centralized MPC. To perform the simulations, we use a machine whose characteristics are summarized in Table 5. The results of the comparison are shown in Table 6. In this table, the sub-index of r indicates the rain scale factor, which is explained in the simulation conditions of both case studies (see Sections VII-A and VII-B). For instance, $r_{1.10}$ denotes that the rain amplitude of the baseline case is increased 10%.

Notice that in all cases the controller based on replicator dynamics is the fastest. In fact, for case study 1, replicator dynamics is three times faster than decentralized LQR and 14 times faster than centralized MPC (the slower method). These differences are higher for case study 2, where replicator dynamics is six times faster than decentralized LQR and 122 times faster than centralized MPC. On the other hand, it is worth mentioning that replicator dynamics exhibit better scalability. Indeed, if we study the computational time taken in case study 1 (UDS with 7 reservoirs) compared to case study 2 (UDS with 16 reservoirs), it can be noticed that the increment is 32 times for centralized MPC, seven times for decentralized LQR and only 3.7 times for replicator dynamics. The fact that LQR and replicator dynamics scale much better than MPC is a consequence of the decentralized nature of the first two controllers, whose parallelization plays a key role (i.e., each UDS sub-system is capable of computing its own control inputs independently). In addition, another factor that makes replicator dynamics beat the two optimization-based methods in terms of scalability is the different paradigm that replicator dynamics use to update the control inputs. Specifically, when the UDS scale increases, optimization-based controllers require to solve minimization problems with a larger number of decision variables (under this situation, centralized MPC is highly penalized because it must deal with all decision variables of the whole UDS). On the contrary, replicator dynamics only need the evaluation of more algebraic equations to update the control input. As explained before, this evaluation requires significantly lower computations than solving a constrained optimization problem.

Table 5. Characteristics of the machine used for simulations

Item	Value
Processor type	Core i7-8700K
Processor's cores	6
Processor's frequency	3.7 GHz
RAM memory	32 GB DDR4
Operating system	Windows 11
Simulation program	Matlab R2021b
ODE solver	Dormand-Prince method
Optimization method	QPKWIK [56]

VIII. CONCLUSIONS

Floods can be mitigated using real time control of Urban Drainage Systems (UDSs). Since the size of these kinds of systems is generally of large-scale nature, designing a central control (which drives all the actuators and has information from all UDS's sensors) is problematic. Instead, it is desirable to design decentralized control schemes, i.e., schemes with multiple controllers, where each one of them has only partial information of the system. We have shown that graph models and population dynamics can be employed to design local controllers that independently handle partitions of the UDS. The proposed scheme allocates wastewater among UDS collectors (modeled as reservoirs) in an efficient way. Specifically, we have proven that the designed local controllers perform wastewater allocation in such a way that the collectors of the UDS tend to reach the same remaining storing capacity while preserving closed-loop system stability. Simu-

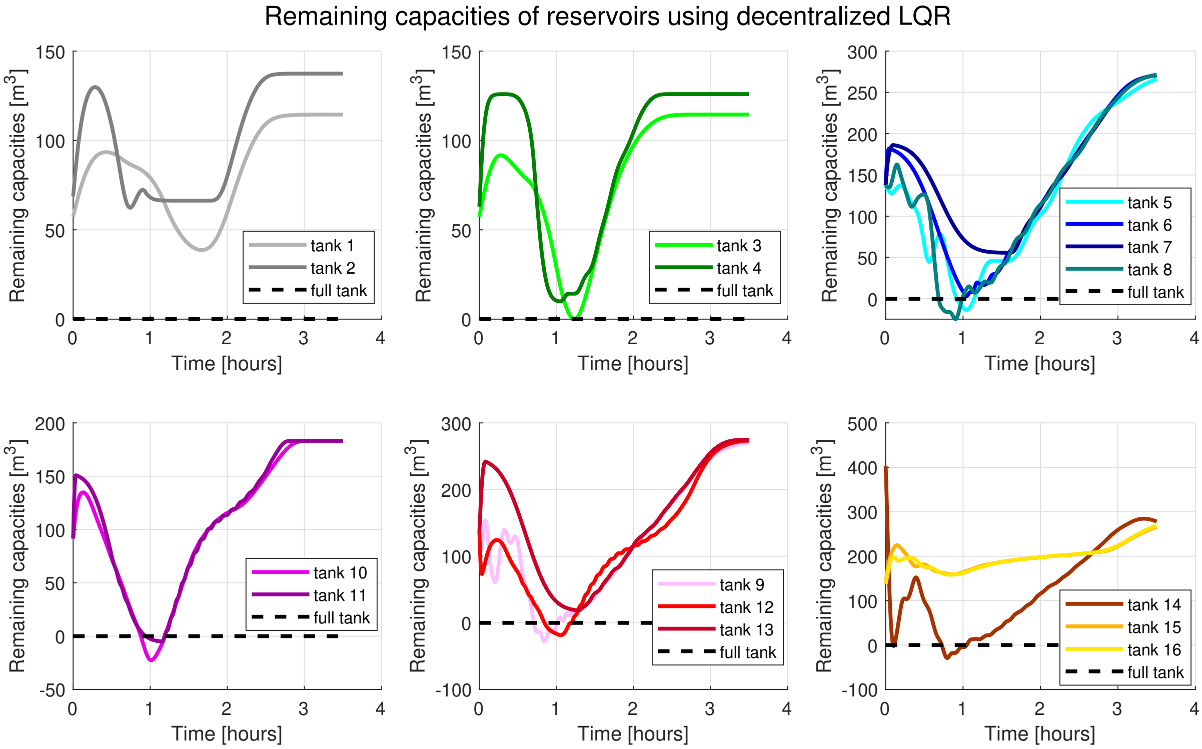


Figure 15. Evolution of the remaining capacities of reservoirs of the Bogotá (Colombia) case study under decentralized LQR.

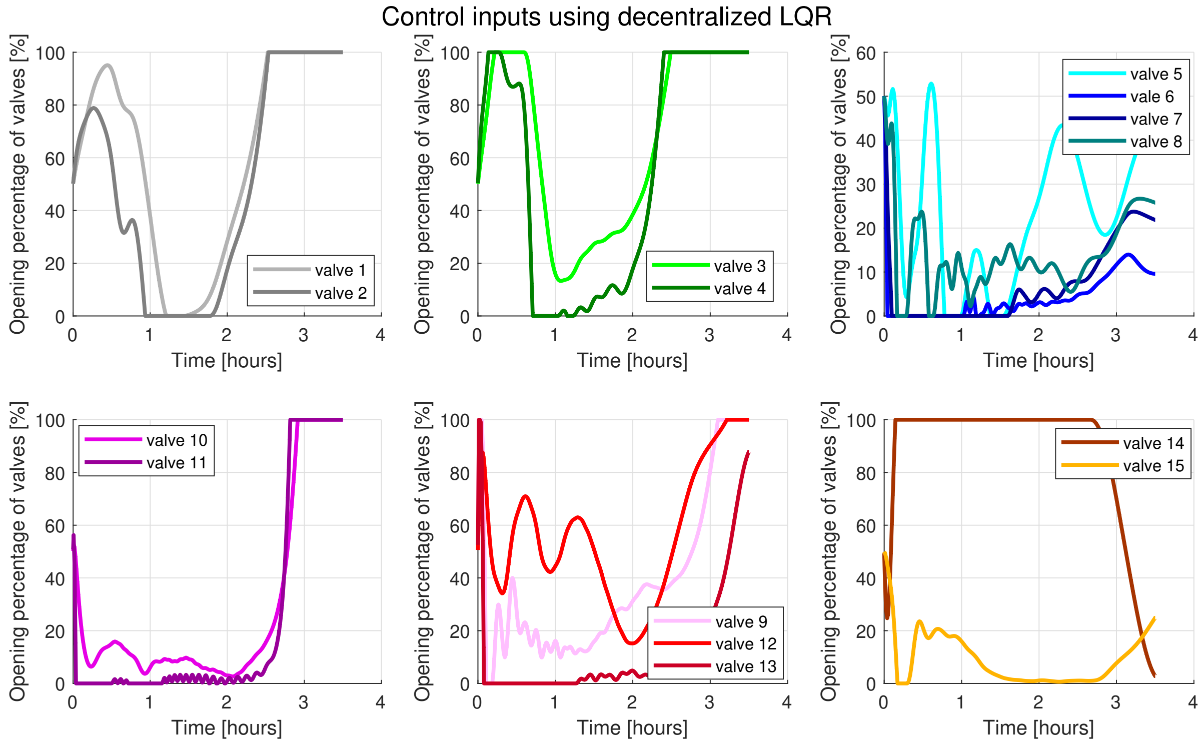


Figure 16. Opening percentage of reservoirs' valves for the Bogotá (Colombia) case study under decentralized LQR.

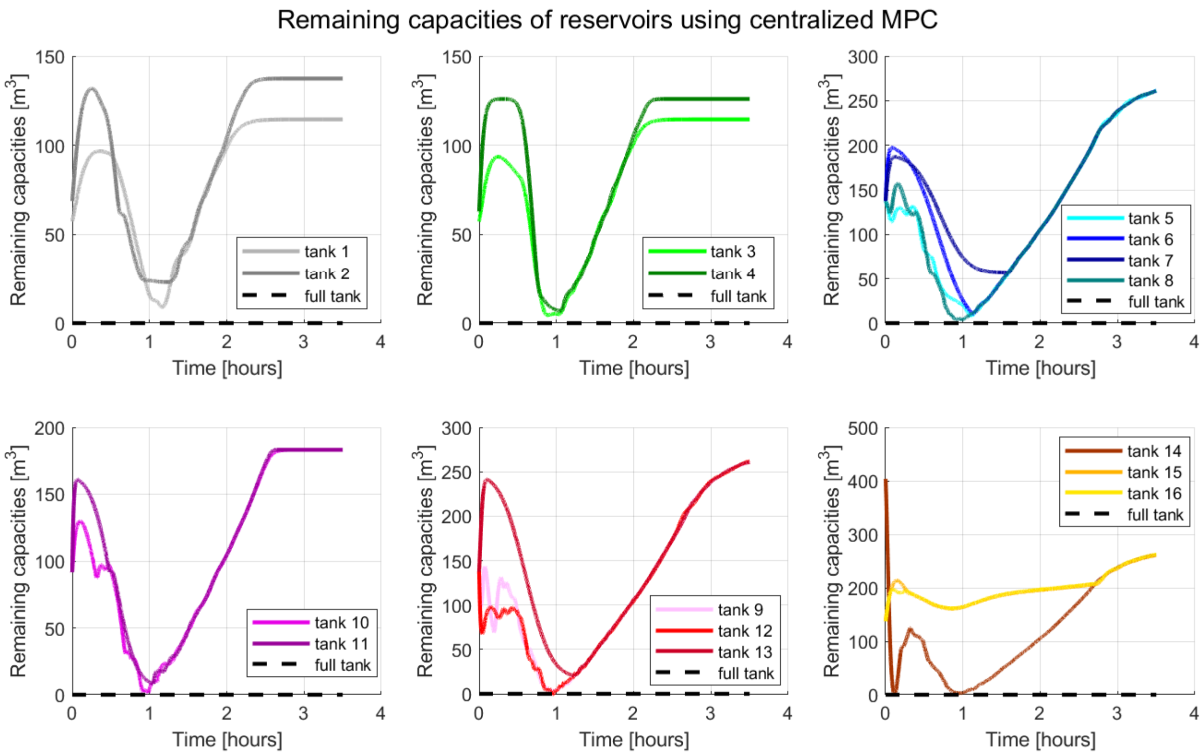

Remaining capacities of reservoirs using centralized MPC

Figure 17. Evolution of the remaining capacities of reservoirs of the Bogotá (Colombia) case study under centralized MPC.

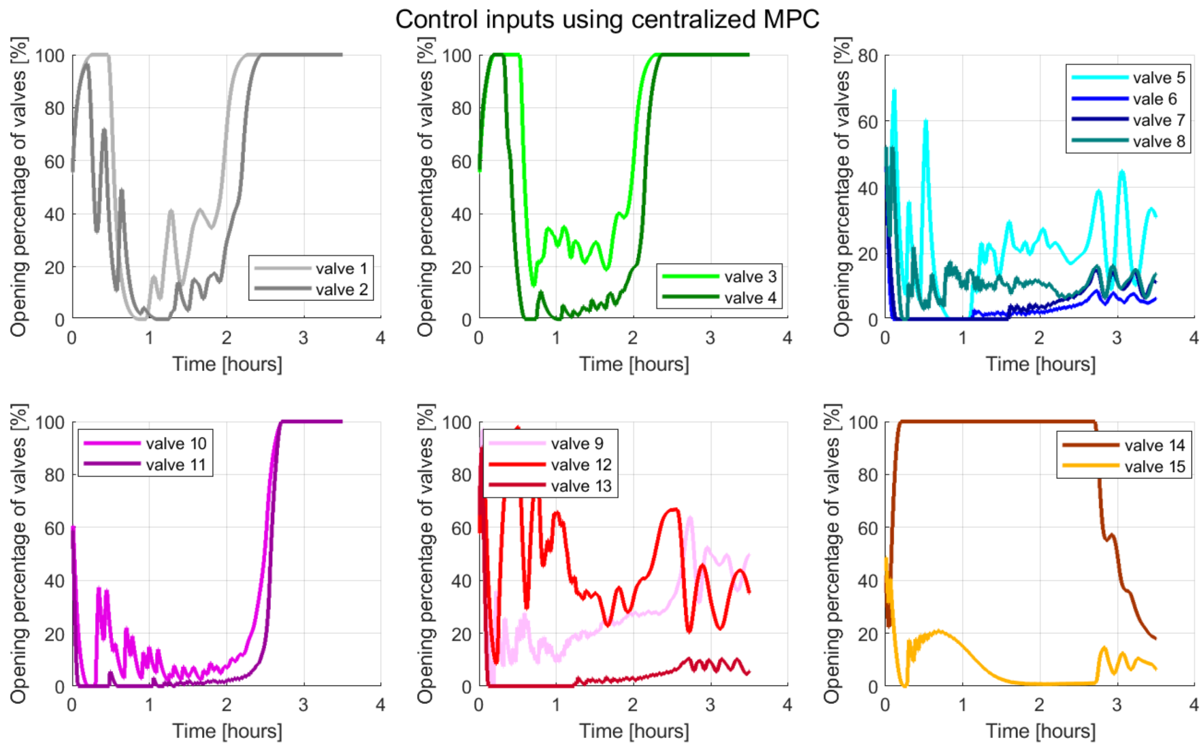

Control inputs using centralized MPC

Figure 18. Opening percentage of reservoirs' valves for the Bogotá (Colombia) case study under centralized MPC.

Table 6. Comparison of computational time taken by UDSs controllers

Controller	Computational time [s]									
	Case study 1					Case study 2				
	$r_{0.95}$	$r_{1.00}$	$r_{1.05}$	$r_{1.10}$	average	$r_{0.95}$	$r_{1.00}$	$r_{1.05}$	$r_{1.10}$	average
Replicator dynamics	0.56	0.56	0.55	0.55	**0.55**	2.01	2.07	2.06	2.15	**2.07**
LQR	1.82	1.80	1.80	1.86	**1.82**	12.68	12.75	12.78	12.80	**12.75**
MPC	7.78	7.84	8.03	8.09	**7.93**	248.36	252.21	255.78	257.83	**253.54**

lations have shown that the performed wastewater allocation mitigates flooding even in the face of heavy rain scenarios. As future work, we propose to improve the performance of the controlled system by incorporating prediction of the UDS state and forecasting of external inputs. Besides, we propose to study tuning algorithms for finding the optimal parameters of the local controllers.

ACKNOWLEDGEMENTS

We would like to thank Julián Barreiro-Gomez for his valuable inputs.

N. Quijano would like to thank the UK- PACT Project (2019-2021), Systemic perspectives on low-carbon cities in Colombia - An integrated urban modelling approach for policy and regulatory analysis.

The work of C. Ocampo-Martinez and N. Quijano is supported by the Spanish project PID2020-115905RB-C21 (L-BEST) funded by MCIN/ AEI /10.13039/501100011033.

References

[1] H. Xu, C. Ma, K. Xu, J. Lian, and Y. Long, "Staged optimization of urban drainage systems considering climate change and hydrological model uncertainty," Journal of Hydrology, vol. 587, p. 124959, 2020.

[2] S. Hettiarachchi, C. Wasko, and A. Sharma, "Increase in flood risk resulting from climate change in a developed urban watershed–the role of storm temporal patterns," Hydrology and Earth System Sciences, vol. 22, no. 3, pp. 2041–2056, 2018.

[3] T. D. Fletcher, H. Andrieu, and P. Hamel, "Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art," Advances in water resources, vol. 51, pp. 261–279, 2013.

[4] D. Waters, W. E. Watt, J. Marsalek, and B. C. Anderson, "Adaptation of a storm drainage system to accommodate increased rainfall resulting from climate change," Journal of Environmental Planning and Management, vol. 46, no. 5, pp. 755–770, 2003.

[5] L. García, J. Barreiro-Gomez, E. Escobar, D. Téllez, N. Quijano, and C. Ocampo-Martinez, "Modeling and real-time control of urban drainage systems: A review," Advances in Water Resources, vol. 85, pp. 120–132, 2015.

[6] A. L. Mollerup, P. S. Mikkelsen, and G. Sin, "A methodological approach to the design of optimising control strategies for sewer systems," Environmental Modelling & Software, vol. 83, pp. 103–115, 2016.

[7] S. Eggimann, L. Mutzner, O. Wani, M. Y. Schneider, D. Spuhler, M. Moy de Vitry, P. Beutler, and M. Maurer, "The potential of knowing more: A review of data-driven urban water management," Environmental science & technology, vol. 51, no. 5, pp. 2538–2553, 2017.

[8] J. Vazquez, D. Bellefleur, D. Gilbert, and B. Grandjean, "Real time control of a combined sewer network using graph theory," Water science and technology, vol. 36, no. 5, pp. 301–308, 1997.

[9] B. Joseph-Duran, M. N. Jung, C. Ocampo-Martinez, S. Sager, and G. Cembrano, "Minimization of sewage network overflow," Water Resources Management, vol. 28, no. 1, pp. 41–63, 2014.

[10] J. M. Sadler, J. L. Goodall, M. Behl, M. M. Morsy, T. B. Culver, and B. D. Bowes, "Leveraging open source software and parallel computing for model predictive control of urban drainage systems using EPA-SWMM5," Environmental Modelling & Software, vol. 120, p. 104484, 2019.

[11] H. Beak, T.-H. Kim, J. Ryu, and J. Oh, "Diversity-enhanced particle swarm optimizer and its application to optimal flow control of sewer networks," in 2013 Science and Information Conference, pp. 977–978, IEEE, 2013.

[12] X. Dong, S. Huang, and S. Zeng, "Design and evaluation of control strategies in urban drainage systems in Kunming city," Frontiers of Environmental Science & Engineering, vol. 11, no. 4, pp. 1–8, 2017.

[13] J. A. van der Werf, Z. Kapelan, and J. Langeveld, "Quantifying the true potential of real time control in urban drainage systems," Urban Water Journal, vol. 18, no. 10, pp. 873–884, 2021.

[14] M. Marinaki and M. Papageorgiou, "Linear-quadratic regulators applied to sewer network flow control," in Proceedings of the 2003 European Control Conference (ECC), pp. 2407–2412, IEEE, 2003.

[15] C. Ocampo-Martinez, Model predictive control of wastewater systems. Springer Science & Business Media, 2010.

[16] K. M. Balla, C. Schou, J. D. Bendtsen, C. Ocampo-Martinez, and C. S. Kallesøe, "A nonlinear predictive control approach for urban drainage networks using data-driven models and moving horizon estimation," IEEE Transactions on Control Systems Technology, (in press).

[17] G. Cembrano, J. Quevedo, M. Salamero, V. Puig, J. Figueras, and J. Martí, "Optimal control of urban drainage systems. a case study," Control engineering practice, vol. 12, no. 1, pp. 1–9, 2004.

[18] S. S. Farahani, S. Soudjani, R. Majumdar, and C. Ocampo-Martinez, "Formal controller synthesis for wastewater systems with signal temporal logic constraints: The Barcelona case study," Journal of Process Control, vol. 69, pp. 179–191, 2018.

[19] C. Ocampo-Martinez, V. Puig, G. Cembrano, and J. Quevedo, "Application of predictive control strategies to the management of complex networks in the urban water cycle [applications of control]," IEEE Control Systems Magazine, vol. 33, no. 1, pp. 15–41, 2013.

[20] J. Li, "A data-driven improved fuzzy logic control optimization-simulation tool for reducing flooding volume at downstream urban drainage systems," Science of the Total Environment, vol. 732, p. 138931, 2020.

[21] A. Mullapudi, B. P. Wong, and B. Kerkez, "Emerging investigators series: building a theory for smart stormwater systems," Environmental Science: Water Research & Technology, vol. 3, no. 1, pp. 66–77, 2017.

[22] C. Ocampo-Martinez, V. Puig, J. Grosso, and S. Montes-de-Oca, "Multi-layer decentralized MPC of large-scale networked systems," in Distributed model predictive control made easy, pp. 495–515, Springer, 2014.

[23] N. Quijano, C. Ocampo-Martinez, J. Barreiro-Gomez, G. Obando, A. Pantoja, and E. Mojica-Nava, "The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory," IEEE Control Systems Magazine, vol. 37, no. 1, pp. 70–97, 2017.

[24] B. Gao and L. Pavel, "On passivity, reinforcement learning, and higher order learning in multiagent finite games," IEEE Transactions on Automatic Control, vol. 66, no. 1, pp. 121–136, 2020.

[25] J. Barreiro-Gomez, F. Dörfler, and H. Tembine, "Distributed robust population games with applications to optimal frequency control in power systems," in 2018 Annual American Control Conference (ACC), pp. 5762–5767, IEEE, 2018.

[26] A. Pashaie, L. Pavel, and C. J. Damaren, "A population game approach for dynamic resource allocation problems," International Journal of Control, vol. 90, no. 9, pp. 1957–1972, 2017.

[27] J. Martinez-Piazuelo, N. Quijano, and C. Ocampo-Martinez, "A payoff dy-

namics model for equality-constrained population games," IEEE Control Systems Letters, vol. 6, pp. 530–535, 2021.

[28] C. Sun and G. Hu, "Continuous-time penalty methods for Nash equilibrium seeking of a nonsmooth generalized noncooperative game," IEEE Transactions on Automatic Control, vol. 66, no. 10, pp. 4895–4902, 2020.

[29] G. Belgioioso, A. Nedić, and S. Grammatico, "Distributed generalized Nash equilibrium seeking in aggregative games on time-varying networks," IEEE Transactions on Automatic Control, vol. 66, no. 5, pp. 2061–2075, 2020.

[30] H. Jaleel and J. S. Shamma, "Distributed optimization for robot networks: From real-time convex optimization to game-theoretic self-organization," Proceedings of the IEEE, vol. 108, no. 11, pp. 1953–1967, 2020.

[31] S. Krilašević and S. Grammatico, "Learning generalized Nash equilibria in multi-agent dynamical systems via extremum seeking control," Automatica, vol. 133, p. 109846, 2021.

[32] F. Genis Mendoza, G. Konstantopoulos, and D. Bauso, "Online pricing for demand-side management in a low-voltage resistive micro-grid via a Stackelberg game with incentive strategies," IET Smart Grid, 2021.

[33] M. Arcak and N. C. Martins, "Dissipativity tools for convergence to Nash equilibria in population games," IEEE Transactions on Control of Network Systems, vol. 8, no. 1, pp. 39–50, 2020.

[34] S. Grammatico, F. Parise, M. Colombino, and J. Lygeros, "Decentralized convergence to Nash equilibria in constrained deterministic mean field control," IEEE Transactions on Automatic Control, vol. 61, no. 11, pp. 3315–3329, 2015.

[35] W. Lee, S. Liu, H. Tembine, W. Li, and S. Osher, "Controlling propagation of epidemics via mean-field control," SIAM Journal on Applied Mathematics, vol. 81, no. 1, pp. 190–207, 2021.

[36] Z. Lu, F. Cai, X. Wu, S. Zhang, and S. Wu, "Evolution of water resource allocation in the river basin between administrators and managers," Hydrology Research, 2022.

[37] J. Barreiro-Gómez, G. Obando, G. Riaño-Briceño, N. Quijano, and C. Ocampo-Martinez, "Decentralized control for urban drainage systems via population dynamics: Bogotá case study," in Proceedings of the 2015 European Control Conference (ECC), pp. 2426–2431, IEEE, 2015.

[38] N. Kändler, I. Annus, A. Vassiljev, and R. Puust, "Real time controlled sustainable urban drainage systems in dense urban areas," Journal of Water Supply: Research and Technology-AQUA, vol. 69, no. 3, pp. 238–247, 2020.

[39] K. Diao, "Multiscale resilience in water distribution and drainage systems," Water, vol. 12, no. 6, p. 1521, 2020.

[40] G. Garofalo, A. Giordano, P. Piro, G. Spezzano, and A. Vinci, "A distributed real-time approach for mitigating CSO and flooding in urban drainage systems," Journal of Network and Computer Applications, vol. 78, pp. 30–42, 2017.

[41] B. Dobson, H. Watson-Hill, S. Muhandes, M. Borup, and A. Mijic, "A reduced complexity model with graph partitioning for rapid hydraulic assessment of sewer networks," Water Resources Research, pp. 1–21, 2021.

[42] Y. Li, D. Stipanović, P. Voulgaris, and Z. Gu, "Decentralized model predictive control of urban drainage systems," WSEAS Transactions on Systems and Control, vol. 14, pp. 247–256, 2019.

[43] F. Farokhi, I. Shames, and K. H. Johansson, "Distributed MPC via dual decomposition and alternative direction method of multipliers," in Distributed model predictive control made easy, pp. 115–131, Springer, 2014.

[44] A. Abbas, I. Carnacina, F. Ruddock, R. Alkhaddar, G. Rothwell, and R. Andoh, "An innovative method for installing a separate sewer system in narrow streets," Journal of Water Management Modeling, 2019.

[45] S. De Toffol, C. Engelhard, and W. Rauch, "Combined sewer system versus separate system–a comparison of ecological and economical performance indicators," Water science and technology, vol. 55, no. 4, pp. 255–264, 2007.

[46] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," Mathematical biosciences, vol. 40, no. 1, pp. 145–156, 1978.

[47] J. W. Weibull, Evolutionary Game Theory. London, England: The MIT Press, 1997.

[48] G. Obando, A. Pantoja, and N. Quijano, "Building temperature control based on population dynamics," IEEE Transactions on Control Systems Technology., vol. 22, no. 1, pp. 404–412, 2014.

[49] E. Mojica-Nava, C. Macana, and N. Quijano, "Dynamic population games for optimal dispatch on hierarchical microgrid control," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 3, pp. 306–317, 2014.

[50] A. Pantoja, G. Obando, and N. Quijano, "Distributed optimization with information-constrained population dynamics," Journal of the Franklin Institute, vol. 356, no. 1, pp. 209–236, 2019.

[51] V. T. Chow, Open channel hydraulics. McGraw-Hill Book Company, Inc; New York, 1959.

[52] G. Conde, N. Quijano, and C. Ocampo-Martinez, "Modeling and control in open-channel irrigation systems: A review," Annual Reviews in Control, vol. 51, no. 1, pp. 153–171, 2021.

[53] W. H. Sandholm, Population games and evolutionary dynamics. Cambridge, Mass. MIT Press, 2010.

[54] J. M. Lemos and L. F. Pinto, "distributed linear-quadratic control of serially chained systems: Application to a water delivery canal," IEEE Control Systems Magazine, vol. 32, no. 6, pp. 26–38, 2012.

[55] N. S. V. Lund, A. K. V. Falk, M. Borup, H. Madsen, and P. Steen Mikkelsen, "Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management," Critical Reviews in Environmental Science and Technology, vol. 48, no. 3, pp. 279–339, 2018.

[56] C. Schmid and L. T. Biegler, "Quadratic programming methods for reduced Hessian SQP," Computers & chemical engineering, vol. 18, no. 9, pp. 817–832, 1994.

GERMÁN OBANDO received the B.S. degree in electronics from Universidad de Nariño, Pasto, Colombia, in 2008; the M.S. degree in electronics engineering from Universidad de los Andes, Bogotá, Colombia, in 2010; and the Ph.D. degree in electronics engineering from Universidad de los Andes and Ecole des Mines de Nantes, France, in 2016. He joined the School of Engineering, Science and Technology of Universidad del Rosario, Bogotá, Colombia, in 2017, where he is currently an Assistant Professor and the director of the B.S. in electronics engineering. In 2020, he was appointed as Distinguished Lecturer from Universidad del Rosario. His current research interests include population dynamics, distributed control and time delayed systems.

NICANOR QUIJANO (IEEE Senior Member) received his B.S. degree in Electronics Engineering from Pontificia Universidad Javeriana (PUJ), Bogotá, Colombia, in 1999. He received the M.S. and PhD degrees in Electrical and Computer Engineering from The Ohio State University, in 2002 and 2006, respectively. In 2007, he joined the Electrical and Electronics Engineering Department, Universidad de los Andes (UAndes), Bogotá, Colombia as an Assistant Professor. He is currently a Full Professor, the director of the research group in control and automation systems (GIAP, UAndes), and an associate editor for the IEEE Transactions on Control Systems Technology, the Journal of Modern Power Systems and Clean Energy, and Energy Systems. He has been a member of the Board of Governors of the IEEE Control Systems Society (CSS) for the 2014 period, and he was the chair of the IEEE CSS, Colombia for the 2011-2013 period. He has published more than 100 scientific papers (journal papers, international conference papers, book chapters), he has co-advised the best European PhD thesis in the control systems area in 2017, and he is the co-author of the best paper of the ISA Transactions, 2018. In 2021, he obtained the Experienced Research Award from the School of Engineering, UAndes. Currently his research interests include hierarchical and distributed network optimization methods for control using learning, bio-inspired, and game-theoretical techniques for dynamic resource allocation problems, especially those in energy, water, agriculture, and transportation.

CARLOS OCAMPO–MARTINEZ received his Ph.D. degree in Control Engineering from the Universitat Politècnica de Catalunya (Barcelona, Spain). In 2007–2008, he held a postdoctoral position at the ARC Centre of Complex Dynamic Systems and Control (University of Newcastle, Australia) and, afterwards at the Spanish National Research Council (CSIC), Institut de Robòtica i Informàtica Industrial, CSIC-UPC (Barcelona) as a Juan de la Cierva research fellow between 2008 and 2011. Since 2011, he is with the Universitat Politècnica de Catalunya, Automatic Control Department (ESAII) as Associate Professor in automatic control and model predictive control. From 2014 until 2018, he has been also Deputy Director of the Institut de Robòtica i Informàtica Industrial (CSIC-UPC), a Joint Research Center of UPC and CSIC. His main research interests include constrained model predictive control, large-scale systems management (partitioning and noncentralized control), process control and industrial applications (mainly related to the key scopes of water and energy, and smart manufacturing under the IoT framework).