

# A Differential Game Approach to Urban Drainage Systems Control

A. Ramirez-Jaime, N. Quijano, *Senior Member, IEEE*, and C. Ocampo-Martinez, *Senior Member, IEEE*.

**Abstract**—Urban drainage systems (UDSs) are complex large-scale systems that carry stormwater and wastewater throughout urban areas. During heavy rain scenarios, UDSs are not able to handle the amount of extra water that enters the network and flooding occurs. Usually, this might happen because the network is not being used efficiently, i.e., some structures remain underused while many others are overused. This paper proposes a control methodology based on differential game theory that aims to efficiently use the existing network elements in order to minimize overflows and properly manage the water resource. The proposed controller is tested on a typical UDS and is compared with a centralized MPC achieving similar results in terms of flooding minimization and network usage, but only using local information on distributed controllers.

## I. INTRODUCTION

Urban drainage systems (UDS) are complex networks of interconnected pipes and nodes that carry stormwater and wastewater to wastewater treatment plants, which treats it and sends it to the environment [4]. In many cases, the design of these networks ends up being under-dimensioned, because of rapid urbanization in cities and climate change scenarios, not being taken into account in early stages of the design process [23]. For that reason, heavy flooding may appear in urban areas, and some serious sanitary problems may occur due to the improper management of wastewater that comes out of the network into street level [4]. Some of the addressed solutions to that problem seek to do a restructuring to the hydraulic design of the network by adding storage elements throughout the system, so that the overflows are totally avoided [19]. Even though these solutions are quite effective, they are extremely expensive in both time and money.

The problem above could be solved, without having to perform many large modifications to the general design of the network, by using real-time control techniques [12]. These techniques seek to find a way to properly manage the active elements of the network, e.g., retention and redirection gates, in order to achieve an efficient management of the wastewater, and thus, assuring that possible overflows are minimized. Optimization-based control techniques have been the most extensively used techniques in the literature to

solve the problem of minimization of overflows in UDS. For instance, model predictive control (MPC) has been widely used to solve the problem [6], [21], due to its flexibility in the selection of performance functions, constraints, and its capabilities to deal with multiple inputs and multiple outputs [15]. However, many of the proposed techniques are based upon centralized schemes for the determination of control actions to be performed, which could derive into heavy computational burden problems [16] and cyber security-related problems [5]. For those reasons, there has been an increased interest in studying distributed control techniques [8]. For instance, [1] proposes a distributed control methodology based on population dynamics, that achieves an efficient use of the network and guarantees a minimization of flooding. However, in that methodology, local controllers are not able to consider arbitrary cost functions, which can be problematic if there are multiple control goals such as, moving wastewater between a wastewater treatment plant and out of the network, while efficiently using the network and minimizing flooding.

This paper proposes a technique that aims to solve problems related to distributed information on local controllers, as well as problems related to the consideration multiple cost functions, by using a game-theoretic approach (i.e., differential games). Differential game (DG) theory [3] gives a natural extension of optimal control to scenarios with multiple controllers that are optimizing its own performance criteria [18], and thus its framework is well suited for optimization-based distributed control applications. This type of games have been used in the literature to solve problems related to the formation control of mobile robots [13], problems related to demand response in power grids [11], and the control of surge tanks [10]. This is due to the fact that DGs have the ability to consider multiple cost functions as well as non-centralized information on distributed controllers. As for the UDS control problem, it has been reported that these networks can be seen as decoupled systems that are being controlled by multiple local agents that interact with each other [1]. Hence, it is a suitable idea to apply the DGs framework to the control of UDS, where multiple local controllers act as players of a game where they interact with each other, in order to guarantee a proper operation of the network in terms of wastewater management.

The main contribution of this paper is the design of a distributed control methodology based on DGs, in the same spirit as in [13], [18], [17], which seeks to determine the optimal behavior of each active element of the UDS by using a consensus-like algorithm, only using local information of the network. The proposed methodology has the advantages of optimization-based techniques used for the control of

A. Ramirez-Jaime and N. Quijano are with Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1ª No 18A-10, Colombia {af.ramirez236, nquijano}@uniandes.edu.co

C. Ocampo-Martinez is with the Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain. cocampo@iri.upc.edu

Authors would like to thank Mexichem (Colombia) for supporting this research through the project “*Drenaje Urbano y Cambio Climático: hacia los sistemas de alcantarillado del futuro.*” Fase II. COLCIENCIAS 633/2013. Also, this work has been partially supported by the spanish project ECOCIS (Ref. DPI2013-48243-C2-1-R).

UDS, e.g., MPC, as well as the ability to have distributed information on the controllers. Moreover, since only local information is used, less data is needed, and thus less computational resources are involved in the computation of the control inputs.

The remainder of this paper is organized as follows. Section II presents the problem statement. Section III presents the DG associated with the control of the UDS, and how its solution looks like in terms of open-loop strategies. Section IV shows a case study of a typical UDS where the proposed controller is going to be tested. Section V shows the results of the controller applied to the proposed network, as well as a comparison with a centralized MPC. Finally, Section VI collects the concluding remarks of the paper.

*Notation:* Bold lower case letters are used to denote column vectors, e.g.,  $\mathbf{v}$ . Calligraphy style letters are used to denote sets, e.g.,  $\mathcal{S}$ . Upper bold case letters are used to denote matrices, e.g.,  $\mathbf{A}$ . Finally,  $\mathbb{I}_n$  represents the identity matrix of  $n \times n$ ,  $0_n$  represents a matrix with null entries of  $n \times n$ ,  $\mathbb{R}_+$  represents all the positive real numbers, and  $\mathbb{Z}_+$  represents all the positive integer numbers.

## II. PROBLEM STATEMENT

Typically, UDS have a strong convergence topology where many pipes end up into a common outlet node, until the drain node is reached. This causes most of the burden to be suffered downstream and quite little burden to be taken upstream of the network. This means that, usually, upstream pipes remain underused, so they could retain some water (by using retention gates) in order to minimize overflows downstream. The proposed scheme aims to solve that problem, by using a decentralized controller based on DGs, so that most of the pipes on the UDS are used efficiently, and the total overflow is minimized.

1) *Urban drainage system model:* Water running through the pipes of a UDS can be modeled by using the Saint-Venant (SV) equations, which use mass and momentum conservation principles, in order to describe the phenomena occurring inside the pipes [7]. These equations describe the flow of water in quite a high level of detail. However, that level of detail is usually not required for control design. For that reason, a control-oriented model based on the *virtual-reservoirs* model is used. In this approach, the UDS is divided into a set of interconnected real and virtual tanks (VT). According to [21], a VT is a storage element that represents the total volume of wastewater inside the pipes associated to a determined portion of the network. The volume is calculated via a mass balance equation, so that the state equation of the  $i$ -th tank is given by  $\dot{v}_i(t) = q_i^{in}(t) - q_i^{out}(t)$ ,  $\forall i \in \{1, \dots, N+1\}$ , where  $v_i$  is the total volume of water of the reservoir,  $q_i^{in}$  and  $q_i^{out}$  are the total inflow and outflow coming in and out of the  $i$ -th reservoir, and  $N+1$  is the total number of reservoirs. For this paper, it is assumed that the reservoirs are linear, and therefore the outflow of every tank is proportional to the volume stored in it, i.e.,  $q_i^{out}(t) = k_i v_i(t)$ , where  $k_i$  is a *volume/flow* conversion (VFC) constant.

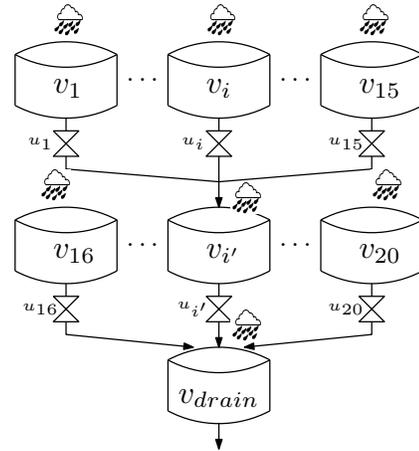


Fig. 1. Typical tree topology of a UDS after the simplification based on VTs is applied.

Finally, it is assumed that there is a retention gate located at the output of every VT. This implies that the manipulated inputs of the system are the outflows of VTs, which can be adjusted by opening or closing the gates. Since the outflow is proportional to the volume of the reservoir, the constraint  $0 \leq u_i(t) \leq k_i v_i(t)$ ,  $\forall t \in \mathbb{R}_+$ , where  $u_i$  is the manipulated outflow of the  $i$ -th reservoir, must hold.

The tree-shape topology of the UDS can be simplified by using the VT model by saying that any portion of a given UDS can be seen as a collection of tanks, whose outflows converge into a common reservoir, until the outlet reservoir is reached. For instance, Figure 1 shows a typical tree topology with 21 VTs of a UDS after the simplification is used. Notice how the whole network eventually converges to the drain reservoir  $v_{drain}$ , and every VT (least the drain VT) has a retention gate located at the output that regulates the outflow.

Hence, the state equation of a single tank can be written as  $\dot{v}_i(t) = -u_i(t) + \sum_{j \in \mathcal{S}_i} u_j(t) + d_i(t)$ , where  $\mathcal{S}_i$  is the set of tanks whose outflows go directly to the  $i$ -th tank, and  $d_i$  is the total inflow from rainfall entering the  $i$ -th reservoir. It is important to notice that  $d_i$  acts as a disturbance that alters the state of the  $i$ -th tank. Since the last tank of the network does not have a retention gate at the output, its state equation is written as  $\dot{v}_{N+1}(t) = -k_{N+1} v_{N+1}(t) + \sum_{j \in \mathcal{S}_{N+1}} u_j(t) + d_{N+1}(t)$ , where  $\mathcal{S}_{N+1}$  is the set of tanks whose outflows go directly into the drain tank. This equation states that the drain tank cannot control its outflow.

2) *Information graph:* A directed graph can be used to represent the interactions among the tanks of a UDS [14]. This representation gives useful insights on how water moves throughout the pipes, and it is also useful in distributed control design because it can describe communication structures among local controllers. On an usual representation, a vertex of the graph corresponds to a reservoir of the network, and edges represent the flow of water among the reservoirs of the network. The edges of an usual UDS graph representation, e.g., [14], are oriented in the direction of gravity, i.e., from an upstream tank to a downstream one. For this paper however, the directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{v_1, v_2, \dots, v_N, v_{drain}\}$  and  $\mathcal{E} = \{(v_i, v_j) \in$

$\mathcal{V} \times \mathcal{V} \mid v_j \in \mathcal{S}_i$ }, represents the UDS. Notice that the only difference with the standard representation is the orientation of the edges. Define  $\mathcal{N}_i = \{v_j \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$  as the neighborhood of the  $i$ -th reservoir, i.e., all the reservoirs to where the outflow of the  $i$ -th tank is going to. This neighborhood describes the information that is available to each agent located at each vertex of the graph, i.e., the local controller responsible for the manipulation of the outflow of single reservoir. The reason for the selection of this graph architecture is that, in order to determine its best action, every local controller should know all the states that its local actions are altering, i.e., all the tanks of its neighborhood.

### III. DIFFERENTIAL GAME CONTROLLER

The main goal of the controller for this type of problems is the minimization of overflows of the network during any precipitation event. To this end, the outflows of the VTs must be regulated so that the normalized volumes of the reservoirs, i.e.,  $\bar{v}_i(t) = v_i(t)/v_i^{max}$ , where  $v_i^{max}$  is the maximum volume of the  $i$ -th reservoir, remain below 1 [1]. Given that downstream reservoirs receive run-off from rainfall and many different tanks of the network, their normalized volume is usually greater than upstream reservoirs. Moreover, since upstream tanks only receive water from rainfall, they tend to stay underused, i.e., their normalized volumes remain below 1. Thus, it is a suitable idea to retain water upstream until it can be evacuated safely out of the network, i.e., when there is no risk of flooding at any part of the UDS.

The proposed scheme uses a distributed consensus algorithm based on a DG, in the same spirit as in [13], which seeks agreement on the normalized volume of the tanks. This allows for the retention of the proper amount of water upstream, and guarantees an even use of the reservoirs.

According to [3], dynamic game theory studies multiplayer decision making in situations where not only the actions that players (also known as agents) make are important, but also the order in which they are made. This means that the game is going to evolve over time following the actions that have been made by the agents. Analogously, DG theory studies multiplayer dynamic decision making in situations where the evolution of the game can be described by a set of first-order differential equations. Then, it can be said that DG theory studies the optimal control of dynamical systems that have several independent manipulated inputs. This framework allows for the design of distributed optimal control strategies for dynamical systems with several inputs (both manipulated and non-manipulated) [2].

For the proposed DG, an agent is a local controller that is responsible for the control of one retention gate. This agent can be seen as the dynamical system composed of the  $i$ -th tank and the  $i$ -th retention gate. This agent has available the volume stored in the reservoirs of its neighborhood, e.g., agent  $i$ -th has available the volumes of the tanks in  $\mathcal{N}_i$ . The goal of this agent is to change the outflow of a reservoir in order to achieve an even normalized volume on the tanks of its neighborhood. Since the decisions that a single agent makes have an impact on the game, and thus on other agents,

the standard optimal control tools cannot be applied directly, and DG theory must be used instead.

According to [3], to properly define a DG, it is necessary to define a state equation that describes the evolution of the game, and a set of cost functionals to be optimized by the players. The state equations of the tanks can be written in matrix form as

$$\dot{\mathbf{v}}(t) = \mathbf{A}\mathbf{v}(t) + \sum_{i=1}^N \mathbf{B}_i u_i + \mathbf{d}(t), \quad (1)$$

where  $\mathbf{v} = [v_1, v_2, \dots, v_N, v_{drain}]^T \in \mathbb{R}^{N+1}$ ,  $\mathbf{d} = [d_1, d_2, \dots, d_N, d_{drain}]^T \in \mathbb{R}^{N+1}$ , and  $\mathbf{A}$  and  $\mathbf{B}_i$  are matrices of proper dimensions. Define  $\mathbf{v}$  as the state of the game. Using this formulation, each agent computes one  $u_i$  and seeks the minimization of

$$J_i = \int_0^T \left\{ \sum_{j \in \mathcal{N}_i} w_{ij} (\bar{v}_i(t) - \bar{v}_j(t))^2 + r_i u_i^2(t) \right\} dt,$$

where  $w_{ij} > 0$  and  $r_i > 0$  are weighting factors, and  $T$  is the duration of the game. Therefore, cost function  $J_i$  can be written in compact form as

$$J_i = \int_0^T \left\{ \bar{\mathbf{v}}^T(t) \mathbf{Q}_i \bar{\mathbf{v}}(t) + r_i u_i^2(t) \right\} dt, \quad (2)$$

where  $\mathbf{Q}_i \geq 0$ . Cost function (2) states that the  $i$ -th agent tries to seek an agreement on the normalized states of its neighborhood, while using a minimum amount of energy in the process. The simultaneous minimization of the functionals (2) subject to the state equation (1) describes a linear-quadratic (LQ) differential game (DG) [3].

#### A. Nash equilibrium

The solution to the previous DG requires the simultaneous minimization of cost functionals that are, in general, not the same. Hence, the notion of optimality is not as clear as in a standard optimal control theory, because there is no single criteria for what an optimum is. In traditional game theory, the notion of optimality is augmented into the notion of equilibrium, and thus allowing the search of a solution to the previous problem [3]. There are several different types of equilibria that can be found in a DG. For instance, if one of the agents announces its strategy before hand and every other agent reacts to that doing, the optimal behaviour of the agents is known as a *Stackelberg* equilibrium [3]. For this work however, only the *Nash* equilibrium (NE) is studied. An NE is a set of strategies where no agent can improve its payoff by changing its strategy while others keep theirs fixed [20]. According to [9], a set of actions  $(u_1^*, u_2^*, \dots, u_N^*)$  is an NE for an  $N$ -player game, where each player is trying to minimize  $J_i$ , if for all  $(u_1, u_2, \dots, u_N)$  the inequalities  $J_i(u_1^*, u_2^*, \dots, u_i^*, \dots, u_N^*) \leq J_i(u_1^*, u_2^*, \dots, u_i, \dots, u_N^*)$ ,  $\forall i \in \{1, 2, \dots, N\}$  hold.

Then, it can be said that an NE of the game is a set of admissible strategies where  $u_i^*$  is the best response for the  $i$ -th agent, when every other player is using its equilibrium strategy, i.e., when all players are rational. Since the proposed controller derives into an LQ DG, it makes sense to study the NE within that framework.

The study of DGs requires the knowledge of the information pattern associated to each player. The information pattern is the information that a player is allowed to have throughout the duration of the game. Two information patterns are usually analyzed on DG theory: open-loop information patterns, and feedback information patterns [3]. The difference between these patterns is whether or not an agent is allowed to have the current state of the game. It is important to point out that, although on the open-loop information pattern agents are not able to measure the state vector of the game, they do know what the initial condition is. In this paper, only open-loop information patterns and their associated NE are studied, due to the simplicity of its analysis.

The following theorems have been adapted from [9, Th. 7.1], [9, Th. 7.2], and [9, Col 7.3.], for this particular application, and defining  $\mathbf{S}_i = \mathbf{B}_i r_i^{-1} \mathbf{B}_i^\top$ .

*Theorem 1:* The  $N$ -player DG described by (1) and (2) has an unique open-loop NE for every initial state  $\mathbf{v}(0)$  if and only if  $\det(\mathbf{H}(T)) \neq 0$ , where

$$\mathbf{H}(T) = [\mathbb{I}_{N+1} \quad 0_{N+1} \quad \dots \quad 0_{N+1}] e^{-\mathbf{M}T} \begin{bmatrix} \mathbb{I}_{N+1} \\ 0_{N+1} \\ \vdots \\ 0_{N+1} \end{bmatrix}, \text{ and}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{S}_1 & \mathbf{S}_2 & \dots & \mathbf{S}_N \\ -\mathbf{Q}_1 & -\mathbf{A}^\top & 0_{N+1} & \dots & 0_{N+1} \\ -\mathbf{Q}_2 & 0_{N+1} & -\mathbf{A}^\top & \dots & 0_{N+1} \\ \vdots & & & \ddots & \\ -\mathbf{Q}_N & 0_{N+1} & \dots & 0_{N+1} & -\mathbf{A}^\top \end{bmatrix}.$$

*Theorem 2:* If  $\mathbf{H}(T)$  is invertible, then the set of coupled Riccati equations  $\dot{\mathbf{P}}_i = -\mathbf{A}^\top \mathbf{P}_i - \mathbf{P}_i \mathbf{A} - \mathbf{Q}_i + \sum_{j=1}^N \mathbf{P}_i \mathbf{S}_j \mathbf{P}_j$  with  $\mathbf{P}_i(T) = 0_{N+1}$ , have an unique solution in  $[0, T]$  and the set of strategies  $u_i^*(t) = -r_i^{-1} \mathbf{B}_i^\top \mathbf{P}_i(t) \Phi(t, 0) \mathbf{v}(0)$ , characterizes the NE of the game, where  $\Phi(t, 0) = (\mathbf{A} - \sum_{i=1}^N \mathbf{S}_i \mathbf{P}_i) \Phi(t, 0)$ ,  $\Phi(t, t) = \mathbb{I}_{N+1}$ , is the state transition matrix of the closed-loop system.

Theorems 1 and 2 determine the existence and uniqueness of the solution to the simultaneous minimization of the cost functionals (2) subject to the state equation (1) in terms of a NE, for every initial state of the system. This means that, the solution of the game for the UDS problem depends heavily on parameters that can be selected, e.g.,  $T$  and  $r_i$ . Thus, they can be chosen so that the solution always exists for every  $\mathbf{v}(0)$ . This is particularly useful because the proposed methodology uses an open-loop information pattern to compute the optimal strategies, and a receding horizon approach is needed to give feedback to the solutions [13]. Moreover, since Theorem 2 presents a way to compute the open-loop strategies, they can be calculated easily for recursive approaches, such as the receding horizon approach.

As noted by [13], it might seem that, in order to compute the optimal strategies  $u_i^*$ , every agent requires the whole state

vector  $\mathbf{v}(0)$ . However, since matrix  $\mathbf{Q}_i$  only has non-zero entries at the positions of  $\mathcal{N}_i$  and  $i$  itself, and given that the matrix  $\mathbf{A}$  has a diagonal structure, the solutions  $\mathbf{P}_i$  of the coupled Riccati equations only have non-zero elements at the positions of the  $i$ -th agent and its neighborhood. Therefore,  $u_i^*$  is in fact a distributed control law.

### B. Receding horizon DG

In general, open-loop control strategies are not able to react against those disturbances, i.e.,  $d_i$  in Equation 1, that may alter the state of the system. Hence, the open-loop strategies developed previously are not able to work successfully for the addressed problem, since the control of UDS depends heavily on the disturbances (rainfall) that alter the system. However, a receding horizon scheme [13] can be used to heal that problem and add feedback to the overall law.

The set of strategies  $u_i^*$  are determined at  $t = 0$  for the time interval  $[0, T]$ , which means that the system only knows how it should behave during that period. However, that set of strategies can be recomputed at  $t = t_1$  for time interval  $[t_1, T + t_1]$ , so that the system knows how it should behave at a different time interval. To recompute the strategy, the system has to measure the initial state (which is now  $\mathbf{v}(t_1)$ ) at a new time, hence a feedback appears. This process is repeated for  $\{t_2, t_3, \dots\}$  until any desired final time is reached. This scheme is known as a receding horizon scheme and allows to have feedback on DG with open-loop information patterns.

The algorithm used in this paper is shown in Algorithm 1, where  $t_f$  denotes the final time of the simulation scenario.

**Algorithm 1:** Algorithm used for the receding horizon DG.

```

while  $t \neq t_f$  do
    Measure current state vector  $\mathbf{v}(t_i)$ ;
    Solve the DG  $\forall t \in [t_i, T + t_i]$ ;
    Compute the optimal strategies  $u_i^*$  for  $\mathbf{v}(t_i)$ ;
    Apply  $u_i^*$  during  $[t_i, t_{i+1}]$ ;
     $t_i = t_{i+1}$ 
end

```

Algorithm 1 is based upon discrete time increments, while the analysis for the computation of the optimal strategies is done on continuous time. This implies that during two discrete time instants, the system is applying a continuous time function that evolves over time, and thus, the system is using an open-loop law between two consecutive discrete times. This is different from traditional receding horizon approaches [15], because the control input is able to change in between a given time interval. However, during the computational implementation of the controller, it is required to have a time discretization on the system, and thus, on the optimal strategies. If the selected sampling time for the implementation of the controller is short enough, it can be assumed that the strategies do not change during a time interval. Hence, for the implementation of the system, constant functions are applied during  $[t_i, t_{i+1}]$ .

#### IV. CASE STUDY

The proposed controller is tested with the network shown in Figure 2. This UDS is composed of 4 sub-catchments that drain into a tree-like network that, in turn, converges into a common outlet node. This network gives a convenient representation of how a full-size UDS would look like, because of its strong convergence topology. Moreover, this network allows to study one of the most common problem associated with UDSs, which is the uneven use of the pipes of the system, which leads into poor wastewater management, and in most cases, flooding. Hence, this network is a suitable testbed for determining the performance of controllers of UDSs.

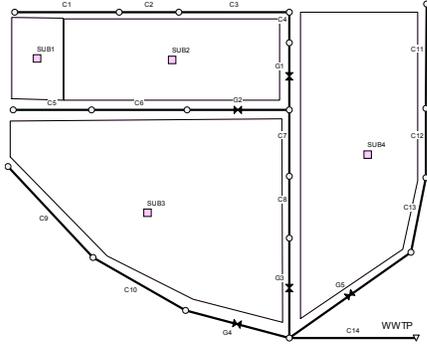


Fig. 2. Proposed network for the testing of the control strategy.

The system is simplified into a set of 6 interconnected VTs using the virtual-reservoir model, where a single tank corresponds to all the pipes in between retention gates, or a retention gate and a inlet or outlet node. Since an information graph is necessary to describe the information available to each agent, and following the definition of Section II, Figure 3 shows the associated graph of the proposed case study.

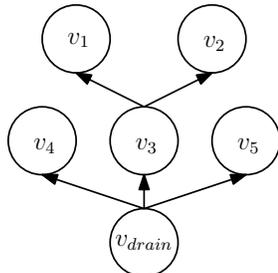


Fig. 3. Information graph that determines the information available to each agent in the case study.

This model requires the calibration of two parameters: the VFC and  $v^{max}$  for each reservoir. Table I presents the parameters associated with the simplified network. They have been obtained using simulated data gathered from MatSWMM [22], which is a co-simulation tool for UDS with *Matlab* and *EPA-SWMM*.

TABLE I  
PARAMETERS OF THE SIMPLIFIED MODEL

Reservoir	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_d$
VFC $\times 10^{-3}$ [s $^{-1}$ ]	0.4	0.8	3.3	0.5	0.8	1.8
$v_i^{max} \times 10^4$ [m $^3$ ]	1.95	0.29	0.36	0.28	0.30	0.30

Due to the fact that the sub-catchments have different geographical locations, they receive different amounts of

rainfall, and thus the total inflow entering each reservoir is different. Figure 4 shows the rain scenario proposed for this application where,  $d_i$  represents the total inflow entering the  $i$ -th reservoir.

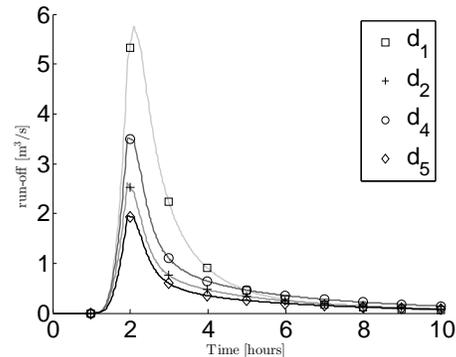


Fig. 4. Proposed precipitation event for the case study.  $d_i$  corresponds to the total inflow entering the  $i$ -th reservoir of the system.

The key idea is to test how the network reacts with the proposed rain scenario, which in fact generates overflow downstream of the UDS. Then, it is going to be determined whether or not the proposed methodology is able to manage the tanks efficiently so that the flooding is minimized and the water resource evenly distributed.

#### V. RESULTS AND DISCUSSION

In order to determine the actual performance of the proposed scheme, the system is tested for three different cases: *i*) the UDS with no controller on the loop, i.e., all gates are left fully open; *ii*) the UDS being controlled via a centralized model predictive control (MPC); and *iii*) the UDS being controlled via a distributed control strategy based on a DG. The performance of the controllers is analyzed upon how efficiently the reservoirs are being used, and what the total flooding is.

The centralized MPC used for comparison minimizes the cost function

$$J_{MPC} = \sum_{k=0}^{H_p-1} \|\bar{\mathbf{v}}(k)\|_{\mathcal{L}}^2 + \|\mathbf{u}(k)\|_{\mathbf{R}}^2,$$

where  $\mathbf{u} = [u_1, u_2, \dots, u_N]^T$ ,  $\mathcal{L}$  is the Laplacian matrix of the graph with edge weights  $w_{ij}$ ,  $\mathbf{R} > 0$  is a diagonal matrix with all  $r_i$  on the diagonal,  $H_p$  is the prediction horizon, and  $k \in \mathbb{Z}_+$  denotes the discrete time. It is important to point out that  $H_p$  has been selected so that the prediction window matches the duration of the DG, i.e.,  $T$ . The reason for the selection of this cost function is to have a consensus-like algorithm in the centralized MPC, so that both controllers, i.e., MPC and DG-based, have the same overall goal.

As for case *i*), Figure 5a shows the open-loop response of the system, i.e., when there are no controllers manipulating the retention gates, for a 10 hours window. It is shown that there is not a proper management of the reservoirs, since they are not evenly used. For instance, some reservoirs, such as 1 and 3, remain underused, while the drain tank presents an overuse of about 50% of its maximum capacity. This is due to the fact that upstream reservoirs only receive water from rainfall, whereas the downstream one receives

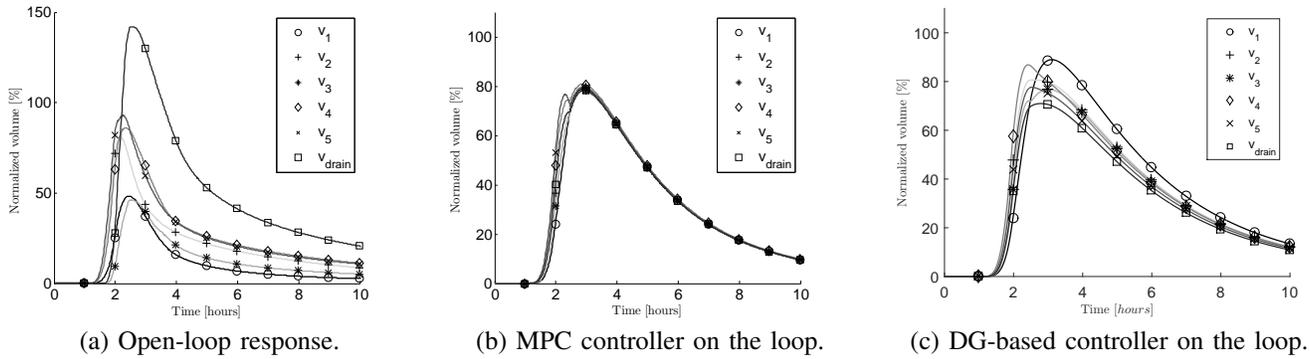


Fig. 5. Simulation results when no controller is used on the system, and when 2 controller are implemented. a) Open-loop response b) Centralized MPC controller; and c) Distributed DG controller. The normalized volumes of the tanks for a 10 hours window is shown.

wastewater from many different tanks which, in term, relieve from rainfall. Therefore, upstream tanks can retain water in order to release some of the burden existing downstream, which leads to a better usage of the existing network.

Cases *ii*) and *iii*) evaluate the performance of the system when two different controllers are used. Both controllers have the same objective: to seek and agreement on the normalized volume of the tanks. The main difference between both schemes is that the MPC uses centralized information in order to calculate the optimal outflows, whereas the DG only uses local information to achieve its goal. Figure 5 shows the evolution of the normalized volumes for a 10 hours window, when the two controllers, i.e., MPC and DG-based, are applied. Both controllers are able to completely remove the overflow found in the open-loop scenario. As for the MPC (Figure 5b), the normalized volumes become almost identical after a short time. This is a sign of proper management of the reservoir, which proves that the control strategy fulfills its goal. The controller based on the DG (Figure 5c) also completely removes the flooding from the network, and is able to manage the VTs so that their normalized volumes move close together.

The proposed scheme achieves a similar performance compared to a centralized controller in terms of flooding minimization and wastewater management. Nonetheless, the MPC requires a lot of extra computational resources, which is a major problem in a large-scale system. For instance, the simulation of the system for a 10 hours time window takes 1 time unit to complete for the DG-based controller within the loop, whereas it takes 27 time units to complete with the MPC controller on the loop. These data have been obtained by using normalized values with respect to the fastest time, from the times collected using Matlab routines. The reason for this normalization is to have a simpler comparison between the measured times. Hence, a methodology such as the one proposed in this work thrives in large-scale problems where having optimization-based controllers is convenient.

## VI. CONCLUSIONS

This paper has proposed a distributed controller based on DG theory, which uses a consensus-like algorithm to achieve an efficient use of the existing network, and to guarantee a minimization of overflows. The proposed controller allows

to have distributed information on local controllers, which leads to less computational burden compared to centralized schemes. Likewise, the proposed scheme allows to consider multiple performance objectives on local controllers, which allows for more flexibility in controller design. The proposed controller has been tested on a typical UDS which is simplified using the virtual-reservoir model. It has been shown that the proposed controller is able to completely remove flooding from the system, and is able to achieve a consensus on the states of the system. Moreover, the proposed controller has been tested with a typical tool for UDS control, i.e., the MPC, and it has been shown that the proposed controller has similar performance in terms of wastewater distribution, but using much lower computational resources.

## REFERENCES

- [1] J. Barreiro-Gomez, 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 European Control Conference (ECC)*, Linz, Austria, 2015.
- [2] T. Başar and P. Bernhard. *H-∞ optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media; Boston, MA, 2008.
- [3] T. Basar and G. Olsder. *Dynamic noncooperative game theory*, volume 200. SIAM; Warrendale, PA, 1995.
- [4] D. Butler and J. Davies. *Urban drainage*. Spon Press; London, 2004.
- [5] A. Cardenas, S. Amin, and S. Sastry. Secure control: Towards survivable cyber-physical systems. In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*, pages 495–500, Beijing, China, 2008.
- [6] G. Cembrano, J. Quevedo, M. Salameiro, V. Puig, J. Figueras, and J. Martí. Optimal control of urban drainage systems. a case study. *Control Engineering Practice*, 12(1):1–9, 2004.
- [7] V. T. Chow. *Open channel hydraulics*. McGraw-Hill Book Company, Inc; New York, 1959.
- [8] P. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.
- [9] J. Engwerda. *LQ dynamic optimization and differential games*. John Wiley & Sons; New York, 2005.
- [10] P. Falugi, P. A. Kountouriotis, and R. B. Vinter. Differential games controllers that confine a system to a safe region in the state space, with applications to surge tank control. *IEEE Transactions on Automatic Control*, 57(11):2778–2788, 2012.
- [11] N. Forouzandehmehr, M. Esmalifalak, H. Mohsenian-Rad, and Z. Han. Autonomous demand response using stochastic differential games. *IEEE Transactions on Smart Grid*, 6(1):291–300, 2015.
- [12] 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*, 2015.

- [13] D. Gu. A differential game approach to formation control. *IEEE Transactions on Control Systems Technology*, 16(1):85–93, 2008.
- [14] B. Joseph-Duran, M. N. Jung, C. Ocampo-Martinez, S. Sager, and G. Cembrano. Minimization of sewage network overflow. *Water Resources Management*, 28(1):41–63, 2014.
- [15] J. M. Maciejowski. *Predictive control: with constraints*. Pearson education; Englewood Cliffs, NJ, 2002.
- [16] J.M. Maestre, D. Muñoz De La Peña, and E.F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153–176, 2011.
- [17] T. Mylvaganam and A. Astolfi. A differential game approach to formation control for a team of agents with one leader. In *Proceedings of the American Control Conference (ACC), 2015*, pages 1469–1474, Chicago, USA, 2015.
- [18] T. Mylvaganam, M. Sassano, and A. Astolfi. A constructive differential game approach to collision avoidance in multi-agent systems. In *Proceedings of the American Control Conference (ACC), 2014*, pages 311–316, Portland, USA, 2014.
- [19] I. V. Nagy, K. Asante-Duah, and I. Zsuffa. *Hydrological dimensioning and operation of reservoirs: practical design concepts and principles*, volume 39. Springer Science & Business Media, 2013.
- [20] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.
- [21] C. Ocampo-Martinez. *Model predictive control of wastewater systems*. Springer Verlag, 2010.
- [22] G. Riaño-Briceño, A. Ramirez-Jaime, J. Barreiro-Gomez, N. Quijano, and C. Ocampo-Martinez. Co-simulation for the design of controllers in urban drainage systems. In *Proceedings of the IEEE Colombian Conference on Automatic Control (CCAC)*, Manizales, Colombia, 2015.
- [23] P. Willems, K. Arnbjerg-Nielsen, J. Olsson, and V. Nguyen. Climate change impact assessment on urban rainfall extremes and urban drainage: methods and shortcomings. *Atmospheric research*, 103:106–118, 2012.