Cooperative Linker for the Distributed Control of the Barcelona Drinking Water Network

Valeria Javalera-Rincon [1][a], Vicenc Puig Cayuela [2][b], Bernardo Morcego Seix[2][c], Fernando Orduña-Cabrera[1][d]

[1] Advanced Systems Analysis and Ecosystem Services and Management programs, International Institute for Applied Systems Analysis, Schlossplatz 1, A-2361, Laxenburg, Austria.

[2]Advanced Control Systems Group, Universitat Politècnica de Catalunya (UPC), Rambla Sant Nebridi, 10, 08222 Terrassa, Spain.

javalera@iiasa.ac.at, {vicenc.puig, bernardo.morcego} @upc.edu, orduna@iiasa.ac.at

Keywords: Multi-Agent Systems, Large Scale Systems, Linkage of Models, Reinforcement Learning, Distributed Control, Water Networks, Large Scale Systems.

Abstract: This work shows how a Linker agent coordinates a cooperative MAS environment to seek a global optimum. The approach is applied to the Barcelona Drinking Water Network (DWN) administrated by AGBAR where the main problem was to coordinate the control of three different sectors of the network. Each part has a local controller (local agent) to solve the local water demands, but it also has to cooperate with the other agents to satisfy the water demands of the whole network. The cooperative Linker agent implemented, learns by using a Reinforcement Learning algorithm, called PlanningByExploration Behaviour with penalization (Javalera et al., 2019), to converge towards an optimal (or suboptimal) value of each of the variables that connect the local agents. For the training and simulation of the Linker agents real historical data of the Barcelona DWN provided by AGBAR were used, as well as the data to model the distributed topology of the DWN. Moreover, some results of the simulations of this approach in contrast with the results of a centralized Model Predictive Controller are depicted.

1 INTRODUCTION

The Barcelona Drinking Water Network (DWN), managed by Aguas de Barcelona, S.A. (AGBAR), supplies drinking water to Barcelona city and the metropolitan area. However, due to the complexity and the computational effort required for its optimal control, AGBAR needs for a distributed control architecture that helps to solve the problem.

The requirement is to break down the whole water network into smaller networks, solve them separately, and then combine their solutions to get a global result for the original task. However, the sub-problems (the smaller networks) are not independent. Some coordination between the partitions of the network is necessary to consider the interrelationships between them. The effort required to deal with these partitions and their coordination can be allocated to various processors, which constitute a distributed computing system. In this way, distributed control is a type of Multi-Agent System. This work presents a realistic application of the LINKER architecture (Javalera 2016) (Javalera et al., 2019) previously called MA-MPC architecture (Javalera et al., 2010).

One of the main problems of distributed control of Large Scale Systems (LSS) is how these dependence relations between sub-systems are preserved. In this

[a] https://orcid.org/0000-0001-8743-9777

[b] https://orcid.org/0000-0002-6364-6429

[c] https://orcid.org/0000-0002-6944-7519

[d] https://orcid.org/0000-0002-8558-0053

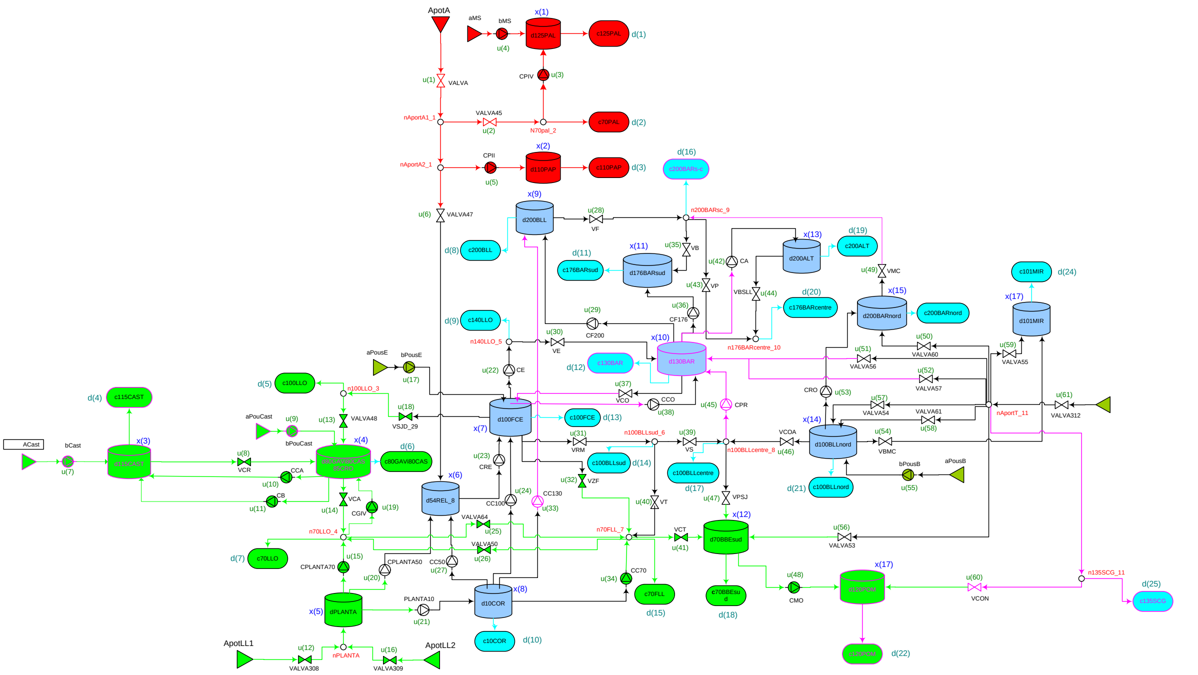

Figure 1: System diagram of the Barcelona DWN aggregate network.

case, these relations are pipes that connect two different control zones of the decentralized water transport network. These connections represent control variables, and the distributed control has to be consistent for both zones, and the optimal value of these variables will have to accomplish a common goal.

The present work addresses the Distributed Control (DC) problem by the application of the Linker Architecture, making use of the LINKER Methodology to implement it.

The structure of the paper is the following: Section 2 introduces the proposed methodology. Section 3 some details of the analysis phase of the proposed methodology are given, while Section 4 presents the design phase. Section 5 shows the results of the experimentation phase applied to the considered case study. Finally, Section 6 summarizes the main conclusions and provide future research paths.

2 METHODOLOGY

A methodology has been developed to accurately define and integrate the LINKER Architecture (Javalera et al., 2010). First attempts to establish this methodology can be found in (Javalera et al., 2010) where a distributed MPC for a hypothetical drinking water network was developed using the proposed framework and compared against a centralized MPC controller.

The LINKER methodology comprises five phases: Analysis, Design, Experimentation, Implementation, and Testing. The description of all the steps of the LINKER methodology and the related processes are described in the next sections of this paper when applied to the Barcelona DWN case study.

3 ANALYSIS

The purpose of the analysis phase is to define the problem and the requirements of the system. It is the basis of all the processes of the LINKER methodology. In the analysis phase, there are five steps to be defined: System description, the definition of control objectives, the definition of functional requirements, definition of restrictions and considerations and definition of the partitioning. The processes are sequential; each process is the basis for the next one. Following the application of the Analysis phase to the Barcelona DWN is introduced.

3.1 A System description

The Barcelona DWN, managed by Aguas de Barcelona, S.A. (AGBAR), not only supplies drinking water to Barcelona city, even more, support the metropolitan area. The sources of water are the Ter and Llobregat rivers, which are regulated at their head by some dams with an overall capacity of 600 cubic hectometers. Currently, there are four drinking water treatment plants (WTP): the Abrera and Sant Joan Despí plants, which extract water from the Llobregat river, the Cardedeu plant, which obtains water from Ter river, and the Besòs plant, which treats the underground flows from the aquifer of the Besòs river. There are also several underground sources (wells) that can provide water through pumping stations. Those different water sources currently offer a flow of around seven m3/s. The water flow from each source is limited and with varying prices of water depending on water treatments and legal extraction canons.

The structure of the Barcelona DWN has two layers; The upper layer, named transport network, this layer aims to links the water treatment plants with the reservoirs distributed all over the city. The lower layer named distribution network, this layer is sectored in subnetworks. Each subnetwork links a tank with each consumer. This application case study aims to work in the transport network. The control system of the transport network is also organized in two layers. The upper layer manages the global control of the network, establishing the set-points of the regulatory controllers at the lower layer. Regulatory controllers are of PID type, while the supervisory layer controller is of MPC type. Regulatory controllers hide the network non-linear behaviour to the supervisory controller. This fact allows the MPC supervisory controller to use a control-oriented linear model.

From the whole drinking water network of Barcelona, described above, this work considers an aggregated version of this model that is an entirely representative version of the full network. Aggregated means that some sectors of the network are collected in a unique part, hence some tanks are raised in a single representative tank and the respective actuators in a single representative pump or valve. This operation has been made to simplify the complexity of the model to have a more manageable but at the same time an essential system, in which the control strategy of this study was applied. AGBAR provided the demands episode of the network.

3.2 Control Objectives

Optimal control in water network deals with the problem of generating flow control strategy from the sources to the consumer areas to satisfy the demand of water while optimizing performance goals such as network safety volumes and flow control stability. Thus, the following operational objectives should be fulfilled by the distributed controllers by order of priority:

Safety term: The satisfaction of water demands should be satisfied at any time instant, this is guaranteed through the equality constraints of the water mass balances at demand sectors. However, some infeasibility avoidance mechanisms should be introduced in the management of the tank volumes such that this volume does not fall below a security amount resulting in demands which cannot be satisfied, this leads to the management of the tank volumes above a specific security volume, which ensures that the network can always supply the demand flows.

Smooth control actions: Pumps and valves should operate smoothly to avoid large transients in the pressurized pipes that can lead to their damage. To obtain such smoothing effect, the MPC controller includes in the objective function a term that penalizes control signal variation $\Delta u(k)$.

Functional requirements: The functional requirements of this system are presented in Table I, the control objectives are reflected in FR3, FR4, and FR8. That means that the priority of the control is to maintain the system inside the security levels, a desirable reference is also considered but the priority is FR3 and FR4. The latter one refers to a smooth control, that means that control actions should increase /decrease in small quantities.

3.3 Restriction and considerations

The safety objective leads to the management of the tank volumes above a specific security volume, which ensures that the network can always supply the demand flows. That is the minimum volume restriction in tanks. A maximum safety level (to avoid spills) should also be applied. Physical limits of valves and pumps should be considered.

3.4 Definition of the partitioning

For this case of study, the Barcelona DWN aggregate network presented in Fig. 1, has been used. From this figure, is clear that the network is comprised of 17 tanks (state variables), 61 actuators (26 pumping stations and 35 valves), 11 nodes and 25 main sectors of water demand (model disturbances). Nodes (of the water network) correspond to the points where water flows are merged or divided within the network. Thus, the nodes represent mass balance relations and are modelled as equality constraints related to inflows and outflows of the nodes.

Table 1: Functional requirements of the Barcelona DWN.

Req No.	Name of the requirement.	Description.
FR1	Type of partitioning.	As defined in Fig 2.
FR2	Distributed control.	One controller for each partition.
FR3	Safety levels.	The tank levels should keep between the defined limits.
FR4	Smooth control.	Control actions should increase / decrease in small quantities.
FR5	Avoid conflicts and collisions.	Avoid conflicts and collisions between sub-systems.
FR6	Satisfy demands.	All demands have the same priority.
FR7 **FR8**	Global optimization Follow a reference	Seek the global optimality of the system. Follow a desirable reference.

Using the partitioning obtained in (Ocampo et al., 2011), the aggregate model of the Barcelona DWN is decomposed in three sub-systems, as depicted in Fig. 1 in different colors. The detailed information about physical parameters and other system values are reported in (Fambrini et al., 2009).

Table 2 collects the resultant dimensions for each sub-system and the corresponding comparison with the dimensions of the vectors of variables for the entire aggregate network.

Sub-system 1: composed by tanks x_i , $i \in \{1, 2\}$, inputs u_j , $j \in \{1, 2, 3, 4, 5\}$, demands d_l , $l \in \{1, 2, 3\}$, and nodes n_q, $q \in \{1, 2\}$. It is represented in Figure 2 with red color.

Sub-system 2: composed by tanks x_i , $i \in \{3, 4, 5, 12, 17\}$, inputs u_j , $j \in \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 25, 26, 32, 34, 40, 41, 47, 48, 56, 60\}$, demands d_l , $l \in \{4, 5, 6, 7, 15, 18, 22\}$, and nodes n_q, $q \in \{3, 4, 7\}$. It is represented in Figure 2 with blue color.

Sub-system 3: composed by tanks x_i , $i \in \{6, 7, 8, 9, 10, 11, 13, 14, 15, 16\}$, inputs u_j , $j \in \{6, 17, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31, 33, 35, 36, 37, 38, 39, 42, 43, 44, 45, 46, 49, 50, 51, 52, 53, 54, 55, 57, 58, 59, 61\}$, demands d_l , $l \in \{8, 9, 10, 11, 12, 13, 14, 16, 17, 19, 20, 21, 23, 24, 25\}$, and nodes n_q, $q \in \{5, 6, 8, 9, 10, 11\}$. It is represented in Figure 2 with green colour.

Table 2: Dimension comparison between the sub-systems and the whole network.

Elements	**Subsyst** (Red)	**Subsyst 2** (Green)	**Subsyst 3** (Blue)	**Whole Model**
Tanks	2	5	10	17
Actuators	5	22	34	61
Demands	3	7	15	25
Nodes	2	3	6	11

As it can be seen, there are inputs u_j that are part of more than one sub-system. In the LINKER control architecture, these are the so-called *shared variables. Shared variables* are control variables that appear in the model of at least two sub-systems in the problem. Their values should be consistent in the sub-systems they appear.

The shared variables in this system (see Figure 1) are: *Sub-system 2–Sub-system 3*: *u18, u20, u21, u32, u34, u40, u47, u56, u60; Sub-system 1–Sub-system 3: u6.*

4 DESIGN

The design phase comprises three processes: definition of the LINKER architecture, the description of the local agents and the meaning of Linker agents. The definition of the LINKER architecture is made first, once defined the architecture, the definition of the local agents and Linker can be made. The whole problem formulation is done in this phase. This problem formulation is based on the information gathered in the *analysis phase*.

Before proceeding with the Design phase, it is important to define what is a local agent and a Linker Agent.

local agent. A local agent (or just an agent) is the entity that is in charge of controlling one specific partition of the system. There is one agent for each system partition (pi). Each agent is arranged to cooperate so that the Linker agent solves the optimization of a common goal through a reinforcement learning algorithm. The cooperative behaviour of local agents is a primary issue in the LINKER Architecture. To behave in such a collaborative way, local agents implement three actions:

1) They provide the data required by the Linker agent.

2) They accept the value(s) provided by the Linker agents of its shared variable(s).

3) They solve the local control problem of its partition, adjusting the value(s) of its shared control

variable(s) in order to coordinate the solution of the negotiation.

Linker agent. A Linker agent is the entity that is in charge of determining the value of one or more *shared variables* between two *local agents*. A Linker agent exists for every pair of local agents that have one or more shared variables in common. Each Linker determines the optimal value of one or more shared variables in the set *V*. Each shared variable is solved seeking a global optimum for both local agents which are agreed to cooperate. The Linker carries out its optimization based on the reinforcements given at each step and on the experience obtained. This experience is stored in a knowledge base.

4.1 Definition of the LINKER architecture

As it was established in Section 3.4, the system is sub-divided in three partitions. This means that three local agents are required for this system. A local agent (named M_1, M_2 and M_3 respectively) was assigned to each partition (sub-system). Figure 2 shows the local agents and the relations between them in the *relation diagram of the system*.

A Linker was placed between the local agents with shared variables between them. Two negotiator agents were required. Figure 3 shows the resulting general structure of the DWN system diagram.
The LINKER Architecture is defined as:

$$\gamma = \{\mathrm{M}, \mathrm{N}, \mathrm{P}, \mathrm{W}, \mathrm{V}, \mathrm{U}, \mathrm{b}\} \qquad (1)$$

where:
M is the set of *local agents*, in this case defined by

$$M = \{M_1, M_2, M_3\} \qquad (2)$$

N is the set of *Linker*, in this case defined by

$$\mathrm{N} = \{\mathrm{n}_1, \mathrm{n}_2\} \qquad (3)$$

P is the set of system partitions in this case defined by

$$\mathrm{P} = \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3\} \qquad (4)$$

Where, in this case each partition of the Barcelona DWN (sub-system) p_i is described by a deterministic linear time-invariant (LTI) model that is expressed in discrete-time as follows

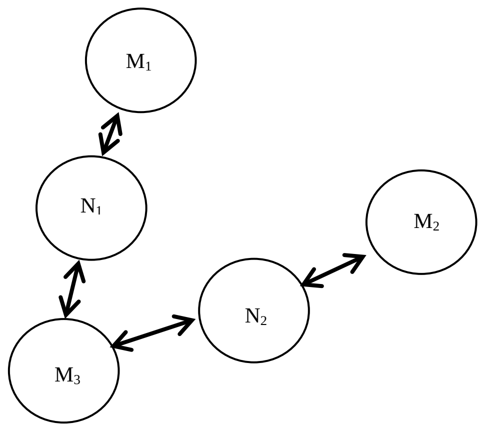

Figure 2: Relation diagram of the Barcelona aggregate DWN.

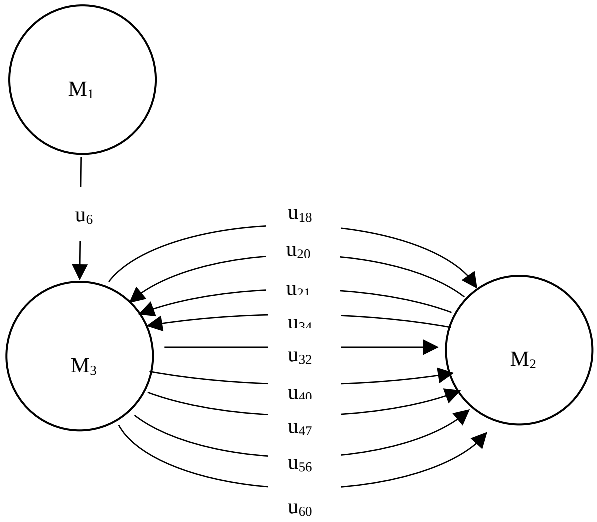

Figure 3: General structure of the Barcelona BWN LINKER implementation.

$$\mathrm{x_i}(\mathrm{k}+1) = \mathrm{A_i}\mathrm{x_i}(\mathrm{k}) + \mathrm{B_{ui}}(\mathrm{k}) + \mathrm{B_{d,i}}\mathrm{d_i}(\mathrm{k}) \qquad (5)$$

$$\mathrm{y_i}(\mathrm{k}) = \mathrm{C_i}\mathrm{x_i}(\mathrm{k}) + \mathrm{D_{u,i}}\mathrm{u_i}(\mathrm{k}) + \mathrm{D_{d,i}}\mathrm{d_i}(\mathrm{k}) \qquad (6)$$

where variables *x, y, u* and *d* are the state, output, input and disturbance vectors of appropriate dimensions, respectively; *A*, *B*, *C* and *D* are the state, input, output and direct matrices, respectively. Sub-indexes *u* and *d* refer to the type of inputs the matrices model, either control inputs or exogenous inputs (disturbances). Control variables are classified as

internal or shared according if they belong only to the sub-system or are shared with other sub-systems.

W represents the set of *nodes* in the system, in this case, there are nodes in all sub-systems, and they have to be taken into account in the model of its respective partition. For now, the set of nodes in the architecture is defined as

$$W=\{w_1,w_2, w_3\} \tag{7}$$

Where w_1, w_2 and w_3 are the sets of *nodes* of sub-system 1, sub-system 2 and sub-system 3 respectively.

V represents the set of *shared variables* described above. In this case V is defined as:

$$V=\{V_1,V_2, V_3\} \tag{8}$$

Where V_1, V_2 and V_3 are the sets of *shared variables* of sub-system 1, sub-system 2 and sub-system 3 respectively.

U is the set of control variables that appear in the model of only one sub-system in the problem, these variables are called *Internal variables*. In this case, the set of internal variables is defined by:

$$U=\{U_1,U_2,U_3\} \tag{9}$$

Where U_1, U_2 and U_3 are the sets of *internal variables* of sub-system 1, sub-system 2 and sub-system 3 respectively.

Finally, *b* represents the *agent platform*, this platform provides the agents with a homogenous medium to communicate and the user with a way to manage agents.

4.2 Definition of local agents

The *local agents* have three main elements: *models, a local controller,* and a *communication module*. Next, these elements will be defined for *local agents* M_1, M_2 and M_3 of the system.

Plant model and *disturbance model* are used in this case to implement the MPC technique of the *local agent*. They are also involved in the learning process as it will be explained later. The model of each agent is described by a deterministic linear time-invariant model expressed in discrete-time defined in Eq. (5) and Eq. (6). A local MPC controller is in charge of the control of each partition P_i, formed by all its internal variables, constraints, objective functions, *Prediction Horizon* (H_p) *(*Interval of finite future time in which the MPC computes the predictive values by using the model in (5) and *Control Horizon* (H_c) *(*Interval of finite future time in which the MPC computes the control values by using the model in (5) and (6)). The *Communication module* is the interface that communicates and synchronizes the local agent with the related Linker agent(s). The models are constructed taking in to consideration the elements of each subsystem described above and their connection in the network of figure 1.

The calculus of states, reward and the prediction horizon H_p are the same for all agents and are defined next.

$$s=\sum_{f=0}^{Hp} J(f)=\sum_{f=0}^{Hp} J_x(f)+\sum_{f=0}^{Hp} J_{\Delta u}(f) \tag{10}$$

where

$$J_x=\vec{e}^{T} w_x \vec{e} \tag{11}$$

$$J_{\Delta u} = \overrightarrow{\Delta u}^{T} w_{\Delta u} \overrightarrow{\Delta u} \tag{12}$$

$$w_{\Delta u} = w_x = 1 \tag{13}$$

$$H_p=24 \tag{14}$$

4.3 Definition of the Linker

The *Linker* applies learning techniques in order to find the optimal (or can we be suboptimal) values of the shared variables of two agents, considering their objectives with the same priority. The system is based on the coordination and cooperation of agents, which share data with the Linker and accept the actions dictated by it. The interaction between the Linker and the agents consists in the following steps: the Linker sends a control action to the agents at each sampling time; the agents set that value as constraint in their respective internal control variables and solve their local problem associated to its partition; agents communicate their new sate to Linker; and the Linker calculates a reward associated to the states. This reward is higher if the actions taken lead to a good state for both agents. The accumulated reward is the experience or the knowledge obtained by the Linker through the training process. The optimization algorithm of the negotiator agent is based on its experience and on maximizing the reinforcements received at every action taken in the past on similar situations.

The Linker agents implements the PlanningbyExploration Behaviour (PBEB), described in depth in (Javalera, 2016) and (Javalera et al, 2019). In the PBEB the agent explores the control action space randomly, assigning large negative rewards to those actions that lead to infeasible states. The exploitation phase is made through the greedy behaviour; see (Javalera et al, 2019) (Javalera, 2016). The internal architecture of a Linker agent comprises the following elements: *Communication module, knowledge base* and *behaviours module*. The *communication module* of the Linker is the analogous of the *communication module* of the local agents. It deals with the interaction between Linker and the related agents involved in the solution of one or more shared variables. A *Q-table* is a tri-dimensional matrix that represents the knowledge related to one particular shared variable. It maintains the *Q-value* gained for each possible pair of states (of the agents related to that shared variable) and an action.

In this way, N_1 is in charge of shared variable u_6 and N_2 is in charge of u_{18}, u_{20}, u_{21}, u_{34}, u_{32}, u_{40}, u_{47}, u_{56} and u_{60}.

5 RESULTS

The objective of PBEB algorithm is to learn by exploration, trying random actions but using just the meaningful experience and penalizing the steps that lead to unfeasible states. A training of PBEB of only 50 iterations using a negative reward of -1000 was applied to obtain the results below. Simulations use same random initial state and reference. The results obtained through the proposed framework are compared with those obtained when a centralized MPC strategy is used. AGBAR has supplied the model parameters and measured disturbances (demands). Demand data correspond to consume of drinking water of the city of Barcelona during the year 2007.

Tank volume evolutions presented in Fig. 4 show that using the LINKER Architecture applying PBEB all tanks remain in the security levels and eight of ten tanks could even follow the desirable reference. That means that agents can solve functional requirements FR3 and FR4 but FR8 (follow a beneficial reference) less accurately than the centralized controller, however, it remains close to the reference.

Table 3: Average $J_{\Delta u}$ of the LINKER and centralized MPC solutions

$J_{\Delta u}$	**M₁**	**M₂**	**M₃**	**Total**
Centralized MPC	4,7837	1,7244	132,4717	138,9798
LINKER	1,4916	0	69,4476	70,9393

Table 3 shows the total of $J_{\Delta u}$ average (the accumulated value of all control actions) of LINKER agents, was almost half (53.55%) of the total average $J_{\Delta u}$ of the centralized MPC solution. That means that The LINKER architecture provides a more economical solution that the centralized MPC. That also represents the improvement in requirement FR4,

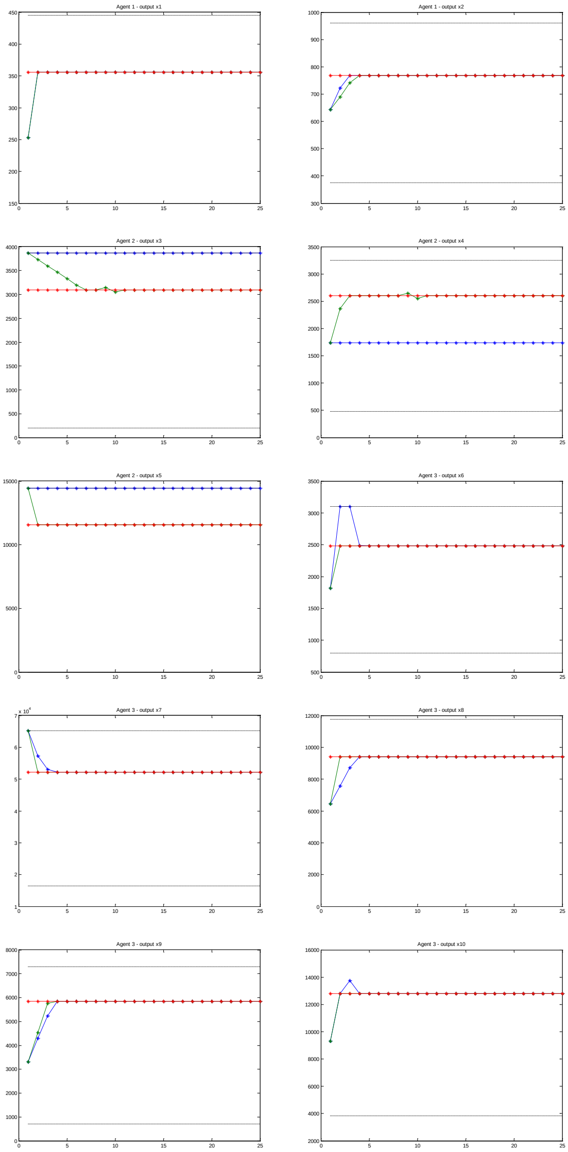

Figure 4: Examples of simulations results of tank volume evolutions. From tank x1 to x10. Blue line represents LINKER solution and green line centralized MPC. Doted lines are min and max volumes of tanks and red line is a desired volume (not mandatory).

smooth control actions, which is essential for the maintenance of the actuators of the water network. Figure 5 compares the actions applied by the LINKER and the centralized MPC during the simulation of figure 4.

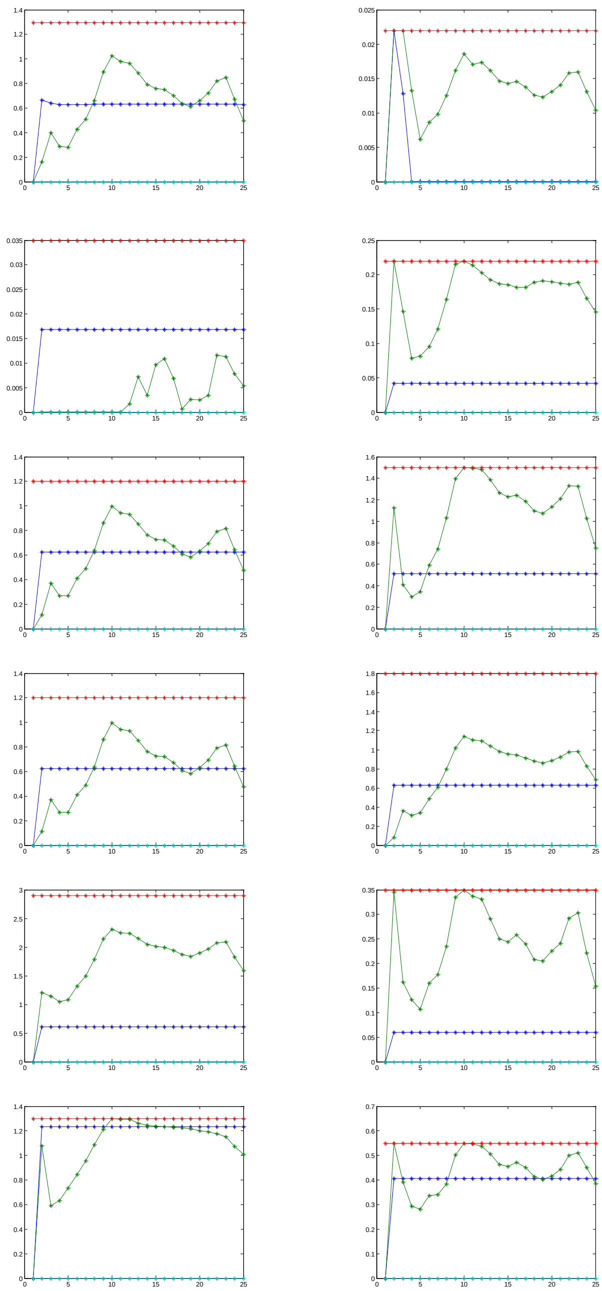

Figure 5: Evolution of some of the control actions applied by The LINKER (blue) and the centralized MPC (green) during simulation of figure 4. Max value (Red) and min value (Cyan)

6 CONCLUSIONS

The implementation of the LINKER Architecture and the PBEB in the case of the Barcelona DWN leads to a good solution where all the states are kept within limits with a cost $J_{\Delta u}$ of almost half (53.55%) of the centralized solution. Ten of seventeen (the 58.8%) tanks of the entire system could even follow the desirable reference (that was not mandatory). That means that the system accomplishes the objectives of keeping within the security levels and maintaining a smooth control better than to track the reference. It seems that with a more balanced partitioning the DWN performance could still improve.

ACKNOWLEDGEMENTS

This work was financed by the National Council of Science and Technology (CONACyT) and PRODEP of Mexico.

REFERENCES

Fambrini and Ocampo-Martinez. (2009). Modelling and decentralized Model Predictive Control of drinking water networks. Technical Report IRI-TR-04-09. Barcelona. Institut de Robòtica i Informàtica Industrial (CSIC-UPC).

Javalera, V., Puig V., Morcego B. and Orduña F. (2019) Reinforcement Learning Approach for Cooperative Control of Multi-Agent Systems. Proceedings of the International Conference of Agents and Artificial Intelligence.

Javalera V. (2016). Distributed Large Scale Systems: A Multi-Agent RL-MPCArchitecture. PhD Thesis, Universitat Politècnica de Catalunya.

Javalera V., Morcego B. and Puig V. (2010) A multi-agent MPC architecture for distributed large scale systems. Valencia Spain. 2nd International Conference on Agents and Artificial Intelligence.

Javalera V., Morcego B. and Puig V. (2010) Distributed MPC for large scale systems using agent based reinforcement learning. Lille France IFAC Symposium Large Scale Systems.

Ocampo-Martinez, Barcelli D., Puig V and Bemporad A. (2011). Hierarchical and decentralised model predictive control of drinking water networks: Application to the barcelona case study. IET Control Theory & Applications.