Optimal Control for Water Distribution Networks with Unknown Dynamics

Jorge Val * **Rafał Wisniewski** * **Carsten S. Kallesøe** *,**

* *Department of Electronic Systems, Automation and Control, Aalborg University, 9210 Aalborg, Denmark. (e-mail: {jvl,raf,csk}@es.aau.dk)*
** *Grundfos A/S, Poul Due Jensens Vej, Bjerringbro, Denmark. (e-mail: ckallesoe@grundfos.com)*

Abstract: Optimal control for Water Distribution Networks (WDN) is subject to complex system models. Typically, detailed models are not available or the implementation is too expensive for small utilities. Reinforcement Learning (RL) methods are well known techniques for model-free control. This paper proposes a model-free controller for WDNs based on RL methods and presents experimental evidence of the practicality of the design.

Keywords: Water Distribution Networks; Level Control; Reinforcement Learning.

1. INTRODUCTION

Water Supply Systems (WSS) are critical infrastructures which deliver water from a source to a number of end-users. These systems consist of the following main parts: water sources, treatment plant and storage, transmission stations and distribution network. The WSS studied in this paper consists of the infrastructure after the water treatment plant, where drinking water is transported long distances through a distribution network to the consumer districts. The system overview is illustrated in Fig. 1.

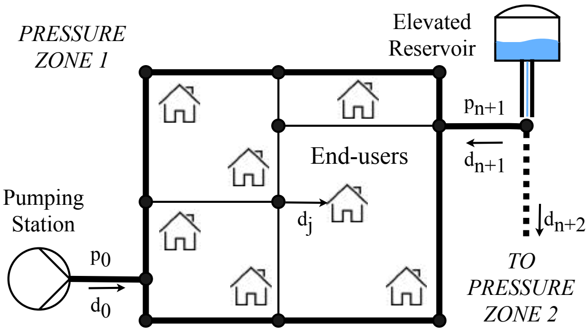

Fig. 1. Illustration of a simplified water distribution network with a pumping station and a storage tank where a city district is supplied through a ring topology network.

The elevated reservoirs (ER) play an important role in a water distribution network. The ER contribute to the pressure regulation of the network, additionally these storage units provide extra water capacity to meet demands in different scenarios such as peak demand periods, service works or emergency situations. Having certain storage capacity combined with proper control strategies, provides the system a suitable framework for energy efficient management as shown in many studies Leirens et al. [2010], Wang et al. [2017], most of them in the Model Predictive Control (MPC) framework.

Efficient management of these infrastructures requires complex control algorithms and detailed network models. This requirement increases the commissioning cost of these controllers and makes these strategies unaffordable for most of small utilities. Therefore, plug & play techniques are proposed to give a control solution which adapts to the network complexity, Kallesøe et al. [2017] Jensen et al. [2018].

Reinforcement learning (RL) is a type of machine learning used in multiple disciplines including control of systems. RL methods are employed to find optimal control policies despite of model uncertainties Sutton and Barto [2018], Bertsekas [2007]. Hence, control RL (model-free) approaches can provide a great advantage when implementing a control solution in large-scale systems. Promising results are presented in Ertin et al. [2001], Castelletti et al. [2002] and Ochoa et al. [2019] using RL methods as hierarchical control strategy for other water systems applications.

When dealing with large-scale continuous systems, the amount of state-action pairs required to map values of the system must be considered. RL techniques where the values are stored can become computationally expensive. Instead, function approximation methods evaluate at every step the state-action pair, leading to a compact representation and efficient use of the data samples Lagoudakis and Parr [2003].

Lewis and Vamvoudakis [2011] and Lewis et al. [2012] present Q-learning algorithms that converge to an optimal controller by using function approximations. These methods find an approximate value function which replaces the complete mapping of the enormous state-action space.

This paper presents an online control solution that uses a Q-Learning algorithm for a system with unknown dynamics. Additionally, this paper presents a novel reformulation of the state space for including an integral control action on the controller response. Part of the RL algorithm is based on the Linear Quadratic Tracking (LQT) controller presented in Kiumarsi et al. [2014]. This approach assumes that a full state feedback is available and the reference signal is given by a linear function. In order to validate that this optimal control solution is able to adapt to different network structures and scenarios, the algorithm is tested in a laboratory testbed which emulates a reference WDN. This reference model is based on a realistic network structure which can be typically found in small utilities like Bjerringbro in Denmark. It consists of a single pumping station, a storage tank and the different consumers are interconnected in a ring topology network. Numerical results are obtained in a simulation of Bjerringbro's WDN. Subsequently, experimental results are obtained at the Smart Water Infrastructure (SWI) laboratory at Aalborg University. This modular testbed allows to replicate real infrastructures in a smaller scale. The laboratory is adapted to qualitatively emulate the particular study case.

The rest of this paper is organised as follows. Section 2 recapitulates LQR formulation using Bellman equation. Section 3 describes the model of the WDN. Section 4 reviews the control algorithm design. Section 5 presents the simulation and experimental results as well as an overview of the testbed used. Section 6 sums up the contributions of the work and relevant ideas for future work.

2. PRELIMINARIES

The work presented in the following section is based on the contribution of Lewis and Vamvoudakis [2011] and Lewis et al. [2012] on optimal control and RL. First, a LQR problem is reformulated with the Bellman function. Then, a Q-learning approach is considered to address a LQR problem without knowledge of the system dynamics. Although the following control approach is considered model-free, the problem structure developed in Section 2.1 is used as reference.

2.1 Bellman function based LQR problem

Consider the following linear discrete-time system in the state-space form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^{n_a}$ are the system states, $u_k \in \mathbb{R}^{m_a}$ are the control inputs, and $y_k \in \mathbb{R}^{p_a}$ are the system outputs and A,B and C are constant matrices with compatible dimensions. The reward function is formulated as a quadratic function of the states as follows

$$V(x_k) = \frac{1}{2}\sum_{i=k}^{\infty} \gamma^{i-k} \rho(x_i, u_i) = \frac{1}{2}\sum_{i=k}^{\infty} \gamma^{i-k} \left[x_i^T Q x_i + u_i^T R u_i\right], \tag{2}$$

where Q>0, R>0 are weights of the cost function $\rho(x, u)$ and $0 < \gamma < 1$ represents a discount factor that reduces the weight of the cost obtained further in the future. Then, the feedback control policy is given by the linear controller

$$u_k = \pi(x_k) = -Kx_k \tag{3}$$

The optimal control policy is found by solving the Linear Quadratic Regulator (LQR) problem by minimising (2) over infinite horizon

$$V^*(x_k) = \frac{1}{2}\min_u \sum_{i=k}^{\infty} \gamma^{i-k} \left[x_i^T Q x_i + u_i^T R u_i\right], \tag{4}$$

using the given state feedback policy u_k, the solution to the Algebraic Riccati Equation (ARE) gives the matrix P such that

$$V^*(x_k) = \frac{1}{2} x_k^T P x_k, \qquad P = P^T > 0 \tag{5}$$

Alternatively, a formulation of this problem can be described by the Bellman equation

$$V(x_k) = \frac{1}{2}\rho_k(x_k, Kx_k) + \gamma V(x_{k+1}), \tag{6}$$

where $V(x_{k+1})$ is the cost of the policy K evaluated at the next time step. This paper uses a similar version of (6), a q-function where the state x_k and control action u_k are explicitly expressed:

$$q(x_k, u_k) = \frac{1}{2}\rho_k(x_k, u_k) + \gamma V(x_{k+1}) \tag{7}$$

By introducing the associated cost function from the LQR problem and (5), the q-function can be expressed as

$$\begin{aligned} q(x_k, u_k) &= \frac{1}{2}(x_k^T Q x_k + u_k^T R u_k) + \gamma x_{k+1}^T P x_{k+1} \\ &= x_k^T Q x_k + u_k^T R u_k + \gamma (Ax_k + Bu_k)^T P (Ax_k + Bu_k) \end{aligned} \tag{8}$$

Then, (8) can be expressed in a matrix form as follows

$$q(x_k, u_k) = \frac{1}{2}\begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \gamma A^T P A + Q & \gamma A^T P B \\ \gamma B^T P A & \gamma B^T P B + R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \tag{9}$$

Rearranging (9) in a compact form yields

$$q(x_k, u_k) = \frac{1}{2}\begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \triangleq \frac{1}{2} z_k^T H z_k \tag{10}$$

where $z(x_k, u_k) = [x_k, u_k]^T$. Subsequently, the optimal control policy is given by

$$u_k^* = \operatorname*{argmin}_u q(x_k, u_k) = -H_{uu}^{-1} H_{ux} x_k \tag{11}$$

This is the optimal control action when the system dynamics is completely known and full state feedback x_k is available.

2.2 Q-learning for LQR

In this section, the system dynamics is unknown. Then, the Bellman optimality principle is applied to formulate the q-function (7) in a recursive form.

First, by introducing the Bellman optimality equation $V_k^*(x_k) = \min_u q_k(x_k, u_k)$ into the q-function (7) leads

$$q_{k+1}(x_k, u_k) = \rho_k(x_k, u_k) + \gamma q_k(x_{k+1}, K^* x_{k+1}), \tag{12}$$

where K^* is the optimal policy. In the future the next state is denoted as $x' = x_{k+1}$.

Then, the q-function expression (12) is rearranged based on the RL Temporal Difference (TD) method for prediction proposed in Sutton and Barto [2018]

$$\begin{aligned} q_{k+1}(x_k, u_k) =& q_k(x_k, u_k) \\ &+ \alpha \left[\rho(x_k, u_k) + \gamma \min_u q_k(x', u) - q_k(x_k, u_k)\right], \end{aligned} \tag{13}$$

where α represents the learning rate. Finally, the expression (13) is reformulated to obtain the update law which gives the q value.

$$\begin{aligned} &q_{k+1}(x_k, u_k) = \\ &\qquad (1 - \alpha) q_k(x_k, u_k) + \alpha \left[\rho(x_k, u_k) + \gamma q_k(x', u')\right] \end{aligned} \tag{14}$$

where u' represents the optimal control action with $u' = \pi^*(x)$.

3. SYSTEM MODEL

A WDN consists of a pipe network with different elements such as valves, pumps and elevated reservoirs. The distribution network is divided into several districts - Pressure Zones (PZ), see Fig. 1. The end-users water consumption (demands) are generally an unknown input or disturbance to the system.

3.1 Network Model

The studied network model is restricted to a ring topology which is a structure typically found in small water utilities. This model can be simplified by unifying the end-users (nodes) that are geographically close because the pressure loss due to pipe resistance is relatively low between them Maschler and Savic [1999]. Fig. 1 shows a standard ring network where the multiple end-user demands are represented by aggregated demands from the main pipes d_j, the controlled inflow from the pumping station is denoted by d_0 and the tank inflow by d_{n+1}. Due to mass conservation in the network, the relation between supply flow d_0, the reservoir flow d_{n+1} and the end-user water consumption d_j can be denoted as

$$d_0 + d_{n+1} = -\sum_{j=1}^{n} d_j, \tag{15}$$

where $d_j \leq 0$ and n is the number of end-user demands. Then, by assuming that the distribution of daily water consumption between the end-users is alike, the demand profile for all the consumers can be described by

$$d_j = \beta_j \overline{d} \qquad \forall j = 1, \ldots, n \tag{16}$$

where β_j is a constant describing the distribution, $\sum_{j=1}^{n} \beta_j = 1$ and $\overline{d}$ is the total district demand in a PZ. The pressure at the reservoir node p_{n+1} is given by the level h in the reservoir and the geodesic level h_0.

$$p_{n+1} = \mu(h + h_0) \tag{17}$$

where μ is a constant scaling the water level and pressure unit and h is the tank level that belongs to an interval restricted by the height of the reservoir. The reservoir level rate depends on the flows leaving the reservoir (d_{n+1} and d_{n+2})

$$A_t \dot{h} = -d_{n+1} - d_{n+2}, \tag{18}$$

where A_t is the constant cross sectional area of the elevated reservoir and the outflow to other PZs d_{n+2}. For simplicity, in the laboratory test this outflow is not further considered.

4. CONTROL

The management of WDNs must ensure the supply of water to the end-users with sufficient pressure head and quality, this task must be performed while considering multiple objectives during the daily operation. Some studies performed in Ocampo-Martinez et al. [2013] state some control objectives: economic, safety, smoothness and water quality.

In this paper only safety is considered in the control strategy, this means that the operational goal is to guarantee the water supply to the end-users. This control task is challenging due to the uncertainty of the water consumption. Therefore, storage tanks must contain enough water to meet future stochastic demands.

4.1 Internal Model Principle

One of the contributions of Kiumarsi et al. [2014] is the solution to the LQT problem and quadratic form of the LQT value function where the problem is formulated as a quadratic form in terms of the system states x and trajectory reference r. In this paper, an additional extension of the state space is proposed for introducing an integral action ξ which rejects the constant disturbances - demands. The augmented system model is built as follows. First, the physical model above (18) is expressed in a state space form for the control design

$$\begin{aligned} \dot{h} &= A_c h + B_c u + W_c d \\ y_c &= C_c h, \end{aligned} \tag{19}$$

where $h \in \mathbb{R}$ represents the tank level, $u \in \mathbb{R}$ the controlled inflow d_0 and $d \in \mathbb{R}$ the end-user demand, with A_c,B_c and C_c constant matrices with compatible dimensions. Then, a reference trajectory r is defined by a linear function

$$\dot{r} = Lr, \tag{20}$$

where $r \in \mathbb{R}$, then defining the integral error

$$\dot{\xi} = y_c - r \tag{21}$$

Equations (19), (20) and (21) are combined to build the following augmented state space model

$$\begin{bmatrix} \dot{h} \\ \dot{r} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} A_c & 0 & 0 \\ 0 & L & 0 \\ C_c & -I & 0 \end{bmatrix} \begin{bmatrix} h \\ r \\ \xi \end{bmatrix} + \begin{bmatrix} B_c \\ 0 \\ 0 \end{bmatrix} [u] + \begin{bmatrix} W_c \\ 0 \\ 0 \end{bmatrix} [d] \tag{22}$$

Finally, expressing the state space representation (22) in a more compact form for discrete time

$$\begin{aligned} x_{k+1} &= A_e x_k + B_e u_k + W_e d_k \\ y_k &= C_e x_k, \end{aligned} \tag{23}$$

where $x = [h,\ r,\ \xi]^T$ is the augmented state vector. A cost (reward) function similar to the previously stated in (2) is built by using the augmented system output from (23)

$$V(x_k) = \frac{1}{2} \sum_{i=k}^{\infty} \gamma^{i-k} \left[y_i^T Q y_i + u_i^T R u_i \right]. \tag{24}$$

By reformulating (24) with $C_e = \begin{bmatrix} C_c & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, the cost function includes the tracking error in terms of x and u.

$$\begin{aligned} V(x_k) &= \frac{1}{2} \sum_{i=k}^{\infty} \gamma^{i-k} [(C_c h_i - r_i)^T Q_1 (C_c h_i - r_i) \\ &+ \xi_i^T Q_2 \xi_i + u_i^T R u_i] \\ &= \frac{1}{2} \sum_{i=k}^{\infty} \gamma^{i-k} (x_i^T Q_e x_i + u_i^T R u_i) = \frac{1}{2} \sum_{i=k}^{\infty} \gamma^{i-k} \rho(x_i, u_i) \end{aligned} \tag{25}$$

with $Q_1 > 0$, $Q_2 > 0$ and R>0 and

$$Q_e = \begin{bmatrix} C_c^T Q_1 C_c & -C_c^T Q_1 & 0 \\ -Q_1 C_c & Q_1 & 0 \\ 0 & 0 & Q_2 \end{bmatrix}. \tag{26}$$

4.2 q-function Approximation using linear architectures

A linear architecture is selected for the approximation over other black-box methods such as Neural Networks. Although the latter methods can provide a more generalised solution, a linear architecture is easier to implement since its behaviour is more transparent, facilitating the troubleshooting task when the algorithm fails.

The q-function proposed in (10) is linearly approximated by a set of Basis Functions (BF) ϕ and the corresponding coordinate vector θ or weights. The BFs are a combination of monomial basis. Thus, learning upon the state vector structure from (10) which is quadratic, a finite set of monomial basis of 2^{nd} degree polynomials, formed with x and u, is chosen as follows. For a multi-index $a \in \mathbb{Z}^{n_a} \geq 0$, with $\mid a \mid= a_1 + \cdots + a_{n_a}$,

$$\hat{q}(x,u) = \sum_{|b|=2} \theta_{(b,0)} x^b + \sum_{|a|=1} \theta_{(a,1)} x^a u + \theta_{(0,2)} u^2. \tag{27}$$

Then, by representing (27) in a vector form

$$\hat{q}(x,u) = \phi^T(x,u)\theta, \tag{28}$$

where ϕ is an n_b-dimensional column vector of BFs and θ is an n_b-dimensional coordinate vector and $n_b = m_a n_a + p_a n_a + m_a$

$$\phi = [x_1^2, x_1 x_2, \ldots, x_{n_a}^2, x_{n_a} u, u^2]^T \tag{29}$$

Subsequently, the approximated control law can be described as $u = \hat{\pi}(\theta, x)$, where $\hat{\pi}(\theta, x)$ can be computed by

$$u'_k = \operatorname*{argmin}_u \hat{q}(x_k, u_k) = \operatorname*{argmin}_u \phi^T(x_k, u_k)\theta \tag{30}$$

This yields to the feedback control policy given by the linear controller

$$u'_k = \hat{K}(\theta) x_k \tag{31}$$

Alternatively, since (27) is quadratic with respect to x and u, a moment matrix $\hat{H}$ can be formed with the coordinates of the BFs such that

$$\hat{q}(x_k, u_k) = z_k^T \hat{H}(\theta) z_k, \tag{32}$$

where $z_k = [x_k, u_k]^T$ and $\hat{H}$ matrix is a symmetric matrix parametrised with the coordinate vector θ as follows $\hat{H} = \begin{bmatrix} \theta_1 & \frac{\theta_2}{2} & \cdots \\ \frac{\theta_2}{2} & \theta_3 & \cdots \\ \vdots & \vdots & \theta_l \end{bmatrix}$ where $\hat{H} \in \mathbb{R}^{n_b(n_b+1)/2}$

Note that q-function (10) and approximated q-function (32) have the same quadratic structure.

4.3 Parameter Update

For the following method, a sample is organised as a tuple of (x_k, u_k, ρ_k, x') and a data batch as a set of collected samples $(\overline{x}_{l_s}, \overline{u}_{l_s}, \overline{\rho}_{l_s}, \overline{x}'_{l_s} \mid s = 1, \ldots, n_l)$ where n_l is the batch size and the index l is the batch iteration number.

The coordinate vector θ is initially unknown, therefore the parameters must be recursively learned. For this, the q-value approximation (28) is introduced into the update law (13)

$$\begin{aligned} \phi^T(x_k, u_k)\theta_{k+1} &= (1-\alpha)\phi^T(x_k, u_k)\theta_k \\ &\quad + \alpha\left[\rho(x_k, u_k) + \gamma \phi^T(x', u')\theta_k\right] \end{aligned} \tag{33}$$

Then, by evaluating (33) recursively, a batch of samples is obtained. The update law for a batch is denoted as

$$\begin{aligned} \Phi_l^T(x,u)\theta_{l+1} &= (1-\alpha)\Phi_l^T(x,u)\theta_l \\ &\quad + \alpha\left[J_l(x,u) + \gamma \Phi_l^T(x', u')\theta_l\right] \end{aligned} \tag{34}$$

where $\Phi \in \mathbb{R}^{n_b \times n_l}$ is a matrix of BFs ϕ, $J \in \mathbb{R}^{n_l}$ is the vector of rewards ρ collected on a batch iteration l. In order to solve the expression (34), a linear Least-Squares Temporal Difference (LSTD) method, similar to Lagoudakis and Parr [2003], is followed to solve the q-function

$$\begin{aligned} \theta_{l+1} &= (1-\alpha)\theta_l \\ &\quad + \alpha G_l^{-1} \Phi_l(x,u)\left[J_l(x,u) + \gamma \Phi_l^T(x', u')\theta_l\right] \end{aligned} \tag{35}$$

Note that a persistent excitation must be added to the control signal such that the term $G_l = \Phi_l \Phi_l^T$ is invertible. The equation (35) is solved by recursively executing the steps described in Algorithm 1.

Algorithm 1 LSTD for Q-function.

```
Input: $\gamma$, $\alpha$, $n_s$,
Approximation mapping of the BFs,
Initialisation: $l \leftarrow 0, x_0, \theta_0$ where $\hat{\pi}(\theta_0)$ must be an admissible policy.
repeat   at every iteration k = 0,1,2, ...
    apply $u_k = K_l x_k$ and measure $x_{k+1}$
    $\Upsilon_{l_s} \leftarrow \rho(x_k, u_k) + \gamma \hat{q}(x_{k+1}, K_l x_{k+1})$
    if $k = (l+1)n_s$ then
        $\theta_{l+1} \leftarrow (1-\alpha)\theta_l + \alpha G_l^{-1} \Phi_l \Upsilon_l$
        $\hat{\pi}(\theta_{l+1}, x) \leftarrow \text{argmin}_u \, \Phi_l^T(x,u)\theta_{l+1}$
        $l \leftarrow l + 1$
    end if
until $\|\theta_{l+1} - \theta_l\| < \epsilon$
```

5. RESULTS

To validate the practicality of the proposed control strategy, algorithm 1 is tested in a computer simulation, then deployed in the Smart Water Laboratory. In this application example the pressure in the network is regulated by controlling the level in the tank. The pressure at the node p_{n+1} is set conservatively enough to meet the flow demands. The weights of the reward function (24) are set to prioritise the minimisation of the tracking error over control action.

The discount factor γ is set close to 1 nearly to the optimal solution, while the learning rate α is sufficiently small such that the old information prevails over new information collected.

5.1 Numerical Results

A simulation environment is developed with the purpose of verifying the proposed control algorithm and training for further implementation. This computer simulation reproduces the water network model from Bjerringbro, a simplified version of the aforementioned network is illustrated in Fig. 1.

As shown in Fig. 2, the tank level has an oscillatory transient where the system dynamics are controlled with a non-optimal policy. Once the learning is considered satisfactory, the persistent excitation on the control action

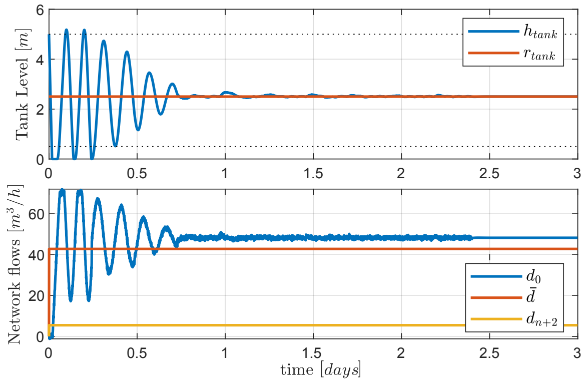

Fig. 2. Simulation Results. Top: Tank Level (blue), reference level(red) Bottom: Controlled input flow (blue), Water Demands (red) (yellow)

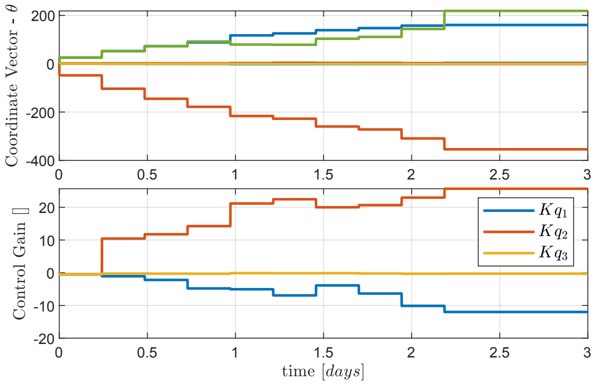

Fig. 3. Simulation Results. Top: Coordinate vector. Bottom: Control Policy

is no longer applied and the tank level stabilises at the reference target despite of the demands $\bar{d}$ and d_{n+2}. This excitation consists of a sum of sines and cosines of different frequencies. In Fig. 3, the coordinate vector parameters θ converge to a satisfactory policy.

5.2 Experimental Results

The testbed scheme consists of a set of Laboratory Units (LU) that can be interconnected to reproduce the desired network. As mentioned earlier, data from Bjerringbro WDN is used to emulate a real water utility. This WDN consists of a single pumping station and storage units, see Fig. 4.

The WDN is built in the laboratory by two aggregated consumers in the City Districts (CD1 and CD2), a pumping station (Pu1), an Elevated Reservoir (ER) and multiple pipe units to reproduce the network structure. A local controller ensuring fast flow control is implemented at Pu1. CD1 and CD2 are equipped with a valve regulating the water consumption. Different geodesic levels h_0 at each critical node (Pu1, ER, CD1, CD2) are simulated by air-pressurising the collecting containers with the equivalent head pressure. The LUs are equipped with multiple sensors and actuators. Each of them has a soft-PLC in charge of the data acquisition, local control and communication. The soft-PLCs at the LUs are interfaced with *CODESYS Control*. Furthermore, the LUs are interconnected to a Central Control Unit (CCU) that can be used for central management of the modules.

The control algorithm 1 for optimal level control is tested in the described laboratory setup. An admissible initial policy is given based on simulation training. As shown in Fig. 5, the tank level is regulated around the reference after some adaptation period. A small error is observed in steady state due to the different accuracy of the flow sensors. Fig. 6 shows the update of the q-function parameters based on the new data, adapting the optimal policy to the new system.

6. DISCUSSION AND FUTURE WORK

The q-learning algorithm succeeded in finding an approximated optimal policy. However, the learning process in a real system is uncertain. This exploration typically leads to saturation of the control actuators and violation of the safety boundaries on the testbed. This factor is a limitation when implementing the controller on systems that have physical boundaries compared with other solutions such as MPC.

The integral action successfully rejects disturbances when the demand profiles are constant. In real scenarios, stochastic disturbances occur, which must be considered in the control design. Due to the real system non-linearity and stochastic disturbances, which are not considered in this control approach, the algorithm does not reach a smooth convergence of the parameters. However, it can be observed that the variation of the controller gains remains to a stable value during the learning, see Fig. 6.

In the future, in order to improve the applicability to a high-dimensional system, this control approach can be improved by considering periodic disturbances in the control design. Moreover, a controller for WDNs must include input and output constraints that set the safe operation boundaries.

7. CONCLUSION

A model-free solution is proposed to regulate the level in the ER in a WDN. This adaptive-optimal control is successfully implemented on a small-scale WDN since the tank level is regulated despite not having the network model. Furthermore, a novel approach is presented, an integral action in the control policy that compensates steady-state constant disturbances. This solution offers an easy-commissioning tool which can reduce the implementation costs.

ACKNOWLEDGEMENTS

We would like to thank *Poul Due Jensen Fond* for funding this project and *Bjerringbro vandforsyning* for providing the network data.

REFERENCES

Dimitri P. Bertsekas. *Dynamic programming and optimal control. Vol. 2.* Athena scientific optimization and computation series. Athena Scientific, 3rd edition, 2007.

Andrea Castelletti, Giorgio Corani, Andrea-Emilio Rizzoli, Rodolfo Soncini-Sessa, and E. Weber. Reinforcement learning in the operational management of a water

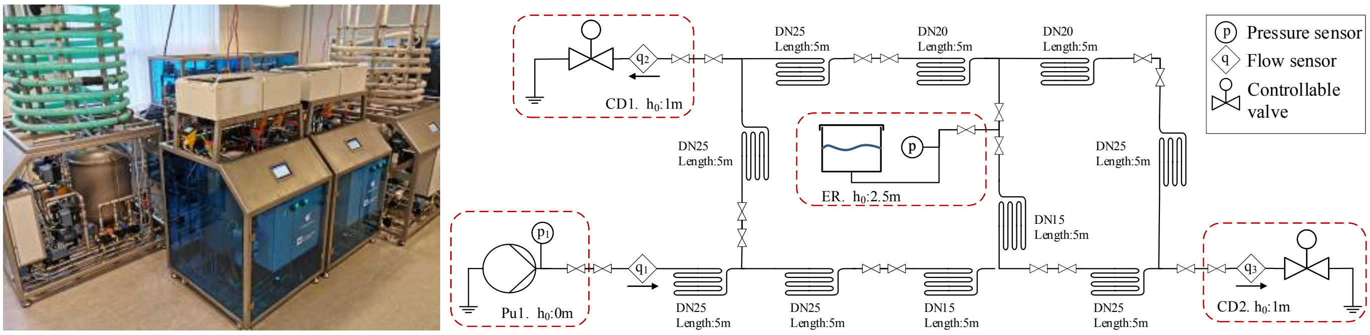

Fig. 4. Left: Photo of the SWI laboratory. Right: Detailed topology of the laboratory setup.

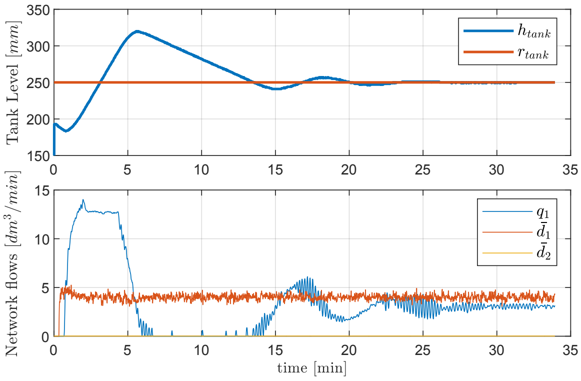

Fig. 5. Experimental Results. Top: Tank Level (blue), reference level(red) Bottom: Controlled input flow (blue) Water Demand (red)

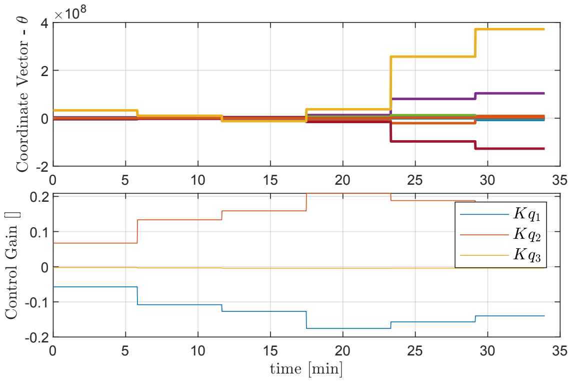

Fig. 6. Experimental Results. Top: Coordinate vector. Bottom: Control Policy

system. In *Modelling and Control in Environmental Issues*. Pergamon Press, 2002.

Emre Ertin, Anthony Dean, Mathew Moore, and Kevin Priddy. Dynamic optimization for optimal control of water distribution systems. *Proceedings of SPIE - The International Society for Optical Engineering*, 2001. doi: 10.1117/12.421163.

Tom Nørgaard Jensen, Carsten Kallesøe, Jan Dimon Bendtsen, and Rafal Wisniewsk. Plug-and-play Commissionable Models for Water Networks with Multiple Inlets*. In *2018 European Control Conference (ECC)*, 2018. doi: 10.23919/ECC.2018.8550092.

Carsten Skovmose Kallesøe, Tom Nørgaard Jensen, and Jan Dimon Bendtsen. Plug-and-Play Model Predictive Control for Water Supply Networks with Storage. *IFAC-PapersOnLine*, 50(1), 2017. doi: 10.1016/j.ifacol.2017.08.616.

Bahare Kiumarsi, Frank L. Lewis, Hamidreza Modares, Ali Karimpour, and Mohammad-Bagher Naghibi-Sistani. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica*, 50(4), 2014. doi: 10.1016/j.automatica.2014.02.015.

Michail G Lagoudakis and Ronald Parr. Least-Squares Policy Iteration. *Journal of Machine Learning Research*, 2003.

S Leirens, C Zamora, R R Negenborn, and B De Schutter. Coordination in urban water supply networks using distributed model predictive control. In *American Control Conference*, 2010. doi: 10.1109/ACC.2010.5530635.

Frank L. Lewis and Kyriakos G. Vamvoudakis. Reinforcement Learning for Partially Observable Dynamic Processes: Adaptive Dynamic Programming Using Measured Output Data. *IEEE Transactions on Systems, Man, and Cybernetics*, 41(1), 2011. doi: 10.1109/TSMCB.2010.2043839.

Frank L. Lewis, Draguna Vrabie, and Kyriakos G. Vamvoudakis. Reinforcement Learning and Feedback Control: Using Natural Decision Methods to Design Optimal Adaptive Controllers. *IEEE Control Systems Magazine*, 32(6), 2012. doi: 10.1109/MCS.2012.2214134.

Tobias Maschler and Dragan A Savic. Simplification of water supply network models through linearisation. Technical Report 99/01, University of Exeter, 1999.

Carlos Ocampo-Martinez, Vicenç Puig, Gabriela Cembrano, and Joseba Quevedo. Application of Predictive Control Strategies to the Management of Complex Networks in the Urban Water Cycle. *Control Systems, IEEE*, 2013. doi: 10.1109/MCS.2012.2225919.

Daniel Ochoa, Gerardo Riaño-Briceño, Nicanor Quijano, and Carlos Ocampo-Martinez. Control of Urban Drainage Systems: Optimal Flow Control and Deep Learning in Action. In *American Control Conference (ACC)*, 2019. doi: 10.23919/ACC.2019.8814958.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, 2nd edition, 2018.

Ye Wang, Vicenç Puig, and Gabriela Cembrano. Nonlinear economic model predictive control of water distribution networks. *Journal of Process Control*, 56, 2017. doi: 10.1016/j.jprocont.2017.05.004.