Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea magistrale in
Ingegneria dell'automazione

Modeling and control of a multiple-pool channel

Relatore:
Prof. Angelo Cenedese

Correlatore:
Dr. Marco Fabris

Laureanda:
Eleonora Bordin

11 aprile 2022

Anno Accademico 2021-2022

Abstract

Questa tesi tratta l'analisi di un sottoinsieme della rete di canali del Cavallino, situata lungo la costa Veneziana che si estende da Punta Sabbioni al porto di Piave Vecchia. In particolare, questo lavoro si focalizza su modellizzazione, stima e controllo di una sequenza di canali, assumendo che le misure relative al livello della superficie dell'acqua e la posizione delle strutture di controllo siano disponibili. Una procedura basata su identificazione dei sistemi consente di analizzare e selezionare il modello grey box migliore tra quelli proposti tra ARX e OE, con l'obiettivo di stimare l'andamento del livello dell'acqua. L'obiettivo del controllo è regolare il livello dell'acqua della rete agendo sulle strutture idriche di controllo. Di conseguenza, viene proposto un tipo di controllo decentralizzato. In particolare, viene sfruttato il controllo multivariabile del livello dell'acqua a monte della struttura di controllo, tenendo in considerazione sia chiuse in superficie che sotto il livello dell'acqua. La soluzione proposta è implementata nell'ambiente Matlab e Simulink e si basa su controllori PI aumentati con un filtro passabasso, al fine di controllare il livello dell'acqua in caso di perturbazione di una sequenza di canali consecutivi.

Abstract

This project deals with the analysis of a subset of the water channel network of Cavallino, that is the section of the Venetian coast that extends from Punta Sabbioni to Piave Vecchia harbor. In particular, this work focuses on modeling, estimation and control design of a multiple channel pools system, assuming that water level measurements and control structure position are available. A system-identification-based procedure is considered for the analysis and selection of an ARX and an OE grey box model, to estimate and control purposes. The control aim is to regulate the water level of the channel network by acting on the hydraulic structure position. Then, a decentralized control is implemented. In particular, a multivariable local upstream control strategy is exploited, involving a model that takes into account both weirs and gates hydraulic control structures. Lastly, a solution is implemented in MATLAB and Simulink, based on PI controllers augmented with lowpass filters, in order to control the water level of multiple pools connected in series.

Contents

List of Figures

List of Tables

1

Introduction

The goal of this work is to develop and implement modeling, estimation and control design for open-channel systems. In the last years, storms, downpours and floods have become increasingly frequent and devastating occurrences. These events are a clear consequence of climate change; indeed extraordinary weather conditions will become more and more frequent, compromising the agricultural sector and even infrastructures. Moreover, sudden and violent downpours may represent a huge inconvenience for the population and may cause devastating damages to cities. Water distribution networks are complicated systems that present nonlinear dynamics; therefore they need suitable modeling and control strategies to obtain a wise management of water resources. The main issue is that the dynamics of water channels is modeled by complex nonlinear partial differential equations. Moreover, there are multiple inputs and outputs systems that may be controlled and these systems are subjected to disturbances, due to water withdrawals or weather perturbations. In addition, the dynamics of the water flow deals with delays between a control action and its effect. Progress in systems theory yields tools to cope with the study and design of complex hydraulic system. The automatization of open-channel systems represents a solution to smart water delivery. To this purpose, this thesis is developed starting from irrigation open-channel systems analysis, aiming at the improvement of water delivery systems. In particular, the Cavallino water channel network (see Figure 1.1) is analyzed.

1.1 State of the art

In this section, it is introduced the state of the art pertaining to the topics related to modeling and control of multiple channel pools. A brief introduction and the knowledge of literature about the open-channel irrigation systems is fundamental to delineate and tackle the water delivery problem that affects the Cavallino water channel network, shown in Figure 1.1.

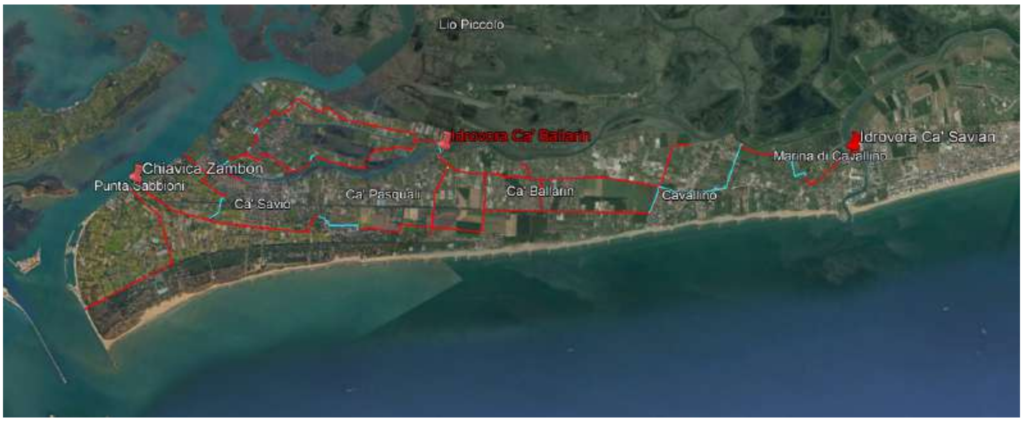


Figure 1.1: Cavallino water channel network

1.1.1 Open-channel irrigation systems modeling

The physical model of a water channel based on the laws of conservation of momentum and mass of fluid can be accurately described by the Saint Venant equations (SVEs) that well characterize the unsteady water motion in free channels, under suitable assumptions as reported in Chapter 2. Open-channel irrigation models can be distinguished as model that are derived by means of analytical simplification and models that result from the dynamics approximation.

Models obtained from simplifications of the SVEs

Open-channel irrigation systems are developed to supply a suitable amount of water request without wastage. Authors in [1] introduce an explicit spatial discretization of the SVEs. Every channel can be split into sections and it is possible to define one differential equation for the water level and another one for the water flow. On the other hand, stability relies on the discretization step size choice. In [2] an implicit Preissman finite-difference scheme is exploited, such that the stability of the model does not depend on the step size. The approach proposed in [3] uses Preissmann finite-difference scheme to implement a control algorithm based on inverse solution of the nonlinear SVEs. In [4] the proposed model comes from the linearization of the SVEs. Moreover, approximating the frequency response, it can be determined a transfer function between the flow rate variations at the upper end and the flow rate variations at the lower end of a channel. Finally, an analytical solution for the water level and the flow is obtained. The approach in [5] takes into account an approximated model called integrator delay zero model that is a simple and common technique to model a canal for control purposes. It consists of an integrator and a delay in low frequencies, and it models the high frequencies by a constant gain and a delay. Most of the approximated modeling schemes require operational information of the system. Moreover, in case of finite-difference strategies, it is worth noting that complex models with high order are achieved. These aspects may represent inconveniences in control system design.

Approximated models

Approximated models like integrator delay model (ID), grey-box models and black-box models have been elaborated from feasible assumptions, by means of physical laws and empirical information. Approximated models represent an important alternative to SVEs with the purpose of design control-oriented models for open-channel systems. In [6] it is presented the integrator delay model, influenced by [4] and it is proposed backwater profile, that is a phenomenon representing the downstream accumulation of water. Thus, the channel can be split into two sections that correspond to a uniform flow and to a reservoir respectively. It can be assumed that the water depth in correspondence of the uniform section depends on the flow while the backwater section can be modeled as a mass balance with an inflow delay. Such model is observed in many studies with control purpose such as [7], [8], [9]. Available measured data represent an important source in order to design control-oriented models for open-channel systems. System identification is a procedure to model the dynamic behavior of a system or a system component based on measured data. Then, system identification can be employed to design models without physical information of the system (black-box models), or models that require physical information of the system (grey-box models) [8]. The black box method explains the relationship between measured inputs and measured outputs when parameters are changed. The grey box method is a configuration of the model where the parameters are determined by means of exact physical principles. There exists many parametric model structures to assist in modeling an unknown system. The most common parametric model structures are a subset of general linear models, such as output-error (OE), autoregressive exogenous (ARX), autoregressive moving average with exogenous inputs (ARMAX), Box–Jenkins (BJ) structures. Differently from ARX model, the ARMAX model structure includes disturb dynamics. In addiction, ARMAX models are useful if you have to control a disturb that enters early in the process, such as at the input. According to results of experiments showed in [10], the model structure that best represents a channel dynamics is the ARMAX structure. In [11] it is introduced a simplified model to control open water channels that are short, flat and deep, which are supposed to be dominated by resonance behavior. The integrator resonance model (IR model) describes resonance-sensitive channels; it includes an integrator and the first resonance mode of a long reflecting wave. Then, the controller avoids triggering the resonance mode as much as possible. In particular, in such article it is also proved that neglecting the resonance behavior in the controller design can cause poor performance of the closed loop behavior. The method proposed in [12] introduces three data-driven modeling tools to represent channel dynamics, which are artificial neural networks (NARX models), local linear models and fuzzy systems. As an experimental result, all models manage to capture the significant dynamics but the neural networks perform slightly better than the other two strategies. For what concerns grey-box models, [13] introduces a control-oriented model based on a simplified mass balance, assuming that the water volume in the channel is proportional to the water level and assuming a time delay in the channel inflow. Such model consists of a differential equation that explains the water mass balance, in which the nonlinear flow relation of the control structures is included. Grey-box models have been employed for control purposes in [14], [15], [16] and leak detection in [17]. Moreover, in [13] it is shown that nonlinear models are more precise than linear models. In particular, third order nonlinear models are able to exhibit wave dynamics, while first order nonlinear models are simpler and more suitable for control purpose. These grey-box models have been restricted to systems with weirs structure in free-flow, where the flow depends on the control structure upstream depth. In conclusion, in [18] it is proposed a grey-box model in a system that takes into account weirs and gates structures but this analysis is restricted to a single channel.

1.1.2 Control of Open-channel irrigation systems

The main control target in open-channel irrigation systems is to discharge a suitable amount of water in order to satisfy the demand. Then, in a well-operated system, the outer water needs to be almost equal to the water disturbance. Open-channel systems are complex systems characterized by long delays, high channel interactions and perturbations. There exist multiple control methods that are characterized by different configurations, strategies, architectures, cost functions, control variables that can be mentioned [19]. Figure 1.2 shows a brief description of the main control architectures, strategies, configuration, control variable and objectives, related to open-channel irrigation systems that have been considered in the next sections.

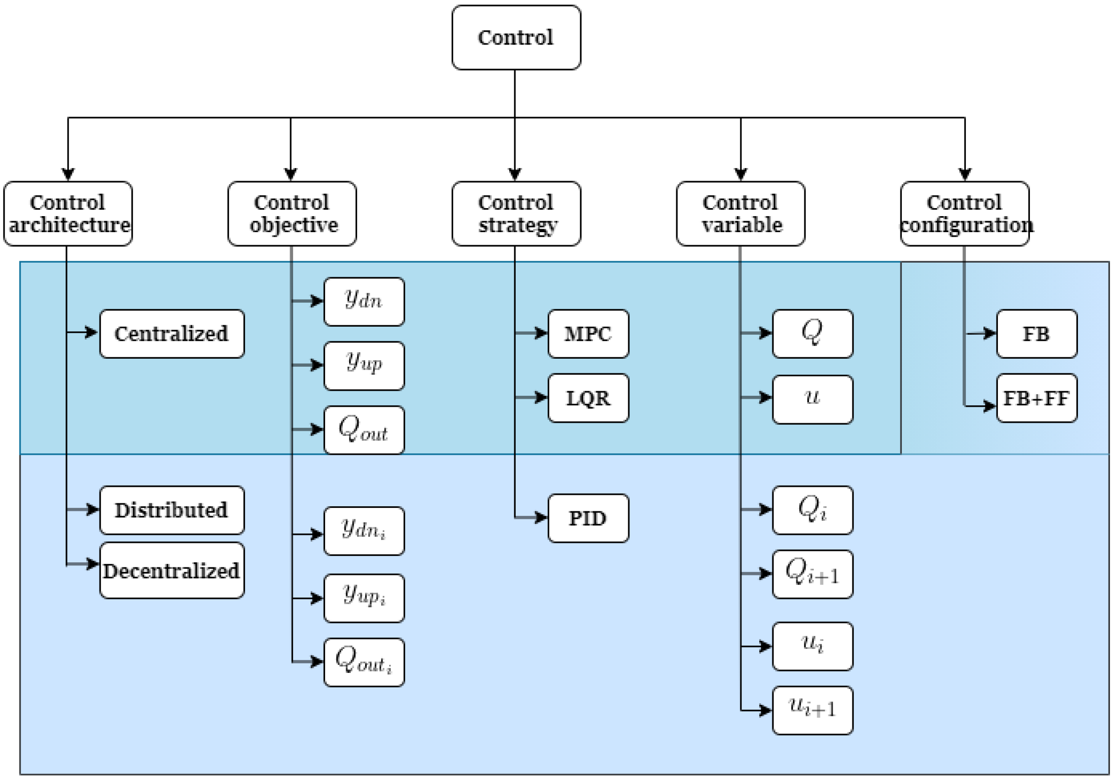


Figure 1.2: Proposed classification of control approaches for open-channel irrigation systems

Control architectures

In general, there exist three main different control architectures that characterize open-channel irrigation systems that are centralized, decentralized and distributed, as shown in figure 1.2. Centralized architectures allow for a central controller to generate a control input, exploiting the measurements, collected by the system as proposed in [20], [21], [22]. These systems permit supervision and remote control of the whole system. On the other hand, they are complicated and more sensitive to a possible hardware crash than localized configurations as reported in [19]. Decentralized architectures [23], [24], show that only local upstream or downstream data of a channel is employed to elaborate the control strategy. Distributed architectures reported in [25], show that the control system exploits both local and adjacent information determining cooperation among local controllers. In general, centralized architectures achieve better results than decentralized and distributed architectures in open-channel irrigation systems. The main advantage of a decentralized or distributed control system is that control is guaranteed even if a piece of information is lost.

Control objectives

The main control target in open-channel irrigation system is to deliver water to the farmers minimizing the wastage. According to the fact that the offtakes are gravity fed, the requirement of being able to deliver water can be translated into a reference regulation of the water levels. Then, the discharges are regulated to each user as reported in [14]. Moreover, water is usually wasted when it crosses the last weir of the channel because it is not possible to recover unused water. Thus, the flow over the last weir should be almost null. It is worth noting that the water level reference changes with operational conditions, but these changes are not frequent. As a result, the perturbation rejection can be considered more relevant than tracking reference changes. It is also fundamental to observe that large weir or gates motions can cause large oscillations, hence gate movements performed in the frequency of the dominant waves need to be avoided. Upstream and downstream controls are frequent when dealing with decentralized and distributed control architectures. Upstream controls consist on keeping a setpoint water level upstream with respect to the regulator, as shown [26], [27], [19], while downstream controls consist on maintaining a setpoint water level downstream with respect to the regulator as shown in [19]. Upstream and downstream controls can be distinguished as close or distant with respect to the regulator. On the contrary, distant upstream control is not common. In particular, in [28] it is estabilished that the water setpoint along the channel is not satisfied. In [29] it is reported a comparison between close upstream and distant upstream control strategies with an adaptive controller. In the first configuration, the experimental results show a good behavior while, the second one exhibit oscillations and poor performances. The adjustment of the upstream level close to the regulator is the most frequent control configuration in open-channel irrigation systems, as shown in [26]. Flow control is required at the inflow of the system, where the inflow is determined according to the user request. Controllers may be multivariable and the controlled variables could be different, such as the upstream water depth denoted with y_{up} [30] , [27], the downstream depth h y_{dn} [31], [22] and the channel inflow Q_{in} or outflow Q_{out} [32] as reported in figure 1.2.

Feedback and feedforward control configurations

In open-channel irrigation systems control, it is possible to select feedback (FB) configuration, feedforward (FF) configuration or a combination of them($FB + FF$), as shown in figure 1.2. In the FB configurations, the channel inflow or outflow is modified to reduce the error between the controlled variable and a set-point water level or flow. In the FF configurations, the channel inflow or outflow is modified taking into account previous information about water request. The FB configurations suggested in [30], [27], [24] show that the rejection of disturbances and uncertainties such as leaks, an unexpected water request, weather perturbations, can be achieved. On the other hand, controller design need to be accurate , avoiding oscillations or instability. The FF configurations allows the presence of less oscillations and a faster response but perturbation rejection is not possible as reported in [33]. The best control performance can be reached by the combination of these configurations ($FB+FF$), resulting in faster responses and disturbances rejection as shown in [24]. Multiple control strategies have been analyzed and reported in the sequel.

PID control strategy

$Proportional - Integral - Derivative$ controllers (PID) are widely employed in control systems industry . There exist multiple studies in literature that exploit PID regulators to keep a fixed reference in open-channel irrigation systems. In [34] it is proposed a routine for tuning upstream PI controllers. In [35] it is analyzed the advantage between design a Proportional-Integral (PI) controller to keep a desired upstream setpoint or a desired downstream setpoint. Moreover, in [36] it is introduced a PI controller with a first order filter in order to filter resonant oscillations produced by neighbor channels. In [37] there is a comparison between the results obtained by a downstream PI controller and by a distant downstream

PI controller. In [38] and [38] PI tuning routines designed exploiting the integrator delay model are suggested.

LQR control strategy

One alternative control strategy for open-channel irrigation systems is focused on optimal control. This approach requires the minimization of a quadratic objective function formulated from the state space system. Given a linear system, a LQR is a vectorial control law obtained by the minimization of a quadratic cost index. The latter quantity is formalized as a trade-off-based penalization between weighted state and weighted control input of the underlying system. The numerical expression of such a controller is computed by solving the Riccati equation associated to the cost index and the system under analysis. Moreover, this approach is useful for controlling systems characterized by a multiple number of states [1], [6] . On the other hand, LQRs satisfy a required behavior in an area close to an operation point.

Model predictive control strategy

The advantage that model predictive control (MPC) presents is relevant for what concern optimality and prediction. There exist multiple studies in literature that focus on MPCs strategies, such as [31], [36], [30], [25], [22], [21]. MPC strategies take into account a prediction model, some constraints, an objective function and an optimization algorithm. The MPC controller has to be computed by solving an optimization problem over the time horizon. It is required a discrete-time model in state space configuration or transfer function. Boundary conditions of the controlled system are included into the constraints for inputs and state variables while the objective function corresponds to the performance index combination of the prediction model and constraints. In conclusion, the optimization procedure finds the optimal solution over a prediction time horizon that minimizes a given objective function.

1.2 Thesis statement, contributions and structures

In this section, the problem formulation is introduced and a solution approach is proposed. Then, the main contribution is compared with the current open-channel irrigation systems and the outline of the thesis is presented.

1.2.1 Problem formulation

The Cavallino peninsula separates the north Venetian lagoon from the Adriatic sea and it is characterized by a complex channel network. In high tide phase, sea water tends to retreat towards rivers along a section. For this reason, there exist water regulators along the main channels which are controlled manually, in order to avoid possible floods. However, in case of downpours or strong perturbations the risk of floods is high. It is thus fundamental to design an automatic control system in order to regulate the water level of the string of pools and to avoid floods. An accurate model for the behavior of the main Cavallino channels need to be designed and an efficient automatic control system is required to adjust and correct the errors without external effort.

1.2.2 Solution approach

The SVEs are complex to handle for modeling, estimate and control purposes. Moreover, geometrical parameters related to the channels are fundamental and sometimes these are missing or not accurate. Then, starting from the water volume mass balance equation, a simpler model is proposed. In particular, a system-identification-based procedure is exploited for estimating the parameters for two grey-box models. In particular, the ARX and the OE model are taken into account. According to the results, the best model among them is the OE model and then a one-step-ahead predictor is validated. In such a case, due to the lack of measurements, a second order model is implemented to generate the input-output data. For what concerns the control design, local upstream control is suggested, with the aim to discharge water avoiding floods. A decentralized approach is preferred because of its main advantages, such as scalability, versatility and robustness to local failures. A simulation can be implemented in Matlab, focusing on a subset of the Cavallino water network, as shown in Figure 1.3.

1.2.3 Contribution

There exists a large literature about the open-channel irrigation network, as reported in Section 1.1.2. Then, to delineate and tackle the water delivery problem that affects the Cavallino water channel network, we draw inspiration from this

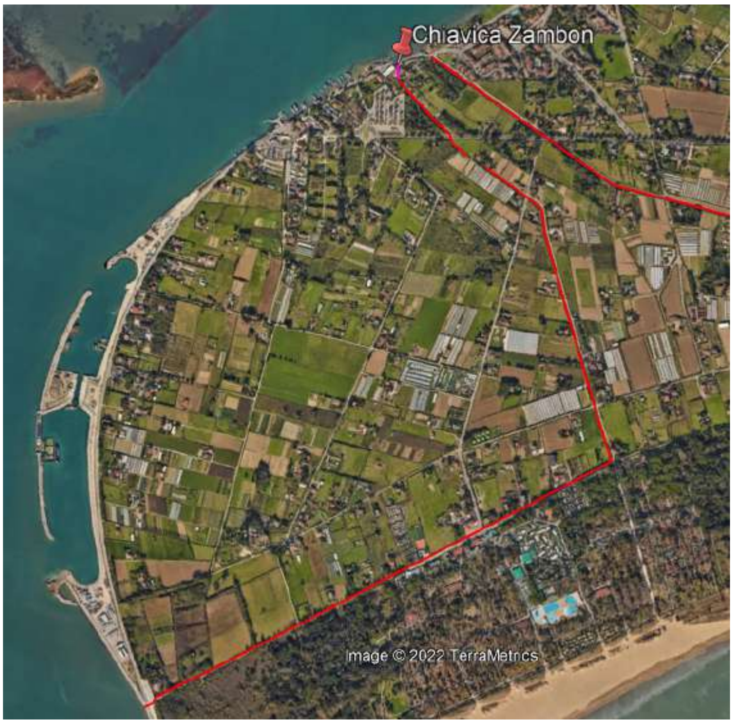


Figure 1.3: Subset of the Cavallino water network

field. Our approach is similar to [13] for what concerns the procedure to identify discrete time model parameters. In addition, we analyze and compare the performances of an ARX model and an OE model for estimation purpose. In particular, we simulate some measurements by means of a discrete time second order model to proceed with the system identification approach. The parametric model estimation involves linear regression in one case and it requires optimization tools in the another one. In particular, the Levenberg-Marquardt algorithm is preferred and then implemented. For what concerns control of the water level, the decentralized control design draws inspiration from [39]. The main contribution provided in this thesis is represented by the development of a local upstream control strategy for a model, that takes into account both weirs and gates hydraulic control structures. A solution based on PI controllers is implemented in Matlab and Simulink, in order to control the water depth of multiple pools connected in series.

1.2.4 Outline

The remainder of this thesis is organized as follows. Chapter 2 introduces the theoretical hydraulics fundamental in order to enhance the basic concept on open channels. Moreover, SVEs are analyzed and their complexity is highlighted. Their implementation is explained in details in the Appendix 6.2. Chapter 3 considers two grey-box models in discrete time, that take into account both weirs and gates hydraulic control structures and a system identification procedure to estimate their parameters is presented. Then, a one-step ahead predictor is proposed. Chapter 4 deals with the main contribution of this thesis, focusing on the development of a decentralized control. Local upstream control is introduced and both feedback and feedforward configurations are analyzed. In conclusion, Chapter 5 shows the numerical simulation implemented in MATLAB and Simulink. A Second order system is implemented in order to simulate the required measurements. Then, system identification procedure developed in Chapter 3 allows to select a suitable model for an estimation purpose, considering the trade-off between complexity and adherence of data. Finally, a procedure for tuning the employed PIL is proposed and a decentralized control is implemented. The results are hence discussed, so that the control performances with and without feedforward are compared and conclusions are reported in Chapter 6.

2

Basic concepts in hydrosystems

In this chapter, basic concepts in hydrosystems are explained and some fundamental definitions are enunciated. The physical model for the water channel is introduced. Unidirectional open channel flow is usually modeled by two hyperbolic equations expressing the conservation of mass and momentum, that are known as the Saint Venant Equations (SVEs).

2.1 Open channel flow classification

Open channel systems are structures characterized by a open top that allow fluids transport from a location to another. An open channel flow presents a free surface, which is subject to atmospheric pressure. Flows can be classified into steady and unsteady. We refer to a steady flow if the flow velocity at a given location does not vary with respect to time. On the contrary, if the local acceleration is different from zero we refer to an unsteady flow. Moreover, flows can be distinguished as uniform or varied. They are referred as uniform if the flow velocity at a given time does not change with respect to a given distance otherwise they are called varied or nonuniform. On the other hand, it is possible to relax this condition, considering a flow uniform as long as the velocity in the direction of the flow is constant along the channel. The rate of variation with respect to distance allows to classify nonuniform flows into gradually varied and rapidly varied flows. As shown in Figure 2.1, a flow is considered critical if the flow velocity is equal to the velocity of a wave characterized by a small amplitude. A gravity wave can be generated by a variation in the flow depth. Then, a flow is defined as subcritical if the flow velocity is lower than the critical velocity while can be defined supercritical if the flow velocity is greater than such value.

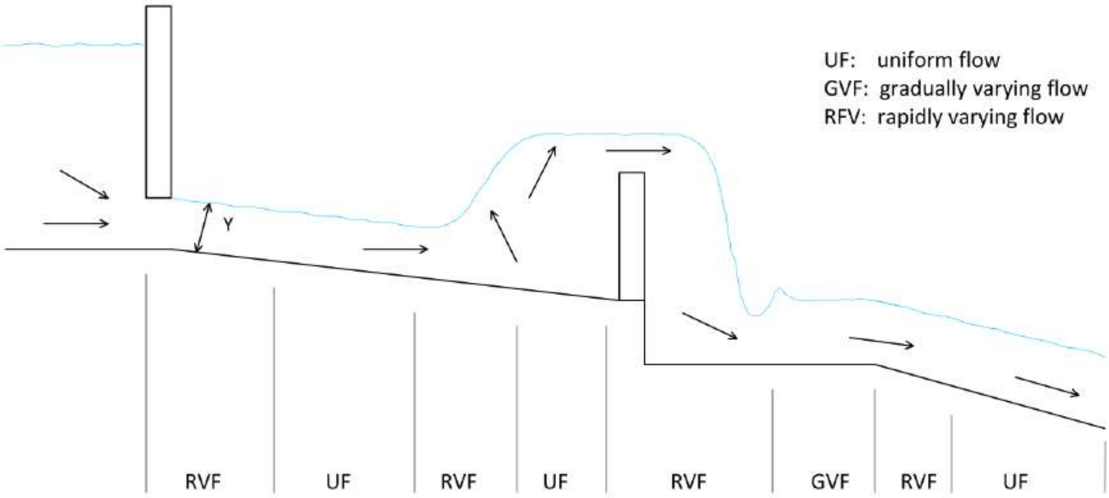


Figure 2.1: Open channel system characterized by subcritical, supercritical and uniform flow

2.2 Open channel terminology

Channels can be distinguished as natural or artificial. A long channel characterized by a minimal slope excavated in the ground is defined with canal. A prismatic channel exhibits a constant cross section and bottom slope. A cross section that is normal to the direction of the water flow is defined as channel section. The flow area A is the cross sectional area of the flow taken normal with respect to the direction of the flow. The depth of the water y is the distance between the water surface and the bottom of the channel in a section. The top width B is the width of channel section at the surface while the bottom width b is the width of the channel section at the lowest point of the channel section. The depth of flow section d can be defined as the depth of flow normal to the direction of flow. The hydraulic radius is defined as $R_h = \frac{A}{P}$, where the wetted perimeter P is defined as the length of intersection line of channel wetted surface with a cross sectional plane normal to the flow [40]. Figure 2.2 reports all the main geometric parameters of an open-channel.

The open-channel flow presents velocity components in the three directions.

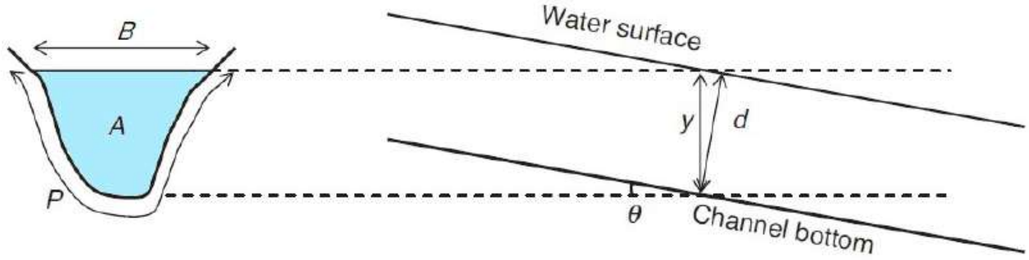


Figure 2.2: Open-channel section

On the other hand, for the most part, open-channel can be considered to be one-dimensional. Also, velocity component direction is the same of the flow. The velocity at different points in a channel section is denoted with v. So, the volume of water that is transferred through a channel section per unit time is denoted with discharge or flow rate [41]. Accordingly, the incremental fluid discharge dQ through an incremental area dA is

$$dQ = vdA \tag{2.1}$$

Then, the discharge can be defined as

$$Q = \int_A dQ = \int_A vdA \tag{2.2}$$

as it can be observed in Figure 2.3. Moreover, the cross-sectional average velocity V can be defined as

$$V = \frac{Q}{A} = \frac{1}{A}\int_A vdA \tag{2.3}$$

2.3 The Saint Venant Equations

Different kind of open channels such as prismatic canals or natural streams usually work under unsteady flow conditions. The unsteady open channel SVEs of motion are known as the continuity equation and the momentum equation that are respectively

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0, \tag{2.4}$$

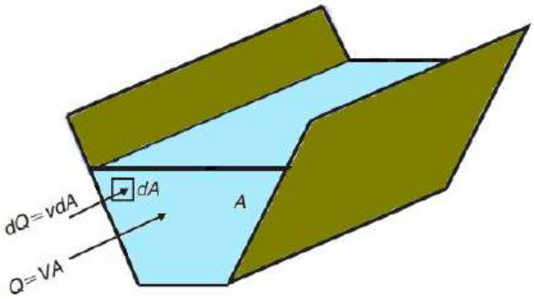


Figure 2.3: Open-channel section

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial y}{\partial x} + gA(\bar{S} - S_0) = 0 \tag{2.5}$$

where A is the cross sectional area of the channel, y is the flow depth, g is the gravitational acceleration, Q is the discharge, V is the average velocity, S_0 is the bottom slope and $\bar{S}$ is the friction slope, x is the displacement in the main flow direction and t is the time. Unsteady-flow equations are complicated and do not yield closed-form analytical solutions. These two equations are determined from mass conservation and momentum conservation if and only if the following assumptions are satisfied.

- No later inflow is considered.
- The flow is incompressible, that is the density of the fluid is constant.
- The flow is one-dimensional.
- Hydrostatic pressure prevails and vertical accelerations are negligible.
- The velocity is uniform over a channel.
- The flow velocity is only in the direction of flow and the components of flow velocity in the transverse and vertical directions are zero.
- The average channel bottom slope is small, that is the measured flow depth y is almost equal to the measured flow depth d perpendicular to the channel bottom.

2.3.1 Continuity equation derivation

Let us consider a volume element where ρ denotes the mass density of the fluid, A is the wetted cross sectional area of the channel, Q is the discharge, Δx is the distance between the upstream and downstream end along the flow direction. Let us denote with the subscript U the upstream section and with D the downstream section. The mass contained in a deformable volume that changes with the system, remains unchanged in time. The mass transfer rate in open-channel flow is the rate with which the mass is moved along a channel section and it is defined as

$$\text{Rate of mass transfer}= \rho Q.$$

The mass conservation law states that the mass of a closed system remains constant over time, meaning that

$$\text{Net rate of mass entering/leaving the volume =Rate of change of mass in the volume.}$$

If the volume of a section is given by $A\Delta x$ then the mass of the volume can be expressed as $\rho A\Delta x$. Let us assume that the water inflow in the volume has a rate ρQ_U and water outflow has a rate ρQ_D over a finite interval of time Δt. Therefore, the law of conservation of mass can be written as

$$\rho Q_U - \rho Q_D = \frac{\Delta(\rho A\Delta x)}{\Delta t} \tag{2.6}$$

Moreover, the conservation of water mass becomes the conservation of water volume if the density is constant, that is the fluid is incompressible. Then it holds

$$\begin{aligned} Q_U - Q_D &= \frac{\Delta(A\Delta x)}{\Delta t} \\ \frac{Q_U - Q_D}{\Delta x} &= \frac{\Delta A}{\Delta t} \\ \frac{\Delta Q}{\Delta x} + \frac{\Delta A}{\Delta t} &= 0 \end{aligned} \tag{2.7}$$

where $\Delta Q = Q_D - Q_U$. According to the fact that Δx and Δt approach zero, then equation (2.7) yields

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = 0 \tag{2.8}$$

where t and x represent the time and space displacement in the flow direction. A sketch for the mass conservation principle is reported in Figure 2.4.

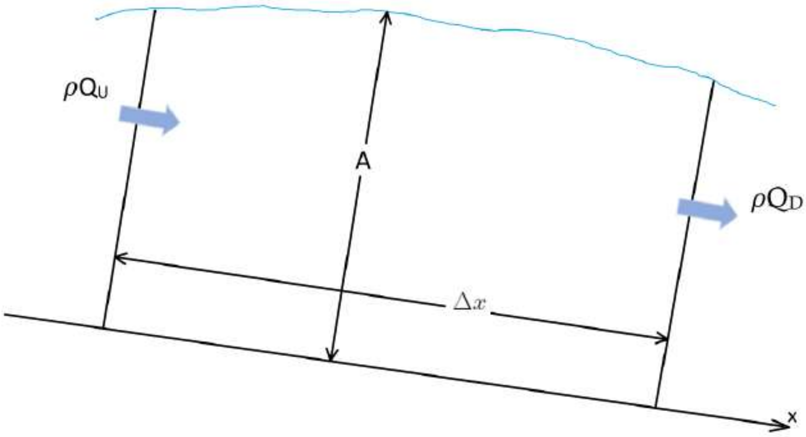


Figure 2.4: Definition sketch for mass conservation principle

2.3.2 Momentum equation derivation

Let consider a volume where ρ denotes the mass density of the fluid, A is the wetted cross sectional area of the channel, Q is the discharge, δx is the distance between the upstream and downstream end and the flow direction follows the $x-$axis. Let us consider the water level of the centroid $\bar{y}$, the mean velocity V, the force due to atmospheric pressure F_p, the friction force F_f and the force generated by the weight of the water F_w. Therefore, according to the conservation of the momentum it holds

Rate of change of momentum within a volume

=

Net rate of momentum transfer into the volume

+

Sum of all forces that act on the volume.

Then the rate of change of momentum can be expressed as

$$\frac{\Delta(\rho \Delta x A V)}{\Delta t} \tag{2.9}$$

The time rate of increase of momentum is given by

$$\rho Q_U V_U - \rho Q_D V_D. \tag{2.10}$$

The force components are pressure force, friction force and force due to the water weight. The pressure force is given by

$$F_{p,U} - F_{p,D} = \rho g A \bar{y}_U - \rho g A \bar{y}_D \tag{2.11}$$

where $\bar{y}$ denotes the centroid water depth. The force produced by the weight of the water is given by

$$F_w = \rho g A \Delta x S_0 \tag{2.12}$$

where $S_0 = \sin\theta$. Finally, the friction force F_f is opposite with respect to the flow direction and is given by

$$F_f = -\rho g A \Delta x \bar{S} \tag{2.13}$$

where $\bar{S}$ is the friction slope. According to Manning friction coefficient n [40], it holds $\bar{S} = \frac{n^2 Q^2}{A^2 R^{4/3}}$. Finally, putting all these components together and dividing by Δx leads to

$$\frac{\Delta(AV)}{\Delta t} + \frac{\Delta(QV)}{\Delta x} + g\frac{\Delta(A\bar{y})}{\Delta x} + gA\bar{S} - gAS_0 = 0 \tag{2.14}$$

where $\Delta(QV)$ and $\Delta(A\bar{y}) = A_d\bar{y}_d - A_u\bar{y}_u$ Hence, according to the assumption that Δx and Δt approach zero, the momentum equation becomes

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(\frac{Q^2}{A}) + gA\frac{\partial y}{\partial x} + gA\bar{S} - gAS_0 = 0 \tag{2.15}$$

where, as reported in [41], it is used the approximation $\frac{\Delta(A\bar{y})}{\Delta x} \simeq A\frac{\partial y}{\partial x}$. In Figure 2.5 a sketch of the derivation of the conservation of the momentum is shown.

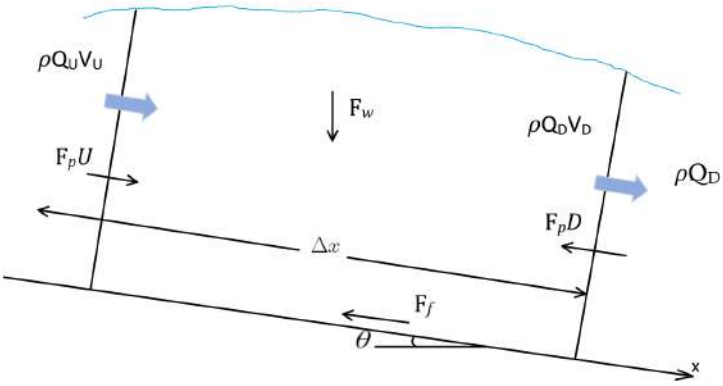


Figure 2.5: Sketch of derivation of the conservation of momentum equation

3

Modeling and estimation

In Chapter 2, we have analyzed the SVEs and we have explained that being partial differential equations, they are not simple to employ for estimation and control purpose. Then, we look for a simpler and straightforward alternative model. In this chapter, the main target is to figure out a model of a string of pools for estimation intention, using the system identification procedure and to examine the accuracy of the candidate models. Due to the lack of necessary measurements, the accuracy of the system identification models is evaluated by comparing the models with data generated from a discrete second order model.
System identification is the science of design mathematical models of dynamic systems from observed input-output data, using statistical methods. It can be interpreted as the interface between the real world and the mathematical world of control theory. The main procedure is examined in the following.

3.1 System identification procedure

The system identification procedure is the list of steps from which we design a mathematical model, starting from data generated by a certain system, to serve certain purposes. Such a procedure (see Figure 3.1) consists of the following main steps:

- **Experiment design**: Experiment design is fundamental for efficiently generating informative data to fit models and investigate the main dynamics of the system. Moreover, preliminary experiments allow to examine the linearity of the system under analysis, to obtain an estimate of the transfer function of the system and to retrieve information about dominating time constants, time delays, stationary gains.

- **Data preprocessing**: The collection of the data needs to be preprocessed by means of detrending process, decimation and filtering process due to possible numerical problems.

- **Model structure design**: Exploiting some a priori knowledge, we guess m model structures $\mathcal{M}_1 ... \mathcal{M}_m$, parametric or nonparametric, in order to find which one better describe the system. We are interested in two model classes for linear time-invariant systems, in particular, those that are transfer-function models and state-space models.

- **Training**: Parametric models require the computation of the estimate of parameters $\hat{\theta}_1 ... \hat{\theta}_m$ using $\mathcal{M}_1 ... \mathcal{M}_m$ respectively, and the data. In such a way, candidate models $\mathcal{M}_1(\hat{\theta}_1) ... \mathcal{M}_m(\hat{\theta}_m)$ are found. There exist several approaches to estimate model parameters, such as the prediction error approach or the maximum likelihood approach.

- **Validation**: A good model choice well fits data and is characterized by a good prediction performance. Several validation criteria, like residual analysis or cross-correlation test, allow to select the best model structure $\mathcal{M}_{OPT}(\hat{\theta}_{OPT})$ among several candidates $\mathcal{M}_1(\hat{\theta}_1) ... \mathcal{M}_m(\hat{\theta}_m)$, which better describes the system and is able to reproduce its behavior considering a new data collection.

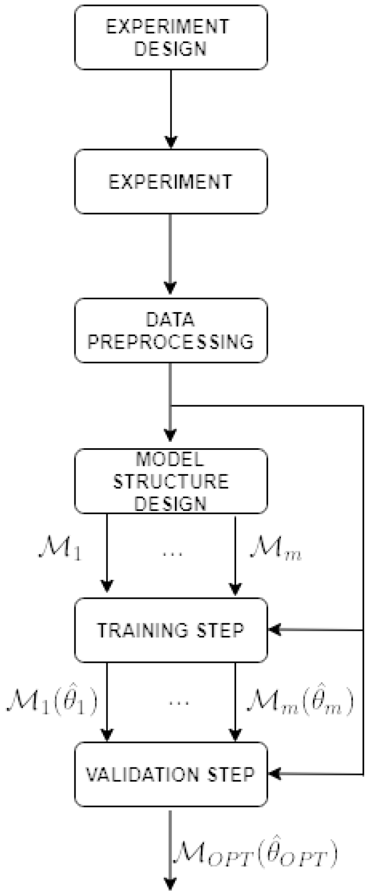


Figure 3.1: System identification procedure

3.2 Experiment design

Let us assume than the channel is automated with hydraulic control structures. It can be assumed that electric power is supplied by solar panels and data communications are carry out via a radio network. The stretch of the channel that extends between two hydraulic structure is defined as a *pool*. The measured variables are the *upstream water level* with respect to the hydraulic structure in m and the *hydraulic structure position* in m. Indeed, water level of each pool can be measured by means of submersible level pressure sensors while hydraulic structures position are measured based on the length of the steel cable between the hydraulic structure and the motor that moves it. Moreover, according to open-channel irrigation system literature [5], [13], the pool systems can be modeled by first order systems, then we can assume that the transfer function between the input and output variables is characterized by one pole. In addition, it is worth noting that there exist a time delay before the water flowing over an hydraulic structure reaches the point where the water level is measured. In order to collect experimental data, it is necessary to define a sampling time T_s, the duration of the experiment T and a reference signal. These variables can be chosen by means of a preliminary test on the open-loop system to gain some knowledge about the system dynamics. One of the most common used nonparametric system identification method is the step response test. The step response test consists on feeding the first order system with a unit step and measuring the step response, as shown in Figure 3.2, in order to obtain some information about the system behavior. In particular, we are interested about a suitable value for the experiment duration and for the period of the hydraulic structure position. Moreover, an experimental value for the discharge delay can be recovered.

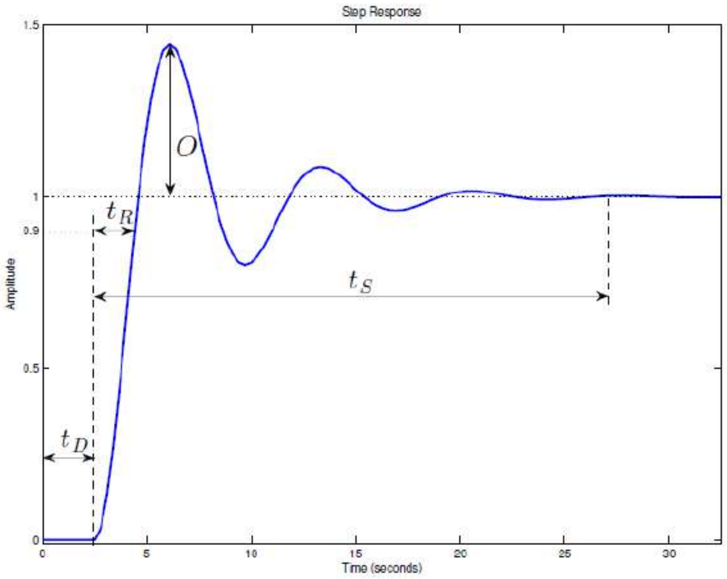


Figure 3.2: Unit step response

Thus, a step input flow should be applied to the second order model to obtain the corresponding step response of a pool system. Then, from the step response the rise time t_R can be computed and a rough estimate of the bandwidth f_B can be obtained as reported in [42] as follows

$$f_B \approx \frac{0.4}{t_R}. \tag{3.1}$$

A rule of thumbs is to choose the sampling time T_s as

$$T_s \leq \frac{1}{10 f_B} \approx \frac{t_R}{40}. \tag{3.2}$$

From the step response, the steady state time t_S can be computed. Then, in order to capture all the dominant modes of the system, the duration of the experiment should be greater than t_S. Then, as reported in [42], the minimum length of the data should be

$$N \approx \frac{t_S}{T_s}. \tag{3.3}$$

The time constant T_c for a first order system is the time it takes for the output response to reach 63% of its final value. The period of upstream control structure position T_g is usually approximated by T_c, i.e. $T_g \approx T_c$, meaning that the control structure should stay in a fixed position for a multiple of T_c. We also assume that there are several overshoot and undershoot hydraulic structures, called weirs and gates respectively. These can be located along the channel as sketched in Figures 3.3,3.4,3.5.

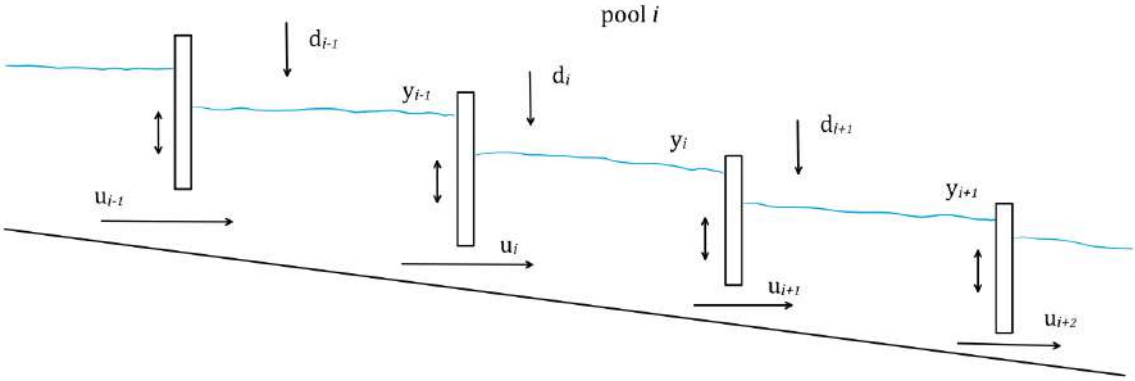


Figure 3.3: Model with gates structures

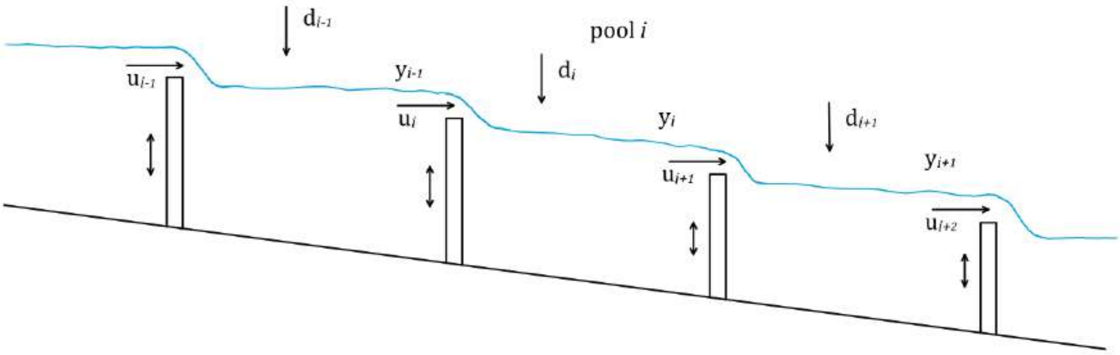


Figure 3.4: Model with weirs structures

The height of water above the weir is refer as h and is called *head over the weir*, and it can be determined from the current upstream water level measure

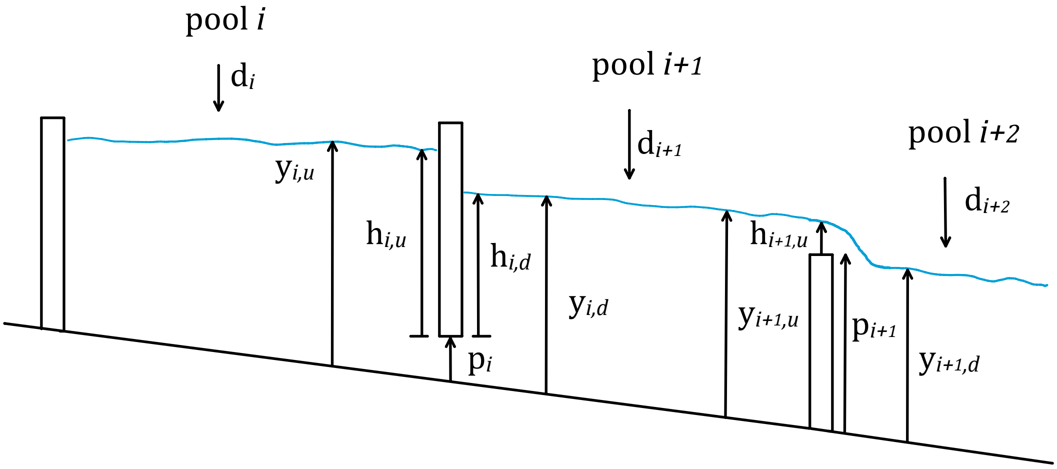


Figure 3.5: Model with gates and weirs structures

and the current weir position, refer as p, that is

$$h(t) = y_U(t) - p(t). \tag{3.4}$$

As a starting point for deriving a system identification model structure that takes into account both weirs and gates (see Figure 3.5), we can focus on the mass balance equation

$$\dot{Q}_v(t) = \alpha(Q_{in}(t) - Q_{out}(t)) \tag{3.5}$$

where Q_v refers to the volume of water in the pool, Q_{in}, Q_{out} denote the inflow and outflow of each pool in $\frac{m^3}{s}$ and $\alpha > 0$. In the literature [43], [44], there exist several relations describing the water discharge according to the flow conditions and shape of the chosen control structure. The considered relations for rectangular crested weirs and rectangular gates are reported in Table 3.1, where $c_{d_i} \approx 0.6$ is a discharge coefficient of pool textiti, p_i is the hydraulic structure position, y_{U_i} is the upstream water level and y_{D_i} is the downstream water level with respect to the hydraulic structure.

Let us assume to consider only submerged undershoot rectangular gates and free flow rectangular crested weirs. In our case measurements are not available, then it is possible to generate input and output data by means of the second order model. Moreover, making the further assumption that the volume of the

Table 3.1: Flow relation for rectangular crested weirs and rectangular gates.

	Free flow	Submerged flow
Rectangular gate	$Q_i = c_{d_i} p_i \sqrt{y_{U_i} - 0.5 p_i}$	$Q_i = c_{d_i} p_i \sqrt{y_{U_i} - y_{D_i}}$
Rectangular crested weir	$Q_i = c_{d_i} (y_{U_i} - p_i)^{3/2}$	$Q_i = c_{d_i} (y_{U_i} - y_{D_i})^{3/2}$

pool is proportional to the water level with respect to the hydraulic structure, we can formulate the following first order model for a pool

$$\begin{aligned} \dot{y}_U(t) &= \theta_1 Q_{in,U}(t-\tau) + \theta_2 Q_{out,U}(t) \\ \dot{y}_D(t) &= \theta_3 Q_{in,D}(t) + \theta_4 Q_{out,D}(t-\tau) \end{aligned} \tag{3.6}$$

where inflow and outflow are described by

$$Q(t) = \begin{cases} c_{weir} \cdot h^{\frac{3}{2}}(t), & \text{if rectangular crested weir;} \quad (3.7) \\ c_{gate} \cdot p(t) \cdot \sqrt{y_U(t) - y_D(t)}, & \text{if rectangular gate.} \quad (3.8) \end{cases}$$

where $c_{weir} = c_d b_w \sqrt{2g}$ and $c_{gate} = c_d A_g \sqrt{2g}$, b_w is the length of the aperture, $c_d \approx 0.6$ and A_g is the section of the aperture as reported in Appendix 6.2. Therefore, considering the time delay τ and introducing Euler approximation for the derivative, we obtain a first order model in discrete time

$$\begin{aligned} y_U((k+1)T_s) &= y_U(kT_s) + T_s\theta_1 Q_{in,U}((k-\tau)T_s) + T_s\theta_2 Q_{out,U}(kT_s) \\ y_D((k+1)T_s) &= y_D(kT_s) + T_s\theta_3 Q_{in,D}(kT_s) + T_s\theta_4 Q_{out,D}((k-\tau)T_s). \end{aligned} \tag{3.9}$$

where T_s is the sampling interval. Then, the parameters to be estimated by means of system identification are

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \tag{3.10}$$

However, such model is not able to describe waves, then more complex models could be required for estimating purpose.

3.3 Model structure selection

As the first step in the system identification procedure suggests, it is fundamental to understand the purpose of the model under investigation. There exist several model applications. For example, the model could be used for control, prediction, error detection or simulation. The purpose of the model affects the choice of identification methods, that justify the number of estimated parameter and the complexity of the model itself. For prediction purpose, it is important that the model catch all the dynamics while for control design it is required an accurate model around the desired crossover frequency. Indeed, the best suitable model structure should be selected considering a trade-off between complexity and performance. Let us assume to collect input $\mathbf{u}(t) \in \mathbb{R}^j$ and output $y(t) \in \mathbb{R}$ measures for $t = 1, \cdots, N$ during the experiment, that is we are dealing with a multi-input single-output system

$$u^N = \begin{bmatrix} u_1(1) & \cdots & u_j(1) \\ \vdots & & \vdots \\ u_1(N) & \cdots & u_j(N) \end{bmatrix}, \quad y^N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \tag{3.11}$$

Different structures of transfer-function models, or polynomials models are available:

- Equation error or Auto-Regressive model structure (ARX);
- Output-Error model structure (OE);
- Moving-Average-Auto-Regressive model structure (ARMAX);
- Box Jenkins model structure (BJ);

In the following, two parametric model structures will be introduced.

3.3.1 ARX Models

The input-output relationship is a linear difference equation

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_A} y(t-n_A) = &+ b_{0_1} u_1(t-1-\tau) + \cdots + b_{0_j} u_j(t-1-\tau) + \\ &+ b_{n_{B_1}} u_1(t-n_B+1-\tau) + \cdots + b_{n_{B_j}} u_j(t-n_B+1-\tau) + \\ &+ e(t) \end{aligned} \tag{3.12}$$

where $y(t) \in \mathbb{R}$, $\mathbf{u}(t) \in \mathbb{R}^j$, $n_B \neq 0$, $n_A \neq 0$, $\tau \geq 0$ is the input delay and $e(t)$ is white noise with variance σ^2 that enters the system as a direct error. Then, equation (3.12) can be rewritten in a more compact way

$$\mathbf{y}(t) = -\sum_{k=1}^{n_A} a_k y(t-k) + \sum_{k=0}^{n_B-1} \mathbf{b}_k \mathbf{u}(t-k-\tau) + e(t). \tag{3.13}$$

Let us define the polynomials in z^{-1} variable $A \in \mathbb{R}$ and $\mathbf{B} \in \mathbb{R}^{j \times j}$,

$$\mathbf{B}(z) = \begin{pmatrix} B_1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & B_i & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & B_j \end{pmatrix} \tag{3.14}$$

$$\begin{aligned} A(z) =& 1 + \sum_{k=1}^{n_A} a_k z^{-k} \Rightarrow deg(A(z)) = n_A \\ B_i(z) =& \sum_{k=0}^{n_B-1} b_k z^{-k} \Rightarrow deg(B_i(z)) = n_B - 1 \end{aligned} \tag{3.15}$$

Then, the above input-output relationship of an ARX model structure becomes

$$A(z)y(t) = \mathbf{B}(z)\mathbf{u}(t-\tau) + e(t) \Rightarrow \tag{3.16}$$

$$y(t) = \frac{\mathbf{B}(z)}{A(z)}\mathbf{u}(t-\tau) + \frac{1}{A(z)}e(t) = \mathcal{F}(z)\mathbf{u}(t-\tau) + \mathcal{G}(z)e(t) \tag{3.17}$$

where

$$\mathcal{F}(z) = \frac{\mathbf{B}(z)}{A(z)}, \quad \mathcal{G}(z) = \frac{1}{A(z)}. \tag{3.18}$$

A sketch of the ARX structure is shown in Figure 3.16. It can be observed that we can collect all the parameters into a vector

$$\theta = [a_1 \cdots a_{n_A}, b_{0_1} \cdots b_{(n_B-1)_1}, b_{0_j} \cdots b_{(n_B-1)_j}]^T \tag{3.19}$$

where number of parameters p in θ of the MISO system is

$$p = n_A + n_B \times j \tag{3.20}$$

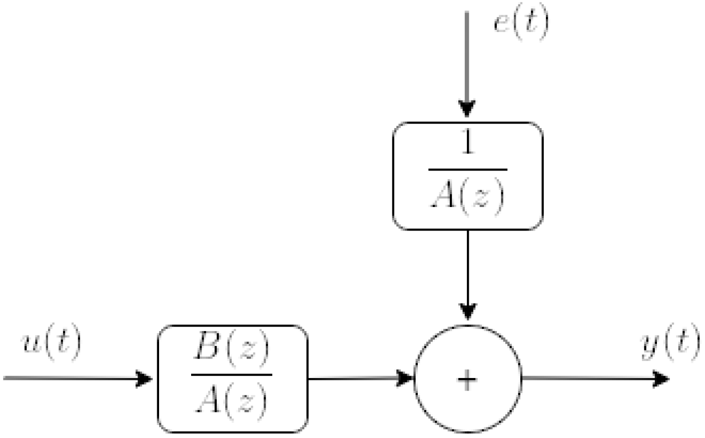


Figure 3.6: ARX model

Let us define the regressor vector $\varphi(t) \in \mathbb{R}^p$

$$\varphi(t) = [-y(t-1) \cdots -y(t-n_A), u_1(t-\tau) \cdots u_1(t-n_B+1-\tau), u_j(t-\tau) \cdots u_j(t-n_B+1-\tau)]^T \tag{3.21}$$

then equation (3.13) can be rewritten as a linear regression model

$$y(t) = \varphi(t)^T \theta + e(t) \tag{3.22}$$

3.3.2 OE Models

The input-output relationship is a linear difference equation

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_A} y(t-n_A) = &+ b_{0_1} u_1(t-1-\tau) + \cdots + b_{0_j} u_j(t-1-\tau) + \\ &+ b_{n_{B_1}} u_1(t-n_B+1-\tau) + \cdots + \\ &+ b_{n_{B_j}} u_j(t-n_B+1-\tau) + \\ &+ e(t) + a_1 e(t-1) + \cdots + a_{n_A} e(t-n_A) \end{aligned} \tag{3.23}$$

where $y(t) \in \mathbb{R}$, $\mathbf{u}(t) \in \mathbb{R}^j$, $\tau \geq 0$ is the input delay and $e(t)$ is white noise with variance σ^2, $n_B \neq 0$, $n_A \neq 0$. According to equation (3.15), the above input-output relationship of an OE model structure becomes

$$A(z)y(t) = \mathbf{B}(z)\mathbf{u}(t-\tau) + A(z)e(t) \Rightarrow \tag{3.24}$$

$$y(t) = \frac{\mathbf{B}(z)}{A(z)}\mathbf{u}(t-\tau) + e(t) = \mathcal{F}(z)\mathbf{u}(t-\tau) + e(t) \tag{3.25}$$

where

$$\mathcal{F}(z) = \frac{B(z)}{A(z)}. \tag{3.26}$$

A sketch of the OE structure is shown in Figure 3.24.

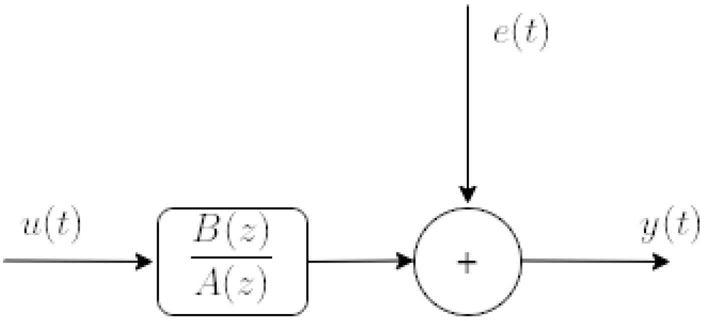


Figure 3.7: OE model

Let us define the regressor vector $\varphi(t) \in \mathbb{R}^p$

$$\varphi(t) = [-y(t-1) \cdots - y(t-n_A), u_1(t-\tau) \cdots u_1(t-n_B+1-\tau), u_j(t-\tau) \cdots u_j(t-n_B+1-\tau)]^T \tag{3.27}$$

then equation (3.25) can be rewritten as a regression model

$$y(t) = \varphi(t)^T\theta + e(t) + \sum_{k=1}^{n_A} a_k e(t-k). \tag{3.28}$$

3.4 Delay estimation

From the step test, it is clear that the mass balance system can be modeled as a time delay system. We can estimate approximate time lags for each of the inflows/outflows computed shifting each input data set by τ minutes, where $\tau \in T$, T $= \left\{3, 5, 6, 7, 8\right\}$ and evaluating a statistical correlation test on the shifted data versus the downstream flow, as suggested in [23]. The amount τ of the shifted data which shows the highest correlation is then taken to be as the suitable time lag of that input. The range of τ is defined for each pool of the river because it is known approximately from the step test. Then, the time delays can be computed from the cross-correlation between the measurements at the upstream and the downstream end of each pool. The cross-correlation can be computed as reported in [42]

$$R_{cc}(\tau) = \frac{1}{N}\sum_{i=1}^{N} \tilde{Q}_{in}(i)\tilde{Q}_{out}(i-\tau) \tag{3.29}$$

where

$$\tilde{Q}(i) = \frac{1}{\sigma(Q(i))}\left[Q(i) - \tfrac{1}{N}\textstyle\sum_{j=1}^{N} Q(j)\right] \tag{3.30}$$

and σ is the standard deviation of the flow. Then, an estimate of τ can be computed by means of optimization procedure. It is worth noting that such a solution is only locally optimal, because we cannot guarantee the convexity of the cross-correlation function. Then, it holds

$$\hat{\tau} = \underset{\tau \in \mathrm{T}}{\operatorname{argmax}}\, R_{cc}(\tau) \tag{3.31}$$

3.5 PEM method

For the estimation of time-varying parameters, the prediction error minimization (PEM) method is a criterion that is the analogue of the least-square principle in the static case. In particular, past values of the input and the output are required to perform a one-step ahead prediction through optimal prediction theory and to obtain the error with respect to the measured output. The usual performance metric of the fitting is the least squares and parameters result from the optimization procedure. Given a model $\mathcal{M}(\theta)$ the prediction error at time t is given by

$$\varepsilon(t) := y(t) - \hat{y}(t|t-1) \tag{3.32}$$

and the overall mean-square error (MSE) is defined as

$$V(\theta) := \frac{1}{N}\sum_{t}^{N} \varepsilon(t)^2 \tag{3.33}$$

where $V : \mathbb{R}^p \to \mathbb{R} : \theta \mapsto V(\theta) \in \mathbb{R}$. In the predictive approach to system identification, the parameters of $\mathcal{M}(\theta)$ are tuned minimizing the cost $V(\theta)$ over all $\theta \in \Theta$, that is

$$\hat{\theta}_{PEM}(y^N, u^N) = \underset{\theta \in \Theta}{\operatorname{argmin}}\, V(\theta) \tag{3.34}$$

If the model quality is accurate, the prediction error has to be white. Let us consider the general model structure $\mathcal{M}$

$$\mathcal{M}(\theta): \quad \mathbf{y}(t) = \mathcal{F}(z)\mathbf{u}(t) + \mathcal{G}(z)\mathbf{e}(t) \tag{3.35}$$

where $\mathbf{e}$ is white noise with variance σ^2 and zero mean and $\mathcal{F}(\infty) = 0$. At time t we have past data, then the optimal prediction of y(t) under model $\mathcal{M}(\theta)$ is

$$\hat{y}(t|t-1) = \mathcal{G}(z)^{-1}\mathcal{G}_1(z)y(t) + \mathcal{G}(z)^{-1}\mathcal{F}(z)u(t) \tag{3.36}$$

where $\mathcal{G}_1(z) = \mathcal{G}(z) - 1$, as reported in [42].

3.5.1 PEM with ARX model structure

The one-step ahead predictor of a MISO system can be determined by means of the general formula (3.36), where we assign

$$\mathcal{F}(z) = z^{-1}\frac{\mathbf{B}(z)}{A(z)}, \quad \mathcal{G}(z) = \frac{1}{A(z)}, \quad \mathcal{G}_1(z) = \frac{1 - A(z)}{A(z)} \tag{3.37}$$

Then, the one-step ahead predictor $\mathcal{M}(\hat{\theta})$ becomes

$$\begin{aligned} \hat{y}(t|t-1) &= \Big[1 - A(z)\Big]\, y(t) + \mathbf{B}(z)\mathbf{u}(t-\tau) \\ &= -a_1 y(t-1) - \cdots - a_{n_A} y(t-n_A) + \\ &+ b_{0_1} u_1(t-1-\tau) + \cdots + b_{0_j} u_j(t-1-\tau) + \\ &+ b_{n_{B_1}} u_1(t-n_B+1-\tau) + \cdots + b_{n_{B_j}} u_j(t-n_B+1-\tau) \end{aligned} \tag{3.38}$$

and equation (3.38) can be rewritten exploiting the regressor vector as

$$\hat{y}(t|t-1)) = \varphi(t)^T\hat{\theta}_{PEM}. \tag{3.39}$$

It can be observed that $\hat{y}(t|t-1)$ is a linear combination of past values of the input and output and that is linear in the unknown θ. Let us define

$$\mathbf{\Phi} := \begin{bmatrix} \varphi(1)^T \\ \vdots \\ \varphi(N)^T \end{bmatrix} \qquad \bar{\varepsilon}_\theta = \begin{bmatrix} \varepsilon_\theta(1)^T \\ \vdots \\ \varepsilon_\theta(N)^T. \end{bmatrix} \tag{3.40}$$

Therefore, we obtain $\bar{\varepsilon}_\theta = y^N - \Phi\theta$ and so

$$V_N(\theta) = \frac{1}{N}\left\|\bar{\varepsilon}_\theta\right\|^2 = \frac{1}{N}\left\|y^N - \Phi\theta\right\|^2 \tag{3.41}$$

and so the estimated parameters are given by

$$\hat{\theta}_{PEM}(y^N, u^N) = \underset{\theta\in\Theta}{\operatorname{argmin}}\, V_N(\theta) = \underset{\theta\in\Theta}{\operatorname{argmin}}\left\|y^N - \Phi\theta\right\|^2 \tag{3.42}$$

whose corresponding unique solution is given by the least-squares estimate

$$\hat{\theta}_{PEM}(y^N, u^N) = (\Phi^T\Phi)^{-1}\Phi^T y^N \tag{3.43}$$

provided that Φ has full column rank.

3.5.2 PEM with OE model structure

The one-step ahead predictor $\mathcal{M}(\hat{\theta})$ is

$$\hat{\mathbf{y}}(t|t-1) = \frac{B(z)}{A(z)}\mathbf{u}(t-\tau) \tag{3.44}$$

and it can be observed that is a linear combination of past values of the input but it depends also on past predictions, since:

$$A(z)\hat{y}(t|t-1)) = \mathbf{B}(z)\mathbf{u}(t-\tau) \Rightarrow 0 = -A(z)\hat{y}(t|t-1) + \mathbf{B}(z)\mathbf{u}(t-\tau) \tag{3.45}$$

Then, it holds

$$\begin{aligned}\hat{y}(t|t-1) &= \hat{y}(t|t-1) - A(z)\hat{y}(t|t-1) + \mathbf{B}(z)\mathbf{u}(t-\tau)\\ &= -a_1\hat{y}(t-1|t-2) - \cdots - a_{n_A}\hat{y}(t-1-n_A|t-2-n_A)+\\ &+ b_{0_1}u_1(t-1-\tau) + \cdots + b_{0_j}u_j(t-1-\tau)+\\ &+ b_{n_{B_1}}u_1(t-n_B+1-\tau) + \cdots + b_{n_{B_j}}u_j(t-n_B+1-\tau)\end{aligned} \tag{3.46}$$

Then, using the simulated output from the model, the relationship is effectively no longer linear. The main difference with least square regression is that the predictor depends on past predicted values and that the vector of parameters $\hat{\theta}_{PEM}$ is obtained minimizing a quadratic criterion by means of nonlinear programming algorithm. Indeed, the model is not linear in θ so there exist no analytic solution for $\hat{\theta}_{PEM}$. Given a general deterministic objective function

$$f : \mathbb{R}^p \mapsto \mathbb{R} : \theta \mapsto f(\theta) \in \mathbb{R} \tag{3.47}$$

the aim is to find $\theta_{PEM} \in \mathbb{R}^p$ such that

$$\hat{\theta}_{PEM} = \operatorname*{argmin}_{\theta \in \Theta} f(\theta) \tag{3.48}$$

$$\Theta = \left\{ \theta \in \mathbb{R}^p \right\}. \tag{3.49}$$

In the following we drop the subscript PEM for readability and we denote the gradient of f the p-dimensional vector

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \vdots \\ \frac{\partial f}{\partial \theta_p} \end{bmatrix} \in \mathbb{R}^p \tag{3.50}$$

and the Hessian of f is the matrix

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial \theta_1 \partial \theta_1} & \cdots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \theta_p \partial \theta_1} & \cdots & \frac{\partial^2 f}{\partial \theta_p \partial \theta_p} \end{bmatrix} \in \mathbb{R}^{p \times p}. \tag{3.51}$$

In the following, three off-line optimization algorithm are introduced in order to find $\hat{\theta}$ for the OE model.

3.5.3 Gradient descent method

In a non-linear optimization problem it is not always easy to find the optimum solution θ^* in closed form, where

$$\theta^* = \operatorname*{argmin}_{\theta \in \mathbb{R}^p} V(\theta) \tag{3.52}$$

and $V : \mathbb{R}^p \mapsto \mathbb{R} : \theta \mapsto V(\theta) \in \mathbb{R}$. As reported in [45], we need to search for a minimization sequence represented by the update rule and the descent condition respectively

$$\begin{cases} \theta^{(i+1)} = \theta^{(i)} - \alpha^{(i)} \Delta\theta^{(i)} \\ V(\theta^{(i+1)}) \leq V(\theta^{(i)}) \end{cases} \tag{3.53}$$

where

- i is the current iteration
- $\alpha^{(i)}$ is the current positive step size
- $\Delta\theta^{(i)}$ [1] is the current descent direction

3.5.4 Steepest-descent method

One of the simplest method based on gradient descent is Steepest-descent. The update rule is

$$\theta^{(i+1)} = \theta^{(i)} - \alpha^{(i)} \nabla V(\theta^{(i)}) \tag{3.54}$$

where the descent direction is chosen opposite to the gradient and

$$\Delta\theta^{(i)} = -\nabla V(\theta^{(i)}). \tag{3.55}$$

Since the step size is not always suitable, a weighting matrix $D^{(i)} \in \mathbb{R}^{n\times n}$ can be introduced as follows

$$\theta^{(i+1)} = \theta^{(i)} - \alpha^{(i)} D^{(i)} \nabla V(\theta^{(i)}) \tag{3.56}$$

Let us compute the Jacobian of the objective function $V(\theta)$

$$\begin{aligned}
\frac{\partial V(\theta)}{\partial \theta} &= \frac{1}{N}\sum_{t=1}^{N} \frac{\partial \varepsilon_\theta(t)^2}{\partial \theta} = \frac{1}{N}\sum_{t=1}^{N} \frac{\partial \varepsilon_\theta(t)^2}{\partial \varepsilon_\theta(t)} \frac{\partial \varepsilon_\theta(t)}{\partial \theta} \\
&= \frac{2}{N}\sum_{t=1}^{N} \varepsilon_\theta(t) \frac{\partial \varepsilon_\theta(t)}{\partial \theta} = \frac{2}{N}\sum_{t=1}^{N} \varepsilon_\theta(t) \frac{\partial y(t) - \hat{y}(t|t-1)}{\partial \theta} \\
&= -\frac{2}{N}\sum_{t=1}^{N} \varepsilon_\theta(t) \frac{\partial \hat{y}(t|t-1)}{\partial \theta} = -\frac{2}{N}\mathbf{J}^T \mathcal{E}
\end{aligned} \tag{3.57}$$

where

$$J_\theta(t) := \frac{\partial \hat{y}(t|t-1)}{\partial \theta} \in \mathbb{R}^p \tag{3.58}$$

$$\mathbf{J} := \begin{bmatrix} J_\theta(1) \\ \vdots \\ J_\theta(N) \end{bmatrix} \qquad \mathcal{E} := \begin{bmatrix} \varepsilon_\theta(1) \\ \vdots \\ \varepsilon_\theta(N) \end{bmatrix} \tag{3.59}$$

[1] To be precise, $-\Delta\theta^{(i)}$ is the descent direction

The Steepest-descent algorithm is generally stable, but it is very slow at the neighborhood of the optimum. There exist several methods that are derived from Steepest-descent according to the choice of the weighting matrix, such as Newton method, Gauss-Newton method, Levenberg-Marquardt method.

3.5.5 Gauss-Newton method

The Gauss-Newton method is a modification of the Newton method to solve nonlinear regression problems for sum-of-squares objective functions. In such a case we are dealing with

$$V(\theta) = \frac{1}{N} \left\| \varepsilon_\theta(t) \right\|^2 . \tag{3.60}$$

The Newton method requires the computation of the Hessian $\nabla^2 V$, providing that it is invertible. The Gauss-Newton method introduces an approximation of its in order to avoid Hessian computation and, in particular, it can be expressed in term of J. Moreover, this method requires that the objective function is approximately quadratic in the parameters near the optimal solution. The Gauss-Newton algorithm is characterized by its fast convergence, but it is often unstable. The function evaluated at the perturbed parameters can be locally approximated by means of a first-order Taylor series expansion as follows

$$\hat{y}(\theta + \Delta\theta) \simeq \hat{y}(\theta) + \mathbf{J}\Delta\theta \tag{3.61}$$

Then, the objective function is

$$\begin{aligned} V(\theta + \Delta\theta) \simeq & (\mathcal{E} - \mathbf{J}\Delta\theta)^T (\mathcal{E} - \mathbf{J}\Delta\theta) \\ \simeq & \mathcal{E}^T\mathcal{E} - 2\Delta\theta^T \mathbf{J}^T \mathcal{E} + \Delta\theta^T \mathbf{J}^T \mathbf{J} \Delta\theta \end{aligned} \tag{3.62}$$

Then, the Jacobian of the objective function becomes

$$\begin{aligned} \frac{\partial V(\theta + \Delta\theta)}{\partial \Delta\theta} \simeq & \partial \frac{(\mathcal{E} - \mathbf{J}\Delta\theta)^T (\mathcal{E} - \mathbf{J}\Delta\theta)}{\partial \Delta\theta} \\ \simeq & 2(\mathbf{J}^T\mathbf{J})\Delta\theta - 2\mathbf{J}^T\mathcal{E} \end{aligned} \tag{3.63}$$

Therefore, $\Delta\theta = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathcal{E}$ and the weighting matrix for the update rule of Gauss-Newton method is $\mathbf{D}^{(i)} = (\mathbf{J}^T\mathbf{J})^{-1} = (2\nabla\hat{y}\nabla\hat{y})^{-1}$, where it can be observed that the Hessian of the function $V(\theta)$ has been approximated as $\nabla^2 V \simeq 2\nabla\hat{y}\nabla\hat{y}$.

3.5.6 Levenberg-Marquardt method

The Levenberg-Marquardt method is an iterative algorithm that adaptively computes the parameter switching between Gauss-Newton and Steepest descent methods. The main idea is to introduce a parameter $\lambda > 0$ in the weighting matrix, such that it holds

$$\Delta\theta = (\mathbf{J}^T\mathbf{J} + \lambda \text{ diag } (\mathbf{J}^T\mathbf{J}))^{-1}\mathbf{J}^T\mathcal{E} \tag{3.64}$$

and the parameter update rule becomes

$$\theta^{(i)} = \theta^{(i-1)} - (\mathbf{J}^T\mathbf{J} + \lambda \text{ diag } (\mathbf{J}^T\mathbf{J}))^{-1}\mathbf{J}^T\mathcal{E}. \tag{3.65}$$

The choice of λ determines the behavior of the algorithm. In particular, small values of the damping parameter λ corresponds to a Gauss-Newton update while large values corresponds to a Steepest descent update. The damping parameter λ is initialized to be large in order to reach small steps in the first iterations in the Steepest-descent direction. Such a parameter is increased if the previous iteration produces a small reduction of the objective, that is the parameter vector θ is far from their optimal value, either as the solution improves, it is decreased.

3.5.7 Numerical implementation

At iteration i, step $\Delta\theta$ is computed by comparing $V(\theta)$ and $V(\theta+\Delta\theta)$. The step is accepted if a metric is greater than a threshold $\omega > 0$. Such a metric represents a measure of the actual improvement in V when compared to the improvement of a Levenberg-Marquardt update and it is defined as follows

$$\begin{aligned} \chi_i(\Delta\theta) &= \frac{V(\theta) - V(\theta+\Delta\theta)}{\mathcal{E}^T\mathcal{E} - (\mathcal{E} - \mathbf{J}\Delta\theta)^T - (\mathcal{E} - \mathbf{J}\Delta\theta)} \\ &= \frac{V(\theta) - V(\theta+\Delta\theta)}{\Delta\theta^T(\lambda_i \text{ diag } (\mathbf{J}^T\mathbf{J})\Delta\theta + \mathbf{J}^T\mathcal{E})} \end{aligned} \tag{3.66}$$

Then, if the step is accepted, namely there is an improvement because $V(\theta+\Delta\theta) > V(\theta)$, the Jacobian $\mathbf{J} \in \mathbb{R}^{p\times 1}$ can be numerically approximated by means of forward differences or central differences respectively

$$J_{j1} = \frac{\partial \hat{y}}{\partial \theta_j} = \begin{cases} \dfrac{\hat{y}(t,\theta+\delta\theta_j) - \hat{y}(t,\theta)}{\|\delta\theta_j\|} & \text{if } \delta\theta_j < 0 \qquad (3.67) \\ \dfrac{\hat{y}(t,\theta+\delta\theta_j) - \hat{y}(t,\theta-\delta\theta_j)}{2\,\|\delta\theta_j\|} & \text{otherwise} \qquad (3.68) \end{cases}$$

where the $j-th$ element of $\delta\theta_j = \zeta(1+|\theta_j|)$ and ζ is a small perturbation. On the other hand, if there are several parameters, a finite differences Jacobian is computationally expensive. If the Jacobian is re-computed exploiting finite differences only occasionally, convergence can be reached with fewer function evaluations. Then, this approximation is exploited if there is an improvement or for iterations that are even multiple of θ dimension. Thus, in all other iterations, the Jacobian can be updated by means of Broyden formula [46]

$$\mathbf{J} = \mathbf{J} + \frac{(\hat{y}(\theta+\Delta\theta) - \hat{y}(\theta) - \mathbf{J}\Delta\theta)\Delta\theta^T}{\Delta\theta^T\Delta\theta}. \tag{3.69}$$

Then, the choice of the adaptive parameterλ determines the behavior of the Levenberg-Marquardt algorithm. Such parameter is updated according to the metric value, that is if $\chi_i(\Delta\theta)$ then the parameter will be reduced, otherwise it will be increased and the algorithm proceeds to the next iteration. In particular, such parameter updates as follows

$$\lambda_{i+1} = \begin{cases} \max\left[\frac{\lambda_i}{\alpha}, 10^{-7}\right] & \text{if } \chi_i(\Delta\theta) > \gamma_3 \qquad (3.70) \\ \min\left[\frac{\lambda_i}{\beta}, 10^{7}\right] & \text{otherwise} \qquad (3.71) \end{cases}$$

where $\alpha > 0$, $\beta > 0$, $gamma_3$.

3.5.8 Stop criteria

Convergence can be reached when one of the following criteria is satisfied,

- Convergence in the gradient: $\max\left|\mathbf{J}^T\mathcal{E}\right| < \gamma_1$;
- Convergence in parameters: $\max\left|\frac{\Delta\theta}{\theta}\right| < \gamma_2$;

assuming that $\gamma_1 > 0$ and $\gamma_2 > 0$. In addition, the algorithm should stop once a maximum number of iteration i_{max} is reached although the convergence criteria are not satisfied. These three conditions are associated to a flag f_{STOP} in the pseudocode 1.

3.5.9 Pseudocode

The pseudocode of the Levenberg-Marquardt optimization procedure for the OE model is proposed in Pseudocode 1.

Algorithm 1 Levenberg-Marquardt

```
Input: $u_1, u_2, y_{meas}, \hat{y}_0, \hat{\theta}_0, flag, \gamma_3, w_1, \lambda_0$
Output: $\hat{y}, \hat{\theta}$
$i \leftarrow 0$
$V_{old} \leftarrow w_1$
$\lambda \leftarrow \lambda_0$
$\hat{y} \leftarrow$ compute iteratively as Eq. (3.16).
$\mathbf{J} \leftarrow$ compute as Eq. (3.67), (3.68) or (3.69)
$\mathcal{E} \leftarrow y_{meas} - \hat{y}$
$\mathbf{V} \leftarrow \mathcal{E}^T\mathcal{E}$
$V_{old} \leftarrow V$
while $(\sim f_{STOP})$ do
    $i \leftarrow i + 1$
    $\Delta\theta \leftarrow$ compute as Eq. (3.64)
    $\theta_{new} \leftarrow \theta + \Delta\theta$
    $\hat{y} \leftarrow$ compute iteratively as Eq. (3.46)
    $\mathcal{E} \leftarrow y_{meas} - \hat{y}$
    $\mathbf{V}_{new} \leftarrow \mathcal{E}^T\mathcal{E}$
    $\chi \leftarrow$ compute as Eq. (3.66)
    if $\chi > \gamma_3$ then
        $V_{old} \leftarrow V$
        $\theta_{old} \leftarrow \theta$
        $\hat{y}_{old} \leftarrow \hat{y}$
        $\theta \leftarrow \theta_{new}$
        $\hat{y} \leftarrow$ compute iteratively as Eq. (3.46)
        $\mathbf{J} \leftarrow$ compute as Eq. (3.67), (3.68) or (3.69)
        $\mathcal{E} \leftarrow y_{meas} - \hat{y}$
        $\mathbf{V} \leftarrow \mathcal{E}^T\mathcal{E}$
        $\lambda \leftarrow$ compute as Eq. (3.70),(3.71)
    else
        $\mathbf{V} \leftarrow V_{old}$
        $\hat{y}_{old} \leftarrow \hat{y}$
        $\hat{y} \leftarrow$ compute iteratively as Eq. (3.46)
        $\mathbf{J} \leftarrow$ compute as Eq. (3.67), (3.68) or (3.69)
        $\mathcal{E} \leftarrow y_{meas} - \hat{y}$
        $\mathbf{V} \leftarrow \mathcal{E}^T\mathcal{E}$
        $\lambda \leftarrow$ compute as Eq. (3.70),(3.71)
    end if
end while
```

3.6 Model validation

Validation is a procedure that provides a criterion to select a model structure $\mathcal{M}$ among a set of candidate model structures $\mathcal{S}$.

3.6.1 Hold-out cross validation

The main idea is to split the data in two data sets, one for the training step and one for the validation step, if the overall data is sufficiently large. The goal is to test the model's capability to predict new data that is not used in estimating it, in order to avoid overfitting and to understand how the model will generalize a new data set. For any model class $\mathcal{S}$, the model $\mathcal{M}(\hat{\theta})$ that better reproduces the training data is selected, and then its performance is evaluated by means of validation criteria such as mean square error. The model that minimizes such criterion among different candidates is selected as the most suitable one. Then, given data

$$u^N = \begin{bmatrix} u_1(1) & \cdots & u_j(1) \\ \vdots & & \vdots \\ u_1(N) & \cdots & u_j(N) \end{bmatrix}, \quad y^N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \tag{3.72}$$

where $\mathbf{u}(t) \in \mathbb{R}^j$ and $y(t) \in \mathbb{R}$, we can partition them into two subsets:

- **Training data set**

$$u_T^{N-k} = \begin{bmatrix} u_1(1) & \cdots & u_j(1) \\ \vdots & & \vdots \\ u_1(N-k) & \cdots & u_j(N-k) \end{bmatrix}, \quad y_T^{N-k} = \begin{bmatrix} y(1) \\ \vdots \\ y(N-k) \end{bmatrix} \tag{3.73}$$

- **Validation data set**

$$u_V^k = \begin{bmatrix} u_1(N-k+1) & \cdots & u_j(N-k+1) \\ \vdots & & \vdots \\ u_1(N) & \cdots & u_j(N) \end{bmatrix}, \quad y_V^k = \begin{bmatrix} y(N-k+1) \\ \vdots \\ y(N) \end{bmatrix} \tag{3.74}$$

where the typical choice is $k = \frac{N}{2}$. Then, for the two candidate models ARX and OE $\mathcal{M} \in \mathcal{S}$:

1. we compute the PEM estimate using training data set; in this way we obtain $\hat{\theta}_{PEM}(y_T^{N-k}, u_T^{N-k})$

2. we test the prediction capability of $\mathcal{M}(\hat{\theta}_{PEM}(y_T^{N-k}, u_T^{N-k}))$, computing the prediction error

$$\varepsilon_{\hat{\theta}_{PEM}}(t) = y(t) - \hat{y}_{\hat{\theta}_{PEM}}(t|t-1), \quad t = N-k+1, \cdots, N \tag{3.75}$$

or equivalently

$$\mathcal{E}^k_{\hat{\theta}_{PEM}} = y^k_V - \hat{y}^k_V \tag{3.76}$$

where

$$\mathcal{E}^k_{\hat{\theta}_{PEM}} = \begin{bmatrix} \varepsilon^k_{\hat{\theta}_{PEM}}(N-k+1) \\ \vdots \\ \varepsilon^k_{\hat{\theta}_{PEM}}(N) \end{bmatrix} \quad \hat{y}^k_V = \begin{bmatrix} \hat{y}^k_{PEM}(N-k+1|N-k) \\ \vdots \\ \hat{y}^k_{PEM}(N|N-1) \end{bmatrix} \tag{3.77}$$

3. we compute the fit percent term as

$$J_{FIT}(\mathcal{M}) = \left(1 - \frac{\left\|\mathcal{E}^k_{\hat{\theta}_{PEM}}\right\|}{\left\|\hat{y}^k_V - \bar{y}_V\right\|}\right) \cdot 100 \tag{3.78}$$

where

$$\bar{y}_V = \begin{bmatrix}1 \cdots 1\end{bmatrix}^T \left(\tfrac{1}{k}\textstyle\sum_{t=N-k+1}^N y(t)\right). \tag{3.79}$$

In conclusion, we choose the model structure in $\mathcal{S}$ that maximizes the fit term.

In fact, there exist several indices of performance that can be considered. In particular, the root mean square error $RMSE$ and correlation coefficient R_{cc} are used to this aim and are defined respectively as

$$RMSE = \sqrt{\frac{1}{N}\left(\mathcal{E}^T\mathcal{E}\right)} \tag{3.80}$$

$$R_{cc} = \frac{\frac{1}{N}\left(y - \underline{y}\right)\left(\hat{y} - \underline{\hat{y}}\right)}{\sqrt{\frac{1}{N}\left(y - \underline{y}\right)^2} \times \sqrt{\frac{1}{N}\left(\hat{y} - \underline{\hat{y}}\right)^2}} \tag{3.81}$$

where $\underline{y} = (1/N)\sum_{i=1}^N y_i$ is the mean value of y.

3.6.2 Model structure selection

Akaike information criterion (AIC) is model structure criterion, useful to select as best model structure $\mathcal{M}$ the one which minimizes the cost :

$$J_{AIC}(\mathcal{M}) = 2p + \ln J(\hat{\theta}) \tag{3.82}$$

where

- p is the number of parameters in θ
- $J(\theta) = \frac{1}{N} \sum_{t=1}^{N} \varepsilon(t, \theta)^2$

It is worth noting that AIC criterion consists in the sum of two terms:

- the complexity term $2p$, that favors simple models;
- the fit-term, that favors models well explaining the data;

Therefore, this criterion selects a model within the presence of a trade-off between the adherence of data and complexity of the system.

4

Decentralized control design

In Chapter 3, we have proposed two grey-box models for multiple pool channel for estimation purpose. In this chapter, a simpler model is chosen for control design. In particular, a decentralized control solution is proposed in order to regulate the water level of the string of pools, acting on the hydraulic structure position. Both rectangular crested weirs and rectangular gates are considered. We take inspiration by some literature about open-channel irrigation system [15], [35], [47], [39]. The complexity of the model proposed is chosen taking into account the context of control design for set-point regulation and perturbation rejection. It is worth noting that, since the free space upon the set point of a pool is not supposed to cope with waves, an additional constraint is to guarantee that the wave dynamics are not excited. Waves are undesirable, then it is necessary to guarantee that each local control-loop bandwidth lies at a frequency lower than the corresponding dominant wave phenomenon. Since these considerations are satisfied during the design of feedback controllers, a first-order model is sufficient for control purpose. It is fundamental to implement a dynamic feedback control in order to efficiently discharge water and reject the perturbations. It can be introduced local upstream control, that is one control policy classically implemented in irrigation system field, capable of capturing the first mode of wave phenomena. Conventional controllers such as PI-regulators are relatively easy to tune and they often reach good performance. Then, a simple feedback configuration is proposed for local upstream water control. Moreover, the addition of feedforward allows to attenuate the water level error propagation and amplification.

4.1 Control model design

Let us denote by y_i the *upstream water level* of pool i, $i = 1, ..., N$, u_i the *upstream control*, u_{i+1} is the *downstream control*, d_i the rain perturbation. In the following, $G_i(s)$ and $\tilde{G}_i(s)$, respectively represent the transfer functions from u_i to y_i and from u_{i+1} to y_i, while s denotes the Laplace domain. Moreover, the transfer function from the perturbation d_i to y_i is also represented by $\tilde{G}_i$, since we are dealing with an additive disturb on the downstream discharge of the pool. Therefore, the pool model in frequency domain is

$$\mathcal{P}_i : \quad y_i(s) = G_i(s)u_i(s) + \tilde{G}_i(s)(u_{i+1}(s) + d_i(s)) \tag{4.1}$$

with the boundary condition $u_1 \equiv 0$. Thus, the channel model can be thought of as a string of pool models, as shown in Figure 4.1.

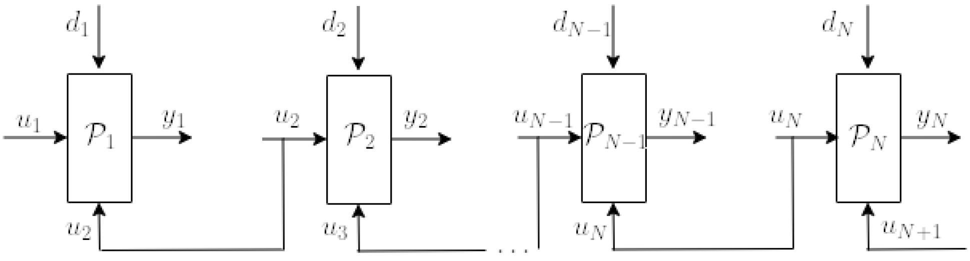


Figure 4.1: Channel modeled as a string of N pools

This analytical model presents a very good frequency domain approximation of the Saint-Venant transfer matrix for a pool. For low frequencies, it can be observed that the behavior of the transfer matrix is dominated by the integrator and the delays, while for high frequencies, the delay and the gravity waves are dominant in the transfer matrix. For simplicity, the gravity waves are approximated by a constant gain in high frequencies, that is the oscillating modes are not modeled as such. Then, these modes are controlled by the control structure and, only the low frequency component of a pool has to be taken into account. We need to analyze and design a controller that acts on low frequencies and that is consistent with the actuator bandwidth. The idea is that the controllers on the hydraulic structures along the channel eliminate the effect produced by structures in low frequency. Such a regulation leads to recovering the model related to each pool, controlled by the discharge at each control structure. This is the reason for which we can assume that $G_i(s)$ and $\tilde{G}_i(s)$ can be derived from the integrator delay zero model:

$$G_i(s) = \frac{e^{-\tau_i s}}{A_b s} \tag{4.2}$$

$$\tilde{G}_i(s) = -\frac{1}{A_b s} \tag{4.3}$$

where τ_i is the propagation delay of pool and A_b is the backpropagation area. Such a simple model is suitable for control purpose because it capture the main dynamics of a pool in low frequency.

4.2 Local upstream control

There exist two main decentralized control policies for a canal pool, that are distant downstream control and local upstream control (see Figure 4.2).

Distant downstream control is one of the most common used control approach for open-channel irrigation system, because it shows a parsimonious water management. It consists in controlling the downstream water level using the upstream control variable. Local upstream control shows a high performance with respect to unpredicted disturbances but it shows a worst water efficiency with respect to distant downstream approach. We are interested in local upstream control of a pool, that deals with the regulation of the upstream water level y_i with respect to

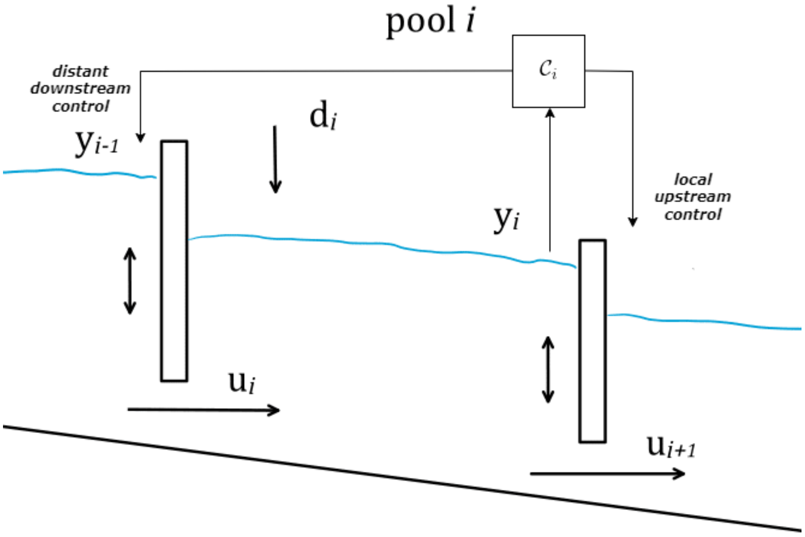


Figure 4.2: Distant downstream and local upstream control

the hydraulic structure, by means of the control variable u_{i+1}, that is the output discharge of the pool. Let us denote by r_i the reference for the water level of pool i and by e_i the corresponding tracking error. A sketch of the system is then represented in Figure 4.3.

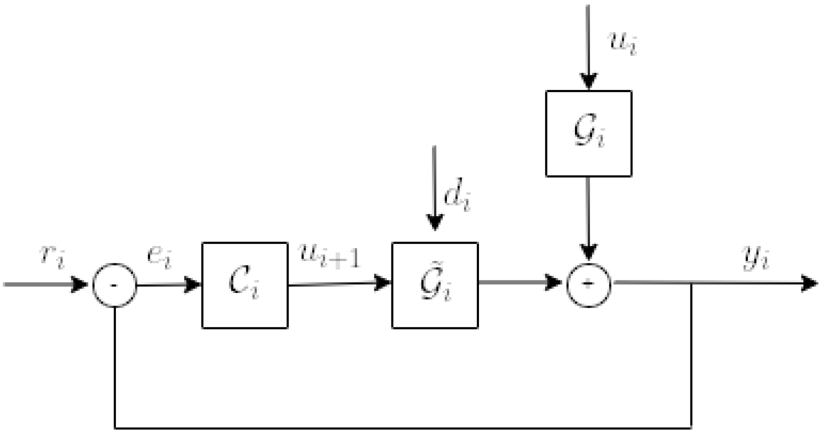


Figure 4.3: Sketch of a generic local upstream control system

Let us define the open-loop transfer function

$$H(s) := C(s)\tilde{G}(s) \tag{4.4}$$

where $C(s)$ is a linear controller, and assume that the closed-loop is BIBO stable. Thus, for the frequency response theorem we obtain the expression for component of steady state tracking error e_S

$$e_S(s) = \left(\frac{\tilde{G(s)}}{1+\tilde{G(s)}C_{(s))}}\right) d(s). \tag{4.5}$$

Thus, in order to have perfect asymptotic rejection to $d(s)$ we should impose $e_S(s) \equiv 0$. Therefore, the control aim is to determine a linear controller such that

$$e_S(s) \equiv 0 \quad \Leftrightarrow \quad \left|(1+\tilde{G}(s)C(s))^{-1}\tilde{G}(s)\right| \equiv 0 \tag{4.6}$$

over the largest frequency bandwidth. The achievable bandwidth is limited by the actuators limitations because there are no time delay in $\tilde{G}(s)$. Thus, local upstream control allows a high performance with respect to unpredicted perturbation but shows a low water efficiency because all the disturbs are propagated downstream without modifying the upstream discharge.

4.2.1 Decentralized Feedback without Feedforward

The decentralized feedback control scheme is proposed in Figure 4.4.

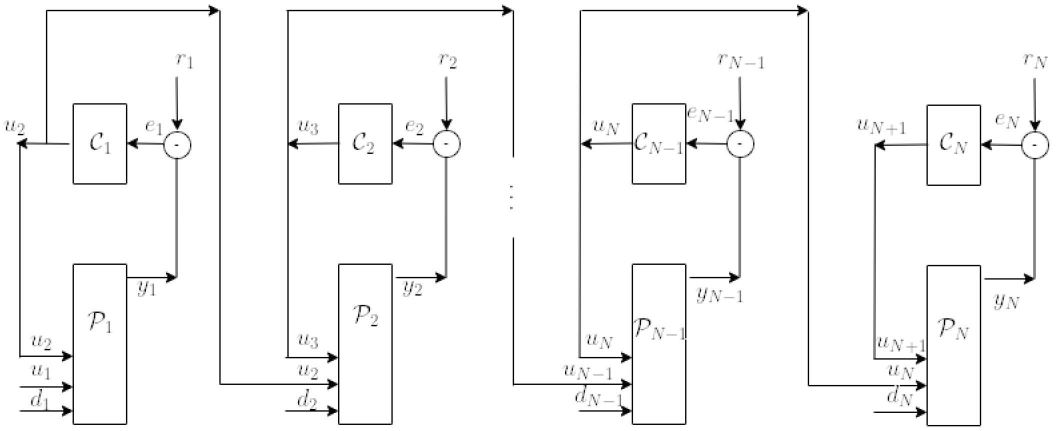


Figure 4.4: Feedback control scheme

We want to show how a control system which is decomposed into subcontrollers can be exploited to solve multivariable control problems. For sake of simplicity, we adopt here single-input single-output (SISO) controllers of the form

$$u_{i+1}(s) = C_i(s)e_i(s) = C_i(s)(r_i(s) - y_i(s)) \tag{4.7}$$

and they are designed according to the local water depth reference r_i and the measured water depth y_i. The design of each regulator C_i is performed considering both a good perturbation rejection and a good local upstream water depth control. It is worth noting that u_i should be considered as a component of the disturb. Let us observe that the local closed-loop transfer functions are yielded by

$$W_{r_i \to e_i}(s) := \frac{1}{1 + C_i(s)\tilde{G}_i(s)}; \tag{4.8}$$

$$W_{d_i \to e_i}(s) := \frac{\tilde{G}_i(s)}{1 + C_i(s)\tilde{G}_i(s)}; \tag{4.9}$$

$$W_{d_i \to u_{i+1}}(s) := \frac{C_i(s)\tilde{G}_i(s)}{1 + C_i(s)\tilde{G}_i(s)}. \tag{4.10}$$

According to the final value theorem, C_i should have at least one pole in zero to attain a zero steady state error in case of a step perturbations d_i. Moreover, it is important to guarantee that the local loop-gain $\left|C_i(jw)\tilde{G}_i(jw)\right|$ is large in correspondence of low frequencies, where the reference and the perturbation are relevant. The bandwidth of the loop gain $\left|C_i(jw)\tilde{G}_i(jw)\right|$ need to lie under the frequency of the local wave dynamics, that are not captured by the model. Then, a PID regulator can satisfy these requirements. Moreover, it presents phase lead at the desired loop gain bandwidth for stability, robustness and additional roll-off to that guarantee a low gain in correspondence of the dominant wave frequency. In order to figure out the downstream error propagation we can consider a string of identical pool, with the same time delay τ and identical decentralized feedback regulators $C(s)$. Since these assumptions hold, the transfer function from the reference to the water level error is given by the closed loop transfer functions

$$W_{r_i \to e_i}(s) := \frac{1}{1 + C(s)\tilde{G}(s)} \tag{4.11}$$

while the error propagates according to the transfer function

$$W_{e_i \to e_{i+1}}(s) := \frac{G(s)}{\tilde{G}(s)} \frac{C(s)}{1 + C(s)\tilde{G}(s)} \tag{4.12}$$

It is worth noting that in such a case there can be coupling of control action into downstream pools.

4.2.2 Decentralized Feedback with Feedforward

The propagation and amplification of water level errors can lead to actuator saturation and such a problem can be interpreted as a further disturbance to the flows out of each pool. This means that the control action applied to hydraulic structure i to balance the perturbation in pool i affects the flow u_{i+1} out of pool i and so water level error in pool $i+1$ increases. According to the fact that a measure of control input u_{i+1} can be achieved at hydraulic structure i, these disturbances can be considered known. Decentralized feedback control scheme with additional feedfoward is thus proposed in Figure 4.5.

In particular, it holds that

$$u_{i+1}(s) = C_i(s)e_i(s) + F_i(s)u_i(s), \tag{4.13}$$

where the feedforward compensator $F_i(s)$ is a stable transfer function. Let us consider a string of identical pool, with the same time delay τ and identical decentralized feedback regulators $C_i(s)$ and feedforward compensators $F_i(s)$. Since

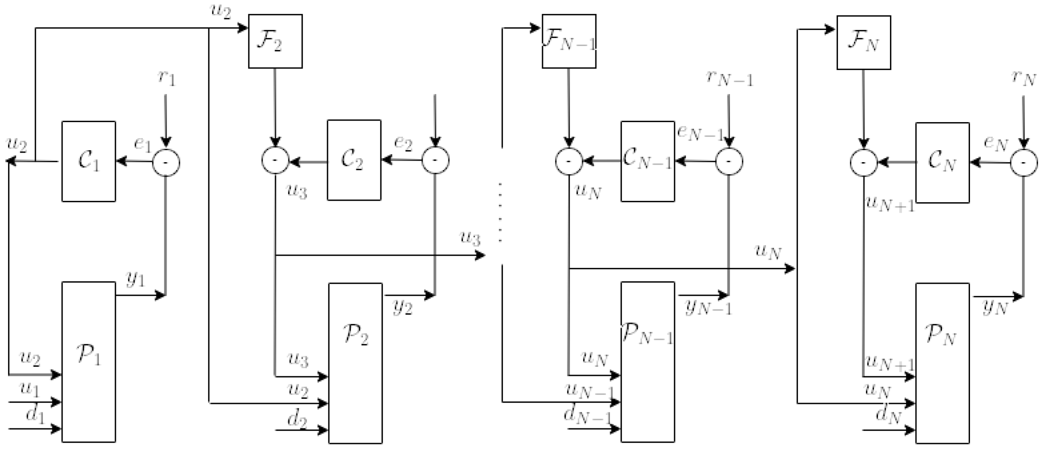


Figure 4.5: Feedback Feedforward control scheme

these assumptions hold, water level error propagation is characterized by

$$W_{u_{i+1}\to e_i}(s) := \frac{\tilde{G}(s) - F(s)\tilde{G}(s)}{1 + C(s)\tilde{G}(s)} \tag{4.14}$$

and the relationship

$$u_{i+1}(s) = C(s)e_i(s) + F(s)u_i(s). \tag{4.15}$$

Thus, the relationship between water level error of neighboring pools can be stated as

$$e_{i+1}(s) = W_{e_i\to e_{i+1}}(s)e_i(s) \tag{4.16}$$

where

$$W_{e_i\to e_{i+1}}(s) = \frac{G(s)}{\tilde{G}(s)}\frac{C(s) - F(s)}{1 + C(s)\tilde{G}(s)}. \tag{4.17}$$

In general, decomposed control approach can easily turn into a complex system, complicated to maintain and to manage. For what concern the control performance, it could be simpler to arrange the controller design problem as an optimization problem. As a consequence, it could be translated into a centralized multivariable controller design problem. On the other hand, there exist several reasons to choose cascaded and decentralized control. One of the most relevant reason is the cost associated with finding good plant models, which is fundamental for employing multivariable control. Moreover, with this approach, each controller is commonly tuned one at a time, with a minimum of modeling effort. Another advantage is that these systems are easy to understand by operators and tuning parameters of the controller have a local effect. Then, the main task is to retrieve a control configuration which permits the (sub)controllers to be tuned independently based on a minimum of model information.

4.3 Multi-variable Decentralized control

Local upstream control policy can be extended to multiple pool channel control by means of decentralized control structures. This approach is advantageous because controllers are easy to be implemented and tuned and they require an easy maintenance. According to equation (4.1), a multiple-pool channel can be represented by the model

$$y(s) = G(s)u(s) + \tilde{G}(s)d(s) \tag{4.18}$$

where G is the bidiagonal matrix

$$G(s) = \begin{pmatrix} G_1(s) & \tilde{G}_1(s) & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & G_i(s) & \tilde{G}_i(s) & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & G_n(s) & \tilde{G}_n(s) \end{pmatrix} \tag{4.19}$$

and

$$\tilde{G}(s) = \begin{pmatrix} \tilde{G}_1(s) & 0 & 0 & 0 & 0 & \\ 0 & \ddots & 0 & 0 & 0 & \\ 0 & 0 & \tilde{G}_i(s) & 0 & 0 & \\ 0 & 0 & 0 & \ddots & 0 & \\ 0 & 0 & 0 & 0 & 0 & \tilde{G}_n(s) \end{pmatrix}. \tag{4.20}$$

Without loss of generality, we can focus on a four-pools channel system, where $y(t) \in \mathbb{R}^4$, $u(t) \in \mathbb{R}^5$ and $d(t) \in \mathbb{R}^4$. Then, the controlled variables vector, the control input vector and the disturbance vector are respectively

$$y(t) := \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \end{bmatrix} \quad u(t) := \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \\ u_5(t) \end{bmatrix} \quad d(t) := \begin{bmatrix} d_1(t) \\ d_2(t) \\ d_3(t) \\ d_4(t) \end{bmatrix}. \tag{4.21}$$

Therefore, the string of pools is described by the following system

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} G_1 & \tilde{G}_1 & 0 & 0 & 0 \\ 0 & G_2 & \tilde{G}_2 & 0 & 0 \\ 0 & 0 & G_3 & \tilde{G}_3 & 0 \\ 0 & 0 & 0 & G_4 & \tilde{G}_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} + \begin{bmatrix} \tilde{G}_1 & 0 & 0 & 0 \\ 0 & \tilde{G}_2 & 0 & 0 \\ 0 & 0 & \tilde{G}_3 & 0 \\ 0 & 0 & 0 & \tilde{G}_4 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \tag{4.22}$$

where we drop the time domain for readability.

4.3.1 Stability and performance

Let us evaluate the stability of the multivariable system (4.22) with decentralized upstream control. In such a case, the regulator matrix is chosen to be constant and is given by

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 \\ K_{21} & 0 & 0 & 0 \\ K_{31} & K_{32} & 0 & 0 \\ K_{41} & K_{42} & K_{43} & 0 \\ K_{51} & K_{52} & K_{53} & K_{54} \end{pmatrix} \tag{4.23}$$

where $K_{21}, K_{32}, K_{43}, K_{54}$ are the monovariable local upstream controllers for each pool, while all the other entries are additional decoupling terms, that express the interaction between pools. Then, the open-loop of the system (4.22) becomes

$$H := GK = \begin{pmatrix} \tilde{G}_1 K_{21} & 0 & 0 & 0 \\ G_2 K_{21} + \tilde{G}_2 K_{31} & \tilde{G}_2 K_{31} & 0 & 0 \\ G_3 K_{31} + \tilde{G}_3 K_{41} & G_3 K_{32} + \tilde{G}_3 K_{42} & \tilde{G}_3 K_{43} & 0 \\ G_4 K_{41} + \tilde{G}_4 K_{51} & G_4 K_{42} + \tilde{G}_4 K_{52} & G_4 K_{43} + \tilde{G}_4 K_{53} & \tilde{G}_4 K_{54} \end{pmatrix} \tag{4.24}$$

and the control input vector for the multivariable model (4.22) becomes

$$u := \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} 0 \\ K_{21} e_1 \\ K_{32} e_2 \\ K_{43} e_3 \\ K_{54} e_4 \end{pmatrix} \tag{4.25}$$

According to the transfer function computed in equations (4.9) and (4.12), we can define the relationship $\mathcal{N}$ between the tracking error vector and the disturbance vector as follows

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \begin{pmatrix} \mathcal{N}_{11} & 0 & 0 & 0 \\ \mathcal{N}_{21} & \mathcal{N}_{22} & 0 & 0 \\ \mathcal{N}_{31} & \mathcal{N}_{32} & \mathcal{N}_{33} & 0 \\ \mathcal{N}_{41} & \mathcal{N}_{42} & \mathcal{N}_{43} & \mathcal{N}_{44} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix} \tag{4.26}$$

where if $i = j$, the entries along the diagonal are

$$\mathcal{N}_{ij} = -\tilde{G}_i(1 + \tilde{G}_i K_{i+1,j})^{-1} \tag{4.27}$$

while if $i \neq j$ the entries off-diagonal are

$$\mathcal{N}_{j,i} = \mathcal{N}_{i,i} K_{i+1,i} \frac{G_{i+1}}{\tilde{G}_{i+1}} \mathcal{N}_{j,i+1} \tag{4.28}$$

Then, in our particular case the entries of matrix $\mathcal{N}$ are

$$\begin{aligned}
\mathcal{N}_{11} &:= W_{d_1 \to e_1}(s) = -\tilde{G}_1(1 + \tilde{G}_1 K_{21})^{-1}; \\
\mathcal{N}_{21} &:= W_{d_1 \to e_1 \to e_2}(s) = \mathcal{N}_{11} K_{21} \left(\tfrac{G_2}{\tilde{G}_2}\right) \mathcal{N}_{22}; \\
\mathcal{N}_{22} &:= W_{d_2 \to e_2}(s) = -\tilde{G}_2(1 + \tilde{G}_2 K_{32})^{-1}; \\
\mathcal{N}_{31} &:= W_{d_1 \to e_1 \to e_2 \to e_3}(s) = \mathcal{N}_{11} K_{21} \left(\tfrac{G_2}{\tilde{G}_2}\right) \mathcal{N}_{32}; \\
\mathcal{N}_{32} &:= W_{d_2 \to e_2 \to e_3}(s) = \mathcal{N}_{22} K_{32} \left(\tfrac{G_3}{\tilde{G}_3}\right) \mathcal{N}_{33}; \\
\mathcal{N}_{33} &:= W_{d_3 \to e_3 \to e_2 \to e_3}(s) = -\tilde{G}_3(1 + \tilde{G}_3 K_{43})^{-1}; \\
\mathcal{N}_{41} &:= W_{d_1 \to e_1 \to e_2 \to e_3 \to e_4}(s) = \mathcal{N}_{11} K_{23} \left(\tfrac{G_2}{\tilde{G}_2}\right) \mathcal{N}_{42}; \\
\mathcal{N}_{42} &:= W_{d_2 \to e_2 \to e_3 \to e_4}(s) = \mathcal{N}_{22} K_{32} \left(\tfrac{G_3}{\tilde{G}_3}\right) \mathcal{N}_{43}; \\
\mathcal{N}_{43} &:= W_{d_3 \to e_3 \to e_4}(s) = \mathcal{N}_{33} K_{43} \left(\tfrac{G_4}{\tilde{G}_4}\right) \mathcal{N}_{44}; \\
\mathcal{N}_{44} &:= W_{d_4 \to e_4}(s) = -\tilde{G}_4(1 + \tilde{G}_4 K_{54})^{-1}.
\end{aligned} \tag{4.29}$$

We should focus on the performance of such a decentralized controller, by examining the gain of $\mathcal{N}(jw)$. In the multivariable case, the module of a transfer function is extended by means of the use of a matrix norm, for example the largest singular value. As reported in [48], the maximum singular value of a transfer function is very useful in terms of frequency-domain performance and robustness. Thus, the performance of the closed-loop system (4.26) can be investigated evaluating the singular values of matrix $\mathcal{N}(jw)$. The singular values of a matrix $\mathcal{N}$ can be written as

$$\lambda_i(\mathcal{N}) = \sqrt{\sigma_i(\mathcal{N}^*\mathcal{N})} = \sqrt{\sigma_i(\mathscr{N})} \tag{4.30}$$

where we have defined

$$\mathscr{N} := \mathcal{N}^*\mathcal{N}, \tag{4.31}$$

where $\mathcal{N}^*$ is the complex conjugate transpose of $\mathcal{N}$ and σ and λ refer to the eigenvalues of $\mathscr{N}(jw)$ and $\mathcal{N}(jw)$ respectively. The largest singular value $\bar{\lambda}$ is a matrix norm, then it can be shown that

$$\left\|e(jw)\right\| = \left\|\mathcal{N}(jw)d(jw)\right\| \leq \bar{\lambda}(\mathcal{N}(jw)) \left\|d(jw)\right\| \tag{4.32}$$

Hence, for definition of the largest singular value, there is a couple of disturbances $(\bar{d}_1, \bar{d}_2)$ such that the tracking error norm achieves the upper bound of the inequality. Accordingly, the largest singular value represents an estimate of the performance of the system. It can be highlighted that the closed loop transfer function $\mathcal{N}$ is a lower triangular matrix, then it holds

$$\prod_{i=1}^{k} \sigma_i = \prod_{i=1}^{k} \lambda_i^2 = \prod_{i=1}^{k} \left|\mathcal{N}_{i,i}\right|^2 \tag{4.33}$$

and

$$\begin{aligned} tr(\mathscr{N}) &= \sum_{i=1}^{k} \sigma_i = \sum_{i=1}^{k}\sum_{j=1}^{k} \left|\mathcal{N}_{i,j}(jw)\right|^2 = \left|\mathcal{N}_{11}\right|^2 + \left|\mathcal{N}_{21}\right|^2 + \left|\mathcal{N}_{22}\right|^2 + \left|\mathcal{N}_{31}\right|^2 + \cdots + \left|\mathcal{N}_{44}\right|^2 \\ &\geq \sum_{i=1}^{k} \left|\mathcal{N}_{i,i}(jw)\right|^2 = \left|\mathcal{N}_{11}\right|^2 + \left|\mathcal{N}_{22}\right|^2 + \left|\mathcal{N}_{33}\right|^2 + \left|\mathcal{N}_{44}\right|^2 \end{aligned} \tag{4.34}$$

where tr denotes the trace of the matrix. It can be observed that the trace of $\mathscr{N}(jw)$ depends on the coupling terms $K_{31}, K_{41}, K_{42}, K_{51}, K_{52}, K_{53}$ and in particular it holds

$$\bar{\lambda}(\mathcal{N}(jw)) > max(\left|\mathcal{N}_{11}(jw)\right|, \cdots, \left|\mathcal{N}_{44}(jw)\right|) \tag{4.35}$$

where $\bar{\lambda}$ is the maximum eigenvalue of $\mathcal{N}$. This confirms that the coupling tends to degenerate the whole performance in terms of perturbation rejection. In order to solve such an issue, the off-diagonal entries of $\mathcal{N}$ should be zero.

4.3.2 Feedforward decoupler

The disturbance that is propagated by the first pool in the forward direction is perfectly known, because it coincides with the control input. Then, such an issue can be solved by means of an additional feedforward term in the regulator, without degenerating stability and robustness of the closed loop 4.26. Hence, the control input vector for the multivariable model (4.22) becomes

$$\mathbf{u} := \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} 0 \\ K_{21}e_1 \\ K_{32}e_2 + K^F(s)u_2 \\ K_{43}e_3 + K^F(s)u_3 \\ K_{54}e_4 + K^F(s)u_4 \end{pmatrix}. \tag{4.36}$$

Then, the relationship $\mathcal{N}$ between the tracking error and the perturbation 4.26 introduces the feedforward term, then if $i = j$, the entries along the diagonal are

$$\mathcal{N}_{ij} = -\tilde{G}_i(1 + \tilde{G}_i K_{i+1,j})^{-1} \tag{4.37}$$

while if $i \neq j$ the entries off-diagonal are

$$\mathcal{N}_{j,i} = \mathcal{N}_{i,i} K_{i+1,i} \left(\tfrac{G_{i+1}}{\tilde{G}_{i+1}} + K_{j,i} \right) \mathcal{N}_{i+1,i+1} \tag{4.38}$$

where

$$K^F_{j,i} = K_{j+1,i} K^{-1}_{i+1,i}. \tag{4.39}$$

If we set the feedforward terms to

$$\begin{aligned} K^F_{21} &= K_{31}K^{-1}_{21}; \\ K^F_{31} &= K_{41}K^{-1}_{21}; \\ K^F_{32} &= K_{42}K^{-1}_{32}; \\ K^F_{41} &= K_{51}K^{-1}_{21}; \\ K^F_{42} &= K_{52}K^{-1}_{32}; \\ K^F_{43} &= K_{53}K^{-1}_{43}; \end{aligned} \tag{4.40}$$

then, the entries of $\mathcal{N}$ become

$$\begin{aligned}
&\mathcal{N}_{11} := W_{d_1 \to e_1}(s) = -\tilde{G}_1(1+\tilde{G}_1 K_{21})^{-1};\\
&\mathcal{N}_{21} := W_{d_1 \to e_1 \to e_2}(s) = \mathcal{N}_{11} K_{21}\left(\tfrac{G_2}{\tilde{G}_2} + K_{31}K_{21}^{-1}\right)\mathcal{N}_{22};\\
&\mathcal{N}_{22} := W_{d_2 \to e_2}(s) = -\tilde{G}_2(1+\tilde{G}_2 K_{32})^{-1};\\
&\mathcal{N}_{31} := W_{d_1 \to e_1 \to e_2 \to e_3}(s) = \mathcal{N}_{11} K_{21}\left(\tfrac{G_2}{\tilde{G}_2} + K_{41}K_{21}^{-1}\right)\mathcal{N}_{32};\\
&\mathcal{N}_{32} := W_{d_2 \to e_2 \to e_3}(s) = \mathcal{N}_{22} K_{32}\left(\tfrac{G_3}{\tilde{G}_3} + K_{42}K_{32}^{-1}\right)\mathcal{N}_{33};\\
&\mathcal{N}_{33} := W_{d_3 \to e_3 \to e_2 \to e_3}(s) = -\tilde{G}_3(1+\tilde{G}_3 K_{43})^{-1};\\
&\mathcal{N}_{41} := W_{d_1 \to e_1 \to e_2 \to e_3 \to e_4}(s) = \mathcal{N}_{11} K_{23}\left(\tfrac{G_2}{\tilde{G}_2} + K_{51}K_{21}^{-1}\right)\mathcal{N}_{42};\\
&\mathcal{N}_{42} := W_{d_2 \to e_2 \to e_3 \to e_4}(s) = \mathcal{N}_{22} K_{32}\left(\tfrac{G_3}{\tilde{G}_3} + K_{52}K_{32}^{-1}\right)\mathcal{N}_{43};\\
&\mathcal{N}_{43} := W_{d_3 \to e_3 \to e_4}(s) = \mathcal{N}_{33} K_{43}\left(\tfrac{G_4}{\tilde{G}_4} + K_{53}K_{43}^{-1}\right)\mathcal{N}_{44};\\
&\mathcal{N}_{44} := W_{d_4 \to e_4}(s) = -\tilde{G}_4(1+\tilde{G}_4 K_{54})^{-1}.
\end{aligned} \tag{4.41}$$

The interaction between pools effect can be reduced with a suitable choice for the decoupling terms of the controller, that is

$$\begin{aligned}
K_{31} &= K_{21}^F K_{21};\\
K_{41} &= K_{31}^F K_{21};\\
K_{42} &= K_{32}^F K_{32};\\
K_{51} &= K_{41}^F K_{21};\\
K_{52} &= K_{42}^F K_{32};\\
K_{53} &= K_{43}^F K_{43};
\end{aligned} \tag{4.42}$$

Then, in order to obtain exact decoupling, the off-diagonal elements of $\mathcal{N}$ should be zero, that is

$$K_{ji}^F = -\frac{G_{i+1}(s)}{\tilde{G}_{i+1}(s)} = e^{-\tau_{i+1}} \tag{4.43}$$

that is a causal transfer function, then also the decoupling terms are causal.

4.4 Tuning of local upstream PI controllers

The controller that we take into account is a PI augmented with a low pass filter (PIL). Indeed, the inflow of water from the rain to the pools is equivalent to a load disturbance and then the controller needs to be able to reject load disturbances. Moreover, it is important to track water level reference changes. The dimensionless PI controller is denoted with

$$C(s) = K_P \left(1 + \tfrac{1}{T_I s}\right) \tag{4.44}$$

where K_P is the proportional gain and T_I the integral time. This corresponds to a continuous controller such that the control u can be computed as

$$u(t) = K_P(y^*_{REF}(t) - y(t)) + \frac{K_P}{T_I}\int_0^t (y^*_{REF}(\nu) - y(\nu))d\nu \tag{4.45}$$

where y^*_{REF} is the water level reference for the upstream water elevation. A PI is chosen for our scenario because the integral action is fundamental to reject step disturbances, while the low pass filter is fundamental to attenuate the waves high frequencies. Let us denote by C_i the local upstream controller for water level of pool i in continuous time. The transfer function of the controller becomes

$$\begin{aligned} C_i(s) =& \frac{K_c(1+T_c s)}{T_c s} \cdot \frac{1}{1+T_f s} \\ =& \left(\tfrac{K_c}{T_c s}\right)\left(\tfrac{1+T_c s}{1+T_f s}\right). \end{aligned} \tag{4.46}$$

It can be noticed that integrators are present in both in the controller and the plant transfer function

$$\tilde{G}_i = \frac{c_{out}}{s}, \tag{4.47}$$

where $c_{out} < 0$ is the output discharge coefficient found by means of system identification procedure. Then, the phase is $-180°$ initially and so it is required a phase lead. Given the desired phase margin ϕ_m, the necessary phase lead depends on the ratio

$$\beta_m = \frac{T_f}{T_c} = \frac{1 - sin(\phi_m + \Delta\phi_m)}{1 + sin(\phi_m + \Delta\phi_m)} \tag{4.48}$$

where an additional phase lead $\Delta\phi_m$ is taken into account because it is necessary to compensate the phase drop due to the gain amplification effect. As explained in details in [15], the maximum phase lead correspond to the geometric mean frequency $w_m = \frac{1}{T_c\sqrt{\beta_m}}$ and there is a gain amplification of $A_m = \frac{1}{\sqrt{\beta_m}}$ at this frequency. Accordingly, we want to obtain the maximum phase lead at the new crossover frequency, then we choose w_m to be the frequency where

$$\left|\tilde{G}_i(jw) \cdot \left(\tfrac{K_c}{T_c jw}\right)\right|_{w=w_m} = \frac{c_{out}K_c}{w_m^2 T_c} = \frac{1}{A_m}. \tag{4.49}$$

Let us substitute $w_m = \frac{1}{T_c\sqrt{\beta_m}}$, then it holds

$$K_c = \frac{1}{c_{out} T_c \sqrt{\beta_m}}. \tag{4.50}$$

Moreover, it is possible to require a specific value M_{wave} of the gain of the controller at the frequency w_{wave},i.e.

$$\left|C_i(jw)\right|_{w=w_{wave}} \frac{K_c\sqrt{1+T_c^2 w_{wave}^2}}{T_c w_{wave}\sqrt{1+\beta_m^2 T_c^2 w_{wave}^2}} = M_{wave}. \tag{4.51}$$

This procedure is recommended to avoid wave amplification. Then, if we substitute Eq. (4.50) in Eq. (4.51) we get a sixth order polynomial in T_c

$$\alpha_m \beta_m^2 w_{wave}^2 T_c^6 + \alpha_m T_c^4 - w_{wave}^2 T_c^2 - 1 = 0 \tag{4.52}$$

where $\alpha_m = (M_{wave} c_{out} \sqrt{\beta_m w_{wave}})^2$. Once the phase margin and the controller gain at the wave frequency are given, we can solve Eq. (4.48) and (4.52). Then, we can set $T_f = \beta_m T_c$ and finally we can solve Eq. (4.50), checking that $T_c > \frac{1}{c_{out}}$. Then, the asymptotic Bode diagram of $C_1(s)\tilde{G}_1(s)$ is reported in Figure 4.6. For what concerns the feedforward transfer function, a first order Butterworth filter can be selected.

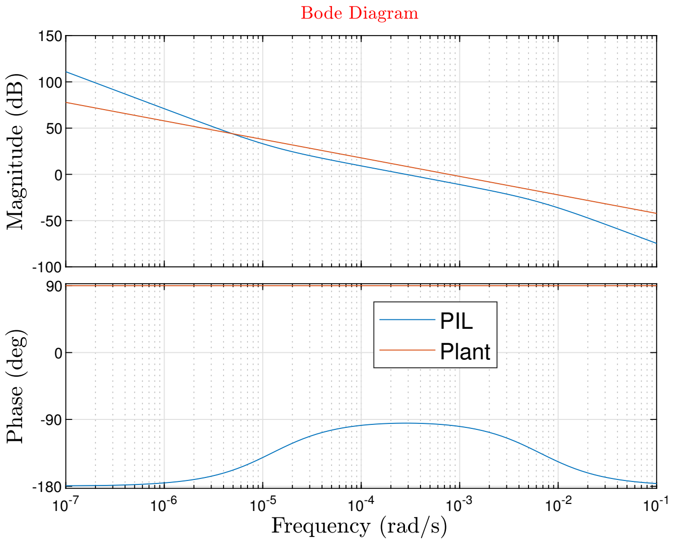


Figure 4.6: Bode plot of the open loop system and of the PIL

$$F(s) = K_B \frac{1}{\left(\frac{s}{w_0}\right)^2 + 2\varepsilon_f\left(\frac{s}{w_0}\right) + 1} \tag{4.53}$$

where $K_B > 0$, $\varepsilon_f = 0.707$, w_0 is the cutoff frequency and is chosen half the wave frequency. This filter provides a maximal flatness gain at low frequencies and so it does not dramatically change the frequency response of the system in the low frequencies.

5

Numerical results

In this chapter, we try to apply the system identification theory presented in Chapter 3 and control theory proposed in Chapter 4. Indeed, a MATLAB and Simulink simulation is proposed with the attempt to find a suitable model and control design for our case study, shown in details in Figures 5.1, 5.2. In the last years, storms, downpours and floods have become increasingly frequent and devastating occurrences, then a violent disturbance will affect the system in order to observe its effect.

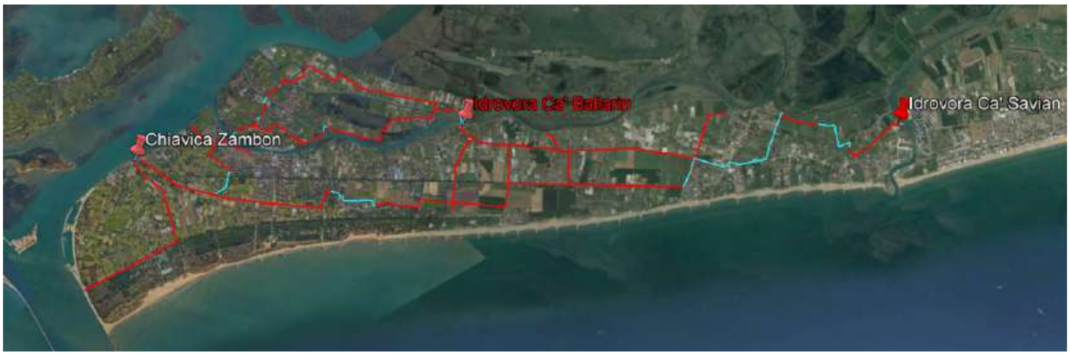


Figure 5.1: Cavallino water channel network

In our analysis, Chiavica Zambon is assumed to be closed, in order to avoid any inflow from the sea. Thus, we restrict our discussion to the pool string from G1-G22, as shown in Figure 5.2.

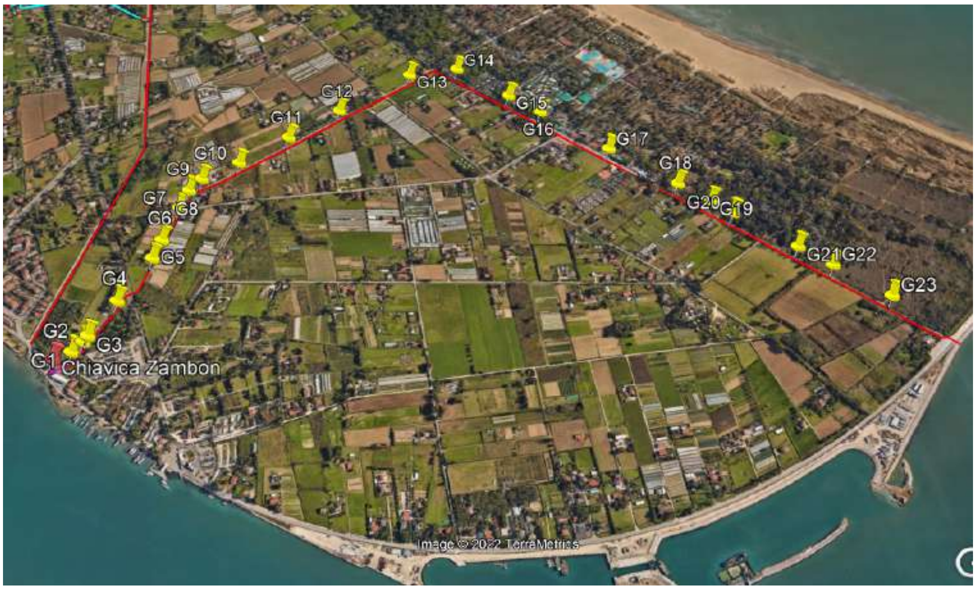


Figure 5.2: Details of a subsection of Cavallino water channel network

5.1 Data generation

The SVEs are widely used for modeling hydraulic systems but for sake of simplicity a second order model [13] is exploited for generating surface water level data. For our analysis, we use the approximated pools parameters summarized in Table 5.1, where l is the length of Cavallino pools, h_0 is the steady state water level, h_{MAX} is the maximum water level that can be reached without overflow, b and B are respectively the smaller and the bigger base of the trapezoidal section of the pools.

Usually, channels should be segmented in order to analyze separately single stretches, due to the possible geometrical variations along the channel. On the other hand, we are dealing with short and not very deep pools, then the trapezoidal section of each pool and the friction can be assumed constant and we can treat the whole pool as a single segment, without losing much accuracy. All the pools are characterized by an undershoot gate and one last weir at the downstream end, where the flow can be approximated as reported in Chapter 6.2. The width and the high of each gate are approximated to $w_g = 0.5m$ and $h_g = 0.4m$. In the attempt to obtain a close approximation of the upstream and downstream surface water level, we can compute a rough estimate of the parameters of the first order model. In particular, the side slope of each pool is assumed constant.

$Pool$	l[m]	h_0 [m]	h_{MAX} [m]	b[m]	B[m]
1	40.50	1.46	1.66	2.50	6.08
2	20.00	0.86	1.06	2.50	5.94
3	126.5	1.54	1.74	2.50	7.28
4	164.5	1.66	1.86	2.50	8.01
5	78.50	1.42	1.62	2.50	6.73
6	117.5	1.63	1.83	2.50	7.96
7	73.00	1.50	1.70	2.50	7.08
8	55.00	1.67	1.87	2.50	8.54
9	117.5	1.79	1.99	2.50	8.61
10	169.5	1.55	1.75	2.50	8.84
11	185.0	1.54	1.74	2.50	7.41
12	262.0	1.59	1.79	2.50	7.66
13	200.0	1.19	1.39	2.76	4.78
14	197.0	0.92	1.12	2.99	4.37
15	127.0	1.05	1.25	2.55	4.65
16	253.0	1.51	1.71	2.00	5.13
17	240.0	1.70	1.90	2.00	6.14
18	110.0	1.73	1.93	2.00	5.75
19	70.00	1.44	1.64	2.00	4.78
20	200.0	1.56	1.76	2.00	8.55
21	105.0	1.56	1.76	2.00	8.89
22	170.0	1.38	1.58	2.76	7.57

Table 5.1: Dataset of geometrical parameters of pools 1-22

Therefore, one has

$$s = \frac{B-b}{2h}. \tag{5.1}$$

and then the greatest base B of the trapezoidal section can be expressed in function of the slope as

$$B = 2sh + b. \tag{5.2}$$

Accordingly, the inflow of an additional water volume V affects the additional high of each pool in the following way

$$V = hl\frac{2B + 2sh}{2} \tag{5.3}$$

Then, the incremental high h can be expressed as a function of the incremental volume V as

$$h = \theta(V)V. \tag{5.4}$$

The simulation duration is $T = 500$ min, the sampling time is reasonable $T_s = 1$ min and the number of samples is $\frac{T}{T_s}$. Thus, the discrete time model for the data generation of the upstream and downstream surface water level is a second order system

$$\begin{aligned} y_U((k+1)T_s) =& y_U(kT_s) + T_s\theta_{in}Q_{in,U}((k-\tau)T_s) - T_s\frac{\theta_{in}}{10}Q_{in,U}((k-\tau-1)T_s)+ \\ &+ T_s\theta_{out}Q_{out,U}(kT_s) - T_s\frac{\theta_{out}}{10}Q_{out,U}((k-1)T_s) \\ y_D((k+1)T_s) =& y_D(kT_s) + T_s\theta_{in}Q_{in,D}(kT_s - T_s\frac{\theta_{in}}{10}Q_{in,D}((k-1)T_s)+ \\ &+ T_s\theta_{out}Q_{out,D}((k-\tau)T_s - T_s\frac{\theta_{out}}{10}Q_{out,D}((k-\tau-1)T_s)). \end{aligned} \tag{5.5}$$

where τ is a reasonable delay time reported in Table 5.2. It is assumed that all the gates are closed and there exists a difference of water level between consecutive pools, as reported from the initial conditions in Table 5.2.

From the data in Table 5.1 it can be observed that a disturbance could be more dangerous for a few pools than others, because some coefficients differs of almost one order. This means that the effects on the pools are different, because the smaller is the coefficient θ, the smaller effect on the water level increment there will be. In this simulation, all the gates are closed and the surface water level of the pools are sequentially decreasing. Thus, for Bernoulli law, as soon as the gates are opened, the water flows along the pools reach a steady state scenario, where all the pools show the same surface water level. For what concerns the boundary flow condition at the upstream and downstream, we choose a binary signal, reported in Figure 5.3. The only exception is given by the first pool that shows only outflow and no inflow. The simulation results are reported in Figures 5.4, 5.5, 5.6.

In general, it can be observed that the water level is constant initially but as

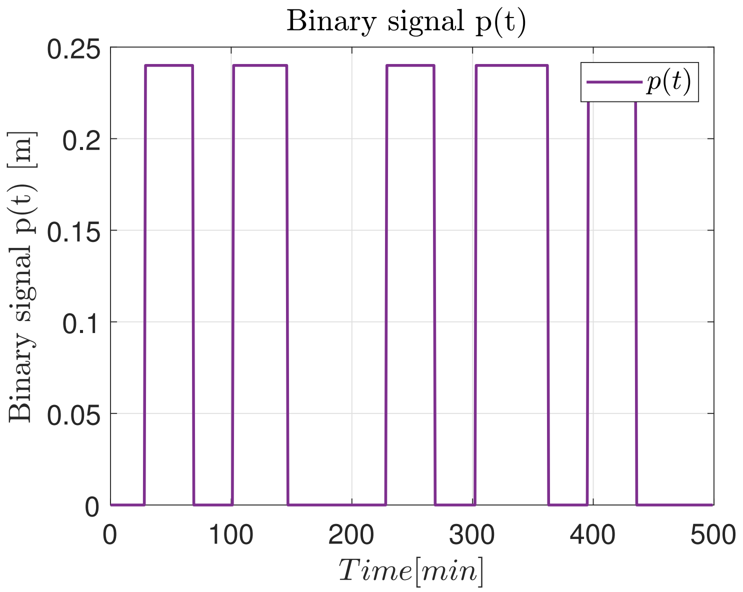
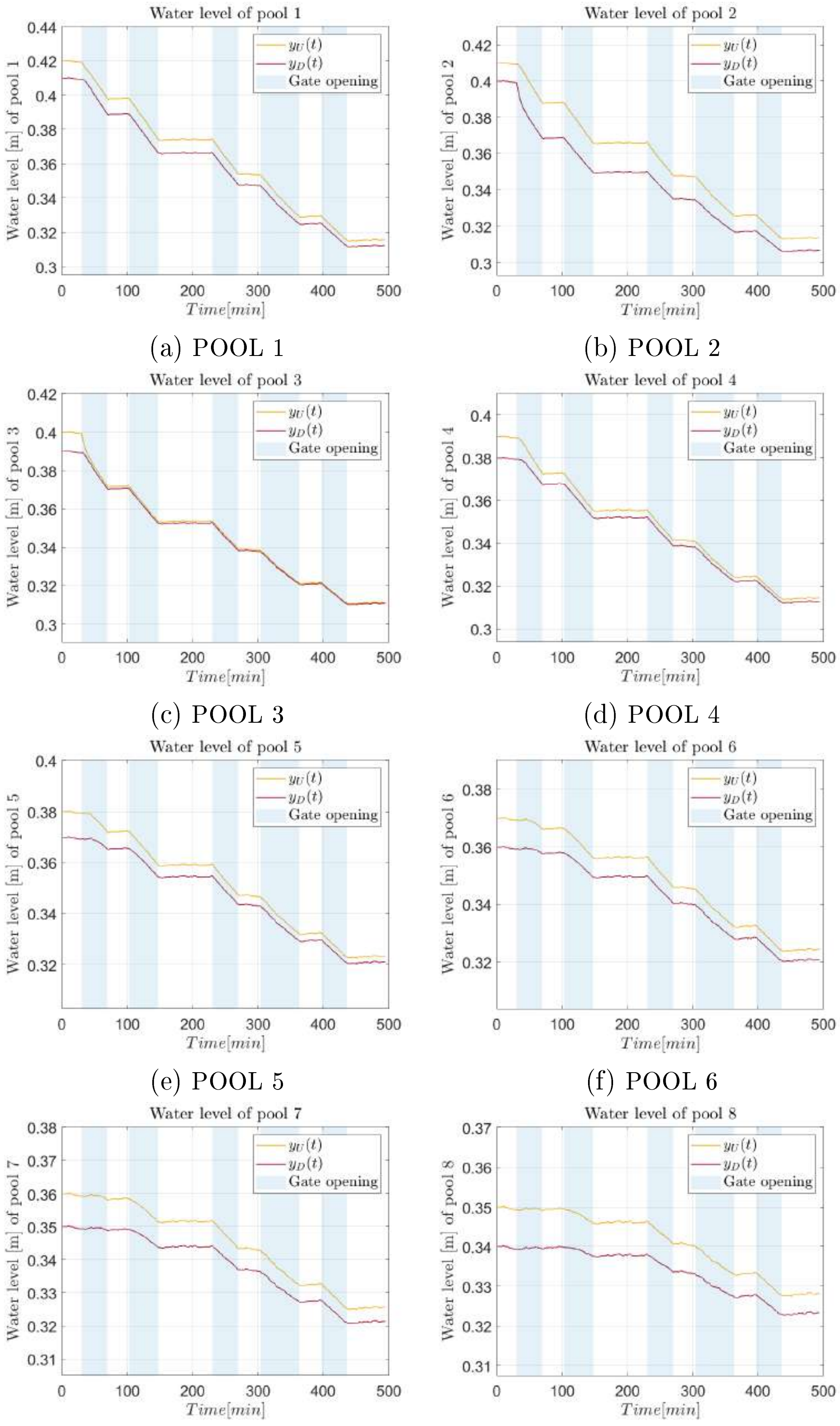


Figure 5.3: Binary signal to control the aperture of all the gates

soon as the gates are opened, the upstream water level tends to decrease while the downstream water level of a pool decreases only if the following pool has a smaller water volume. In general, the last pools show large variation of the water level due to the large coefficient θ of the outflow of the last pool. Large variations can be observed also in the first pools, due to the fact that the first pool do not have any inflow. The scenario is different if we notice the pools in the middle of the string, due to the fact that the difference between the water levels is really small. Indeed, the flow amplitude is strictly dependent to the difference between water levels of consecutive pools and to the dimension of the gates aperture.

Pool	$y = [y_D, y_U]$[m]	$\theta = [\theta_{in}, \theta_{out}]$ []	τ[min]
1	$\left[0.42, 0.41\right]$	$\left[4.07 \cdot 10^{-3}, -4.07 \cdot 10^{-3}\right]$	2
2	$\left[0.41, 0.40\right]$	$\left[8.51 \cdot 10^{-3}, -8.51 \cdot 10^{-3}\right]$	2
3	$\left[0.40, 0.39\right]$	$\left[1.09 \cdot 10^{-3}, -1.09 \cdot 10^{-3}\right]$	3
4	$\left[0.39, 0.38\right]$	$\left[7.60 \cdot 10^{-4}, -7.60 \cdot 10^{-4}\right]$	2
5	$\left[0.38, 0.37\right]$	$\left[1.89 \cdot 10^{-3}, -1.89 \cdot 10^{-3}\right]$	3
6	$\left[0.37, 0.36\right]$	$\left[1.07 \cdot 10^{-3}, -1.07 \cdot 10^{-3}\right]$	2
7	$\left[0.36, 0.35\right]$	$\left[1.94 \cdot 10^{-3}, -1.94 \cdot 10^{-3}\right]$	2
8	$\left[0.35, 0.34\right]$	$\left[2.13 \cdot 10^{-3}, -2.13 \cdot 10^{-3}\right]$	3
9	$\left[0.33, 0.32\right]$	$\left[9.89 \cdot 10^{-4}, -9.89 \cdot 10^{-4}\right]$	3
10	$\left[0.32, 0.31\right]$	$\left[6.67 \cdot 10^{-4}, -6.67 \cdot 10^{-4}\right]$	3
11	$\left[0.31, 0.30\right]$	$\left[7.30 \cdot 10^{-4}, -7.30 \cdot 10^{-4}\right]$	4
12	$\left[0.30, 0.29\right]$	$\left[4.99 \cdot 10^{-4}, -4.99 \cdot 10^{-4}\right]$	3
13	$\left[0.29, 0.28\right]$	$\left[1.05 \cdot 10^{-3}, -1.05 \cdot 10^{-3}\right]$	3
14	$\left[0.28, 0.27\right]$	$\left[1.16 \cdot 10^{-3}, -1.16 \cdot 10^{-3}\right]$	3
15	$\left[0.27, 0.26\right]$	$\left[1.70 \cdot 10^{-3}, -1.70 \cdot 10^{-3}\right]$	3
16	$\left[0.26, 0.25\right]$	$\left[7.70 \cdot 10^{-4}, -7.70 \cdot 10^{-4}\right]$	3
17	$\left[0.25, 0.24\right]$	$\left[6.79 \cdot 10^{-4}, -6.79 \cdot 10^{-4}\right]$	3
18	$\left[0.24, 0.23\right]$	$\left[1.58 \cdot 10^{-3}, -1.58 \cdot 10^{-3}\right]$	4
19	$\left[0.23, 0.22\right]$	$\left[3.00 \cdot 10^{-3}, -3.00 \cdot 10^{-3}\right]$	3
20	$\left[0.22, 0.21\right]$	$\left[5.85 \cdot 10^{-4}, -5.85 \cdot 10^{-4}\right]$	4
21	$\left[0.21, 0.20\right]$	$\left[1.07 \cdot 10^{-3}, -1.07 \cdot 10^{-3}\right]$	3
22	$\left[0.20, 0.19\right]$	$\left[7.77 \cdot 10^{-4}, -7.77 \cdot 10^{-4}\right]$	3

Table 5.2: Parameters of the data generation model



(a) POOL 1

(b) POOL 2

(c) POOL 3

(d) POOL 4

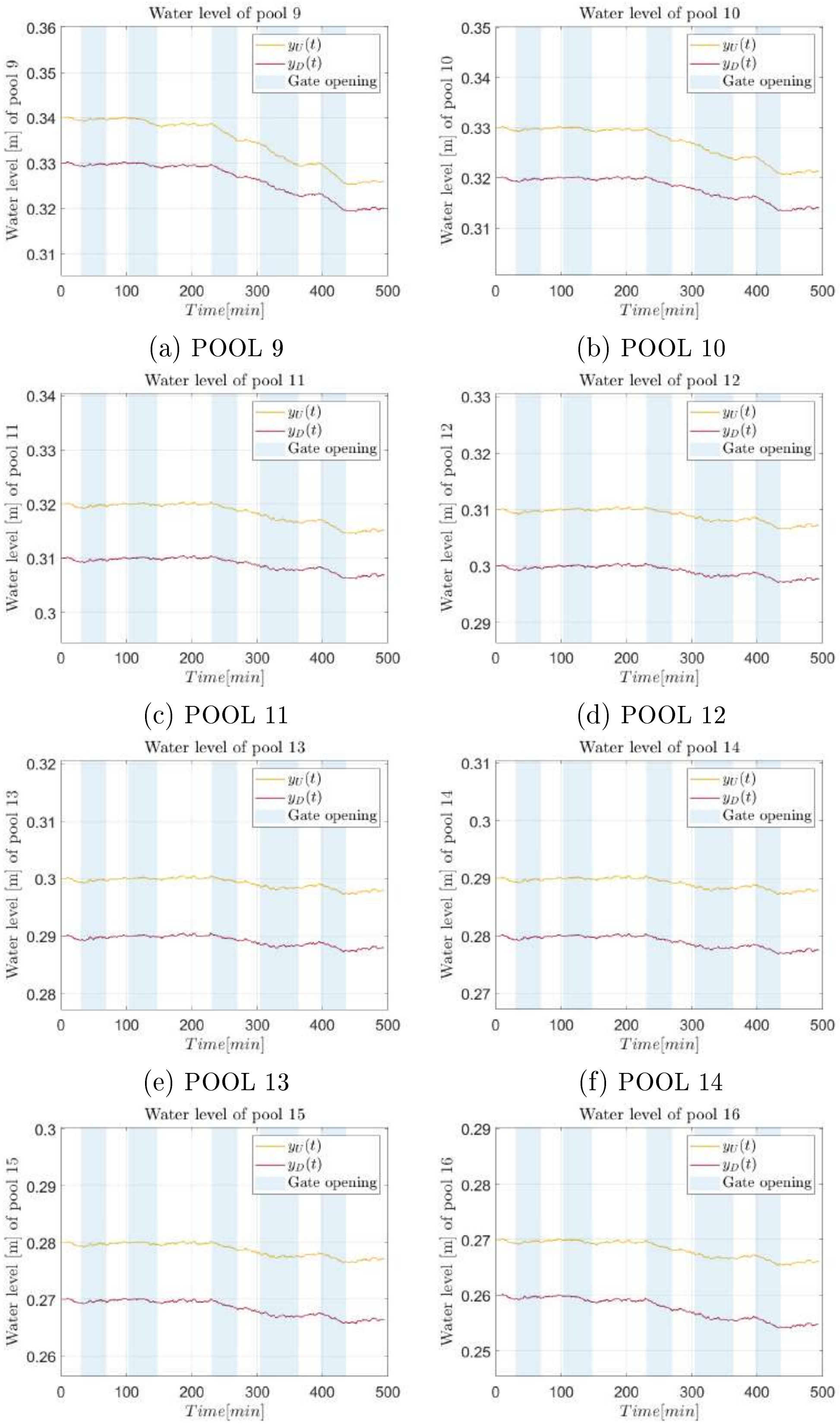
(e) POOL 5

(f) POOL 6

(g) POOL 7

(h) POOL 8

Figure 5.4: Upstream and downstream surface water level (pools from 1 to 8)



(a) POOL 9

(b) POOL 10

(c) POOL 11

(d) POOL 12

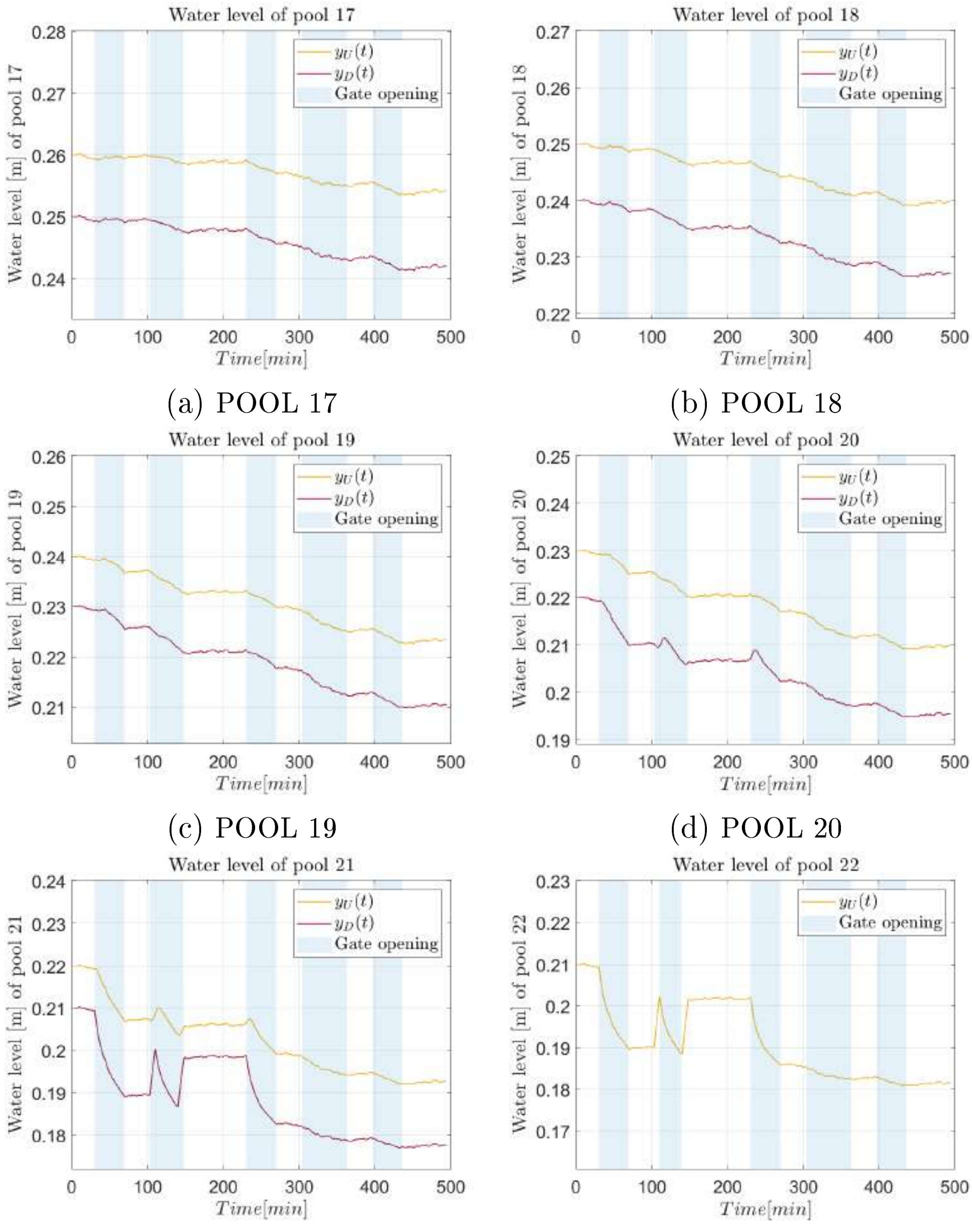
(e) POOL 13

(f) POOL 14

(g) POOL 15

(h) POOL 16

Figure 5.5: Upstream and downstream surface water level (pools from 9 to 16)



(a) POOL 17

(b) POOL 18

(c) POOL 19

(d) POOL 20

(e) POOL 21

(f) POOL 22

Figure 5.6: Upstream and downstream surface water level (pools from 17 to 22)

5.2 Identification

Discrete time models are preferred for system identification purpose, because data samples are collected in discrete time. There exist several linear model structures as reported in Chapter 3 and in particular we refer to ARX (Auto Regression with External Input) and OE (Output Error) models, where the transfer functions are rational functions of polynomials.In general, OE models show a good representation of low frequencies properties, while ARX is more sensitive to high frequencies properties. It is worth noting that waves shows higher frequencies with respect to the other pool dynamics. Moreover, accurate models in low frequencies should be enhanced with the attempt of modeling and control pools, that have the main interesting dynamics in the low frequencies because changes are slow. The order of the model is fundamental to describe the whole system dynamics. In such a case, a first order model and a second order model are proposed. The parameters are estimated using the prediction error method with a quadratic criterion, that consists on the minimization of a cost function. As reported in Chapter 3, this function is the mean squared error and vector θ contains the estimated parameters

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{250} \sum_{t=1}^{250} (y_i(t) - \hat{y}(t, \theta, \tau))^2. \tag{5.6}$$

One model is obtained for every value of the time delay and the one that show the smaller error on the validation set is selected

$$\hat{\tau} = \underset{\tau \in \{1,2,3,4,5\}}{\operatorname{argmin}} \frac{1}{249} \sum_{t=251}^{500} (y_i(t) - \hat{y}(t, \hat{\theta}, \tau))^2. \tag{5.7}$$

With this approach, all the inflow time delays have been estimated correctly. The ARX model structure allows to find the solution of the optimization problem analytically while the OE structure required an iterative searching method. In this case, parameter estimation is performed for the ARX and the OE models with 3 and with 5 parameters and it is performed by means of MATLAB System Identification toolbox. In order to improve the performance of the OE model, Levenberg-Marquardt algorithm has been employed for the cost minimization problem. The parameters for such algorithm implementation are reported in Table 5.3. Validation mean square error (MSE) and parameters estimation are reported in Tables 5.4, 5.5 and 5.6.

It can be observed that the parameters relative to the inflow are almost the

max_{iter}	γ_1	λ_0	θ_0	err_{meas}	ζ
150	10^{-3}	10^{-2}	$[0.05 - 0.05]$	10^{-3}	10^{-3}

Table 5.3: Levenberg-Marquardt algorithm parameters

$MSE_{V_{ARX}}$	θ_{ARX}	$MSE_{V_{OE}}$	θ_{OE}	J_{ARX}	J_{OE}
$2.59 \cdot 10^{-7}$	$[1, 0.000, -0.213]$	$7.78 \cdot 10^{-7}$	$[1, 0.000, -0.217]$	96.7	96.6
$2.47 \cdot 10^{-7}$	$[1, 0.496, -0.481]$	$9.66 \cdot 10^{-6}$	$[1, -0.205, -0.141]$	96.4	77.6
$1.72 \cdot 10^{-6}$	$[1, 0.016, -0.045]$	$1.12 \cdot 10^{-5}$	$[1, -0.14, -0.04]$	88.3	70.1
$4.51 \cdot 10^{-7}$	$[1, 0.044, -0.042]$	$7.66 \cdot 10^{-7}$	$[1, 0.019, -0.016]$	93.9	92.0
$6.96 \cdot 10^{-7}$	$[1, 0.111, -0.109]$	$1.41 \cdot 10^{-7}$	$[1, 0.004, -0.018]$	91.3	96.0
$1.13 \cdot 10^{-6}$	$[1, 0.065, -0.064]$	$1.48 \cdot 10^{-7}$	$[1, 0.0039, -0.013]$	87.8	95.6
$1.61 \cdot 10^{-6}$	$[1, 0.122, -0.122]$	$1.88 \cdot 10^{-7}$	$[1, 0.021, -0.026]$	82.3	93.9
$2.90 \cdot 10^{-6}$	$[1, 0.150, -0.149]$	$3.31 \cdot 10^{-6}$	$[1, 0.354, -0.353]$	66.2	63.9
$6.09 \cdot 10^{-6}$	$[1, 0.085, -0.084]$	$9.19 \cdot 10^{-6}$	$[1, 0.323, -0.320]$	31.5	15.9
$1.39 \cdot 10^{-5}$	$[1, 0.081, -0.080]$	$1.46 \cdot 10^{-5}$	$[1, 0.331, -0.329]$	-53.0	-56.0
$3.78 \cdot 10^{-5}$	$[1, 0.152, -0.161]$	$9.42 \cdot 10^{-6}$	$[1, 0.222, -0.229]$	-326	-112
$1.06 \cdot 10^{-4}$	$[1, 0.226, -0.226]$	$2.27 \cdot 10^{-5}$	$[1, 0.363, -0.368]$	-1061	-437
$2.1 \cdot 10^{-5}$	$[1, 0.546, -0.546]$	$7.30 \cdot 10^{-5}$	$[1, 2.12, -2.12]$	-1369	-1183
$9.58 \cdot 10^{-5}$	$[1, 0.412, -0.412]$	$1.47 \cdot 10^{-5}$	$[1, 1.090, -1.09]$	-854	-510
$3.60 \cdot 10^{-5}$	$[1, 0.350, -0.349]$	$1.59 \cdot 10^{-5}$	$[1, 1.950, -1.95]$	-355	-437
$1.14 \cdot 10^{-5}$	$[1, 0.118, -0.117]$	$2.63 \cdot 10^{-6}$	$[1, 0.022, -0.029]$	-168	-78.8
$5.94 \cdot 10^{-6}$	$[1, 0.077, -0.076]$	$1.24 \cdot 10^{-7}$	$[1, 0.0006, -0.0006]$	-38.2	70.9
$2.80 \cdot 10^{-6}$	$[1, 0.115, -0.114]$	$4.51 \cdot 10^{-6}$	$[1, -0.053, -0.053]$	52.5	-13.3
$7.96 \cdot 10^{-7}$	$[1, 0.160, -0.160]$	$6.19 \cdot 10^{-7}$	$[1, 0.002, -0.002]$	73.8	71.0
$5.08 \cdot 10^{-7}$	$[1, 0.028, -0.029]$	$1.25 \cdot 10^{-6}$	$[1, 0.003, -0.003]$	75.3	63.8
$5.82 \cdot 10^{-7}$	$[1, 0.052, -0.053]$	$5.86 \cdot 10^{-6}$	$[1, 0.0008, -0.011]$	64.4	17.1
$2.45 \cdot 10^{-6}$	$[1, 0.036, -0.037]$	$9.68 \cdot 10^{-6}$	$[1, 0.023, -0.033]$	25.0	-49.0

Table 5.4: Validation results for ARX, OE models with 3 parameters for upstream water level models

$MSE_{V_{ARX}}$	θ_{ARX}	$MSE_{V_{OE}}$	θ_{OE}	J_{ARX}	J_{OE}
$9.68 \cdot 10^{-6}$	$[1, -0.195, -0.006]$	$6.01 \cdot 10^{-7}$	$[1, -0.202, -0.139]$	94.6	79.5
$1.02 \cdot 10^{-5}$	$[1, -0.078, -0.023]$	$1.35 \cdot 10^{-5}$	$[1, -0.157, -0.045]$	71.9	67.9
$4.82 \cdot 10^{-7}$	$[1, -0.019, -0.006]$	$1.75 \cdot 10^{-6}$	$[1, -0.026, -0.017]$	93.76	88.11
$4.30 \cdot 10^{-6}$	$[1, -0.012, -0.0005]$	$5.63 \cdot 10^{-7}$	$[1, -0.007, -0.015]$	80.4	92.9
$1.42 \cdot 10^{-5}$	$[1, -0.006, -0.0004]$	$1.29 \cdot 10^{-7}$	$[1, 0.007, -0.016]$	58.4	96.0
$1.51 \cdot 10^{-5}$	$[1, -0.005, 0.0004]$	$3.31 \cdot 10^{-6}$	$[1, 0.179, -0.271]$	51.0	77.1
$1.32 \cdot 10^{-5}$	$[1, -0.003, -0.0006]$	$2.49 \cdot 10^{-6}$	$[1, 0.272, -0.271]$	42.3	75.0
$8.57 \cdot 10^{-6}$	$[1, -0.002, -0.0003]$	$4.54 \cdot 10^{-6}$	$[1, 0.226, -0.224]$	29.2	48.8
$4.74 \cdot 10^{-6}$	$[1, -0.0009, 0.0003]$	$1.03 \cdot 10^{-5}$	$[1, 0.234, -0.232]$	20.4	-16.7
$2.12 \cdot 10^{-6}$	$[1, -0.0006, 0.0003]$	$2.46 \cdot 10^{-5}$	$[1, 0.552, -0.5503]$	16.7	-182
$6.93 \cdot 10^{-7}$	$[1, -0.0007, 0.0006]$	$6.92 \cdot 10^{-5}$	$[1, 0.789, -0.788]$	13.5	-761
$3.45 \cdot 10^{-7}$	$[1, -0.0005, 0.0003]$	$8.66 \cdot 10^{-5}$	$[1, 2.20, -2.20]$	18.2	-1195
$2.35 \cdot 10^{-7}$	$[1, -0.0005, 0.0003]$	$6.93 \cdot 10^{-6}$	$[1, 0.253, -0.262]$	22	-320
$2.19 \cdot 10^{-7}$	$[1, -0.0006, 0.0003]$	$5.15 \cdot 10^{-6}$	$[1, 0.097, -0.107]$	30	-234
$1.23 \cdot 10^{-7}$	$[1, -0.0007, 0.0003]$	$7.17 \cdot 10^{-6}$	$[1, 0.609, -0.607]$	42	-215
$2.16 \cdot 10^{-7}$	$[1, -0.0009, 0.0003]$	$1.23 \cdot 10^{-7}$	$[1, -0.0004, -0.0005]$	58.6	68.7
$1.42 \cdot 10^{-7}$	$[1, -0.002, -0.0004]$	$5.19 \cdot 10^{-6}$	$[1, -0.005, -0.005]$	75.6	-47.7
$2.88 \cdot 10^{-7}$	$[1, -0.003, 0.0009]$	$3.07 \cdot 10^{-7}$	$[1, -0.002, -0.002]$	76.0	75.2
$1.53 \cdot 10^{-6}$	$[1, -0.004, 0.0006]$	$1.01 \cdot 10^{-6}$	$[1, -0.003, -0.003]$	59.6	67.1
$7.35 \cdot 10^{-6}$	$[1, 0.0005, -0.004]$	$8.79 \cdot 10^{-7}$	$[1, 0.091, -0.092]$	10.7	69.1
$8.79 \cdot 10^{-5}$	$[1, 0.008, -0.014]$	$2.51 \cdot 10^{-6}$	$[1, 0.035, -0.041]$	-142	33.5

Table 5.5: Validation results for ARX, OE models with 3 parameters for downstream water level models

opposite of those related to the outflow, that makes sense from a physical point of view. Moreover, MSE values on the training set are almost the same order of magnitude. On the other hand, there is an improvement of the MSE computed in the validation set. As expected, the performance of the second order model is more accurate than the one obtained from a first order model. In particular, it can be enhance that for large variation in water level, second order model

$MSE_{V_{ARX}}$	$MSE_{V_{OE}}$	θ_{LM}	$MSE_{V_{LM}}$	J_{LM}
2.59×10^{-7}	2.77×10^{-7}	$[0, -0.220]$	4.46×10^{-7}	95.7
2.49×10^{-7}	9.66×10^{-7}	$[0.419, -0.432]$	6.22×10^{-7}	94.3
1.71×10^{-6}	1.12×10^{-5}	$[0.070, -0.0617]$	3.32×10^{-7}	94.8
4.51×10^{-7}	7.66×10^{-7}	$[0.034, -0.037]$	2.91×10^{-7}	95.0
6.95×10^{-7}	1.41×10^{-7}	$[0.094, -0.095]$	3.52×10^{-7}	93.8
1.31×10^{-6}	1.48×10^{-7}	$[0.054, -0.054]$	3.78×10^{-7}	92.9
1.61×10^{-6}	1.88×10^{-7}	$[0.0958, -0.096]$	4.71×10^{-7}	90.4
2.90×10^{-6}	3.31×10^{-6}	$[0.099, -0.099]$	8.44×10^{-7}	81.7
6.08×10^{-6}	9.19×10^{-6}	$[0.050, -0.050]$	2.70×10^{-7}	85.5
1.40×10^{-5}	1.45×10^{-5}	$[0.016, -0.016]$	5.80×10^{-6}	0.83
3.78×10^{-5}	9.42×10^{-6}	$[0.050, -0.050]$	1.83×10^{-6}	5.4
1.05×10^{-4}	2.27×10^{-5}	$[0.050, -0.050]$	5.58×10^{-6}	-168
9.58×10^{-5}	7.30×10^{-5}	$[0.050, -0.050]$	2.79×10^{-7}	20.2
3.60×10^{-5}	1.47×10^{-5}	$[0.050, -0.050]$	2.47×10^{-7}	20.4
1.14×10^{-5}	1.59×10^{-5}	$[0.050, -0.050]$	4.68×10^{-7}	7.21
5.94×10^{-6}	2.63×10^{-6}	$[0.050, -0.050]$	8.80×10^{-7}	-3.87
2.80×10^{-6}	1.23×10^{-7}	$[0.019, -0.02]$	5.52×10^{-7}	38.3
7.96×10^{-7}	4.51×10^{-6}	$[0.049, -0.050]$	2.73×10^{-7}	71.9
5.07×10^{-7}	6.19×10^{-7}	$[0.049, -0.051]$	1.02×10^{-7}	62.8
5.81×10^{-7}	1.25×10^{-6}	$[0.021, -0.022]$	1.39×10^{-7}	61.7
1.08×10^{-6}	5.86×10^{-6}	$[0.061, -0.060]$	2.39×10^{-7}	83.1
2.45×10^{-6}	9.68×10^{-6}	$[0.043, -0.043]$	3.19×10^{-7}	72.8

Table 5.6: Validation results for ARX, OE with 5 parameters and LM with 2 parameters for upstream water level

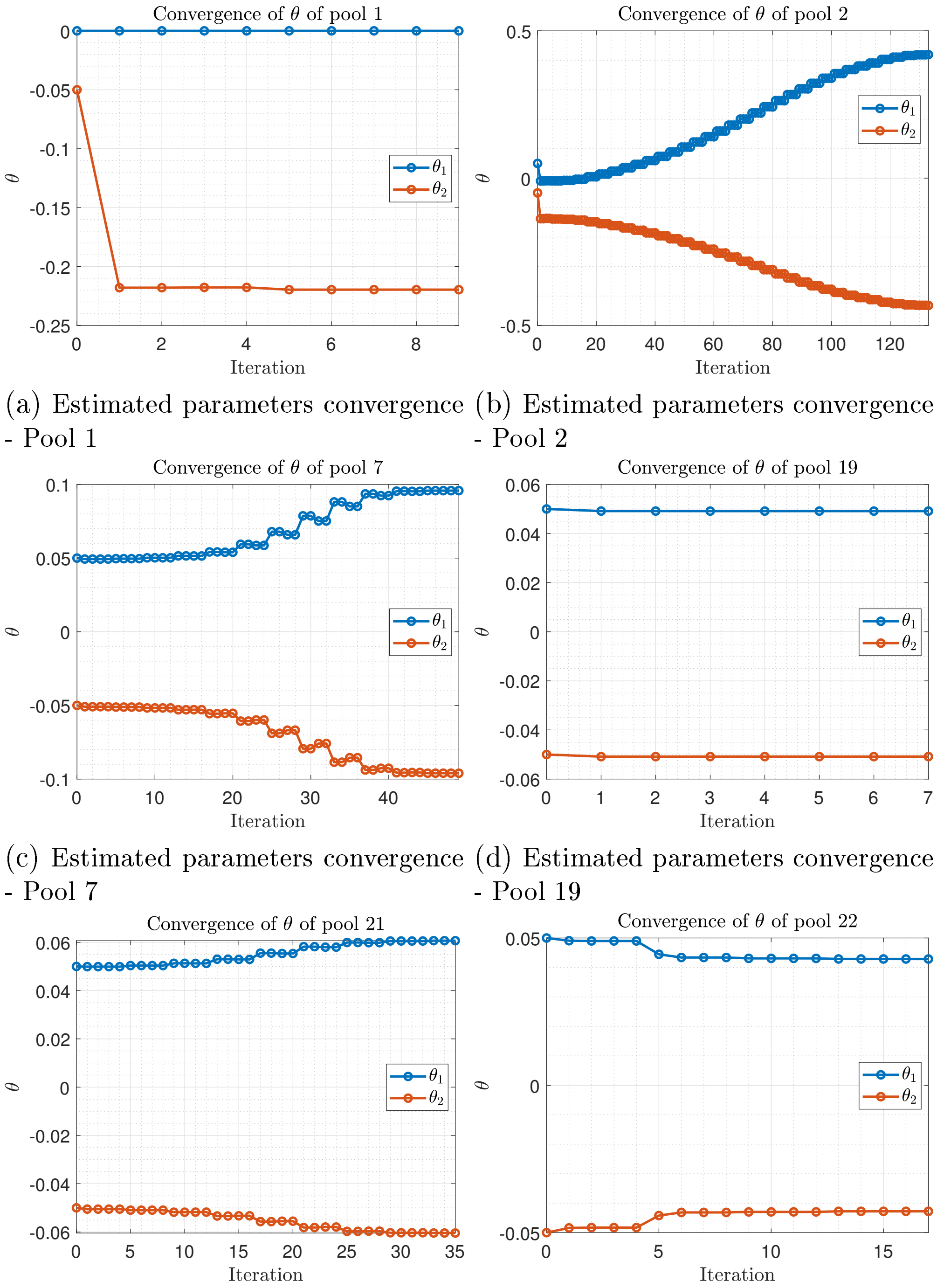
perform better than first order ones. Thus, for estimate purpose, OE second order model could represent the best solution. On the other hand, the difference in performance is minimal. Moreover, it can be highlighted the performance of the OE model with Levenberg-Marquardt approach. Indeed, starting from

$MSE_{V_{ARX}}$	$MSE_{V_{OE}}$	θ_{LM}	$MSE_{V_{LM}}$	J_{LM}
6.00×10^{-7}	8.69×10^{-6}	$[0, -0.220]$	1.55×10^{-6}	91.3
1.03×10^{-5}	1.35×10^{-5}	$[0.419, -0.432]$	3.26×10^{-7}	94.9
4.82×10^{-7}	1.75×10^{-6}	$[0.070, -0.0617]$	8.87×10^{-7}	91.5
4.29×10^{-6}	5.63×10^{-7}	$[0.034, -0.037]$	3.19×10^{-7}	94.7
1.42×10^{-5}	1.30×10^{-7}	$[0.094, -0.095]$	3.74×10^{-7}	93.2
1.51×10^{-5}	3.31×10^{-6}	$[0.054, -0.054]$	4.31×10^{-7}	91.7
1.32×10^{-5}	2.49×10^{-6}	$[0.0958, -0.096]$	5.94×10^{-7}	87.7
8.57×10^{-6}	4.54×10^{-6}	$[0.099, -0.099]$	2.78×10^{-7}	87.2
4.74×10^{-6}	1.03×10^{-5}	$[0.050, -0.050]$	4.44×10^{-6}	22.7
2.12×10^{-6}	2.46×10^{-5}	$[0.016, -0.016]$	2.25×10^{-6}	13.9
6.93×10^{-7}	6.92×10^{-5}	$[0.050, -0.050]$	6.08×10^{-6}	-157
3.45×10^{-7}	8.66×10^{-5}	$[0.050, -0.050]$	2.74×10^{-7}	26.7
2.35×10^{-7}	6.93×10^{-6}	$[0.050, -0.050]$	2.45×10^{-7}	20.4
2.19×10^{-7}	5.15×10^{-6}	$[0.050, -0.050]$	4.25×10^{-7}	3.26
2.37×10^{-7}	7.17×10^{-6}	$[0.050, -0.050]$	8.69×10^{-7}	-10.5
1.42×10^{-7}	5.18×10^{-6}	$[0.050, -0.050]$	6.80×10^{-7}	26.2
2.88×10^{-7}	3.07×10^{-7}	$[0.019, -0.02]$	3.56×10^{-7}	61.1
1.53×10^{-6}	1.01×10^{-6}	$[0.049, -0.050]$	2.86×10^{-7}	75.9
7.35×10^{-6}	8.80×10^{-7}	$[0.049, -0.051]$	1.51×10^{-6}	59.6
7.36×10^{-6}	8.79×10^{-7}	$[0.021, -0.022]$	2.36×10^{-7}	84.0
8.80×10^{-7}	3.33×10^{-5}	$[0.061, -0.060]$	2.93×10^{-7}	77.1

Table 5.7: Validation results for ARX, OE with 5 parameters and LM with 2 parameters for downstream water level

suitable initial condition, the estimation is much more accurate than the simple OE performance. For this reason, the estimated parameters are also employed for the continuous time model in the control scheme implementation. In particular, the convergence of the parameters is reported in Figure 5.7.

It can be observed that some iteration search procedure required the maximum number of iterations to converge while other reach the optimal solution in



(a) Estimated parameters convergence - Pool 1

(b) Estimated parameters convergence - Pool 2

(c) Estimated parameters convergence - Pool 7

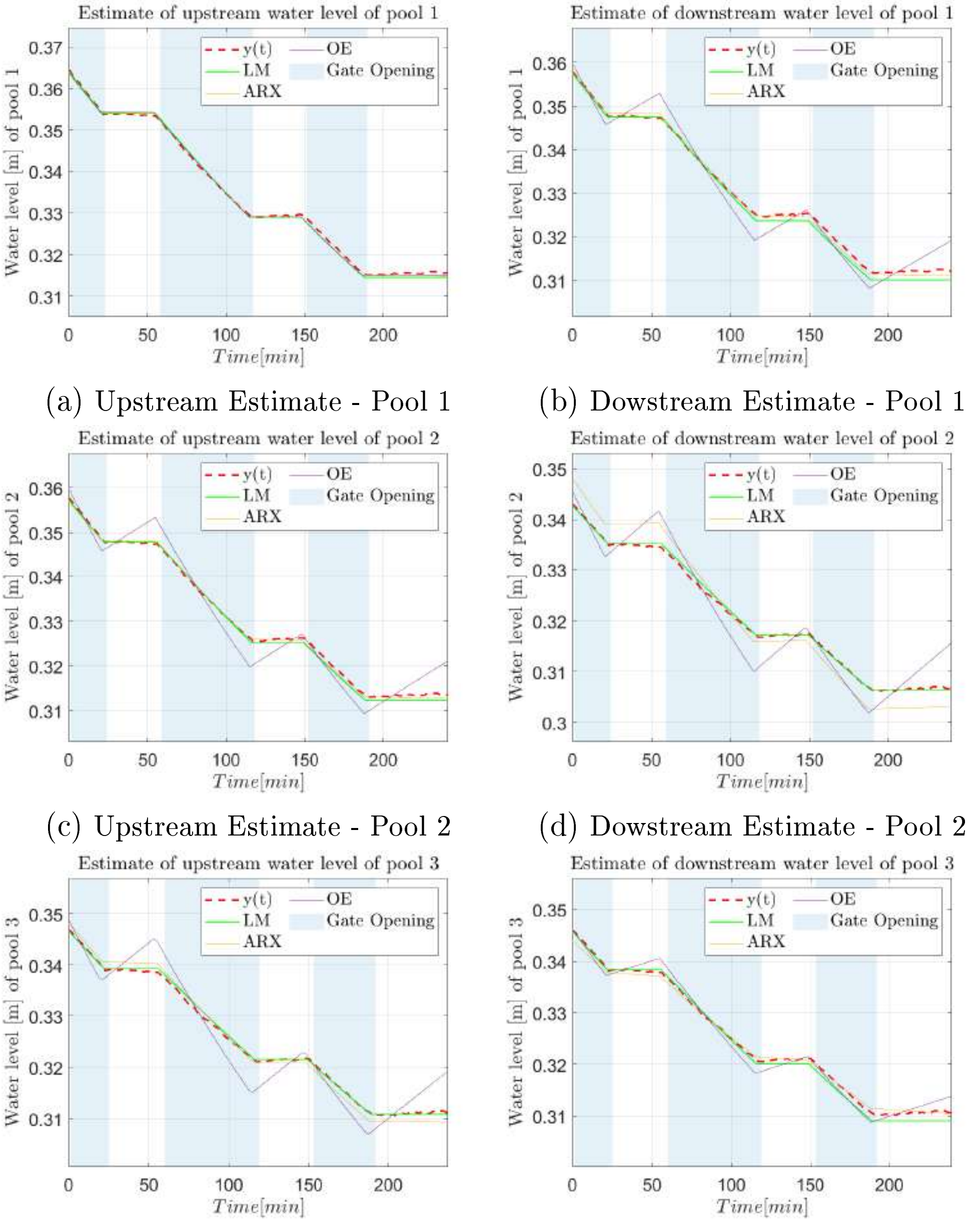
(d) Estimated parameters convergence - Pool 19

(e) Estimated parameters convergence - Pool 21

(f) Estimated parameters convergence - Pool 22

Figure 5.7: Convergence of parameter estimation with Levenberg-Marquardt algorithm for some relevant results.

a few iterations. Instead, for control purpose, these first order system estimated parameters are sufficient to have a good approximation of the string of pools control. In conclusion, validation results can be observed in Figures 5.8, 5.9, 5.10, 5.11, 5.12, 5.13.

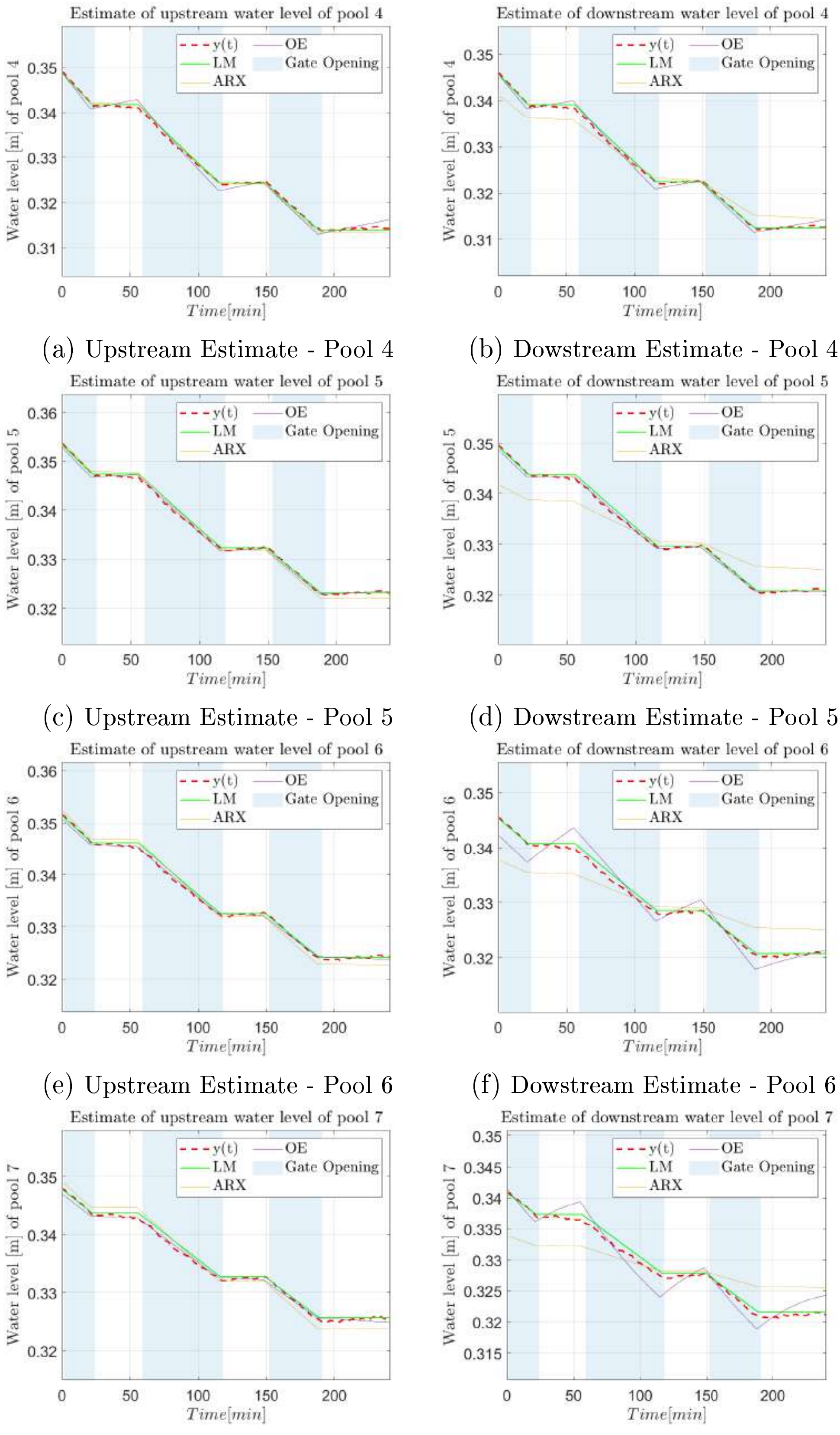


(a) Upstream Estimate - Pool 1 (b) Dowstream Estimate - Pool 1

(c) Upstream Estimate - Pool 2 (d) Dowstream Estimate - Pool 2

(e) Upstream Estimate - Pool 3 (f) Dowstream Estimate - Pool 3

Figure 5.8: Upstream and downstream estimate surface water level (Pools from 1 to 3)



(a) Upstream Estimate - Pool 4

(b) Dowstream Estimate - Pool 4

(c) Upstream Estimate - Pool 5

(d) Dowstream Estimate - Pool 5

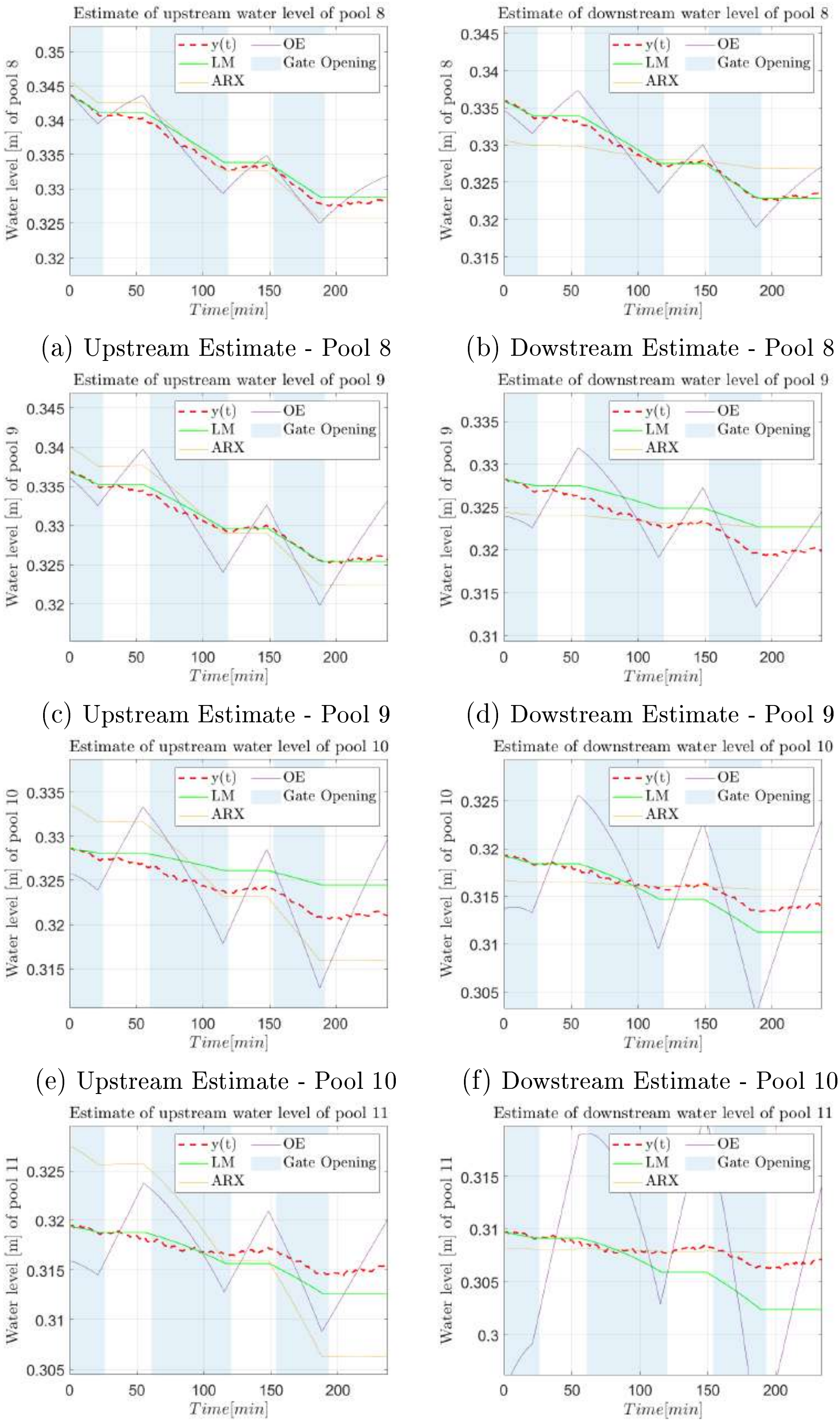
(e) Upstream Estimate - Pool 6

(f) Dowstream Estimate - Pool 6

(g) Upstream Estimate - Pool 7

(h) Dowstream Estimate - Pool 7

Figure 5.9: Upstream and downstream estimate surface water level (Pools from 4 to 7)



(a) Upstream Estimate - Pool 8

(b) Dowstream Estimate - Pool 8

(c) Upstream Estimate - Pool 9

(d) Dowstream Estimate - Pool 9

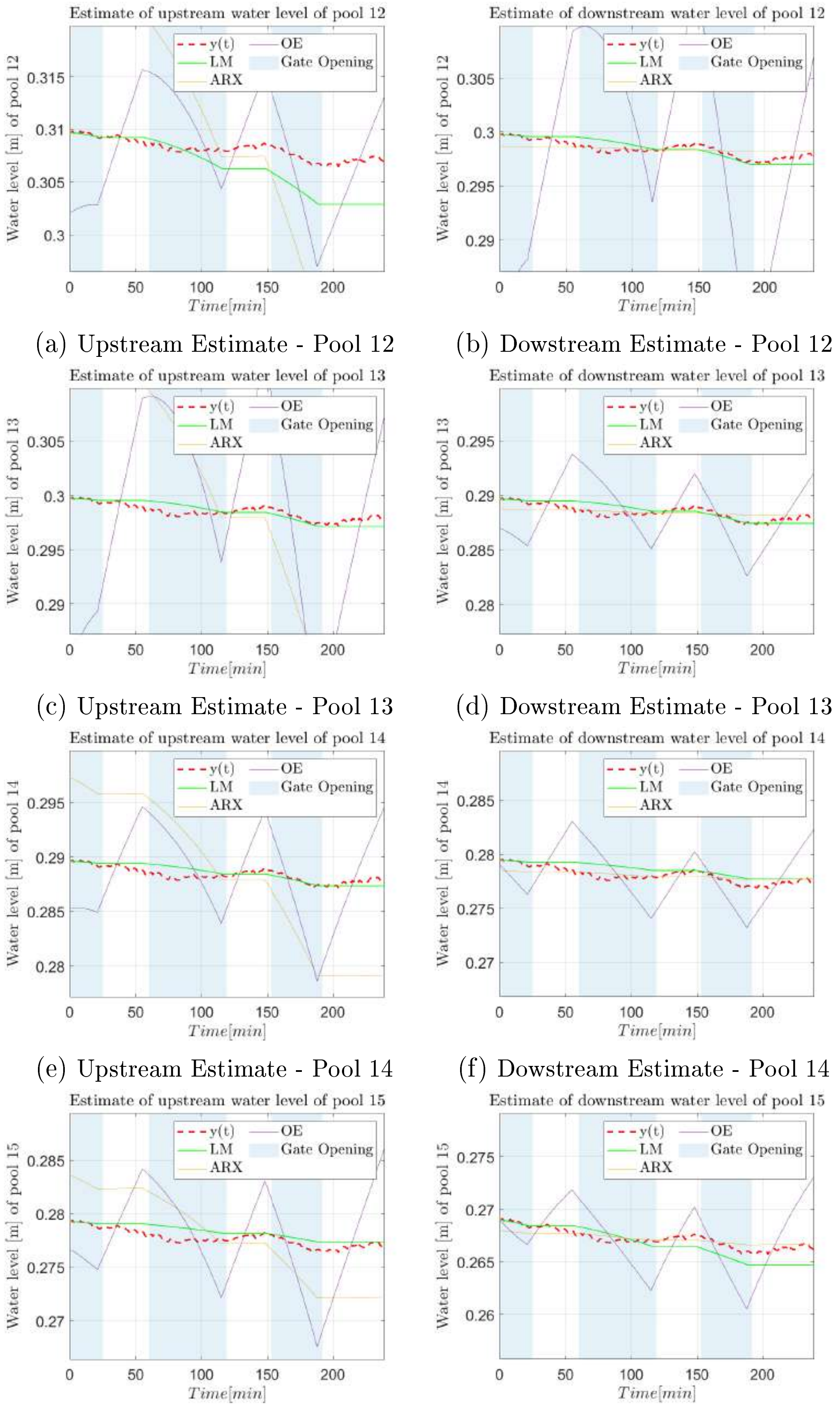
(e) Upstream Estimate - Pool 10

(f) Dowstream Estimate - Pool 10

(g) Upstream Estimate - Pool 11

(h) Dowstream Estimate - Pool 11

Figure 5.10: Upstream and downstream estimate surface water level (Pools from 8 to 11)



(a) Upstream Estimate - Pool 12

(b) Dowstream Estimate - Pool 12

(c) Upstream Estimate - Pool 13

(d) Dowstream Estimate - Pool 13

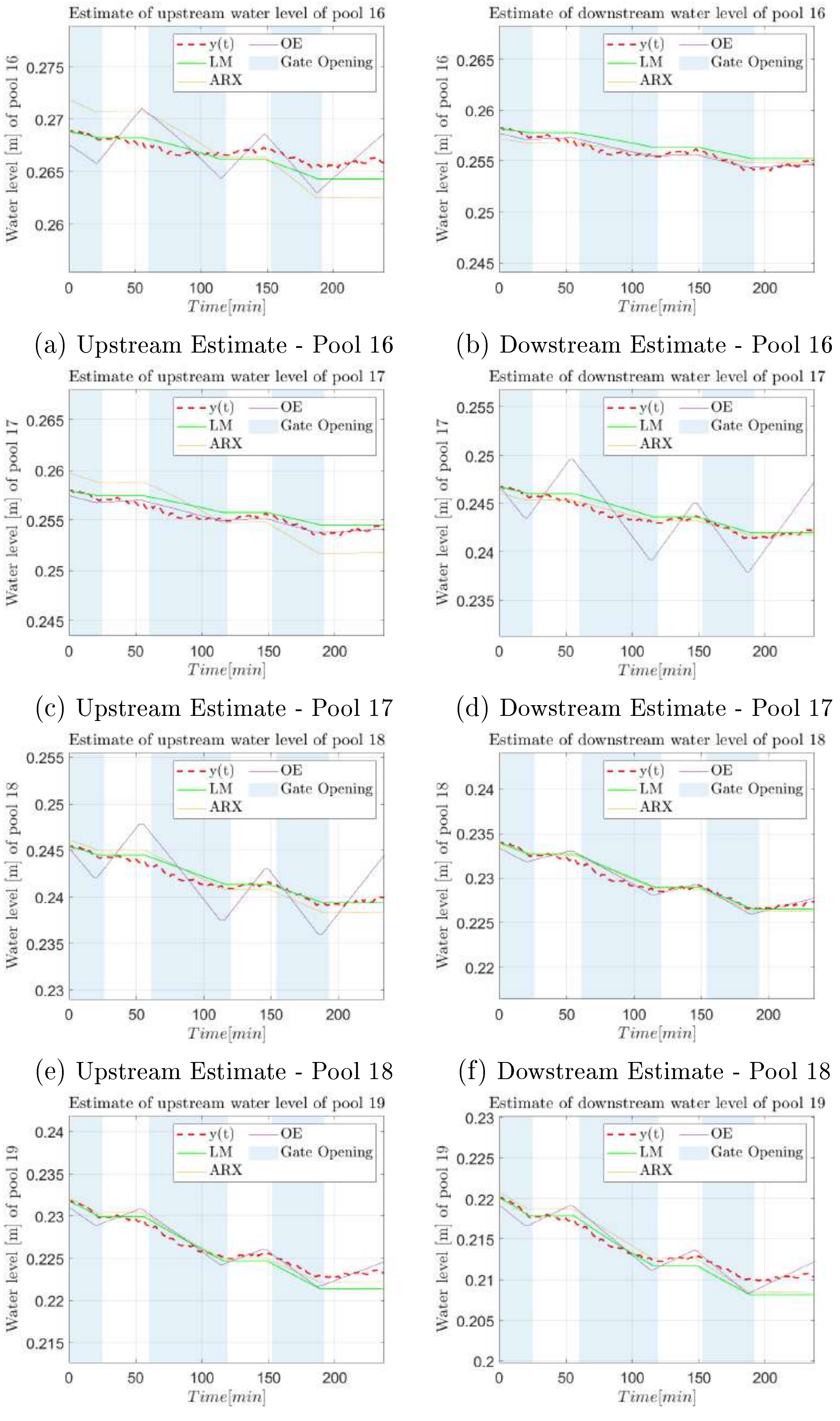
(e) Upstream Estimate - Pool 14

(f) Dowstream Estimate - Pool 14

(g) Upstream Estimate - Pool 15

(h) Dowstream Estimate - Pool 15

Figure 5.11: Upstream and downstream estimate surface water level (Pools from 12 to 15)



(a) Upstream Estimate - Pool 16

(b) Dowstream Estimate - Pool 16

(c) Upstream Estimate - Pool 17

(d) Dowstream Estimate - Pool 17

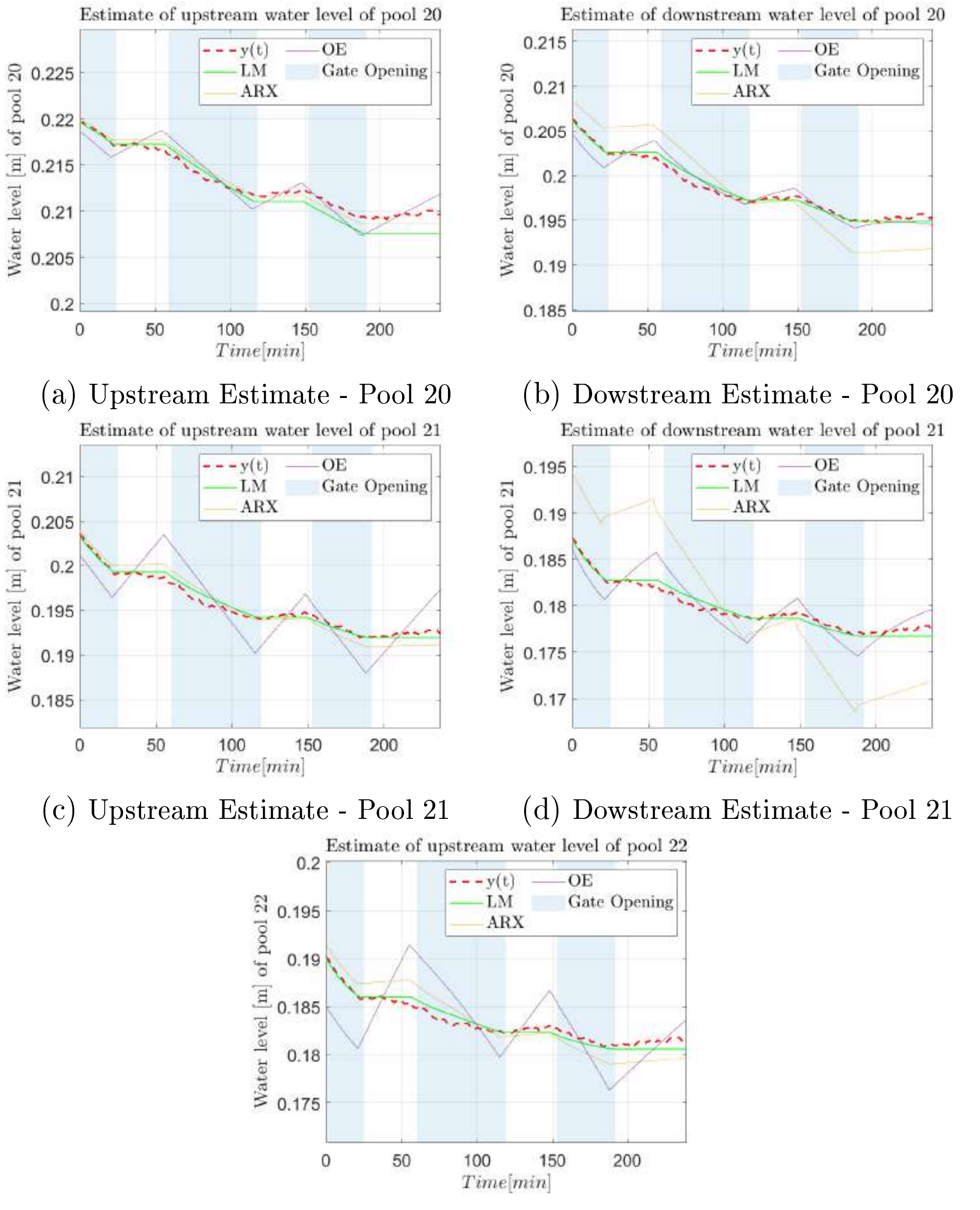
(e) Upstream Estimate - Pool 18

(f) Dowstream Estimate - Pool 18

(g) Upstream Estimate - Pool 19

(h) Dowstream Estimate - Pool 19

Figure 5.12: Upstream and downstream estimate surface water level (Pools from 16 to 19)



(a) Upstream Estimate - Pool 20

(b) Dowstream Estimate - Pool 20

(c) Upstream Estimate - Pool 21

(d) Dowstream Estimate - Pool 21

(e) Upstream Estimate - Pool 22

Figure 5.13: Upstream and downstream estimate surface water level (Pools from 20 to 22)

5.3 Control scheme implementation

The knowledge of literature about the open-channel irrigation systems is fundamental to delineate and tackle the water delivery problem that affects the Cavallino water channel network. In general, our scenario presents some differences with respect to the irrigation system. When we deal with an irrigation systems we assume that there exists a string of pools that extend on different high levels and are filled by means of an upstream tank. In this case the main target is to satisfy the water request that comes from the downstream pool acting on the system as negative disturbances on the last pool. In our scenario, we assume a steady state initial condition where all the gates are opened with the exception of the ones upstream and downstream of the string of pools. Then, we assume to start with the same initial conditions for all the pools. Another difference with irrigation systems is that positive perturbations act on all the pools at the same time, filling the pools with different volume of water. It is worth noting that, for the Bernoulli law, the flow strictly depends on the water level difference between two consecutive pools when we deal with undershoot gates. Moreover, distant downstream control is widely used in irrigation systems; whereas, we prefer to use local upstream control configuration. Indeed, this approach allows to control the output gates of each pool in order to reach the desired pool water level. The automatic controller tuning routine implemented in Chapter 4 allows to find suitable PIL parameters, that are reported in Table 5.8, once we choose the phase margin $\phi_m = 80^\circ$ and the gain of the controller $M_m = -10db$ at the wave frequency $w_{wave} = 0.015\frac{rad}{s}$, while the cutoff frequency of the butterworth filter for the feedforward action is set equal to $\frac{w_{wave}}{2}$ and its gain $K_B = 0.75$

According to the Bernoulli law, the flow control input depends on the water level difference of two consecutive pools. This can be translated into an additional saturation of the control output. Moreover, as soon as the water level of pool i is lower than the water level of pool $i+1$, the control input vanishes, that is the output gate of pool i is closed. Due to these specifics, the control becomes much more slower than what is expected to be. Feedforward is fundamental to avoid the error propagation along the string of pools and to have a better performance in terms of settling time. Then, in order to avoid floods it is fundamental a change of reference with respect to the steady state reference. According to the database in Table 5.1, the maximum water level for each pool is 0.2 m higher than the steady state condition. In our scenario, the steady state condition corresponds

$Pool$	K_c	T_c	T_f
1	-0.222	$2.53 \cdot 10^4$	48.3
2	-0.206	$1.31 \cdot 10^4$	24.9
3	-0.320	$6.57 \cdot 10^4$	125
4	-0.370	$8.16 \cdot 10^4$	155
5	-0.265	$4.56 \cdot 10^4$	86.9
6	-0.323	$6.64 \cdot 10^4$	126
7	-0.263	$4.49 \cdot 10^4$	85.6
8	-0.256	$4.19 \cdot 10^4$	79.9
9	-0.333	$6.97 \cdot 10^4$	133
10	-0.390	$8.79 \cdot 10^4$	168
11	-0.380	$8.35 \cdot 10^4$	159
12	-0.443	$1.04 \cdot 10^5$	198
13	-0.325	$6.73 \cdot 10^4$	128
14	-0.313	$6.30 \cdot 10^4$	120
15	-0.274	$4.92 \cdot 10^4$	93.9
16	-0.368	$8.09 \cdot 10^4$	154
17	-0.387	$8.70 \cdot 10^4$	165
18	-0.281	$5.16 \cdot 10^4$	98.4
19	-0.235	$3.25 \cdot 10^4$	61.9
20	-0.413	$9.48 \cdot 10^4$	180
21	-0.322	$6.63 \cdot 10^4$	126
22	-0.366	$8.05 \cdot 10^4$	153

Table 5.8: PIL parameters

to the water level 0 m, while the chosen disturbance amplitude is

$$d_i(t) = B_i l_i \cdot 5 \times 10^{-4} \frac{m^3}{s} \tag{5.8}$$

where B_i is the greatest base of the trapezoidal pool section and l_i is the length of the pool. The control scheme has been implemented in Simulink as shown in

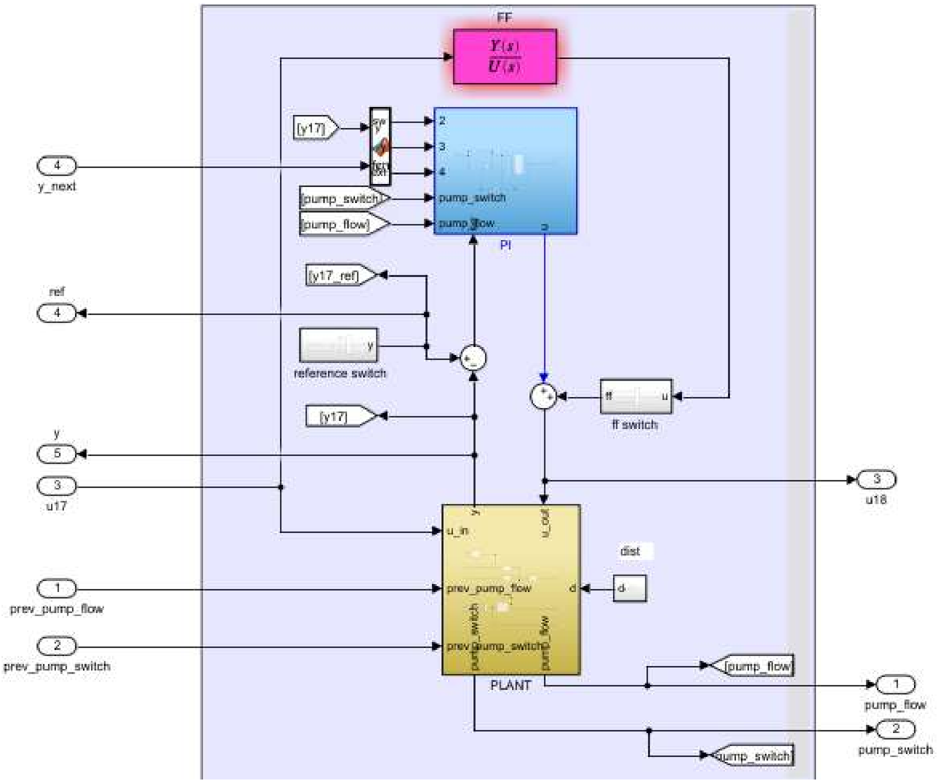


Figure 5.14: Zoom on a single pool model

Figures 5.14, 5.15, 5.16.

The time simulation is T=108000 s≈ 30 hr, while the disturbance act on the system from $\frac{T}{3}$ to $\frac{T}{3}+360s$, that is a reasonable choice for a violent rain perturbation that lasts one hour. The change of reference is chosen with different amplitude for the pools in series. The reference of the upstream pools are chosen smaller in module with respect to the downstream ones. In particular, the references occur 10 hours before the rain starts, if the forecast is known with a high probability, otherwise the references amplitudes should be reduced.

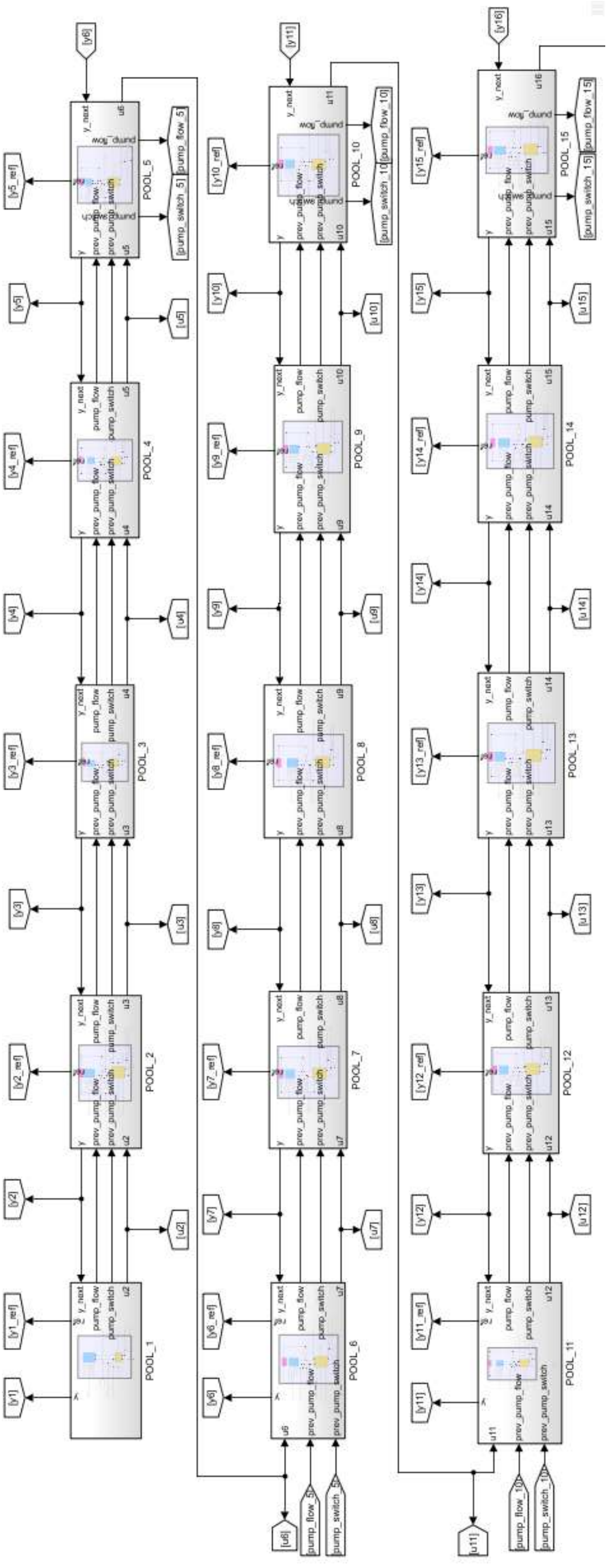


Figure 5.15: Simulink control scheme

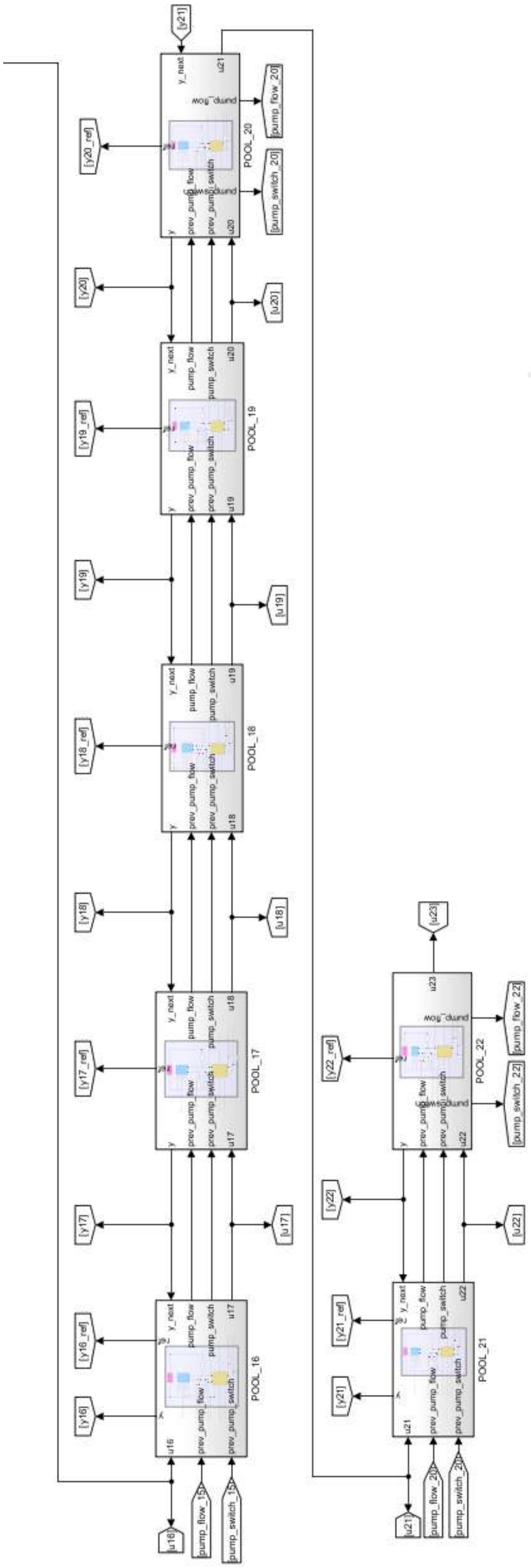
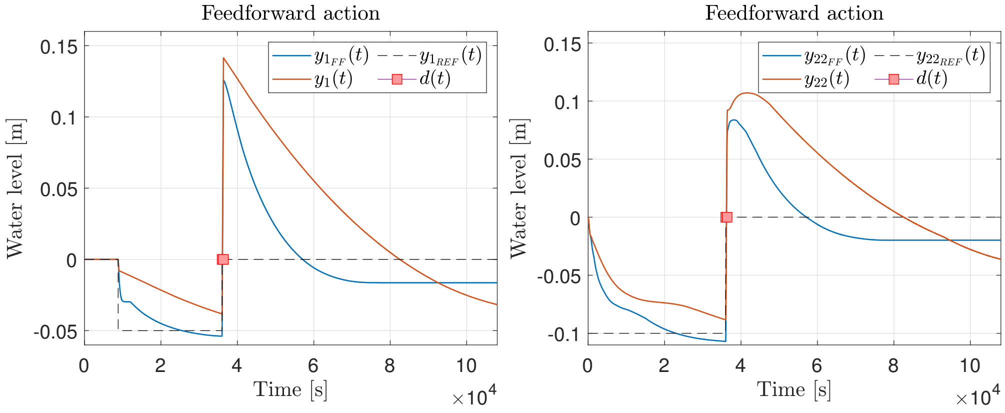


Figure 5.16: Simulink control scheme

A comparison between the water level with and without feedforward action can be observed in Figure 5.17.

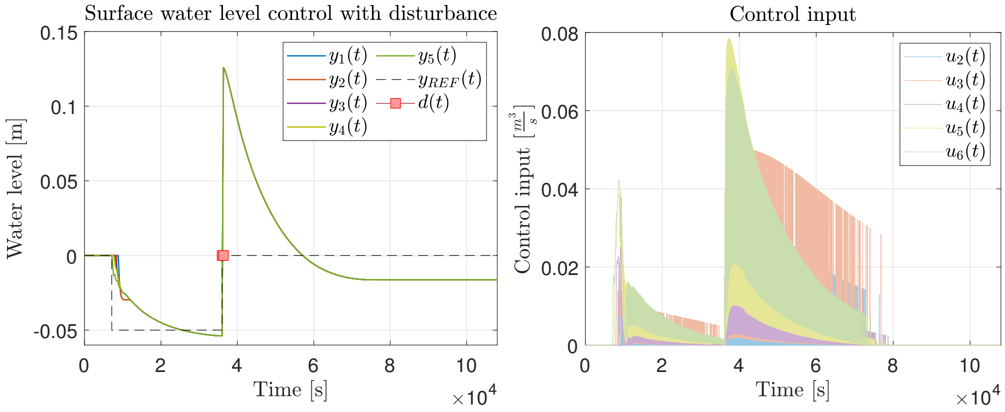


(a) Pool 1

(b) Pool 22

Figure 5.17: Water level simulation with and without feedforward control action

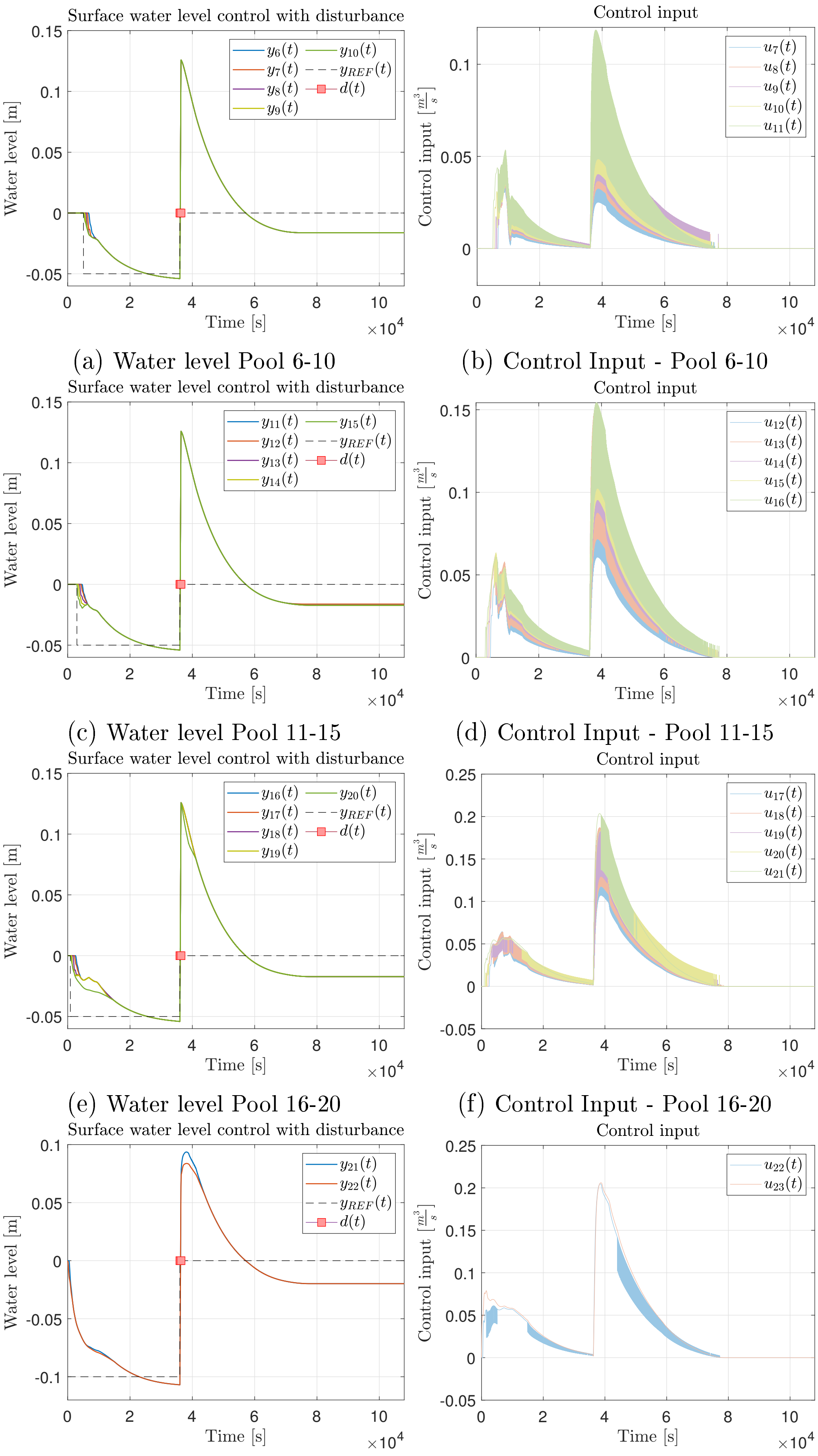
The simulation results with feedforward action are reported in Figures 5.18 and 5.19. It can be observed that the disturbance rejection is slow but with a smart change of reference the maximum water level is not reached, avoiding floods. Moreover, the control input response shows an undershoot of 2 cm, that can be reduced trying to reduce the integral action, i.e. $T_c = T_c \cdot 1.5$ for a 50% reduction of the integral action.



(a) Water level Pool 1-5

(b) Control Input - Pool 1-5

Figure 5.18: Control of a string of Cavallino pools with disturbance.



(a) Water level Pool 6-10

(b) Control Input - Pool 6-10

(c) Water level Pool 11-15

(d) Control Input - Pool 11-15

(e) Water level Pool 16-20

(f) Control Input - Pool 16-20

(g) Water level Pool 21-22

(h) Control Input - Pool 21-22

Figure 5.19: Control of a string of Cavallino pools with disturbance.

6

Conclusions and future work

6.1 Conclusions

In order to deal with this work, we draw inspiration from the open-channel irrigation field, highlighting the main differences with our water discharge scenario. We have presented a system-identification-based procedure to estimate parameters of a discrete time first order model and a control strategy to regulate upstream water level in order to avoid floods in case of strong perturbation of the system. Indeed, as soon as we collect some measurements of the water level and of the gates position, an OE structure model can be tested as candidate model. The Levemberg-Marquardt algorithm allows to solve the minimization of the cost given by the predicted error, showing better results with respect to the ARX model, in terms of mean square error and best fit. Once the parameters are estimated, a one-step-ahead predictor can be easily implemented. The main contribution on this thesis is the upstream control of the string of pools. Feedforward implementation allows to reduce the error amplification and propagation between pools and to attain a better performance in terms of rise time a disturbance rejection. In general, the PIL tuning automatic routine proposed allows to reach the desired phase margin and gain at the waves frequencies. The resulting control performance is slow but the disturbance rejection is satisfied and the lowpass filter cut the waves frequencies. Then, the resulting control input is feasible but there is some undershoot that can be reduced trying to limit the integral action. Moreover, it is required a change of the set point in order to discharge in advance the pool with respect to a violent perturbation, that is a probabilistic event. For this reason, the amplitude of the reference needs to be chosen in the most suitable way, according to the probability of the event, given by the forecast.

6.2 Future works

Some of the possible improvements that we can make are listed in the following lines.

- We could consider a LQR centralize control, in order to reach a better control performance by means of a controller that requires less maintenance with respect to a PIL (4.46);
- We could extend our work to the whole networked, taking into account junctions and the network flow systems;
- We could obtain real measurements by means of sensors, analyzing the minimum number required to get better results;
- We could extend the model considering pipelines.

Appendix

A SVEs implementation

A finite difference approach is analyzed and exploited to evaluate SVEs solution for one-dimensional flow of open channel systems.

A.1 Numerical solution methods

SVEs are nonlinear partial differential equations (PDE) of the hyperbolic type, for which a closed form solution is not possible. Hence, there exist multiple methods that can be used for their integration. Considering a channel of known parameters and measures such as cross-sectional geometry, roughness factor and longitudinal slope, the unknowns in equations (2.4) and (2.5) are the discharge, Q, and the flow depth, y, while the independent variables are time, t, and distance along the channel, x. The channel dynamic is discretized in order to apply numerical solution methods. In particular, a computation grid of uniform size is exploited to define finite difference equations. Vertical lines depict different sections along the channel, while horizontal lines depict the discrete time at which a solution is computed. The space increment is denoted by Δx while the time increments are denoted by Δt and they are assumed to be constant. All the nodes on the first horizontal line satisfy the initial condition of the flow. The nodes on the first vertical line describe the upstream end of the channel while the M-th vertical line describes the downstream end of the channel. In correspondence of these vertical lines there are two boundary conditions to satisfy. Once the initial conditions are known, flow conditions at all the nodes on the horizontal line can be computed for the next time step. The finite difference equations can be determined by means of Taylor series approximations to the partial differential terms of the SVEs. On the other hand, this approach is affected by truncation errors, that may be lead to a wrong solution. Then, it is fundamental to ensure stability of the numerical method. The finite difference approach can be distinguished as explicit and implicit. Explicit methods exhibit spatial partial derivatives replaced in terms of the variables at the known time level, while those in terms of variables at future time level are referred to implicit methods.

A.2 Preissmann scheme

In the implicit finite-difference schemes, the spatial partial derivatives and/or the coefficients are replaced in terms of the values at the unknown time level. Preissmann scheme has the advantages that a variable spatial grid can be exploited (see Figure 6.1). Moreover, such scheme proposes an accurate solution of the linearized form of the main equations for a specific value of Δx and Δt. The partial derivatives for a channel section between nodes i and $i+1$ are approximated as

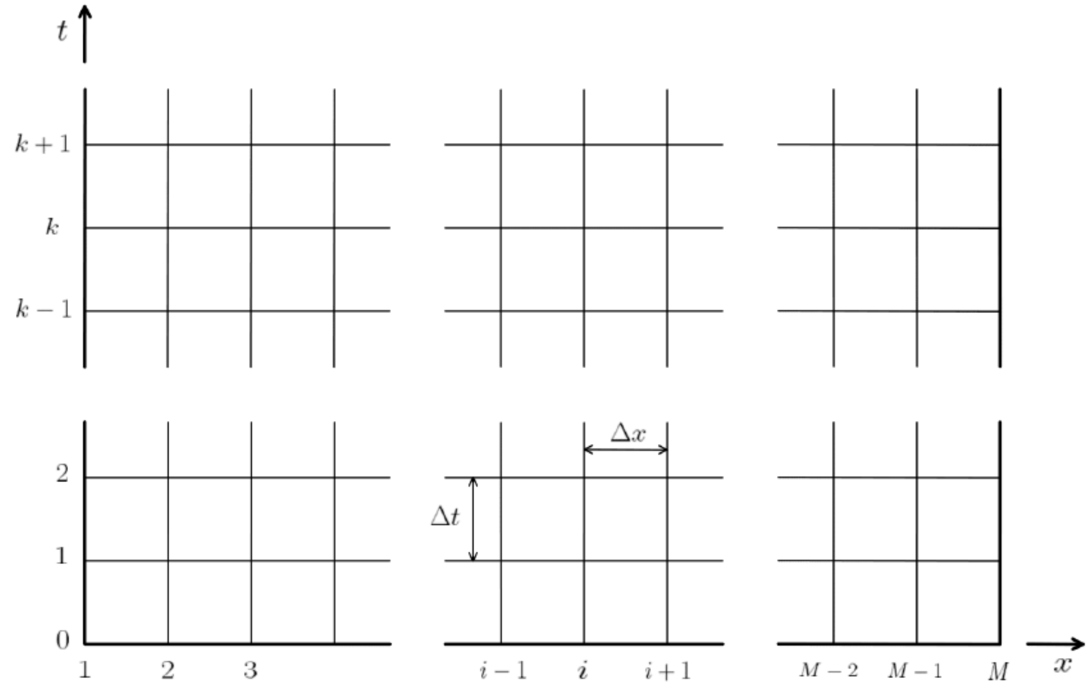


Figure 6.1: Computation grid for numerical solution methods

$$\begin{aligned}
f &= \frac{1}{2}\alpha(f_{i+1}^{k+1} + f_i^{k+1}) + \frac{1}{2}(1-\alpha)(f_{i+1}^{k} + f_i^{k}) \\
\frac{\partial f}{\partial t} &= \frac{(f_{i+1}^{k+1} + f_i^{k+1}) - (f_{i+1}^{k} + f_i^{k})}{2\Delta t} \\
\frac{\partial f}{\partial x} &= \frac{\alpha(f_{i+1}^{k+1} - f_i^{k+1})}{\Delta x} + \frac{(1-\alpha)(f_{i+1}^{k} - f_i^{k})}{\Delta x}
\end{aligned} \tag{6.1}$$

where f refers to the unknown quantities Q and y, while α is a weighting coefficient. This four point implicit approach is stable if t $0.5 < \alpha < 1$. It is worth noting that the cross sectional area A_i^{k+1} and the friction slope $\bar{S}_i^{k+1}$ can be expressed as a function of the unknowns y_i^{k+1} and Q_i^{k+1}. Therefore, the components of the SVEs can be rewritten as the so called four point implicit scheme

$$\begin{aligned}
\frac{\partial Q}{\partial t} &\simeq \frac{(Q_{i+1}^{k+1}+Q_i^{k+1})-(Q_{i+1}^{k}+Q_i^{k})}{2\Delta t}\\
\frac{\partial Q}{\partial x} &\simeq \frac{\alpha(Q_{i+1}^{k+1}-Q_i^{k+1})}{\Delta x}+\frac{(1-\alpha)(Q_{i+1}^{k}-Q_i^{k})}{\Delta x}\\
\frac{\partial A}{\partial t} &\simeq \frac{(A_{i+1}^{k+1}+A_i^{k+1})-(A_{i+1}^{k}+A_i^{k})}{2\Delta t}
\end{aligned} \tag{6.2}$$

$$\begin{aligned}
\frac{\partial (Q^2/A)}{\partial x} \simeq {} & \frac{\alpha\left\{\left[(Q_{i+1}^{k+1})^2/A_{i+1}^{k+1}\right]-\left[(Q_{i}^{k+1})^2/A_{i}^{k+1}\right]\right\}}{\Delta x}+\\
&+\frac{(1-\alpha)\left\{\left[(Q_{i+1}^{k+1})^2/A_{i+1}^{k+1}\right]-\left[(Q_{i}^{k+1})^2/A_{i}^{k+1}\right]\right\}}{\Delta x}
\end{aligned} \tag{6.3}$$

$$\frac{A\,\partial y}{\partial x} \simeq \frac{\alpha(A_{i+1}^{k+1}+A_i^{k+1})(y_{i+1}^{k+1}-y_i^{k+1})}{2\Delta x}+\frac{(1-\alpha)(A_{i+1}^{k}+A_i^{k})(y_{i+1}^{k}-y_i^{k})}{2\Delta x} \tag{6.4}$$

$$A \simeq \frac{\alpha(A_{i+1}^{k+1}+A_i^{k+1})+(1-\alpha)(A_{i+1}^{k}+A_i^{k})}{2} \tag{6.5}$$

$$\begin{aligned}
A\bar{S} \simeq {} & \frac{\alpha(A_{i+1}^{k+1}+A_i^{k+1})}{2}\frac{(\bar{S}_{i+1}^{k+1}+\bar{S}_i^{k+1})}{2}+\\
&+\frac{(1-\alpha)(A_{i+1}^{k}+A_i^{k})}{2}\frac{(\bar{S}_{i+1}^{k}+\bar{S}_i^{k})}{2}
\end{aligned} \tag{6.6}$$

Thus, exploiting these results, the continuity equation in finite difference form becomes

$$\begin{aligned}
C(Q_i^{k+1},Q_{i+1}^{k+1},y_i^{k+1},y_{i+1}^{k+1}) = {} & \frac{(A_{i+1}^{k+1}+A_i^{k+1})-(A_{i+1}^{k}+A_i^{k})}{2\Delta t}+\\
&+\frac{\alpha(Q_{i+1}^{k+1}-Q_i^{k+1})+(1-\alpha)(Q_{i+1}^{k}-Q_i^{k})}{2\Delta x}=0
\end{aligned} \tag{6.7}$$

while the momentum equation in finite difference form, can be written as follows

$$\begin{aligned}
M(Q_i^{k+1}, Q_{i+1}^{k+1}, y_i^{k+1}, y_{i+1}^{k+1}) = & \frac{(Q_{i+1}^{k+1} + Q_i^{k+1}) - (Q_{i+1}^{k} + Q_i^{k})}{2\Delta t} + \\
& + \frac{\alpha((Q_{i+1}^{k+1})^2/A_{i+1}^{k+1} - (Q_i^{k+1})^2/A_i^{k+1}))}{\Delta x} \\
& + \frac{(1-\alpha)((Q_{i+1}^{k})^2/A_{i+1}^{k} - (Q_i^{k})^2/A_i^{k}))}{\Delta x} \\
& + g\alpha \frac{(A_{i+1}^{k+1} + A_i^{k+1})}{2} \frac{(y_{i+1}^{k+1} - y_i^{k+1})}{\Delta x} \\
& g(1-\alpha) \frac{(A_{i+1}^{k} + A_i^{k})}{2} \frac{(y_{i+1}^{k} - y_i^{k})}{\Delta x} \\
& g\alpha \frac{(A_{i+1}^{k+1} + A_i^{k+1})}{2} \frac{(S_{i+1}^{k+1} + S_i^{k+1})}{2} \\
& g(1-\alpha) \frac{(A_{i+1}^{k} + A_i^{k})}{2} \frac{(S_{i+1}^{k} + S_i^{k})}{2} = 0
\end{aligned} \tag{6.8}$$

Moreover, SVEs can be rearranged into compact matrix form that is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \mathbf{S} = 0 \tag{6.9}$$

where

$$\mathbf{U} = \begin{bmatrix} A \\ Q \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} Q \\ \frac{Q^2}{A} + gAy \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 0 \\ gA(\bar{S} - S_0) \end{bmatrix} \tag{6.10}$$

Then, equation (A.2) can be rewritten as follows

$$\begin{aligned}
(\mathbf{U}_i^{k+1} + \mathbf{U}_{i+1}^{k+1}) - (\mathbf{U}_i^{k} + \mathbf{U}_{i+1}^{k}) + 2\frac{\Delta t}{\Delta x} & \left[\alpha(\mathbf{F}_{i+1}^{k+1} - \mathbf{F}_i^{k+1}) + (1-\alpha)(\mathbf{F}_{i+1}^{k} - \mathbf{F}_i^{k})\right] \\
\Delta t & \left[\alpha(\mathbf{S}_{i+1}^{k+1} - \mathbf{S}_i^{k+1}) + (1-\alpha)(\mathbf{S}_{i+1}^{k} - \mathbf{S}_i^{k})\right] = 0
\end{aligned} \tag{6.11}$$

The scheme is unconditionally stable if and only if it is satisfied $\frac{1}{2} < \alpha < 1$. An unconditional stability indicates that there are not boundaries on the size of Δx and Δt to reach stability.

A.3 Trapezoidal cross-sectional area

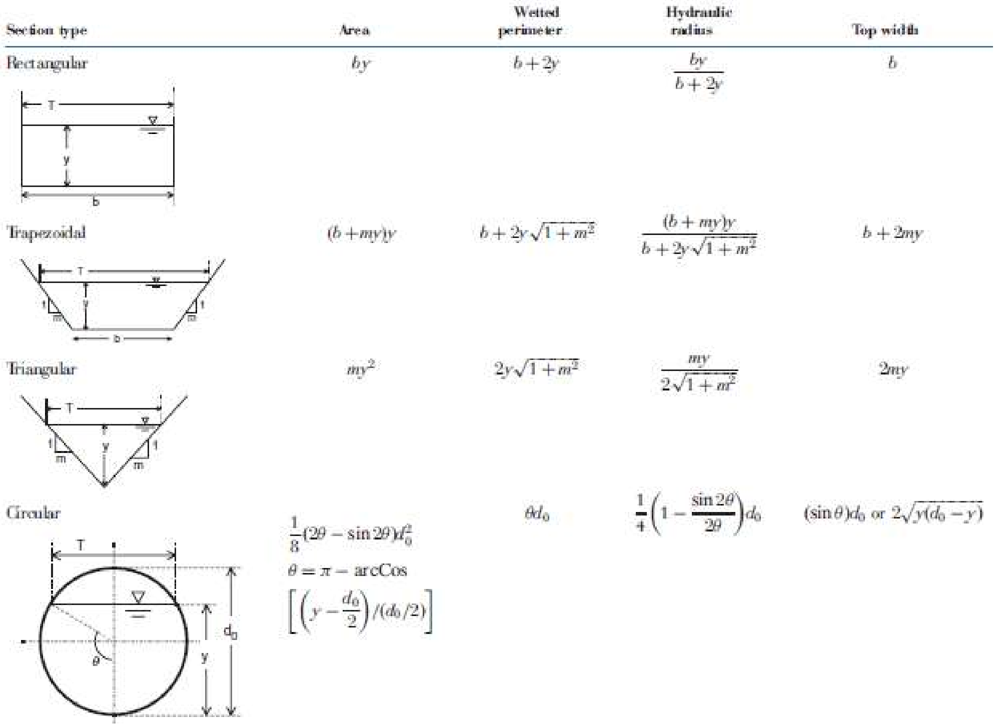
Let us assume that the wetted cross sectional area can be approximated by a trapezoid. Let b, B, P, s denote the bottom width of the channel, the top width, the wetted perimeter and the side slope respectively, where $s = \frac{(B-b)}{2y}$. Thus, it holds

$$\frac{B}{2} = sy + \frac{b}{2} \tag{6.12}$$

Therefore, the wetted cross-sectional area can be rewritten as

$$A = \frac{(b+B)y}{2} = (b+sy)y \tag{6.13}$$

In Figure 6.2 are reported the main geometric parameters according to the shape of the channel.



Section type	Area	Wetted perimeter	Hydraulic radius	Top width
Rectangular	by	$b+2y$	$\frac{by}{b+2y}$	b
Trapezoidal	$(b+my)y$	$b+2y\sqrt{1+m^2}$	$\frac{(b+my)y}{b+2y\sqrt{1+m^2}}$	$b+2my$
Triangular	my^2	$2y\sqrt{1+m^2}$	$\frac{my}{2\sqrt{1+m^2}}$	$2my$
Circular	$\frac{1}{8}(2\theta - \sin 2\theta)d_0^2$ $\theta = \pi - \text{arcCos}$ $\left[\left(y - \frac{d_0}{2}\right)/(d_0/2)\right]$	θd_0	$\frac{1}{4}\left(1 - \frac{\sin 2\theta}{2\theta}\right)d_0$	$(\sin\theta)d_0$ or $2\sqrt{y(d_0 - y)}$

Figure 6.2: Geometric parameters with different cross sectional area

A.4 Boundary conditions

Let us assume that index i, $i = 1, ..., N$, denotes the channel spatial section and index k, $k = 1, ..., T$ denotes the time in the computation grid scheme. Boundary conditions for each $j = 1, ..., N_p$ pool regarding the upstream and downstream end can be stated as follows and can be different according to the control structure chosen for each pool. In the following, boundary conditions for pools with weirs and gates are considered.

A.4.1 Rectangular crested weir structure

Multiple boundary conditions can be stated according to the flow classification. If the system provides a weir at the upstream and downstream, then boundary conditions for an unsteady flow can be stated as

$$Q_{j,i=1}^{k+1} - Q_U^{k+1} = 0 \tag{6.14}$$

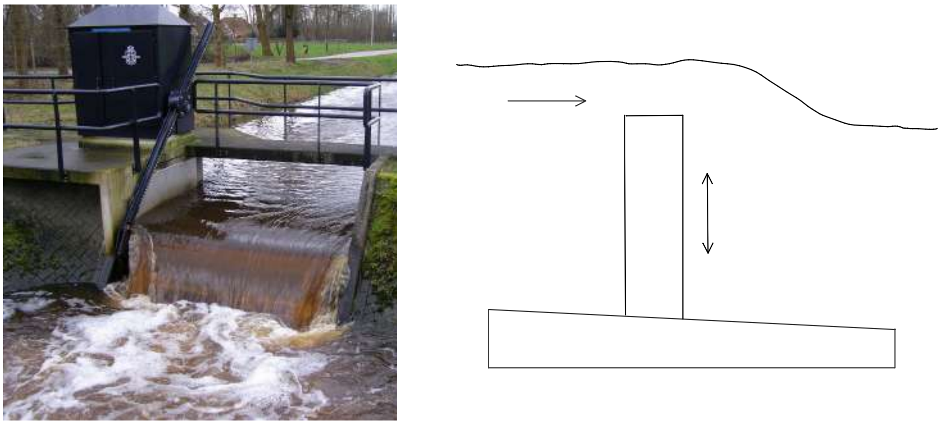
where Q_U^{k+1} is the given upstream inflow rate at time stage $k+1$. For what concerns the downstream end of the channel regulated by weirs, the main constraint is

$$Q_{j,i=N}^{k+1} = \begin{cases} 0, & \text{if } h_N^{k+1} < 0 & (6.15) \\ c_{weir}(h_N^{k+1})^{\frac{3}{2}}, & \text{otherwise} & (6.16) \end{cases}$$

where h_N^{k+1} is the downstream height of the water over the weir, defined as

$$h_i^{k+1} = y_i^{k+1} - p_{weir}^{k+1} \tag{6.17}$$

, where p is the weir opening, $c_{weir} = c_d b_w \sqrt{2g}$ where b_w is the length of the aperture, $c_d \approx 0.6$. The weir control structures is reported in Figure 6.3.



(a) Real weir structure (b) Sketch of a weir structure

Figure 6.3: Weir control structure

A.4.2 Rectangular gate structure

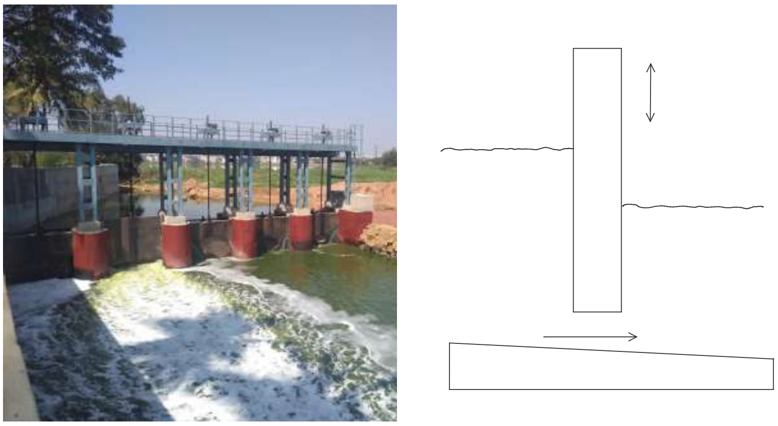
In case of undershot gates, the boundary conditions for an unsteady flow at the upstream can be expressed as

$$Q^{k+1}_{j,i=1} - Q^{k+1}_{U} = 0 \tag{6.18}$$

where Q^{k+1}_{u} is the given upstream inflow rate at time stage $k+1$, while for the downstream end it holds

$$Q^{k+1}_{j,i=N} = c_{gate} p_{gate} \sqrt{y^{k+1}_{U,i=N} - y^{k+1}_{D,i=N}} \tag{6.19}$$

where p_{gate} is the gate opening, y_U and y_D are the water depth upstream and downstream of the gate , $c_{gate} = c_d A_g \sqrt{2g}$, $c_d \approx 0.6$ [44] and A_g is the section of the aperture. The gate control structures is reported in Figure 6.4.



(a) Real gate structure (b) Sketch of a gate structure

Figure 6.4: Gate control structure

A.5 Initial conditions

In order to solve SVEs with finite difference approach, initial condition of the water level and initial flow need to be chosen. For sake of simplicity, it can be assumed a steady state initial condition, which imply that all the time derivatives are set to zero. Therefore, continuity equation (2.4) yields

$$\frac{\partial Q}{\partial x} = 0 \tag{6.20}$$

that means assuming a water flow constant along the spatial grid. In addition, the initial water flow is constant for all the spatial increment Δx then it holds

$$Q^1_{j,i} = Q^1_U \tag{6.21}$$

Moreover, it is worth noting that the bottom width b and the side slope s change along the spatial sections, then it holds

$$\frac{\partial A}{\partial x} = (b+2sy)\frac{\partial y}{\partial x} + y\frac{\partial b}{\partial x} + y^2\frac{\partial s}{\partial x} \tag{6.22}$$

For sake of simplicity, the bottom width and the side slope are considered constant, then SVEs become

$$\begin{aligned} gA(S_0-\bar{S}) &= \frac{\partial}{\partial x}\frac{Q^2}{A} + gA\frac{\partial y}{\partial x} \\ gA(S_0-\bar{S}) &= \frac{1}{A}\frac{\partial Q^2}{\partial x} - \frac{Q^2}{A^2}\frac{\partial A}{\partial x} + gA\frac{\partial y}{\partial x} \\ gA(S_0-\bar{S}) &= -\frac{Q^2}{A^2}\frac{\partial A}{\partial x} + gA\frac{\partial y}{\partial x} \\ gA(S_0-\bar{S}) &= -\frac{Q^2}{A^2}(b+2sy)\frac{\partial y}{\partial x} + gA\frac{\partial y}{\partial x} \end{aligned} \tag{6.23}$$

Finally, the differential form of equation (6.23) is the ordinary differential equation

$$\frac{dy}{dx} = \frac{gA(S_0-\bar{S})}{gA - \frac{Q^2(b+2sy)}{A^2}} \tag{6.24}$$

whose solution can be exploited as initial condition of the finite difference approach.

A.6 Solution procedure

Implicit finite difference equations are expressed in function of y_i^{k+1} and Q_i^{k+1}, while the area and the friction slope can be easily formulated in function of the water level and flow, once the geometrical parameters of the channel are determined. Let us refer B_1 and B_N to the boundary condition equations and let us denote by C_i and M_i the finite difference continuity and momentum equations for the section between nodes i and *i+1*. Thus, we need to solve *2N* non-linear equations in *2N* unknows,

$$\begin{aligned}
B_1\left[Q_1^{k+1}, y_1^{k+1}\right] &= 0\\
C_1\left[Q_1^{k+1}, y_1^{k+1}, Q_2^{k+1}, y_2^{k+1}\right] &= 0\\
M_1\left[Q_1^{k+1}, y_1^{k+1}, Q_2^{k+1}, y_2^{k+1}\right] &= 0\\
C_2\left[Q_2^{k+1}, y_2^{k+1}, Q_3^{k+1}, y_3^{k+1}\right] &= 0\\
M_2\left[Q_2^{k+1}, y_2^{k+1}, Q_3^{k+1}, y_3^{k+1}\right] &= 0\\
&\ldots\ldots\ldots\ldots\ldots\\
&\ldots\ldots\ldots\ldots\ldots\\
C_{N-1}\left[Q_{N-1}^{k+1}, y_{N-1}^{k+1}, Q_N^{k+1}, y_N^{k+1}\right] &= 0\\
M_{N-1}\left[Q_{N-1}^{k+1}, y_{N-1}^{k+1}, Q_N^{k+1}, y_N^{k+1}\right] &= 0\\
B_N\left[Q_N^{k+1}, y_N^{k+1}\right] &= 0
\end{aligned} \tag{6.25}$$

One solution approach could be Newton iterative method. In particular, we specify a set of guess values for the unknowns y_i^{k+1}, Q_i^{k+1} for $i = 1, ..., N$. Then, these values are replaced in the left side of the system of equations (6.25) and residuals different from zero are generated if the proposed solution does not correspond to the optimal solution. Therefore, in the next iteration, new estimated guess values are proposed in order to improve the performance. In particular, some correction terms $\Delta Q_i, \Delta y_i$ for the unknowns are evaluated for $i = 1, ..., N$, in order to reduce the residuals $rB_1, rB_N, rC_i, rM_i, i = 1, ..., N-1$. Then it holds

$$\begin{aligned}
\frac{\partial B_1}{\partial Q_1^{k+1}}\Delta Q_1 + \frac{\partial B_1}{\partial y_1^{k+1}}\Delta y_1 &= -rB_1\\
\frac{\partial C_1}{\partial Q_1^{k+1}}\Delta Q_1 + \frac{\partial C_1}{\partial y_1^{k+1}}\Delta y_1 + \frac{\partial C_1}{\partial Q_2^{k+1}}\Delta Q_2 + \frac{\partial C_1}{\partial y_2^{k+1}}\Delta y_2 &= -rC_1\\
\frac{\partial M_1}{\partial Q_1^{k+1}}\Delta Q_1 + \frac{\partial M_1}{\partial y_1^{k+1}}\Delta y_1 + \frac{\partial M_1}{\partial Q_2^{k+1}}\Delta Q_2 + \frac{\partial M_1}{\partial y_2^{k+1}}\Delta y_2 &= -rM_1\\
\frac{\partial C_2}{\partial Q_2^{k+1}}\Delta Q_2 + \frac{\partial C_2}{\partial y_1^{k+1}}\Delta y_2 + \frac{\partial C_2}{\partial Q_3^{k+1}}\Delta Q_3 + \frac{\partial C_2}{\partial y_3^{k+1}}\Delta y_3 &= -rC_2\\
\frac{\partial M_2}{\partial Q_2^{k+1}}\Delta Q_2 + \frac{\partial M_2}{\partial y_2^{k+1}}\Delta y_2 + \frac{\partial M_2}{\partial Q_3^{k+1}}\Delta Q_3 + \frac{\partial M_2}{\partial y_3^{k+1}}\Delta y_3 &= -rM_2
\end{aligned} \tag{6.26}$$

$$
\begin{aligned}
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
\frac{\partial C_{N-1}}{\partial Q_{N-1}^{k+1}}\Delta Q_{N-1}+\frac{\partial C_{N-1}}{\partial y_{N-1}^{k+1}}\Delta y_{N-1}+\frac{\partial C_{N-1}}{\partial Q_{N}^{k+1}}\Delta Q_{N}+\frac{\partial C_{N-1}}{\partial y_{N}^{k+1}}\Delta y_{N}&=-rC_{N-1}\\
\frac{\partial M_{N-1}}{\partial Q_{N-1}^{k+1}}\Delta Q_{N-1}+\frac{\partial M_{N-1}}{\partial y_{N-1}^{k+1}}\Delta y_{N-1}+\frac{\partial M_{N-1}}{\partial Q_{N}^{k+1}}\Delta Q_{N}+\frac{\partial M_{N-1}}{\partial y_{N}^{k+1}}\Delta y_{N}&=-rM_{N-1}\\
\frac{\partial B_{N}}{\partial Q_{N}^{k+1}}\Delta Q_{N}+\frac{\partial B_{N}}{\partial y_{N}^{k+1}}\Delta y_{N}&=-rB_{N}
\end{aligned}
\tag{6.27}
$$

System of linear equations (6.27) can be rewritten in a more compact form, that is

$$
Jx=f \tag{6.28}
$$

where

$$
J=\begin{bmatrix}
\frac{\partial B_1}{\partial Q_1} & \frac{\partial B_1}{\partial y_1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
\frac{\partial C_1}{\partial Q_1} & \frac{\partial C_1}{\partial y_1} & \frac{\partial C_1}{\partial Q_2} & \frac{\partial C_1}{\partial y_2} & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
\frac{\partial M_1}{\partial Q_1} & \frac{\partial M_1}{\partial y_1} & \frac{\partial M_1}{\partial Q_2} & \frac{\partial M_1}{\partial y_2} & 0 & 0 & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & \frac{\partial C_2}{\partial Q_2} & \frac{\partial C_2}{\partial y_2} & \frac{\partial C_2}{\partial Q_3} & \frac{\partial C_2}{\partial y_3} & \cdots & 0 & 0 & 0 & 0\\
0 & 0 & \frac{\partial M_2}{\partial Q_2} & \frac{\partial M_2}{\partial y_2} & \frac{\partial M_2}{\partial Q_3} & \frac{\partial M_2}{\partial y_3} & \cdots & 0 & 0 & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{\partial C_{N-1}}{\partial Q_{N-1}} & \frac{\partial C_{N-1}}{\partial y_{N-1}} & \frac{\partial C_{N-1}}{\partial Q_N} & \frac{\partial C_{N-1}}{\partial y_N}\\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{\partial M_{N-1}}{\partial Q_{N-1}} & \frac{\partial M_{N-1}}{\partial y_{N-1}} & \frac{\partial M_{N-1}}{\partial Q_N} & \frac{\partial M_{N-1}}{\partial y_N}\\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{\partial B_N}{\partial Q_N} & \frac{\partial B_N}{\partial Q_N}
\end{bmatrix}
\tag{6.29}
$$

$$
x=\begin{bmatrix}\Delta Q_1\\ \Delta y_1\\ \Delta Q_2\\ \Delta y_2\\ \vdots\\ \Delta Q_N\\ \Delta y_N\end{bmatrix}
\qquad
f=\begin{bmatrix}-rB_1\\ -rC_1\\ -rM_1\\ -rC_2\\ -rM_2\\ \vdots\\ -rC_{N-1}\\ -rM_{N-1}\\ -rB_N\end{bmatrix}
\tag{6.30}
$$

Once the matrix form is computed, the corrections term can be determined by matrix inversion, taking into account that we are dealing with a sparse matrix to improve time complexity. Then, the solution for the water level and water flow of each section can be computed iteratively

$$\begin{aligned}(Q_i^{k+1})_{n+1} &= (Q_i^{k+1})_n + (\Delta Q_i)_n \\ (y_i^{k+1})_{n+1} &= (y_i^{k+1})_n + (\Delta y_i)_n\end{aligned} \tag{6.31}$$

where n denotes the iterations until convergence.

References

[1] O.S. Balogun, M. Hubbard, J.J. DeVries. Automatic control of canal flow using linear quadratic regulator theory. *Journal of Hydraulic Engineering*, 114(1):75–102, 1988.

[2] P.O. Malaterre. Pilote: Linear quadratic optimal controller for irrigation canals. *Journal of Irrigation and Drainage Engineering*, 124(4):187–194, 1998. cited By 93.

[3] F. Liu, J. Feyen, P.O. Malaterre, J.P. Baume, P. Kosuth. Development and evaluation of canal automation algorithm clis. *Journal of Irrigation and Drainage Engineering*, 124(1):40–46, 1998. cited By 23.

[4] G. Corriga, S. Sanna, G. Usai. Frequency response and dynamic behaviour of canal networks with self-levelling gates. *Applied Mathematical Modelling*, 4(2):125–129, 1980.

[5] X. Litrico, V. Fromion. Analytical approximation of open-channel flow for controller design. *Applied Mathematical Modelling*, 28(7):677–695, 2004.

[6] J. Schuurmans. Control of water levels in open-channels. 1997.

[7] X. Litrico, V. Fromion. Simplified modeling of irrigation canals for controller design. *Journal of Irrigation and Drainage Engineering*, 130(5):373–383, 2004.

[8] K. Horváth, E. Galvis, J. Rodellar, M.V. Gómez. Experimental comparison of canal models for control purposes using simulation and laboratory experiments. *Journal of Hydroinformatics*, 16(6):1390–1408, 07 2014.

[9] B.T. Wahlin. Performance of model predictive control on asce test canal 1. *Journal of Irrigation and Drainage Engineering*, 130(3):227–238, 2004.

[10] R. Rivas Perez, V. Feliu Batlle, L. Sanchez Rodriguezù. Robust system identification of an irrigation main canal. *Advances in Water Resources*, 30(8):1785–1796, 2007.

[11] P.J. van Overloop, K. Horváth, B. Ekin Aydin. Model predictive control based on an integrator resonance model applied to an open water channel. *Control Engineering Practice*, 27:54–60, 2014.

[12] I. Tavares, J. Borges, M. Mendes, B.M Ayala. Assessment of data-driven modeling strategies for water delivery canals. *Neural Computing and Applications*, 23, 09 2013.

[13] E. Weyer. System identification of an open water channel. *Control Engineering Practice*, 9(12):1289–1299, 2001.

[14] M. Cantoni, E. Weyer, Y. Li, S.K. Ooi, I. Mareels, M. Ryan. Control of large-scale irrigation networks. *Proceedings of the IEEE*, 95(1):75–91, 2007.

[15] S.K. Ooi, E. Weyer. Control design for an irrigation channel from physical data. *Control Engineering Practice*, 16(9):1132–1150, 2008.

[16] I. Mareels, E. Weyer, S.K. Ooi, M. Cantoni, Y. Li , G. Nair. Systems engineering for irrigation systems: Successes and challenges. *Annual Reviews in Control*, 29(2):191–204, 2005.

[17] E. Weyer, G. Bastin. Leak detection in open water channels. *IFAC Proceedings Volumes*, 41(2):7913–7918, 2008. 17th IFAC World Congress.

[18] K. Eurén, E. Weyer. System identification of open water channels with undershot and overshot gates. *IFAC Proceedings Volumes*, 38(1):638–643, 2005. 16th IFAC World Congress.

[19] P.O. Malaterre. Regulation of irrigation canals - characterisation and classification. *Irrigation and Drainage Systems*, 9(4):297–327, 1995.

[20] B.E Aydin, P.J. van Overloop, M. Rutten, X. Tian. Offset-free model predictive control of an open water channel based on moving horizon estimation. *Journal of Irrigation and Drainage Engineering*, 143(3):B4016005, 2017.

[21] O. Begovich, V. M. Ruiz, G. Besançon, G. Georges, C. Aldana. Predictive control with constraints of a multi-pool irrigation canal prototype. *Latin American Applied Research*, 37:177–185, 2007.

[22] K. Horváth, E. Galvis, M. Gómez Valentín, J. Rodellar. New offset-free method for model predictive control of open channels. *Control Engineering Practice*, 41:13–25, 2015.

[23] P. Segovia, L. Rajaoarisoa, F. Nejjari, V. Puig, E. Duviella. Decentralized control of inland navigation networks with distributaries: Application to navigation canals in the north of france. 05 2017.

[24] E. Weyer. Control of irrigation channels. *IEEE Transactions on Control Systems Technology*, 16(4):664–675, 2008.

[25] N. Le-Duy-Lai, P. Ionela, L. Laurent, G.C. Denis. Distributed model predictive control of irrigation systems using cooperative controllers **the work is partially supported and funded by the artemis ar-rowhead european project under grant agreement number 332987. *IFAC-PapersOnLine*, 50(1):6564–6569, 2017. 20th IFAC World Congress.

[26] X. Tian, P.J. van Overloop, X. Litrico. Integrator delay zero model for design of upstream water-level controllers. *Journal of Irrigation and Drainage Engineering*, 143:B4015001, 12 2015.

[27] M. Rijo, C. Arranja. Supervision and water depth automatic control of an irrigation canal. *Journal of Irrigation and Drainage Engineering*, 136(1):3–10, 2010.

[28] J.M. Lemos, I. Sampaio. Distributed lqg control for multiobjective control of water canals. *Operations Research/ Computer Science Interfaces Series*, 58:59–73, 01 2015.

[29] L. Rato, P. Salgueiro, J.M. Lemos, M. Rijo. Adaptive predictive controller applied to an open water canal. In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO,*, pages 357–360. INSTICC, SciTePress, 2007.

[30] M. Breckpot, O.M. Agudelo, B.D. Moor. Flood control with model predictive control for river systems with water reservoirs. *Journal of Irrigation and Drainage Engineering*, 139(7):532–541, 2013.

[31] A. Boran, P.J. van Overloop, M. Rutten, X. Tian. Offset-free model predictive control of an open water channel based on moving horizon estimation. *Journal of Irrigation and Drainage Engineering*, pages 1–9, 2016.

[32] X. Litrico, D. Georges. Robust continuous-time and discrete-time flow control of a dam–river system. (ii) controller design. *Applied Mathematical Modelling*, 23(11):829–846, 1999.

[33] P.J. van Overloop, S. Weijs, S. Dijkstra. Multiple model predictive control on a drainage canal system. *Control Engineering Practice*, 16(5):531–540, 2008.

[34] C.M. Burt, R.S. Mills, R.D. Khalsa, C.V. Ruiz. Improved proportional-integral (pi) logic for canal automation. *Journal of Irrigation and Drainage Engineering*, 124(1):53–57, 1998.

[35] X. Litrico, V. Fromion, J.P. Baume, M. Rijo. Modelling and pi control of an irrigation canal. In *2003 European Control Conference (ECC)*, pages 850–855, 2003.

[36] C.M. Burt P.J. van Overloop, J. Schuurmans, R. Brouwer. Multiple-model optimization of proportional integral controllers on canals. *Journal of Irrigation and Drainage Engineering*, 131(2):190–196, 2005.

[37] D. Lozano, C. Arranja, M. Rijo, L. Mateos. Simulation of automatic control of an irrigation canal. *Agricultural Water Management*, 97(1):91–100, 2010.

[38] X. Tian Xin G. Guan T. Arauz, J.M. Maestre. Design of pi controllers for irrigation canals based on linear matrix inequalities. *Water*, 12(3), 2020.

[39] X. Litrico, V. Fromion. Design of structured multivariable controllers for irrigation canals. volume 2005, pages 1881 – 1886, 01 2006.

[40] M.H. Chaudhry. Open-channel flow. 1999.

[41] A. Osman Akan. 1 - fundamentals of open-channel flow. In A. Osman, editor, *Open Channel Hydraulics*, pages 1–23. Butterworth-Heinemann, Oxford, 2006.

[42] M. Zorzi. Lecture notes in learning dynamical systems. Department of Information Engineering, Università di Padova, Italy, inhedit, 2020-2021.

[43] W. Boiten. Flow measuring structures. *Flow Measurement and Instrumentation*, 4:17–24, 1993.

[44] M.G. Bos. Runoff discharge measurement structures. *NASA STI/Recon Technical Report N*, 78:31395, 04 1976.

[45] A. Cenedese, R. Antonello, M. Fabris, G. Michieletto, N. Lissandrini, L. Varotto, and the class of 2016-17. Lecture notes in networked control for multiagent systems. Department of Information Engineering, Università di Padova, Italy, inhedit, 2020-2021.

[46] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[47] M. Cantoni, E. Weyer, Y. Li, S.K. Ooi, I. Mareels, M. Ryan. Control of large-scale irrigation networks. *Proceedings of the IEEE*, 95:75 – 91, 02 2007.

[48] S. Skogestad, I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley and Sons, Inc., Hoboken, NJ, USA, 2005.

[49] A. Farhadi, A. Khodabandehlou. Distributed model predictive control with hierarchical architecture for communication: application in automated irrigation channels. *International Journal of Control*, 89(8):1725–1741, 2016.

[50] A.Z. Cabeza, J.M. Maestre, M.A. Ridao, E.F. Camacho, L. Sánchez. A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control*, 21(5):787–799, 2011. Special Issue on Hierarchical and Distributed Model Predictive Control.

[51] A. Osman Akan. 8 - introduction to unsteady open-channel flow. In A. Osman, editor, *Open Channel Hydraulics*, pages 315–359. Butterworth-Heinemann, Oxford, 2006.

[52] X.Litrico, V.Fromion. Modeling and control of hydrosystems. 2009.

[53] E. Polak. *Optimization: Algorithms and Consistent Approximations*. Applied Mathematical Sciences. Springer New York, 2012.